PART I INTRODUCTION

LECTURE 1

Noncooperative Games

This lecture uses several examples to introduce the key principles of noncooperative game theory.

1.1 Elements of a Game
1.2 Cooperative vs. Noncooperative Games: Rope-Pulling
1.3 Robust Designs: Resistive Circuit
1.4 Mixed Policies: Network Routing
1.5 Nash Equilibrium
1.6 Practice Exercise

1.1 Elements of a Game

To characterize a game one needs to specify several items:

- The *players* are the agents that make decisions.
- The *rules* define the actions allowed by the players and their effects.
- The *information structure* specifies what each player knows before making each decision.

  Chess is a *full-information* game because the current state of the game is fully known to both players as they make their decisions. In contrast, Poker is a *partial-information* game.
- The *objective* specifies the goal of each player.

For a mathematical solution to a game, one further needs to make assumptions on the *player's rationality*, regarding questions such as:

- Will the players always pursue their best interests to fulfill their objectives? [YES]
- Will the players form coalitions? [NO]
- Will the players trust each other? [NO]

The answers in square brackets characterize what are usually called *noncooperative games*, and will be implicitly assumed throughout this course. This will be further discussed shortly.

Note 1 (Human players). Studying noncooperative solutions for games played by humans reveals some lack of faith in human nature, which has certainly not prevented economists (and engineers) from doing so. However, when pursuing this approach one should not be overly surprised by finding solutions of "questionable ethics." Actually, one of the greatest contributions of noncooperative game theory is that it allows one to find problematic solutions to games and often indicates how to "fix" the games so that these solutions disappear. This type of approach falls under the heading of *mechanism design.*

In many problems, one or more players are modeling decision processes not affected by human reason, in which case one can safely pursue noncooperative solutions without questioning their ethical foundation. Robust engineering designs and evolutionary biology are good examples of this. □

1.2 Cooperative vs. Noncooperative Games: Rope-Pulling

We use the rope-pulling game to discuss the motivation and implications of assuming a noncooperative framework. This game is depicted schematically in Figure 1.1.

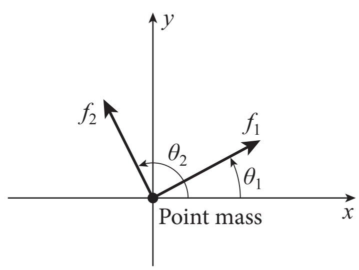

Figure 1.1 Rope-pulling game.

Rules. Two players push a point mass by exerting on it forces f_1 and f_2. Both players exert forces with the same magnitude ($|f_1| = |f_2|$), but they pull in different directions $\theta_1(t)$ and $\theta_2(t)$. The game is played for 1 second.

Note. $\theta_1(t)$ and $\theta_2(t)$ correspond to the decisions made by the players.

Note. The equations in (1.1) effectively encode the rules of the game, as they determine how the player's decisions affect the outcome of the game.

Initially the mass is at rest and, for simplicity, we assume unit forces and a unit mass. According to Newton's law, the point mass moves according to

$$\ddot{x} = \cos\theta_1(t) + \cos\theta_2(t), \qquad \dot{x}(0) = 0, \qquad x(0) = 0 \tag{1.1a}$$

$$\ddot{y} = \sin\theta_1(t) + \sin\theta_2(t), \qquad \dot{y}(0) = 0, \qquad y(0) = 0. \tag{1.1b}$$

1.2.1 Zero-Sum Rope-Pulling Game

Consider the following objective for the rope-pulling game:

Objective (zero-sum). Player P_1 wants to *maximize* $x(1)$, whereas player P_2 wants to *minimize* $x(1)$.

Notation. This is called a *zero-sum* game since players have opposite objectives. One could also imagine that P_1 wants to *maximize* $x(1)$, whereas P_2 wants to *maximize* $-x(1)$. According to this view the two objectives add up to zero.

Solution. We claim that the "optimal" solution for this game is given by

$$\mathsf{P}_1\colon \theta_1(t) = 0, \quad \forall t \in [0, 1], \mathsf{P}_2\colon \theta_2(t) = \pi, \quad \forall t \in [0, 1], \tag{1.2}$$

which results in no motion ($\ddot{x} = \ddot{y} = 0$), leading to $x(1) = y(1) = 0$ [cf. Figure 1.2(a)].

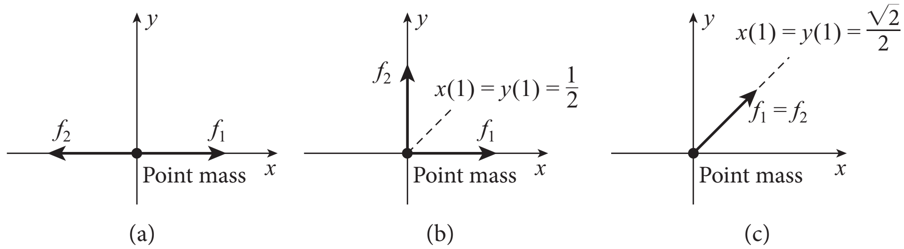

Figure 1.2 Solutions to the rope-pulling game. (a) Solution (1.2); (b) Solution (1.3); (c) Solution (1.4).

One could wonder whether it is reasonable to pull at all, given that the mass will not move. Perhaps the optimal solution is not to push at all. This is not the case for two reasons

1. Not pushing is not allowed by the rules of the game, which call for each player to exert a force of precisely one Newton, as per (1.1).
2. Even if not pulling was an option, it would be a dangerous choice for the player that decided to follow this line of action, as the other player could take advantage of the situation.

   This of course presumes that we are talking about noncooperative games for which players do not trust each other and do not form coalitions.

For now we do not justify why (1.2) is the optimal solution for the objective given above. Instead, we do this for a modified objective, for which the solution is less intuitive.

1.2.2 Non-Zero-Sum Rope-Pulling Game

Consider now a version of the game with precisely the same rules, but a modified objective:

Attention! This is no longer a zero-sum game.

Objective (non-zero-sum). Player P_1 wants to *maximize* $x(1)$, whereas player P_2 wants to *maximize* $y(1)$.

Notation. In games *a solution* is generally a set of policy, one for each player, that jointly satisfy some optimality condition.

Solution (Nash). We claim that the "optimal" solution for this game is given by

$$\mathsf{P}_1\colon \theta_1(t) = 0, \quad \forall t \in [0, 1], \qquad \mathsf{P}_2\colon \theta_2(t) = \frac{\pi}{2}, \quad \forall t \in [0, 1] \tag{1.3}$$

which leads to constant accelerations $\ddot{x} = \ddot{y} = 1$ and therefore $x(1) = y(1) = \frac{1}{2}$ [cf. Figure 1.2(b)].

This solution has two important properties:

P1.1 Suppose that player P_1 follows the course of action $\theta_1(t) = 0$ throughout the whole time period and therefore

$$\ddot{x} = 1 + \cos\theta_2(t), \quad \ddot{y} = \sin\theta_2(t), \quad \forall t \in [0, 1].$$

In this case, the best course of action for P_2 so as to maximize $y(1)$ is precisely to choose

$$\theta_2(t) = \frac{\pi}{2}, \quad \forall t \in [0, 1] \quad \Rightarrow \quad \ddot{y}(t) = 1, \quad \forall t \in [0, 1].$$

Moreover, any deviation from this will necessarily lead to a smaller value of $y(1)$. In this sense, once P_1 decides to stick to their part of the solution in (1.3), a rational P_2 must necessarily follow their policy in (1.3).

P1.2 Conversely, suppose that player P_2 follows the course of action $\theta_2(t) = \frac{\pi}{2}$ throughout the whole time period and therefore

$$\ddot{x} = \cos\theta_1(t), \quad \ddot{y} = \sin\theta_1(t) + 1, \quad \forall t \in [0, 1].$$

In this case, the best course of action for P_1 so as to maximize $x(1)$ is precisely to choose

$$\theta_1(t) = 0, \quad \forall t \in [0, 1] \quad \Rightarrow \quad \ddot{x}(t) = 1, \quad \forall t \in [0, 1].$$

Moreover, any deviation from this will necessarily lead to a smaller value of $x(1)$. Also now, once P_2 decides to stick to their part of the solution in (1.3), a rational P_1 must necessarily follow their policy in (1.3).

A pair of policies that satisfy the above properties is called a *Nash equilibrium solution.* The key feature of Nash equilibrium is that it is *stable,* in the sense that if the two players start playing at the Nash equilibrium, none of the players gains from deviating from these policies.

This solution also satisfies the following additional properties:

P1.3 Suppose that player P_1 follows the course of action $\theta_1(t) = 0$ throughout the whole time period. Then, regardless of what P_2 does, P_1 is guaranteed to achieve $x(1) \geq 0$.

Note. Even if P_2 pulls against P_1, which is not very rational but possible.

Moreover, no other policy for P_1 can guarantee a larger value for $x(1)$ regardless of what P_2 does.

P1.4 Suppose that player P_2 follows the course of action $\theta_2(t) = \frac{\pi}{2}$ throughout the whole time period. Then, regardless of what P_1 does, P_2 is guaranteed to achieve $y(1) \geq 0$.

Moreover, no other policy for P_2 can guarantee a larger value for $y(1)$ regardless of what P_1 does.

In view of this, the two policies are also called *security policies* for the corresponding player.

Note. We shall see later that for zero-sum games Nash policies are always security policies (cf. Lecture 3), but this is not always the case for non-zero-sum games such as this one.

The solution in (1.3) is therefore "interesting" in two distinct senses: these policies form a Nash equilibrium (per P1.1–P1.2) and they are also security policies (per P1.3–P1.4).

Solution (cooperative). It is also worth considering the following alternative solution

$$\mathsf{P}_1\colon \theta_1(t) = \frac{\pi}{4}, \quad \forall t \in [0, 1], \qquad \mathsf{P}_2\colon \theta_2(t) = \frac{\pi}{4}, \quad \forall t \in [0, 1] \tag{1.4}$$

which leads to constant accelerations $\ddot{x} = \ddot{y} = \sqrt{2}$ and therefore

$$x(1) = y(1) = \frac{\sqrt{2}}{2} > \frac{1}{2}$$

[cf. Figure 1.2(c)]. This policy is interesting because both players do strictly better than with the Nash policies in (1.3). However, this is not a Nash policy because suppose that P_1 decides to follow this course of action $\theta_1(t) = \frac{\pi}{4}$ throughout the whole time period and therefore

$$\ddot{x} = \frac{\sqrt{2}}{2} + \cos\theta_2(t), \quad \ddot{y} = \frac{\sqrt{2}}{2} + \sin\theta_2(t), \quad \forall t \in [0, 1].$$

In this case, the best course of action for P_2 to maximize $y(1)$ is to choose

$$\theta_2(t) = \frac{\pi}{2}, \quad \forall t \in [0, 1],$$

instead of the assigned policy in (1.4), because this will lead to $\ddot{y} = \frac{\sqrt{2}}{2} + 1$ and

$$y(1) = \frac{\sqrt{2} + 2}{4} > \frac{\sqrt{2}}{2}.$$

Unfortunately for P_1, this also leads to $\ddot{x} = \frac{\sqrt{2}}{2}$ and

$$x(1) = \frac{\sqrt{2}}{4} < \frac{1}{2} < \frac{\sqrt{2}}{2}.$$

In this sense, (1.4) is a dangerous choice for P_1 because a greedy P_2 will get P_1 even worse than with the Nash policy (1.3) that led to $x(1) = \frac{1}{2}$. For precisely the same reasons, (1.4) can also be a dangerous choice for P_2. In view of this, (1.4) is *not a Nash equilibrium solution*, in spite of the fact that both players can do better than with the Nash solution (1.3).

Note 2. The solution (1.4) is called *Pareto-optimal*. ▷ p. 7

Solutions such as (1.4) are the subject of *cooperative game theory*, in which one allows negotiation between players to reach a mutually beneficial solution. However, this requires faith/trust among the players. As noted above, solutions arising from cooperation are not robust with respect to cheating by one of the players.

For certain classes of games, noncooperative solutions coincide with cooperative solutions, which means that by blindingly pursuing selfish interests one actually helps other players in achieving their goals. Such games are highly desirable from a social perspective and deserve special study. It turns out that it is often possible to "reshape" the reward structure of a game to make this happen. In economics (and engineering) this is often achieved through pricing, taxation, or other incentives/deterrents and goes under the heading of *Mechanism Design.*

Note. We shall see an example in Lecture 12, where a network administrator can minimize the total interference between "selfish" wireless users by carefully charging their use of the shared medium.

Note 2 (Pareto-optimal solution). A solution like (1.4) is called *Pareto-optimal* because it is not possible to further improve the gain of one player without reducing the gain of the other. The problem of finding Pareto-optimal solutions can typically be reduced to a single-criteria constrained optimization. For the non-zero-sum rope-pulling game, all Pareto-optimal solutions can be found by solving the following constrained optimization problem

$$\max_{\theta_1(t),\theta_2(t)} \{x(1) : y(1) \geq \alpha\}$$

with $\alpha \in \mathbb{R}$. Pareto-optimal solutions are generally not unique and different values of α result in different Pareto-optimal solutions. □

Note. In some cases, all Pareto-optimal solutions can also be found by solving unconstrained optimization problems. This is the case for this example, where all Pareto-optimal solutions can also be found by solving $\max_{\theta_1(t),\theta_2(t)} \beta x(1) + (1-\beta)y(1)$. The different solutions are found by picking different values for β in the interval [0, 1].

1.3 Robust Designs: Resistive Circuit

In many engineering applications, game theory is used as tool to solve design problems that do not start as a game. In such cases, the first step is to take the original design problem and "discover" a game theoretical formulation that leads to a desirable solution. In these games, one of the players is often the system designer and the opponent is a fictitious entity that tries to challenge the choices of the designer. Prototypical examples of this scenario are discussed in the example below and the one in Section 1.4.

Consider the resistive circuit in Figure 1.3 and suppose that our goal is to pick a resistor so that the current $i = 1/R$ is as close as possible to 1. The challenge is that when we purchase a resistor with nominal resistance equal to R_{nom}, the actual resistance may exhibit an error up to 10%, i.e.,

$$R = (1 + \delta)R_{\text{nom}},$$

where δ is an unknown scalar in the interval [−0.1, 0.1].

This a called a *robust design problem* and is often formalized as a game between the circuit designer and an unforgiving nature that does her best to foil the designer's objective:

Note. Robust designs generally lead to noncooperative zero-sum games, such as this one. Cooperative solutions make no sense in robust design problems.

- P_1 is the circuit designer and picks the nominal resistance R_{nom} to *minimize* the current error

$$e = \left|\frac{1}{R} - 1\right| = \left|\frac{1}{(1+\delta)R_{\text{nom}}} - 1\right|.$$

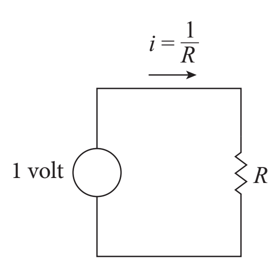

Figure 1.3 Resistive circuit game.

- P_2 is "nature" and picks the value of $\delta \in [-0.1, 0.1]$ to *maximize* the same current error e.

Solution (security). A possible solution for this game is given by

$$\mathsf{P}_1\colon R_{\text{nom}} = \frac{100}{99}, \qquad \mathsf{P}_2\colon \delta = 0.1 \tag{1.5}$$

which leads to a current error equal to

$$e(R_{\text{nom}}, \delta) = \left| \frac{1}{(1+\delta)R_{\text{nom}}} - 1 \right| = \left| \frac{99}{110} - 1 \right| = \left| \frac{99-110}{110} \right| = 0.1.$$

This solution exhibits the following properties:

P1.5 Once player P_1 picks $R_{\text{nom}} = \frac{100}{99}$, the error e will be maximized for $\delta = 0.1$ and is exactly $e = 0.1$.

P1.6 However, if player P_2 picks $\delta = 0.1$, then player P_1 can pick

$$\frac{1}{(1+\delta)R_{\text{nom}}} = 1 \;\Leftrightarrow\; R_{\text{nom}} = \frac{1}{1+\delta} = \frac{1}{1.1} = \frac{100}{110}$$

and get the error exactly equal to zero.

We thus conclude that (1.5) is *not* a safe choice for P_2 and consequently *not a Nash equilibrium*. However, (1.5) is safe for P_1 and $R_{\text{nom}} = \frac{100}{99}$ is therefore a *security policy* for the circuit designer.

Note. It turns out that, as defined, this game does not have any *Nash equilibrium*. It does however have a generalized form of Nash equilibrium that we will encounter in Lecture 4. This new form of equilibrium will allow us to "fix" P_2's policy and we shall see that P_1's choice for the resistor in (1.5) is already optimal.

1.4 Mixed Policies: Network Routing

Consider the computer network in Figure 1.4 and suppose that our goal is to send data packets from source to destination. In the typical formulation of this problem, one selects a path that minimizes the number of hops transversed by the packets. However, this formulation does not explore all possible paths and tends to create hot spots.

An alternative formulation considers two players:

- P_1 is the *router* that selects the path for the packets
- P_2 is an *attacker* that selects a link to be disabled.

The two players make their decisions independently and without knowing the choice of the other player.

Note. As in the example in Section 1.3, the second player is purely fictitious and, in this game, its role is to drive P_1 away from routing decisions that would lead to hot spots. The formulation discussed here is not unique and one can imagine other game theoretical formulations that achieve a similar goal.

Objective. Player P_1 wants to *maximize* the probability that a packet reaches its destination, whereas P_2 wants to *minimize* this probability.

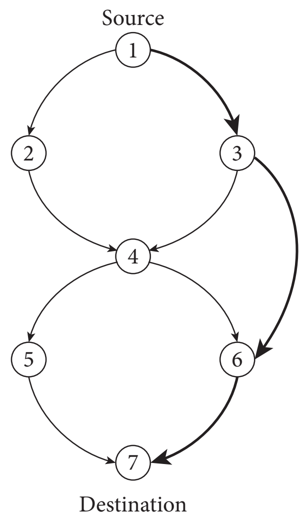

Figure 1.4 Network routing game. The 3-hop shortest path from source to destination is highlighted.

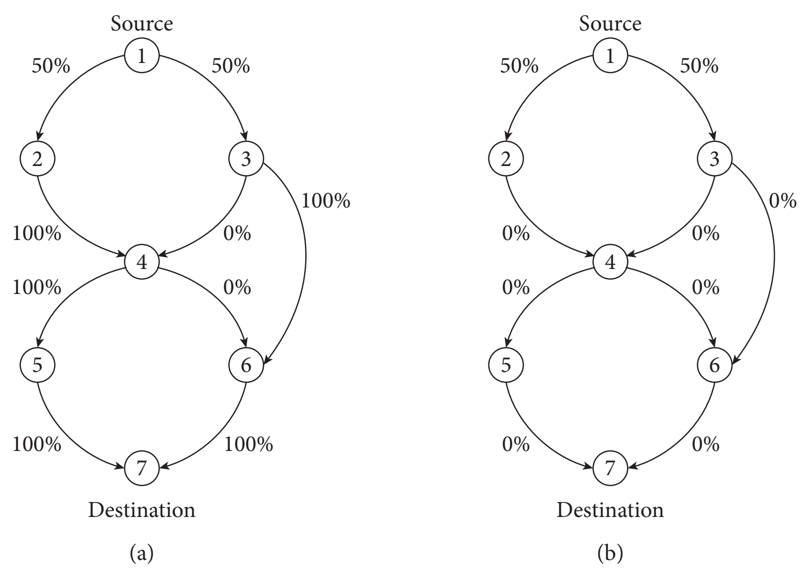

Figure 1.5 Saddle-point solution to the routing game. (a) Stochastic routing policy. The percentages by each link indicate how traffic should be distributed among the outgoing links of a node. (b) Stochastic attack policy. The percentages by each link indicate the probability by which the attacker will disable that link.

Note. See Exercise 1.1. ▹ p. 11

Solution. Figure 1.5 shows a *saddle-point solution* to this game for which 50% of the packets will reach their destination. This solution exhibits the two key properties:

P1.7 Once player P_1 picks the routing policy, P_2's attack policy is the best response from this player's perspective.

P1.8 Once player P_2 picks the routing policy, P_1's attack policy is the best response from this player's perspective.

Note. Here, "no worse" may mean "larger than" or "smaller than," depending on the player.

These are also *security policies* because each policy guarantees for that player a percentage of packet arrivals no worse than 50%. Moreover, no other policies can lead to a guaranteed better percentage of packet arrivals.

Notation. By contrast, policies that do not require randomization are called *pure policies* (cf. Lecture 4).

The policies in Figure 1.5 are *mixed policies* because they call for each player to randomize among several alternatives with "carefully" chosen probabilities. For this particular game, there are no Nash equilibrium that do not involve some form of randomization.

1.5 Nash Equilibrium

Notation. In zero-sum games, Nash equilibrium solutions are called *saddle-point solutions* (cf. Lecture 3).

The previous examples were used to illustrate the concept of *Nash equilibrium*, for which we now provide a "meta definition":

Notation. We often refer to C1.1–C1.2 as the *Nash equilibrium conditions*.

Attention! Condition C1.1 does not require π_2 to be strictly better than all the other policies, just no worse. Similarly for π_1 in C1.2.

Definition 1.1 (Nash Equilibrium). Consider a game with two Players P_1, P_2. A pair of policies (π_1, π_2) is said to be a *Nash equilibrium* if the following two conditions hold:

C1.1 If P_1 uses the policy π_1, then there is no admissible policy for P_2 that does strictly better than π_2.

C1.2 If P_2 uses the policy π_2, then there is no admissible policy for P_1 that does strictly better than π_1.

We call this a "meta definition" because it leaves open several issues that need to be resolved in the context of specific games:

Note. As we progress, we will find several notions of Nash equilibrium that fall under this general "meta definition," each corresponding to different answers to these questions.

1. What exactly is a policy?
2. What is the set of admissible policies against which π_1 and π_2 must be compared?
3. What is meant by a policy doing "strictly better" than another?

As mentioned before, the key feature of a Nash equilibrium is that it is *stable,* in the sense that if P_1 and P_2 start playing at the Nash equilibrium (π_1, π_2), none of the players gains from an unilateral deviating from these policies.

Attention! The definition of Nash equilibrium does not preclude the existence of *multiple Nash equilibria* for the same game. In fact we will find examples of that shortly (e.g., in Lecture 2).

Moreover, there are games for which there are *no Nash equilibria*.

1.6 Practice Exercise

1.1. Find other saddle-point solutions to the network routing game introduced in Section 1.4.

Solution to Exercise 1.1. Figure 1.6 shows another saddle-point solution that also satisfies the Nash equilibrium conditions C1.1–C1.2 in the (meta) Definition 1.1.

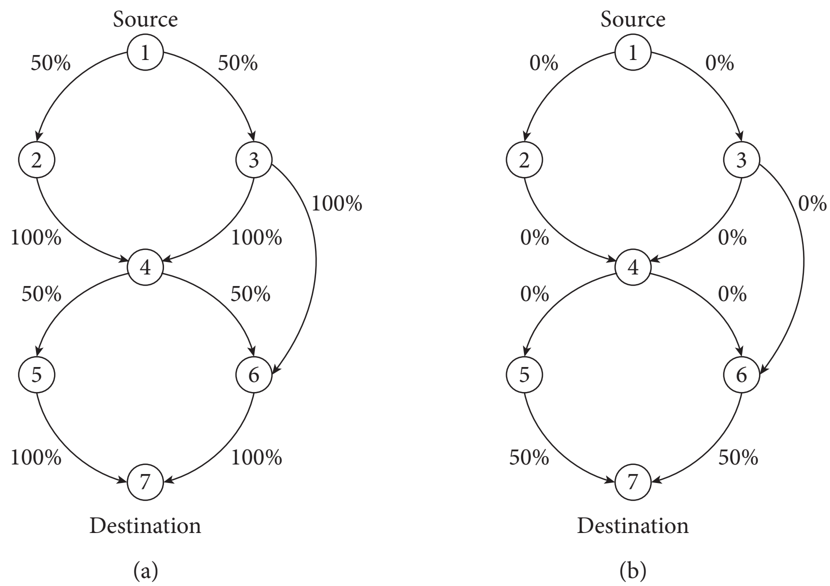

Figure 1.6 Saddle-point solution to the routing game. (a) Stochastic routing policy. The percentages by each link indicate how traffic should be distributed among the outgoing links of a node. (b) Stochastic attack policy. The percentages by each link indicate the probability by which the attacker will disable that link.