Chapter 12

Zero-Sum Games and the Minimax Theorem

Pure conflict—what one wins, the other loses—is where game theory began, and where its theory is most complete. Zero-sum games also connect game theory to optimization (linear programming) and to machine learning (adversarial training, GANs), making this chapter a bridge between Parts II and III.

12.1 Maxmin reasoning

Definition 12.1: Zero-sum game; value

A two-player game is *zero-sum* if $u_2 = -u_1$; write $u = u_1$ and think of player 1 (the *maximizer*) and player 2 (the *minimizer*). Player 1's *security level* (maxmin) and player 2's (minmax) are

$$\underline{v} = \max_{\sigma_1} \min_{\sigma_2} u(\sigma_1, \sigma_2), \qquad \overline{v} = \min_{\sigma_2} \max_{\sigma_1} u(\sigma_1, \sigma_2).$$

Always $\underline{v} \le \overline{v}$. When they coincide, the common number is the *value* of the game, and the optimizers are the players' *optimal strategies*.

In pure strategies the gap can be strict (Matching Pennies: $\underline{v} = -1 < 1 = \overline{v}$). Von Neumann's theorem says randomization closes it:

Theorem 12.1: Minimax Theorem (von Neumann 1928)

Every finite two-player zero-sum game has a value in mixed strategies:

$$\max_{\sigma_1 \in \Delta(A_1)} \min_{\sigma_2 \in \Delta(A_2)} u(\sigma_1, \sigma_2) \;=\; \min_{\sigma_2 \in \Delta(A_2)} \max_{\sigma_1 \in \Delta(A_1)} u(\sigma_1, \sigma_2) \;=\; v.$$

Moreover (σ_1^*, σ_2^*) is a Nash equilibrium if and only if each σ_i^* is an optimal (security-maximizing) strategy; all equilibria yield the same payoff v; and equilibrium strategies are *interchangeable*.

Key Idea: Why zero-sum games are "solved"

In zero-sum games the troubles of general games vanish: equilibrium payoffs are unique (no selection problem), equilibrium strategies are interchangeable (no coordination problem),

and playing your maxmin strategy is unconditionally safe (it guarantees v regardless of the opponent). Game-theoretic advice is unambiguous: play optimally and you cannot be exploited. This is precisely the property modern superhuman poker programs exploit.

Example 12.1: Solving a 2×2 game

u	L	R
T	3	-1
B	0	2

(Entries are payments from 2 to 1.) No saddle point in pure strategies ($\underline{v}_{\text{pure}} = 0$, $\overline{v}_{\text{pure}} = 2$). Mix: player 1 plays T with p making 2 indifferent: $3p = -p + 2(1-p) \Rightarrow p = \frac{1}{3}$; player 2 plays L with q making 1 indifferent: $3q - (1-q) = 2(1-q) \Rightarrow q = \frac{1}{2}$. Value $v = 3 \cdot \frac{1}{3} \cdot \frac{1}{2} \cdot 2 = 1$ (compute: $u = 3pq - p(1-q) + 2(1-p)(1-q) = 1$).

12.2 Minimax and linear programming duality

The Minimax Theorem is linear programming duality in disguise. Player 1 seeks a mixture $x \in \Delta(A_1)$ maximizing the worst-case column payoff:

$$\text{(P)} \quad \max_{x, v} \; v \quad \text{s.t.} \quad \sum_i x_i \, u(i,j) \geq v \;\; \forall j, \quad \textstyle\sum_i x_i = 1, \; x \geq 0,$$

and player 2's problem is exactly the LP dual; strong duality delivers $\underline{v} = \overline{v}$. Conversely, LP duality can be derived from the Minimax Theorem: the two results are equivalent. Historically, Dantzig and von Neumann discovered this connection in a famous 1947 meeting; computationally, it means zero-sum games are solvable in *polynomial time*—in sharp contrast to general-sum games (Chapter 17).

12.3 Solving games by fictitious play and no-regret learning

Optimal strategies can be *learned* by iterative self-play, an idea with modern resonance.

Definition 12.2: Fictitious play (Brown 1951)

Each round, every player best-responds to the *empirical frequency* of the opponent's past actions.

Theorem 12.2: Robinson (1951)

In finite two-player zero-sum games, the empirical frequencies of fictitious play converge to the set of optimal (maxmin) strategies, and the average payoff converges to the value.

Modern algorithms strengthen this: if both players use any *no-regret* learning rule (e.g., multiplicative weights), time-averaged play converges to optimal strategies at rate $O(1/\sqrt{T})$—the engine behind poker solvers (counterfactual regret minimization) and adversarial training of neural networks (Chapter 17).

12.4 Applications of pure conflict

Example 12.2: Colonel Blotto

Two colonels allocate armies across k battlefields; each field is won by the larger force, and the player winning more fields wins. Blotto games model resource allocation in elections (campaign spending across swing states), R&D races, cybersecurity, and sports scheduling. Optimal strategies are highly random even though nothing in the environment is random—unpredictability across fronts is the essence of the solution. (Closed forms are known only for special cases; the continuous symmetric case involves uniform marginals on $[0, 2/k]$.)

Example 12.3: Inspection games

A tax authority audits with probability p (cost c); a taxpayer evades or complies. Evasion caught costs a fine F; uncaught evasion saves tax T. The equilibrium is mixed: the authority audits just often enough to make the taxpayer indifferent ($p^* = T/F$), and evasion occurs just often enough to justify the audit budget. Comparative static worth remembering: raising the fine F reduces *auditing* in equilibrium, not evasion incentives at the margin—the indifference logic of Chapter 5 at work in law enforcement design.

Historical Note: Poker, solved

Heads-up limit Texas hold'em was *essentially solved* in 2015 (Bowling et al., *Science*): a strategy computed by counterfactual regret minimization is so close to the game's maxmin optimum that a human lifetime of play cannot distinguish it from perfection. No-limit variants (Libratus 2017, Pluribus 2019) followed via approximation. The underlying mathematics is exactly this chapter's: zero-sum extensive-form games, behavior strategies, and convergence of no-regret dynamics to minimax equilibrium.

Exercises

Exercise 12.1: Saddle points

u	L	M	R
T	4	2	3
C	1	0	5
B	3	1	2

Find $\underline{v}$ and $\overline{v}$ in pure strategies. Does the game have a saddle point? If so, identify all of them; if not, solve for the mixed-strategy value.

Exercise 12.2: Guess-the-larger-number

Each player secretly writes 1, 2, or 3. If the numbers are equal, the payoff is 0. If they differ by 1, the player with the *larger* number pays 2 to the smaller. If they differ by 2, the player with the larger number receives 1 from the smaller. Write the payoff matrix, verify the game is symmetric with value 0, and find an optimal strategy.

Exercise 12.3: LP formulation

Write down explicitly the linear program (P) for Matching Pennies and its dual, solve both, and confirm the value 0 and optimal strategies $(\frac{1}{2}, \frac{1}{2})$.

Exercise 12.4: Fictitious play by hand

Run six rounds of fictitious play on Matching Pennies starting from both players playing H in round 1 (break best-response ties toward H). Record empirical frequencies and observe the cycling-but-converging pattern.