Part I

Games Theory and Analyses of Randomized Algorithms

CLASSICAL GAMES THEORY - BASIC CONCEPTS

CLASSICAL GAMES THEORY BRIEFLY

BASIC CONCEPTS of CLASSICAL GAME THEORY

We will consider **games with two players**, Alice and Bob. X and Y will be nonempty sets of their game **(pure) strategies** -X of Alice, Y of Bob. Mappings $p_X : X \times Y \to \mathbf{R}$ and $p_Y : X \times Y \to \mathbf{R}$ will be called **payoff functions** of Alice and Bob. The quadruple (X, Y, p_X, p_Y) will be called a (mathematical) **game**.

A **mixed strategy** will be a probability distribution on pure strategies.

An element $(x, y) \in X \times Y$ is said to be a **Nash equilibrium** of the game (X, Y, p_X, p_Y) iff $p_X(x', y) \leq p_X(x, y)$ for any $x' \in X$, and $p_Y(x, y') \leq p_Y(x, y)$ **for all** $y' \in Y$.

Informally, Nash equilibrium is such a pair of strategies that none of the players gains by changing his/her strategy.

A game is called **zero-sum game** if $p_X(x, y) + p_Y(x, y) = 0$ for all $x \in X$ and $y \in Y$.

ONE of THE BASIC RESULTS

One of the basic result of the classical game theory is that not every two-players zero-sum game has a Nash equilibrium in the set of pure strategies, but there is always a Nash equilibrium if players follow mixed strategies.

POWER Of QUANTUM PHENOMENA

It has been shown, for several zero-sum games, that if one of the players can use quantum tools and thereby **quantum strategies**, then he/she can increase his/her chance to win the game.

This way, from a fair game, in which both players have the same chance to win if only classical computation and communication tools are used, an unfair game can arise, or from an unfair game a fair one.

EXAMPLE - PENNY FLIP GAME

Alice and Bob play with a box and a penny as follows:

- Alice places a penny head up in a box.
- Bob flips or does not flip the coin
- Alice flips or does not flip the coin
- Bob flips or does not flip the coin

After the "game" is over, they open the box and Bob wins if the penny is head up.

It is easy to check that using pure strategies chances to win are $\frac{1}{2}$ for each player and there is no (Nash) equilibrium in the case of pure classical strategies.

However, there is equilibrium if Alice chooses its strategy with probability $\frac{1}{2}$ and Bob chooses each of the four possible strategies with probability $\frac{1}{4}$.

VERSION of PRISONERS' DILEMMA from 1992

Two members of a gang are imprisoned, each in a separate cell, without possibility to communicate. However, police has not enough evidence to convict them on the principal charge and therefore police intends to put both of them for one year to jail on a lesser charge.

Simultaneously police offer both of them so called Faustian bargain. Each prisoner gets a chance either to betray the other one by testifying that he committed the crime, or to cooperate with the other one by remaining silent. Here are payoffs they are offered:

- If both betray, they will get into jail for 2 years.
- If one betrays and second decides to cooperate, then first will get free and second will go to jail for 3 years.
- If both cooperate they will go to jail for 1 year.

What is the best way for them to behave? This game is a model for a variety of real-life situations involving cooperative behaviour. Game was originally framed in 1950 by M. Flood and M. Dresher

PRISONERS' DILEMMA - I.

Two prisoners, Alice and Bob, can use, independently, any of the following two strategies: **to cooperate** or **to defect** (not to cooperate).

The problem is that the payoff function (p_A, p_B), in millions, is a very special one (first (second) value is payoff of Alice (of Bob):

Alice / Bob	C_A	D_A
C_B	$(3,3)$	$(5,0)$
D_B	$(0,5)$	$(1,1)$

What is the best way for Alice and Bob to proceed in order to maximize their payoffs?

PRISONERS' DILEMMA - II.

A strategy s_A is called **dominant** for Alice if for any other strategy s'_A of Alice and s_B of Bob, it holds

$$P_A(s_A, s_B) \geq P_A(s'_A, s_B).$$

Clearly, defection is the dominant strategy of Alice (and also of Bob) in the case of Prisoners Dilemma game.

Prisoners Dilemma game has therefore **dominant-strategy equilibrium**

Alice / Bob	C_A	D_A
C_B	$(3,3)$	$(5,0)$
D_B	$(0,5)$	$(1,1)$

BATTLE of SEX GAME

Alice and Bob have to decide, independently of each other, where to spent the evening.

Alice prefers to go to opera (O), Bob wants to watch TV (T) - tennis.

However, at the same time both of them prefer to be together than to be apart.

Pay-off function is given by the matrix (columns are for Alice) (columns are for Bob)

	O	T
O	(α,β)	(γ,γ)
T	(γ,γ)	(β,α)

where $\alpha > \beta > \gamma$.

What kind of strategy they should choose?

The two Nash equilibria are (O, O) and (T, T), but players are faced with tactics dilemma, because these equilibria bring them different payoffs.

COIN GAME

There are three coins: one fair, with both sides different, and two unfair, one with two heads and one with two tails.

The game proceeds as follows.

- Alice puts coins into a black box and shakes the box.
- Bob picks up one coin.
- Alice wins if coin is unfair, otherwise Bob wins

Clearly, in the classical case, the probability that Alice wins is $\frac{2}{3}$.

FROM GAMES to LOWER BOUNDS for RANDOMIZED ALGORITHMS

Next goal is to present, using zero-sum games theory, a method how to prove lower bounds for the average running time of randomized algorithms.

This techniques can be applied to algorithms that terminate for all inputs and all random choices.

TWO–PERSON ZERO–SUM GAMES – EXAMPLE

A two players zero–sum game is represented by an $n \times m$ payoff–matrix M with all rows and columns summing up to 0.
Payoffs for n possible strategies of Alice are given in rows of M.
Payoffs for m possible strategies of Bob are given in columns of M.

$$M_{i,j}$$

is the amount paid by Bob to Alice if Alice chooses strategy i and Bob's choice is strategy j.
The goal of Alice (Bob) is to maximize (minimize) her payoff.
Example - stone-scissors-paper game

PAYOFF–MATRIX

Bob

Alice

	Scissors	Paper	Stone
Scissors	0	1	-1
Paper	-1	0	1
Stone	1	-1	0

→ Table shows how much Bob has to pay to Alice

Rules: Stone looses to paper and wins sissors.Paper looses to sissors and wins to stone.**Sissors** looses to stone and wins to paper.

STRATEGIES for ZERO-INFORMATION and ZERO-SUM GAMES

(Games with players having no information about their opponents' strategies.)

Observe that if Alice chooses a strategy i, then she is guaranteed a payoff of $\min_j M_{ij}$ regardless of Bob's strategy.

An optimal strategy O_A for Alice is such an i that maximises $\min_j M_{ij}$.

$$O_A = \max_i \min_j M_{ij}$$

denotes therefore the lower bound on the value of the payoff Alice gains (from Bob) when she uses an optimal strategy.

An optimal strategy O_B for Bob is such a j that minimizes $\max_i M_{ij}$. Bob's optimal strategy ensures therefore that his payoff is at least

$$O_B = \min_j \max_i M_{ij}$$

Theorem

$$O_A = \max_i \min_j M_{ij} \leq \min_j \max_i M_{ij} = O_B$$

Often $O_A < O_B$. In our last (scissors-...) example, $-1 = O_A < O_B = +1$.

If $O_B = O_A$ we say that the game has a solution – a specific choice of strategies that leads to this solution.

ϱ and γ are so called optional strategies for Alice and Bob if

$$O_A = O_B = M_{\varrho\gamma}$$

Example of the game which has a solution ($O_A = O_B = 0$)

0	1	2
-1	0	1
-2	-1	0

What happens if a game has no solution ?
There is no clear–cut strategy for any player.
Way out: to use randomized strategies.

Alice chooses strategies according to a probability vector $p = (p_1, \dots, p_n)$; p_i is probability that Alice chooses strategy $s_{A,i}$

Bob chooses strategies according to a probability vector $q = (q_1, \dots, q_n)$; q_j is a probability that Bob chooses strategy $s_{B,j}$.

Payoff is now a random variable – **if p, q are taken as column vectors** then

$$E[\text{payoff}] = p^T Mq = \sum_{i=1}^{n} \sum_{j=1}^{m} p_i M_{ij} q_j$$

Let O_A (O_B) denote the best possible (optimal) lower (upper) bound on the expected payoff of Alice (Bob). Then it holds:

$$O_A = \max_p \min_q p^T Mq \quad O_B = \min_q \max_p p^T Mq$$

Theorem (von Neumann Minimax theorem) For any two–person zero–sum game specified by a payoff matrix M it holds

$$\max_p \min_q p^T Mq = \min_q \max_p p^T Mq$$

Observe that once p is fixed, $\max_p \min_q p^T Mq = \min_q \max_p p^T Mq$ is a linear function and is minimized by setting to 1 the q_j with the smallest coefficient in this linear function.

This has interesting/important implications:

If Bob knows the distribution p used by Alice, then his optimal strategy is a pure strategy.

A similar comment applies in the opposite direction. This leads to a simplified version of the minimax theorem, where e_k denotes a unit vector with 1 at the k-th position and 0 elsewhere.

Theorem (Loomis' Theorem) For any two–persons zero–sum game

$$\max_p \min_j p^T Me_j = \min_q \max_i e_i^T Mq$$

YAO'S TECHNIQUE 1/3

Yao's technique provides an application of the game-theoretic results to the establishment of lower bounds for randomized algorithms.
For a given algorithmic problem $\mathcal{P}$ let us consider the following payoff matrix.

		deterministic algorithms $\mathcal{A}_1$ $\mathcal{A}_2$ $\mathcal{A}_3$
I	c_1	
N	c_2	entries
P	c_3	=
U	c_4	resources
T		(i.e. used computation time)
S		

Bob – a designer choosing good algorithms

Alice – an adversary choosing bad inputs

Pure strategy for Bob corresponds to the choice of a deterministic algorithm.
Optimal pure strategy for Bob corresponds to a choice of an optimal deterministic algorithm.

YAO'S TECHNIQUE 2/3

Let V_B be the worst-case running time of any deterministic algorithm of Bob

Problem: How to interpret mixed strategies ?

A *mixed strategy* for Bob is a probability distribution over (always correct) deterministic algorithms—so it is a Las Vegas randomized algorithm.

An optimal mixed strategy for Bob is an optimal Las Vegas algorithm. **Distributional complexity** of a problem is an expected running time of the best deterministic algorithm for the worst distribution on the inputs.

Loomis theorem implies that distributional complexity equals to the least possible time achievable by any randomized algorithm

Reformulation of von Neumann+Loomis' theorem in the language of algorithms

Corollary Let Π be a problem with a finite set I of input instances and $\mathcal{A}$ be a finite set od deterministic algorithms for Π. For any input $i \in I$ and any algorithm $A \in \mathcal{A}$, let $T(i, A)$ denote computation time of A on input i. For probability distributions p over I and q over $\mathcal{A}$, let i_p denote random input chosen according to p and A_q a random algorithm chosen according to q. Then

$$\max_p \min_q E\left[T(i_p, A_q)\right] = \min_q \max_p E\left[T(i_p, A_q)\right]$$

$$\max_p \min_{A \in \mathcal{A}} E\left[T(i_p, A)\right] = \min_q \max_{i \in I} E\left[T(i, A_q)\right]$$

YAO'S TECHNIQUE 3/3

Consequence:

Theorem(Yao's Minimax Principle) For all distributions p over I and q over $\mathcal{A}$.

$$\min_{A \in \mathcal{A}} \mathbf{E}[T(i_p, A)] \leq \max_{i \in I} \mathbf{E}[T(i, A_q)]$$

Interpretation: Expected running time of the optimal deterministic algorithm for any arbitrarily chosen input distribution p for a problem Π is a lower bound on the expected running time of the optimal (Las Vegas) randomized algorithm for Π.

In other words, to determine a lower bound on the performance of all randomized algorithms for a problem P, derive instead a lower bound for any deterministic algorithm for P when its inputs are drawn from a specific probability distribution (of your choice).

IMPLICATIONS OF YAO'S MINIMAX PRINCIPLE

Interpretation again Expected running time of the optimal deterministic algorithm for an arbitrarily chosen input distribution p for a problem Π is a lower bound on the expected running time of the optimal (Las Vegas) randomized algorithm for Π.

Consequence:
In order to prove a lower bound on the randomized complexity of an algorithmic problem, it suffices to choose *any* probability distribution p on the input and prove a lower bound on the expected running time of deterministic algorithms for that distribution.

The power of this technique lies in

1. the flexibility at the choice of p
2. the reduction of the task to determine lower bounds for randomized algorithms to the task to determine lower bounds for deterministic algorithms.

(It is important to remember that we can expect that the deterministic algorithm "knows" the chosen distribution p.)

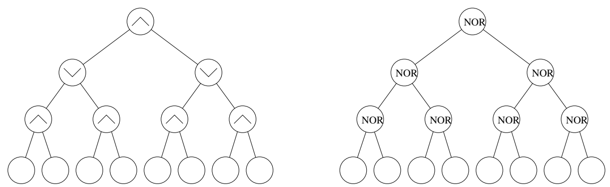
The above discussion holds for Las Vegas algorithms only!

GAMES TREES REVISITED

A randomized algorithm for a game-tree T evaluations can be viewed as a probability distribution over deterministic algorithms for T, because the length of computation and the number of choices at each step are finite.

Instead of AND–OR trees of depth $2k$ we can consider NOR–trees of depth $2k$. Indeed, it holds:

$$(a \vee b) \wedge (c \vee d) \equiv (a \text{ NOR } b)\text{NOR}(c \text{ NOR } d)$$



Note: It's important to distinguish between:

- the expected running time of the randomized algorithm with a fixed input (where probability is considered over all random choices made by the algorithm)
- and
- the expected running time of the deterministic algorithm when proving the lower bound (the average time is taken over all random input instances).

LOWER BOUND FOR GAME TREE EVALUATION - I

Assume now that each leaf of a NOR-tree is set up to have value 1 with probability $p = \frac{3-\sqrt{5}}{2}$ (observe that $(1-p)^2 = p$ for such a p).

Observe that if inputs of a NOR-gate have value 1 with probability p then its output value is also 1 with probability $(1-p)(1-p) = p$.

Consider now only *depth–first pruning algorithms* for tree evaluation.
(They are such depth–first algorithms that make use of the knowledge that subtrees that provide no additional useful information can be "pruned away".)

Of importance for the overall analysis is the following technical lemma:

Lemma Let T be a NOR–tree each leaf of which is set to 1 with a fixed probability. Let $W(T)$ denote the minimum, over all deterministic algorithms, of the expected number of steps to evaluate T. Then there is a depth–first pruning algorithm whose expected number of steps to evaluate T is $W(T)$.

The last lemma tells us that for the purposes of our lower bound, we may restrict our attention to the depth–first pruning algorithms.

LOWER BOUND FOR GAME TREE EVALUATION - II

For a depth–first pruning algorithm evaluating a NOR–tree, let $W(h)$ be the expected number of leaves the algorithm inspects in determining the value of a node at distance h from the leaves.

It holds

$$W(h) = pW(h-1) + (1-p)2W(h-1) = (2-p)W(h-1)$$

because with the probability $1-p$ the first subtree produces 0 and therefore also the second tree has to be evaluated. If $h = \lg_2 n$, then the above recursion has a solution

$$W(h) \geq n^{0.694}.$$

This implies:

Theorem The expected running time of any randomized algorithm that always evaluates an instance of T_k correctly is at least $n^{0.694}$, where $n = 2^{2k}$ is the number of leaves.

The upper bound for randomized game tree evaluation algorithms already shown, at the beginning of this chapter was $n^{0.79}$, what is more than the lower bound n^{694} just shown.

It was therefore natural to ask what does the previous theorem really says?

For example, is our lower bound technique weak? ?

No, the above result just says that in order to get a better lower bound another probability distribution on inputs may be needed.

RECENT RESULTS

Two recent results put more light on the Game tree evaluation problem.

- It has been shown that for our game tree evaluation problem the upper bound presented at the beginning is the best possible and therefore that $\theta(n^{0.79})$ is indeed the classical (query) complexity of the problem.
- It has also been shown, by Farhi et al. (2009), that the upper bound for the case quantum computation tools can be used is $O(n^{0.5})$.