CSE 203A: Randomized Algorithms

Spring 2026

Lecture 7: Yao's Minimax Principle and NAND Tree Evaluation
Date: 4.13
Instructor: Daniel M. Kane
Scribe: Yusen Ma

Lecture Overview

- Review of Yao's Minimax Principle in the context of query complexity.
- Application to NAND tree (game tree) evaluation.
- Separation between deterministic, randomized, and nondeterministic query complexity.
- Introduction of hard distributions for proving lower bounds.

1 Introduction

This lecture continues the study of query complexity and randomized algorithms. The main goal is to understand how randomness affects computational power and how Yao's Minimax Principle helps analyze randomized algorithms via deterministic ones.

2 Brief Reminders about Previous Topics

We consider **query problems**, where:

- The input is accessed via an oracle.
- The cost is measured only by the number of queries.

We focus on **zero-error randomized algorithms (ZPP)**.

Theorem 2.1 (Yao's Minimax Principle)**.** *The complexity of randomized algorithms equals:*

$$\min_{\text{randomized } A} \max_{x} \mathbb{E}[\text{queries of } A \text{ on } x] = \max_{D} \min_{\text{deterministic } A} \mathbb{E}_{x \sim D}[\text{queries of } A]$$

This reduces proving randomized lower bounds to:

- Finding a hard input distribution.
- Analyzing deterministic algorithms under that distribution.

3 Main Lecture Content

3.1 Fixing a Lemma from Randomized Routing

Definition 3.1. A packet has *delay* L if it reaches edge e_t at time $t + L$.

Claim 3.2. *If a packet ends with delay greater than L, then some other packet left the path with delay exactly L.*

Proof Idea. Each additional delay must be caused by another packet occupying an edge. By tracking delay propagation, one shows that for every level of delay there exists a distinct packet responsible. □

3.2 Game Tree and NAND Tree Evaluation

We consider a two-player game:

- Players alternate making binary decisions.
- After n steps, the outcome is determined.

This corresponds to evaluating a binary tree:

- Leaves encode outcomes.
- Internal nodes represent decisions.

Definition 3.3. Label each node as:

- 1 if the current player is winning,
- 0 otherwise.

Then each node equals the **NAND** of its children.

3.3 Deterministic Complexity

Theorem 3.4. *The deterministic query complexity is 2^n.*

Idea. An adversary assigns leaf values adaptively so that no internal node can be determined until all its descendants are queried. □

3.4 Nondeterministic Complexity

Theorem 3.5. *The nondeterministic complexity is $\Theta(2^{n/2})$.*

Idea. To prove a player wins, it suffices to verify all opponent strategies. Since the opponent moves every other step, there are $2^{n/2}$ possibilities. □

3.5 Randomized Algorithm

Algorithm 1: Randomized NAND Evaluation

Input: Node v
Output: Value of v
if *v is a leaf* **then**
  query v
else
  pick a random child c
  evaluate c
  if $c = 0$ **then**
    return 1
  else
    evaluate the other child

3.6 Expected Complexity

Define:

- a_n: expected queries when value is 1
- b_n: expected queries when value is 0

Recurrence:

$$a_n = 2b_{n-1}, \quad b_n = a_{n-1} + \frac{1}{2}b_{n-1}$$

Theorem 3.6. *The expected number of queries is at most $3^{n/2}$.*

Thus:

$$2^{n/2} < 3^{n/2} < 2^n$$

3.7 Lower Bounds via Hard Distributions

We construct a distribution:

- Each leaf is 1 independently with probability p

Choose p such that:

$$p = 1 - p^2$$

This makes the tree **self-similar**.

Lemma 3.7. *For independent leaves, the optimal deterministic algorithm is depth-first search with short-circuiting.*

Idea. Delaying evaluation is suboptimal. Either evaluate immediately or skip entirely. This leads to a DFS strategy being optimal. □

4 Further Remarks

- Yao's principle is a central tool for proving randomized lower bounds.
- NAND tree evaluation is a canonical example showing separations between models.
- The recurrence can be solved exactly using linear algebra methods.

5 Summary

- Yao's Minimax Principle connects randomized and deterministic complexity.
- NAND tree evaluation demonstrates separations between deterministic, randomized, and non-deterministic algorithms.
- Randomization provides exponential improvement over deterministic algorithms.
- Hard input distributions are key to proving lower bounds.