EECS 495: Randomized Algorithms **Lecture 5**
Lower Bounds for Randomized Alg.

Reading: Text: Chapter 2

Take-Aways

Concepts/Techniques:

- Lower bounds for rand. alg: Yao's min-max principle

Application:

- Evaluating game trees

Game Trees

Given: win-lose tree

- players alternate levels
- nodes state whether current mover wins or loses in subgame

Draw game tree.

Note: h levels, $n = 2^h$ leaves

Game Tree Evaluation

Fact: To check if parent is a ...

- winner, must find ONE losing child
- loser, must verify BOTH children are winners

Deterministic

Claim: Deterministic lower-bound: $2^h = n$

Proof: By induction on h.

- Alg needs to know value of all nodes at level $(h-1)$
- For each parent, make first examined child a winner

[[*Build up tree dynamically; ok because alg is deterministic (hence can simulate and build up bad tree in advance).*]]

Non-Deterministic/Checking

- $W(h)$ – time to prove win with height h
- $L(h)$ – time to prove lose with height h

Claim: Can verify in time $n^{1/2}$

Proof:

- $W(0) = L(0) = 1$
- Winning position guesses move: $W(h) = L(h-1)$
- Losing position checks both: $L(h) = 2W(h-1)$
- Thus $W(h) = 2W(h-2) = 2^{h/2} = n^{1/2}$

[[*So any (Las Vegas) alg needs time at least* $n^{1/2}$.]]

Randomized

Idea: Guess which leaf wins.

- $W(T)$ time takes to verify win on tree T (rand. var)
  - *undefined* if T losing
  - $E[W(T)]$ is diff. for diff. T (exp. over rand. choices of alg.)
- $W(h) = \max_{T \text{ of height } h} E[W(T)]$ expected time to calc win on any winning tree of height T
- $W(0) = L(0) = 1$

Ditto $L(h)$.

Losing h-tree:

Claim: $L(h) \leq 2W(h-1)$

Proof: For losing, both children win and must eval both.

Winning h-tree:

Claim: $W(h) \leq L(h-1) + \frac{1}{2}W(h-1)$

Proof:

- Case 1: Both children lose $\rightarrow W(h) \leq L(h-1)$
- Case 2: Exactly one loser child
  - first choice loses, stop: time $L(h-1)$
  - first choice wins, eval second: time $W(h-1) + L(h-1)$
  - so $W(h) \leq \frac{1}{2}L(h-1) + \frac{1}{2}(W(h-1) + L(h-1))$
  - so $W(h) \leq L(h-1) + \frac{1}{2}W(h-1)$
- Case 2 dominates

Fact: $W(h) \leq L(h)$

Combining,

- $W(h) \leq L(h-1) + \frac{1}{2}W(h-1) \leq \frac{3}{2}L(h-1) \leq 3W(h-2)$
- so $W(h) = 3^{h/2} = n^{\log 3/2} = n^{0.79}$

[[*Better than det. alg (which was linear) but worse than non-det. lower bound. Can we do better?*]]

The Minmax Principle

[[*Lower-bound technique from game-theory interpretation of alg. design.*]]

Zero-sum 2-player games

- an $n \times m$ payoff matrix A
  - a_{ij} payoff to row player
  - $-a_{ij}$ payoff to column player
- mixed strat. $y = (y_1, \ldots, y_n)$ for row player
- mixed strat. $x = (x_1, \ldots, x_m)$ for column player
- so expected payoff $y^T A x$

Minmax payoff (row):

- if pick y, guaranteed payoff $\geq \min_x y^T A x$
- pick best such y: $\max_y \min_x y^T A x$

Minmax payoff (col):

- if pick x, guaranteed payment $\leq \max_y y^T A x$
- pick best such x: $\min_x \min_y y^T A x$

Claim: (von Neumann): These are equal.

$$\min_x \max_y y^T Ax = \max_y \min_x y^T Ax$$

(called the *value* of the zero-sum game)

Proof: LP duality!

Row player prob.: $\max_y \min_x y^T Ax$

- objective: fixing y, $y^T A$ become linear coeff. for variables x
- constraints: $\sum_{i=1}^n x_i = 1, x \in [0,1]^n$

Note:

- Polytope is hypercube
- Vertices $\{e_i\}$ where e_i unit vector 1 in i'th position
- WLOG, opt is a vertex

$\rightarrow$ best-response is a *pure* strategy!

Row may assume col plays pure strategy:

$$\max_y \min_x y^T Ax = \max_y \min_j (y^T A)_j$$

and

$$\min_x \max_y y^T Ax = \min_x \max_i (Ax)_i$$

so want to prove

$$\min_x \max_i A_i x = \max_y \min_j y^T A_j$$

[[*To write min/max (or ratio) objectives as LPs, separate tasks, i.e., find min subject to max at least something or vice-versa.*]]

Primal:

- objective: $\min t$
- constraints:
  - $t \geq A_i x$, or $t - A_i x \geq 0$ for all $1 \leq i \leq n$
  - $\sum_{j=1}^m x_j = 1$
  - non-negativity

Dual:

- objective: $\max s$
- constraints:
  - $s - y^T A_j \leq 0$ for all $1 \leq j \leq m$
  - $\sum_{i=1}^n y_i = 1$
  - non-negativity

So:

$$\min_x \max_i A_i x = Primal = Dual = \max_y \min_j y^T A_j$$

[[*Complementary slackness shows no positive prob. on sub-opt strategies.*]]

[[*Other proofs, see Bobby's lecture notes, week 2.*]]

Yao's minimax method

Idea: Interpret alg. design as zero-sum game

- row $\equiv$ adversary, strat. $\equiv$ inputs $\mathcal{I}$
- col $\equiv$ alg. designer, strat. $\equiv$ (det.) algs $\mathcal{A}_D$
- payoff $\equiv$ running time
- opt rand. alg. $\equiv$ opt mixed strat. for designer (col)
- worst-case input $\equiv$ corresponding opt pure strat. for adversary (row)

from where

- worst-case expected running time of opt. rand. alg. $\equiv$ value of zero-sum game
- from von Neumann, equals running time of best det. alg. for dist. over inputs

Yao's minimax principle:

Worst case expected runtime of randomized algorithm for any input
EQUALS
best case running time of a deterministic algorithm for worst distribution of inputs.

Claim: (Yao's minimax principle): Let

- $\mathcal{A}_R$, class of rand. alg.
- $\mathcal{A}_D$, class of det. alg.
- $\mathcal{I}$, set of inputs
- $\mathcal{D}$, class of dist. over inputs

$$min_{A \in \mathcal{A}_R} max_{I \in \mathcal{I}} E_p[A(I,p)] =$$
$$max_{\Delta \in \mathcal{D}} min_{A \in \mathcal{A}_D} E_{I \in_R \Delta}[A(I,\Delta)].$$

To lower bound runtime of best rand. alg., show an input distribution with no good deterministic algorithm.

Note: Det. alg. knows dist.

Example: Find-bill: n boxes, one with \$1

Question: Best det. alg.? runtime n

Question: Good rand. alg.? Probe rand. order, runtime $\frac{n+1}{2}$

Claim: This is best possible.

Proof:

1. Choose dist. on input: uniform random box for bill
2. WLOG look at det. alg. that probe each box at most once
3. By symmetry, assume probe order is $1, \ldots, n$
4. $E[A(I,\Delta)] = \sum_{i=1}^{n} i \times \frac{1}{n} = \frac{n+1}{2}$

Lower bound for game-tree

Hard case for det. alg. is when one child win, other lose. Set win/lose prob. at leaves so high prob. that each internal node has one win and one lose child.

Claim: WLOG, det. alg. finishes evaluating one child before other (depth-first pruning alg.).

Proof: By induction on height of tree.

Idea: Use dist. in which each node is W with equal prob. p:

$$p = (1 - p^2) \rightarrow p = \frac{1}{2}(\sqrt{5} - 1)$$

Claim: Every rand. alg. takes time at least $n^{0.69}$.

Proof: Let $T(h)$ be expected # leaves evaluated on trees of height h.

- with prob. $(1 - p)$ eval. one child, else with prob. p eval. both
- $T(h) = (1 - p)T(h - 1) + 2pT(h - 1) = (1 + p)T(h - 1)$
- $T(h) = (1 + p)^h = n^{\log(1+\sqrt{5})/2} = n^{0.694}$

Question: Better bound?

Idea: Use dependent events to ensure always one winning and one losing child at random