DAVID FRIDOVICH-KEIL

SMOOTH GAME THEORY

DFRIDOVI.GITHUB.IO

First printing, October 2024

Contents

Dedicated to all of my collaborators.

Author's Foreword

This monograph has its origins in my graduate seminar at the University of Texas at Austin, titled "Game-Theoretic Modeling of Multi-Agent Systems." Currently, it is targeted at early-year graduate students and aims to provide a concise introduction to the fundamentals of game theory; this monograph is *not* a substitute for other excellent references.[1] Over time, I hope that it will expand and mature to provide a thorough introduction to perhaps lesser-known corners of game theory which underlie modern advances in the field.

[1] Tamer Başar and Geert Jan Olsder. *Dynamic noncooperative game theory.* SIAM, 1998

As the title suggests, we shall focus our attention upon games which are "smooth," by which we indicate the existence of useful derivative information. Adopting this perspective allows us to frame many of the key ideas in game theory in terms of coupled, generally nonlinear and constrained optimization problems. Again, numerous excellent references[2] for optimization exist, and this monograph is not intended to be a substitute for any of them.

[2] Jorge Nocedal and Stephen J Wright. *Numerical optimization.* Springer, 1999

Why (Smooth) Game Theory?

Game theory is the language we use to describe interdependent decision-making problems. For example, suppose that one actor (say, the University), wants the other (say, a certain junior professor) to work very hard and publish lots of papers, when he would prefer to spend most of the day sitting at the café. These preferences *conflict*, and therefore we should expect each actor to choose a *strategy* which in some way accounts for the presence of the other.

Not all games are games of conflict. For example, if students in a class wish to raise the average grade, they can *collude* with one another to improve their collective performance. However, there might be a large number of potential *coalitions* of like-minded students, and depending upon which students collude with which other students, overall performance may differ.

Time is also an essential ingredient in many games. For example, in chess, each player knows that they will get to move in the future. *Dynamic* games of this type are of particular interest, because as we shall see, they allow each player a far richer set of strategies.

Traditionally, most introductions to game theory are framed in terms of finite problems, in which each player has only a finite number of actions available. While this finiteness may provide a convenient theoretical framing for many core concepts, it dramatically restricts our ability to construct efficient numerical algorithms. Imagine a single-player setting, for example. If a single decision-maker has only a finite number of potential actions, it cannot easily infer the performance of one action from that of another; in the worst case and without any additional structure, one must simply try every available action in order to identify the best one. In multi-player settings, this gets tiresome very quickly.

Therefore, we study "smooth" games because they afford more computationally-tractable algorithms. In a smooth game, as we shall see, not only does each player have uncountably-many actions available, but one can also compute derivatives of each player's objective (and potentially, constraints) with respect to those actions. These derivatives will allow us to construct efficient algorithms which solve—or approximate solutions to—these games.

Conventions

Although it is common to discuss players maximizing utility, we shall take the engineering perspective and frame matters in terms of minimizing costs. It is also common these days to refer to "agents" and "policies." Here, we will again adopt a more traditional mindset and consider "players" and "strategies."

Taxonomy

One may categorize games along several axes:

- Finite/infinite—A game is "finite" if players have only finitely many actions available. It is "infinite" if the actions available to each player form a continuum (e.g., lie in $\mathbb{R}^n$).

- Static/dynamic/differential—A game is "static" if it is played at a single instant in time. It is "dynamic" if play continues over a period of time. We call a game "differential" if it is played in continuous time. "Dynamic" and "differential" games are typically characterized by an underlying state, which contains any information needed to describe the future evolution of the game based upon players' actions.

- Zero/general sum—Games in which players' objectives add to a constant (without loss of generality, zero), are called "zero-sum" and model perfectly adversarial problems. Games with arbitrary player objectives are called "general-sum."

- Unconstrained/constrained—Many games include constraints on players' actions or on the game state. Such constraints may be borne by any subset of the players and need not be the concern of all players jointly.

- Pure/mixed strategies—"Pure" strategies are deterministic. However, foundational results in game theory show that not all games possess equilibria when players are restricted to pure strategies. Such games are often relaxed to include "mixed," or stochastic, strategies.

In this monograph, we shall consider many of these variants. However, as discussed below, we ultimately wish to provide readers with both theoretical and algorithmic grounding for the case of "infinite, dynamic, general-sum, constrained" games played primarily in "pure" strategies.

Organization

Most resources on game theory begin with finite, static, pure strategy games and discuss fundamental solution concepts in that limited setting. We shall do the same; however, in contrast to existing resources which then discuss mixed strategies, we will next discuss infinite static games and the role of constraints. This will lead to a broader discussion of mixed strategies in both finite and infinite static games.

Building upon these ideas, we will introduce the fundamental ideas behind dynamic games, both in finite and infinite cases. Discussion will revolve around the "information structure" of these games, and the relationship between that structure and algorithms we may deploy to find equilibria.

Static, Finite, Pure Strategy Games

We begin by studying static, finite games played in pure strategies, and introduce the following key ideas:

- Normal form—a common notational formalism for these types of games.
- Pure strategies—the simplest type of strategy in a game.
- Upper and lower values—our very first "solution concept."[3]
- Saddle point, Nash, and Stackelberg equilibria.

[3] In games, there are a wide variety of interesting "solution concepts" or "equilibrium concepts." These correspond to different assumptions on game structure, player capabilities, etc.

By the end of this chapter, you should (a) be able to interpret the normal form when players are restricted to pure strategies, (b) have a clear understanding of the role of upper and lower values, and associated security strategies for each player, and (c) intuitively appreciate the relationship between saddle point, Nash, and Stackelberg solution concepts.

Normal form and pure strategies

Normal form is the standard, canonical form we use to describe static, finite games. As we shall see later, the manner in which we express a game can influence the way we think about it, and ultimately construct solutions. This point is well-illustrated in hierarchical games[4] which we shall study in future sections.

[4] Vincent Conitzer. On Stackelberg mixed strategies. *Synthese*, 193(3): 689–703, 2016

Let us begin to explain the normal form with the following, classical example.

Example 1 (Prisoner's dilemma)**.** *There are two prisoners being held on suspicion of committing a crime. The prisoners are guilty; however, the police need a confession because they have insufficient evidence for a conviction. The police tell each prisoner that they have two options: confess (C) or stay quiet (Q). Since each prisoner has these options, there are four possible outcomes:*

CC If both prisoners confess to the crime, they will both be given a 2-year prison sentence.

QQ If both prisoners stay quiet, they will both be given a 1-year prison sentence.

CQ/QC If one prisoner confesses and the other does not, then the one who confesses will avoid jail time, and the one who stays quiet will get a 3-year sentence.

We can arrange these outcomes in a table for each prisoner, as shown in Tables 1 and 2.

P1 \ P2	C	Q
C	2	0
Q	3	1

Table 1: Prisoner 1's sentence length

P1 \ P2	C	Q
C	2	3
Q	0	1

Table 2: Prisoner 2's sentence length

Tables 1 and 2 are referred to as the "normal form" for the game in Example 1. Note that, since the normal form is simply a set of tables, one can always interpret them as matrices. This matrix representation allows us to express the outcome of the game concisely for each player.

Example 2 (Prisoner's dilemma, continued)**.** *Suppose that the first prisoner,* P1, *decides to confess (C), and the second prisoner,* P2, *decides to stay quiet (Q). We can encode these decisions—or "actions"—as the vectors*

$$x_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \text{ and } x_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

where x_i corresponds to P*i, the first entry of x_i corresponds to the action "confess" (C), and the second entry of x_i corresponds to "stay quiet" (Q).*

Thus equipped, we can write the outcome J_i of the game for each prisoner as:

$$J_i(x_1, x_2) := x_1^\top M_i x_2 \,,$$

where the matrices M_1 and M_2 simply replicate the entries in Tables 1 and 2, i.e.

$$M_1 = \begin{bmatrix} 2 & 0 \\ 3 & 1 \end{bmatrix} \text{ and } M_2 = \begin{bmatrix} 2 & 3 \\ 0 & 1 \end{bmatrix}.$$

In Example 2, the vectors $x_i \in \{0,1\}^2$ are called "pure strategies," and the (deterministic) choices C and Q which they encode are called "actions."[5] The use of matrices M_i to encode the game outcome (via expression $J_i(x_1, x_2) = x_1^\top M_i x_2$), suggests the term "bimatrix game" to describe these two-player finite, static games in normal form. A special case of these games—which we shall investigate shortly—is that in which $J_1 \equiv -J_2$ (or equivalently, $M_1 = -M_2$). These games are called "zero-sum games," in the finite, static setting we call them "matrix games."

[5] This distinction may seem arbitrary or pedantic at this point; however, when we discuss both "mixed strategies" and strategies in dynamic games, the need for a distinction will become clear.

Security strategies

Consider a bimatrix game characterized by (arbitrary) matrices M_1 and M_2 for P1 and P2, respectively. If P1 does not know what P2 will do, how can it obtain minimal cost?

Player 1 can play what is called a "security strategy," which solves the following problem:

$$x_1^\dagger \in \operatorname*{argmin}_{x_1} \left(\max_{x_2} x_1^\top M_1 x_2 \right) . \tag{1}$$

Let us examine (1) more closely.

The strategy $x_1^\dagger$ is called a security strategy because it minimizes P1's cost *even when* P2 *is playing adversarially.*

Question 1 (What order of play is encoded in (1)? Equivalently, what information does each player know when choosing a strategy?)**.** Answer: *We read from left to right. Player 1* first *selects a strategy x_1.* Then, P2 *gets to choose a response x_2 which maximizes $x_1^\top M_1 x_2$,* given *knowledge of* P1*'s choice, x_1.*

Reconsider the prisoner's dilemma of Examples 1 and 2. What is P1's security strategy?

Example 3 (Prisoner's dilemma, continued)**.** *Player 1 solves the following problem to determine its security strategy:*

$$x_1^\dagger \in \operatorname*{argmin}_{x_1} \left(\max_{x_2} x_1^\top \begin{bmatrix} 2 & 0 \\ 3 & 1 \end{bmatrix} x_2 \right) ,$$

where each $x_i \in \{(0,1)^\top, (1,0)^\top\}$. There are only a small number of possible combinations, so we can readily identify the solution as

$$x_1^\dagger = \begin{bmatrix} 1 \\ 0 \end{bmatrix} ,$$

with corresponding worst-case $x_2 = (1,0)^\top$. Plainly, the best worst-case outcome for P1 *occurs when both players confess to the police.*

Lower and upper values

Consider a (zero-sum) game in which $J \equiv J_1 \equiv -J_2$. In the context of matrix games, this implies that $M = M_1 = -M_2$. Here, we may offer further insight into the outcomes which correspond to players' security strategies.

We define the "upper value" of the game[6] to be the outcome corresponding to P1's security strategy, i.e.

[6] This is also called the "loss ceiling" or "security level."

$$\overline{V} := \min_{x_1} \left(\max_{x_2} J(x_1, x_2) \right) . \tag{2}$$

Likewise, we define the "lower value" to be

$$\underline{V} := \max_{x_2} \left(\min_{x_1} J(x_1, x_2) \right) , \tag{3}$$

which is the outcome of the game corresponding to P2's security strategy computation. In other words, it is the best (largest outcome) that P2 can achieve when P1 reacts adversarially to $x_2^\dagger$.

These values are called "upper" and "lower" for good reason!

Proposition 1 (Upper and lower values)**.** *For every cost function J and all sets of possible strategies $\mathcal{X}_1$ and $\mathcal{X}_2$ such that $x_i \in \mathcal{X}_i$,*[7] *we have*

$$\overline{V} \geq \underline{V} .$$

[7] So far, we have only considered pure strategies in which $\mathcal{X}_i$ to be the set of standard Cartesian basis vectors. However, we will shortly relax that assumption.

Proof. Examine (2), and replace x_1 with one of P1's security strategies $x_1^\dagger$ such that (2) now reads:

$$\overline{V} = \max_{x_2} J(x_1^\dagger, x_2) \geq J(x_1^\dagger, \tilde{x}_2), \forall \tilde{x}_2 \in \mathcal{X}_2 ,$$

where the latter inequality follows by the definition of "maximum." Similarly, we can substitute one of P2's security strategies $x_2^\dagger$ into (3) to find:

$$\underline{V} = \min_{x_1} J(x_1, x_2^\dagger) \leq J(\tilde{x}_1, x_2^\dagger), \forall \tilde{x}_1 \in \mathcal{X}_1 .$$

When we set $\tilde{x}_i \equiv x_i^\dagger$, these inequalities together imply

$$\overline{V} = \max_{x_2} J(x_1^\dagger, x_2) \geq J(x_1^\dagger, x_2^\dagger) \geq \min_{x_1} J(x_1, x_2^\dagger) = \underline{V} ,$$

which is what we wished to show.[8] □

[8] In the optimization literature, this property is often called "weak duality."

Corollary 1 (The case of matrix games)**.** *When $J(x_1, x_2) := x_1^\top M x_2$ and $\overline{V}, \underline{V}$ are defined accordingly, Proposition 1 continues to hold.*

This result begs the following question:

Question 2 (Is it better to play first or second in a zero-sum game?)**.** Answer: *$\overline{V}$ corresponds to a setting in which* P1 *plays first and* P2 *gets to react with knowledge of $x_1^\dagger$; the reverse is true for $\underline{V}$. Because $\overline{V} \geq \underline{V}$ by Proposition 1, and* P1 *wishes to* minimize *the game outcome, we conclude that it is better to play* second *(and exploit knowledge of the other player's decision) than to play* first *and blindly commit to a strategy.*

Example 4 (Computing upper and lower values)**.** *Consider a matrix game with*

$$M = \begin{bmatrix} 1 & 3 \\ 2 & 1 \end{bmatrix} ,$$

and players are restricted to pure strategies. We can readily compute $\overline{V} = 2$ and $\underline{V} = 1$. As expected, $2 = \overline{V} \geq \underline{V} = 1$.

Solution concepts in normal form games

In this section, we will discuss three solution—or "equilibrium"—concepts for normal form games. Consider the following game:

Example 5 (Saddle point)**.** *Suppose a matrix game is characterized by*

$$M = \begin{bmatrix} 2 & 1 \\ 3 & 2 \end{bmatrix}$$

and players are restricted to pure strategies. Here, we can verify that both $\overline{V} = 2$ *and* $\underline{V} = 2$*. The corresponding security strategies* $x_1^\dagger, x_2^\dagger = (1, 0)^\top$ *form a "saddle point" equilibrium of the game.*

Formally, a saddle point is defined as follows:

Definition 1 (Saddle point)**.** *A saddle point is a pair of strategies* (x_1^*, x_2^*) *which simultaneously achieve the upper- and lower-values of a zero-sum game. These strategies are therefore security strategies and must satisfy*

$$J(x_1^*, x_2) \leq J(x_1^*, x_2^*) \leq J(x_1, x_2^*), \forall x_1 \in \mathcal{X}_1, x_2 \in \mathcal{X}_2\,. \tag{4}$$

The name "saddle point" derives from the geometry of graphs such as that shown in Figure 1, which resembles the shape of a horse's saddle. In this case, the function $J(x_1, x_2) = x_1^2 - x_2^2$ has a saddle point at $(0, 0)$ because:

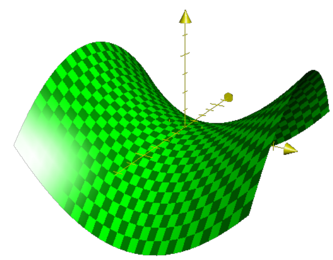

Figure 1: Graph of the function $J(x_1, x_2) = x_1^2 - x_2^2$.

$$J(0, x_2) \leq \underbrace{J(0,0)}_{=0} \leq J(x_1, 0), \forall x_1, x_2 \in \mathbb{R}\,.$$

Note, however, that not every function that geometrically appears to be a saddle will satisfy Definition 1. For example, rotating the graph in Figure 1 about the vertical axis by 90° will make the origin violate Definition 1.

Let us broaden our horizon slightly, and consider non-zero-sum (i.e., general-sum) games. The prisoner's dilemma of Examples 1 and 2 is one such game. The analogue of the saddle point solution in such cases is called a "Nash equilibrium," named in honor of Nobel laureate John Nash.[9]

[9] John F Nash. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950

Definition 2 (Nash equilibrium)**.** *A Nash equilibrium (NE) of an N-player game is a set of strategies* $(x_i^*)_{i=1}^N$ *satisfying*

$$J_i(x_i^*, x_{\neg i}^*) \leq J_i(x_i, x_{\neg i}^*), \forall x_i \in \mathcal{X}_i\,. \tag{5}$$

Note that the notation $x_{\neg i}$ is commonly used to refer to the strategies for all players *other than* Pi, i.e. $(x_j)_{j \neq i}$.

Example 6 (Nash equilibrium)**.** *Consider the bimatrix prisoner's dilemma game, with*

$$M_1 = \begin{bmatrix} 2 & 0 \\ 3 & 1 \end{bmatrix} \text{ and } M_2 = \begin{bmatrix} 2 & 3 \\ 0 & 1 \end{bmatrix},$$

and restrict both prisoners to pure strategies. We can readily verify that the unique Nash equilibrium of this game is given by the point $x_1^* = x_2^* = (1,0)^\top$*, corresponding to both prisoners confessing to the police.*

Interestingly, recall from Example 3 that P1*'s Nash strategy* x_1^* *is also its security strategy* $x_1^\dagger$*. One can also show that the same is true for* P2.

Finally, we introduce the "Stackelberg equilibrium" concept,[10] which can be understood as a general-sum extension of the notion of security strategies. More precisely, Stackelberg equilibria occur in two-player[11] games where one player must commit to a strategy that the other views before deciding its strategy. The player who pre-commits is called the "leader" and the other is called the "follower."

[10] Named for Heinrich von Stackelberg.

Heinrich von Stackelberg. Marktform und gleichgewicht. 1934

[11] One may also construct "hierarchical" Stackelberg games involving $N > 2$ players, and in which leader-follower structure follows an arbitrary directed graph topology. Computing equilibria of such games is quite interesting (and difficult!), but will not be discussed in this monograph.

Example 7 (Stackelberg equilibrium)**.** *Consider the prisoner's dilemma with* M_1 *and* M_2 *given in Example 6, and take* P1 *to be the leader (with* P2 *the follower). In this case,* P1 *reasons that if it chooses* $x_1 = (1,0)^\top$*, then* P2*'s rational response will be* $x_2^*(x_1) = (1,0)^\top$*; i.e., if* P1 *commits to confessing, then* P2 *will also wish to confess and both prisoners will end up with 2 years in prison. Similarly,* P1 *reasons that if it chooses* $x_1 = (0,1)^\top$*, then* P2*'s rational response will be* $x_2^*(x_1) = (1,0)^\top$*; i.e., if* P1 *commits to staying quiet, then* P2 *will still wish to confess—in which case,* P1 *will end up with 3 years in prison and* P2 *will go free. Clearly, between these options,* P1 *should choose* $x_1^* = (1,0)^\top$*, i.e., to confess.*

Mathematically, the leader's reasoning in Example 7 amounts to minimizing cost while accounting for the follower's "best response map"—indicated by $x_2^*(x_1)$.[12]

[12] In practice, it is possible for this map to be set-valued; later in this monograph we shall often neglect these cases.

Definition 3 (Stackelberg equilibrium)**.** *A Stackelberg equilibrium (SE) is a point* (x_1^*, x_2^*) *satisfying*

$$\text{(P1, \textit{Leader})} \qquad x_1^* = \underset{x_1}{\operatorname{argmin}}\ J_1(x_1, x_2^*) \tag{6a}$$

$$\text{(P2, \textit{Follower})} \qquad \text{s.t. } x_2^* = \underbrace{\underset{x_2}{\operatorname{argmin}}\ J_2(x_1, x_2)}_{=:\, x_2^*(x_1)}\ . \tag{6b}$$

In Example 7, we saw that the (unique) Stackelberg solution to the prisoner's dilemma occurs when both players select $x_i^* = (1,0)^\top$, which coincides with the Nash solution from Example 6. This is not always the case, as illustrated below:

Example 8 (Nash $\neq$ Stackelberg, in general)**.** *Consider a game modeling the decision to work from home or commute to the office:* $x_i = (1,0)^\top$ *corresponds to working from home, and* $x_i = (0,1)^\top$ *corresponds to commuting to the office. Each player's cost is of the same form, given by matrices*

$$M_1 = \begin{bmatrix} 2 & 2 \\ 4 & 1 \end{bmatrix} \text{ and } M_2 = \begin{bmatrix} 2 & 4 \\ 2 & 1 \end{bmatrix}$$

which encode the following cost structure: each player pays one unit of cost to commute, two units of cost if they are working in isolation, and a last unit of cost when they are frustrated because they made an effort to come to the office and their office mate stayed home.

We can readily compute the Nash solutions to this game as the points $x_1^*, x_2^* = (1, 0)^\top$ *and* $x_1^*, x_2^* = (0, 1)^\top$*, corresponding to both players working from home or both working from the office, respectively. However, the Stackelberg solution (with* P1 *as leader) is given by* $x_1^* = x_2^*(x_1^*) = (0, 1)^\top$*, in which both players come to the office. In this latter case, we see that by being able to* commit *to come to the office beforehand,* P1 *is able to influence the behavior of* P2 *and achieve lower cost (for both players, in fact).*

Static, Smooth, Unconstrained Games

In this chapter, we will discuss games in which players have *continuous and unconstrained* action spaces, and in which their objectives are *smooth* (i.e., continuous and at least twice differentiable). While we may still seek the saddle point, Nash, and/or Stackelberg equilibria defined in Static, Finite, Pure Strategy Games, in this chapter we shall discuss *local* variants of these solution concepts which may be found efficiently without substantial requirements on game structure.

Why smooth static games?

Smooth static games arise in a plethora of applications, ranging from machine learning to robotics to smart infrastructure and beyond.

Example 9 (Generative adversarial network)**.** *In machine learning, generative adversarial networks (GANs)*[13] *are widely used to model complicated data distributions to enable, e.g., producing realistic (but artificial) images or text. A GAN is comprised of two neural network components, a "generator" and "discriminator." The generator,* $G_\theta : \mathcal{Z} \to \mathcal{W}$, *maps random noise* $z \sim p_{\mathcal{Z}}$ *to the space* $\mathcal{W}$ *in which the data of interest exists (e.g., images of the appropriate dimension). Meanwhile, the discriminator,* $D_\phi : \mathcal{W} \to [0,1]$, *takes in an element of the data space* $\mathcal{W}$ *and estimates the probability that it is genuine (i.e., is not the output of generator* G_θ*).*

[13] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014

We frame this scenario as a smooth, static, zero-sum game played between G_θ *and* D_ϕ*: the generator wishes to fool the discriminator into misclassifying its (synthetic) output as genuine. That is, we can curate a dataset* $\mathcal{D}_\theta = \mathcal{D}_{fake} \cup \mathcal{D}_{real}$, *where* $\mathcal{D}_{fake} = \{G_\theta(z_i) : z_i \sim p_{\mathcal{Z}}\}_{i=1}^{n_{fake}}$ *and* $\mathcal{D}_{real}$ *consists of* n_{real} *genuine data points from* $\mathcal{W}$*. Given* $\mathcal{D}$*, we measure the rate of discriminator errors as*

$$E(\theta, \phi; \mathcal{D}_\theta) := \mathbb{E}_{w \sim \mathcal{D}_\theta} \big[-\mathbb{I}\{w \in \mathcal{D}_{real}\} \log D_\phi(w) - \mathbb{I}\{w \in \mathcal{D}_{fake}\} \log(1 - D_\phi(w)) \big] .$$

This quantity is the Kullback-Leibler divergence between the discriminator

$\mathbb{I}\{\mathcal{E}\}$ is a binary indicator function which takes the value 1 if event $\mathcal{E}$ occurs and 0 otherwise.

output and the true composition of $\mathcal{D}_\theta$. Thus equipped, we can express the game as

$$D_\phi: \quad \theta^* \in \underset{\theta}{\operatorname{argmin}}\, E(\theta, \phi; \mathcal{D}_\theta)$$
$$G_\theta: \quad \phi^* \in \underset{\phi}{\operatorname{argmin}} -E(\theta, \phi; \mathcal{D}_\theta)\,.$$

This game is zero-sum because the players' objectives are precisely opposite (i.e., they add to zero), and the game is smooth because the function $E(\theta, \phi; \mathcal{D}_\theta)$ is a continuous and differentiable function of players' actions θ, ϕ.

The difficulty of training GANs—finding saddle point solutions to this game—is well-established. We will revisit this observation shortly, when we discuss algorithmic considerations.

Example 10 (Multi-agent reinforcement learning)**.** *Reinforcement learning (RL) problems are a class of stochastic optimization problems which correspond to (typically time-invariant) sequential decision making. Multi-agent reinforcement learning (MARL), therefore, constitutes problems in which multiple stochastic decision problems are mutually coupled. A wide variety of formulations exist for these problems, and often research focuses on algorithmic innovations which specialize to particular domain artifacts. Consider the following simplified example, in which player $i \in \{1, 2, \ldots, N\}$ wishes to solve a problem of the form:*

$$\theta_i^* \in \underset{\theta_i}{\operatorname{argmin}}\, \mathbb{E}_{x_1 \sim p_x, u_t^i \sim \pi^i_{\theta_i}(x_t), x_{t+1} \sim f(x_t, u_t^i, u_t^{\neg i})} \left[\sum_{t=1}^{\infty} \gamma^t g_i(x_t, u_t^i, u_t^{\neg i}) \right].$$

We read this expression as follows:[14]

- *The players collectively have a* state x_t *at each time $t \in \{1, 2, \ldots, \infty\}$. The initial state x_1 is drawn from distribution p_x.*
- *At each time t, each* Pi *selects an action u_t^i from a distribution specified by* policy $\pi^i_{\theta_i}(x_t)$.
- *At each time t, the next state x_{t+1} is drawn from the* dynamics *distribution $f(x_t, u_t^i, u_t^{\neg i})$.*[15]
- Pi *wishes to minimize the expectation of its discounted cost: $\gamma \in (0, 1)$ is a discount factor, and g_i is a (bounded) real-valued function which encodes the cost* Pi *incurs at each time t.*

The coupled optimization problems above constitute a N-player, general-sum game. The game is unconstrained and static *in the sense that each player* Pi *only selects the parameters θ_i* once*. We may identify "optimal" policies $\pi^i_{\theta_i}$ for each player by finding, e.g., a Nash equilibrium of this game. So long as all variables are continuous and functions are continuously differentiable, the game is smooth.*

Beyond the fact that many games of interest are smooth and static, we also study smooth, static games because the properties of continuity and differentiability lend themselves to the construction of efficient algorithms.

[14] Note that many of the objects below pertain to "dynamic" games and will be discussed in far greater detail later in this monograph.

[15] Recall that the notation $\neg i$ indicates the collection of all variables *not* pertaining to Pi.

Local Nash equilibria

Algorithms which exploit gradient information reason only about *local* structure of a problem, i.e., how outcomes change in response to small perturbations of players' actions. To capture these effects, we define a local variant of the Nash solution concept.[16]

[16] Lillian J Ratliff, Samuel A Burden, and S Shankar Sastry. On the characterization of local Nash equilibria in continuous games. *IEEE transactions on automatic control*, 61(8):2301–2307, 2016

Definition 4 (Local Nash equilibrium)**.** *A local Nash equilibrium (LNE) is a point* $(x_i^*)_{i=1}^N$ *at which Definition 2 holds only* locally. *Formally, let* $\tilde{\mathcal{X}}_i \subset \mathcal{X}_i$ *be an open set containing the point* x_i^*. *We require that*

$$J_i(x_i^*, x_{\neg i}^*) \leq J_i(x_i, x_{\neg i}^*), \forall x_i \in \tilde{\mathcal{X}}_i$$

for some *(potentially very small) set* $\tilde{\mathcal{X}}_i$.

Just as Nash equilibria generalize saddle points beyond the case of two-player, zero-sum games, one may define a local saddle point (LSP) by restricting Definition 4 to that case.

What does locality imply about equilibria?

The difference between Definitions 2 and 4 appears rather innocuous: all that has changed is the domain on which an inequality must hold. In the case of a single player (i.e., if $N = 1$), a LNE corresponds to a local optimum of the function $J_1 : \mathcal{X}_1 \to \mathbb{R}$. In many cases, such solutions are all that we can ever hope to identify given the computational intractability of identifying global optima in nonconvex programming.

When $N > 1$, however, matters become more interesting. In this case, Definition 4 says that each player is *unilaterally* locally optimal, i.e., they are operating at a local minimum when other players' actions are fixed to $x_{\neg i}^*$. In particular, Pi may prefer to play action $x_i \neq x_i^*$, so long as x_i is not in the neighborhood $\tilde{\mathcal{X}}_i$ of x_i^*.

One may interpret this as a relaxation of the implicit "self-interested rationality" assumption behind the NE: that is, a local Nash equilibrium models self-interested behavior (i.e., Pi minimizing its own objective J_i) when each player can only myopically reason about small changes to its strategy.

In practice, the author [Fridovich-Keil et al., 2020] and others [Le Cleac'h et al., 2022] have found that, despite these subtleties, LNE yield both high-performance and qualitatively-appropriate strategies. For example, a self-driving car can follow a locally Nash strategy to navigate complex traffic patterns while accounting for other drivers' reactions.

How can we recognize a LNE in a smooth, static game?

Suppose that $(x_i^*)_{i=1}^N$ constitute a LNE in a smooth, static game. Following standard arguments in optimization, we construct the following necessary and sufficient conditions:

Proposition 2 (First- and second-order conditions for LNE)**.** *Let $\nabla_{x_i} J_j$ denote the gradient of* Pj*'s cost with respect to the action of* Pi*, and $\nabla^2_{x_i,x_j} J_k$ denote the matrix of mixed partial derivatives (i.e., Hessian) of* Pk*'s cost with respect to* Pi *and* Pj*'s actions. The following conditions must be satisfied for a point $(x_i^*)_{i=1}^N$ to be a LNE:*

$$\nabla_{x_i} J_i(x_i^*, x_{\neg i}^*) = 0, \forall i \in \{1, 2, \dots, N\}\,. \tag{7}$$

All LNE $(x_i^*)_{i=1}^N$ must additionally satisfy

$$\nabla^2_{x_i,x_i} J_i(x_i^*, x_{\neg i}^*) \succ 0, \forall i \in \{1, 2, \dots, N\}\,. \tag{8}$$

These conditions are respectively necessary (7) and sufficient (8) for a point to be a LNE.

Proof. The proof follows from a direct application of standard results in optimization[17] to the i^{th} inequality in Definition 4. □

[17] See, e.g., chapter 2 of Nocedal and Wright [1999].

Given a candidate LNE $(x_i^*)_{i=1}^N$, we can readily verify both conditions in Proposition 2. As we shall soon see, however, common algorithms one may wish to use in order to identify such points do not always satisfy these conditions. In fact, recent work[18] proposes that such issues may be responsible for some of the difficulty of training GANs (cf. Example 9).

[18] Eric Mazumdar, Lillian J Ratliff, and S Shankar Sastry. On gradient-based learning in continuous games. *SIAM Journal on Mathematics of Data Science*, 2 (1):103–131, 2020

Iterative best response

Consider the following well-known, intuitive algorithm (summarized in Algorithm 1) for identifying NE and LNE: beginning with a set of strategies $(x_i)_{i=1}^N$ and proceeding sequentially, each player replaces x_i with its best response $x_i^*(x_{\neg i})$. This algorithm is called iterative best response (IBR).

While attractive for its simplicity, Algorithm 1 has a serious drawback... it does not always converge! We can observe non-convergence even in normal form games, as shown in Example 11.

Algorithm 1: Iterative best response

Input: initial strategies $(x_i)_{i=1}^N$
while *not converged* **do**
  for $i = 1, 2, \dots, N$ **do**
    $x_i \leftarrow x_i^*(x_{\neg i}) := \text{argmin}_{x_i} J_i(x_i, x_{\neg i})$ ▷ Pi's best response
return *converged* $(x_i^*)_{i=1}^N$

Example 11 (IBR does not always converge)**.** *Consider the (matrix) game of rock-paper-scissors (RPS), in which* P1*'s cost can be encoded via matrix*

$$M = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}.$$

Suppose that x_i is initialized to the mixed[19] *strategy $(0.75, 0.25, 0)^\top$ for each i (i.e., both players initially choose to play "rock" with probability 0.75 and "paper" with probability 0.25). After one round of the outer loop in Algorithm 1, we will have $x_1 = (0, 1, 0)^\top$ ("paper") and $x_2 = (0, 0, 1)^\top$ ("scissors"). After the next round: $x_1 = (1, 0, 0)^\top$ ("rock") and $x_2 = (0, 1, 0)^\top$ ("paper"). Continuing, we readily observe that Algorithm 1 will cycle ad infinitum and never converge.*

[19] We will discuss mixed strategies in greater detail shortly.

Example 11 analyzed IBR for a (finite) matrix game, rather than a (continuous) smooth game. However, one can certainly observe the same non-convergence phenomenon in smooth games.

Example 12 (IBR nonconvergence in a smooth game)**.** *Consider the following (smooth) Nash game:*

$$\begin{aligned} \mathrm{P1}: \quad & x_1^* = \operatorname*{argmin}_{x_1 \in \mathbb{R}} \ (\tanh x_1 - \tanh x_2)^2 \\ \mathrm{P2}: \quad & x_2^* = \operatorname*{argmin}_{x_2 \in \mathbb{R}} -(\tanh x_1 - \tanh x_2)^2 . \end{aligned}$$

Let us arbitrarily initialize Algorithm 1 from the point $(x_1, x_2) = (0, 0)$. P1*'s best response to $x_2 = 0$ is $x_1^*(0) = 0$, and* P2*'s best response is $x_2^*(0) = \{-\infty, \infty\}$. Clearly, IBR has already diverged—but matters get worse! Suppose* P2 *chooses $x_2^*(0) = \infty$. At the next round,* P1 *chooses to play $x_1^*(\infty) = \infty$, but then* P2 *will choose to play $x_2^*(\infty) = -\infty$. Continue this reasoning onward, and it becomes clear that IBR will cycle between $\pm\infty$ and never converge.*

Of course, IBR does not always fail, either in finite or smooth games. Consider the following example.

Example 13 (IBR convergence in a smooth game)**.** *Consider a (smooth) Nash game with the following quadratic structure:*

$$\begin{aligned} \mathrm{P1}: \quad & x_1^* = \operatorname*{argmin}_{x_1 \in \mathbb{R}} x_1^2 + x_1(x_2 - 1) \\ \mathrm{P2}: \quad & x_2^* = \operatorname*{argmin}_{x_2 \in \mathbb{R}} x_2^2 + x_2(x_1 + 1) . \end{aligned}$$

Let us arbitrarily initialize Algorithm 1 from the point $(x_1, x_2) = (0, 0)$. Figure 2 plots the iterates (x_1, x_2) after each round of IBR. As shown, IBR rapidly converges to the point $(1, -1)$, at which we may verify the first- and second-order conditions of Proposition 2.

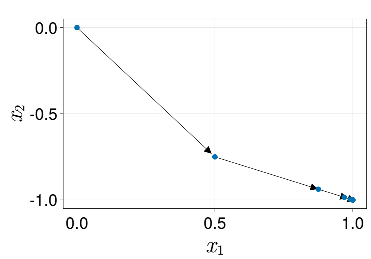

Figure 2: IBR iterates for the game in Example 13.

When will IBR converge? A sufficient condition is given in Proposition 3.

Proposition 3 (IBR convergence)**.** *Consider an N-player game. Suppose that there exists a "potential" function $\phi(x_1, \dots, x_N) \geq 0$ such that for each* Pi*, the following relation holds:*

$$\phi(x_i, x_{\neg i}) - \phi(x_i', x_{\neg i}) = J_i(x_i, x_{\neg i}) - J_i(x_i', x_{\neg i}), \ \forall x_i, x_i', x_{\neg i} \, .$$

Then, assuming that each player's best response map yields a unique action, IBR will converge to a fixed point $(x_i^*)_{i=1}^N$, which will be a Nash equilibrium.

Proof. In every round of IBR in Algorithm 1, each Pi holds $x_{\neg i}$ constant and minimizes $J_i(x_i, x_{\neg i})$. Denote the minimizer by x_i': from the condition above, we are assured that $J_i(x_i, x_{\neg i}) - J_i(x_i', x_{\neg i}) = \phi(x_i, x_{\neg i}) - \phi(x_i', x_{\neg i})$. Therefore, at each round of IBR, the function ϕ decreases monotonically. Because $\phi \geq 0$, this monotonic sequence must approach a limiting value. We are also assured that this limiting value corresponds to a single action x_i^* for each Pi due to the uniqueness of its best response map. Therefore, we conclude that IBR will converge to a set of actions $(x_i^*)_{i=1}^N$, each of which constitutes a best response to the others; therefore, this set must satisfy the Nash conditions of Definition 2. □

Corollary 2 (Local IBR convergence)**.** *Assume the conditions of Proposition 3, and take Line 4 of Algorithm 1 to imply that* Pi *chooses a* local *minimizer of its cost function, rather than a global minimizer. Following the same argument of Proposition 3, it is clear that IBR will converge to a local Nash equilibrium.*

The limiting case: simultaneous gradient play

While Iterative best response may be the "standard" algorithm practitioners employ to identify NE, it can be impractical to minimize J_i at every round. For example, in the GAN training of Example 9, players' decision variables correspond to the parameters of potentially very large convolutional neural networks, and training such networks to convergence can take significant time and resources. Therefore, it is becoming increasingly common to consider limiting variants of IBR, such as Algorithm 2, where at each round, players take only a single gradient step on their objectives.

However, as we shall see below, the simultaneous gradient methods do not always converge to (local) NE, and can in fact converge to spurious non-Nash points.[20] Following Mazumdar et al. [2020], we will analyze the continuous flow of players' strategies $(x_i)_{i=1}^N$, i.e., the limit of Algorithm 2 when step size $\alpha \downarrow 0$.

[20] Eric Mazumdar, Lillian J Ratliff, and S Shankar Sastry. On gradient-based learning in continuous games. *SIAM Journal on Mathematics of Data Science*, 2 (1):103–131, 2020

Algorithm 2: Simultaneous gradient play

Input: initial strategies $(x_i)_{i=1}^N$, step size $s > 0$
while *not converged* **do**
  for $i = 1, 2, \ldots, N$ **do**
    $\delta_i \leftarrow \nabla_{x_i} J_i(x_i, x_{\neg i})$ ▷ Pi's cost gradient
  for $i = 1, 2, \ldots, N$ **do**
    $x_i \leftarrow x_i - \alpha\delta_i$ ▷ gradient step
return *converged* $(x_i^*)_{i=1}^N$

Recall that $\mathbf{x} := (x_i)_{i=1}^N$, and define the function

$$\omega(\mathbf{x}) := \begin{bmatrix} \nabla_{x_1} J_1(\mathbf{x}) \\ \nabla_{x_2} J_2(\mathbf{x}) \\ \vdots \\ \nabla_{x_N} J_N(\mathbf{x}) \end{bmatrix} .$$

Thus equipped, we can rewrite the limiting behavior of Algorithm 2 (with $\alpha \downarrow 0$) as:

$$\dot{\mathbf{x}} \equiv \frac{d\mathbf{x}}{dt} = -\omega(\mathbf{x}) \,, \tag{9}$$

where we interpret the variable t ("time") as measuring progress in the outer loop of Algorithm 2. We are interested in analyzing the points to which $\mathbf{x}$ can converge, and how those points relate to LNE. In order to facilitate this analysis, define the Jacobian of $\omega(\mathbf{x})$ as follows:

$$D(\mathbf{x}) := \begin{bmatrix} \nabla^2_{x_1 x_1} J_1(\mathbf{x}) & \nabla^2_{x_1 x_2} J_1(\mathbf{x}) & \cdots & \nabla^2_{x_1 x_N} J_1(\mathbf{x}) \\ \nabla^2_{x_2 x_1} J_2(\mathbf{x}) & \nabla^2_{x_2 x_2} J_2(\mathbf{x}) & \cdots & \nabla^2_{x_2 x_N} J_2(\mathbf{x}) \\ \vdots & \vdots & \ddots & \vdots \\ \nabla^2_{x_N x_1} J_N(\mathbf{x}) & \nabla^2_{x_N x_2} J_N(\mathbf{x}) & \cdots & \nabla^2_{x_N x_N} J_N(\mathbf{x}) \end{bmatrix} .$$

Let the eigenvalues of $D(\mathbf{x})$, denoted $\{\lambda_i(D(\mathbf{x}))\}$, be sorted in ascending order according to their real part $\Re(\lambda_i)$, and consider the following points of interest:

- Critical points (CPs): points at which $\omega(\mathbf{x}) = 0$. These are stationary points for the differential equation (9).
- Strict saddle points (SSPs): points at which $\omega(\mathbf{x}) = 0$ and $\exists \ell$: $\Re(\lambda_i) < 0, \forall i < \ell$ and $\Re(\lambda_i) > 0, \forall i \geq \ell$. These are points to which the differential equation (9) will almost certainly avoid, due to the negative eigenvalues.

- Locally asymptotically stable equilibria (LASE): points at which $\omega(\mathbf{x}) = 0$ and $\Re(\lambda_i) > 0, \forall i$. These are the points to which the dynamics (9) can converge.

- Local Nash equilibria (LNE): points at which Proposition 2 holds.

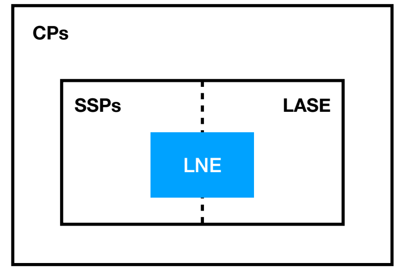

Figure 3: Relationships among points of interest for simultaneous gradient play, for general-sum games.

In general, the relationship among these points of interest are summarized in Figure 3, reproduced from Mazumdar et al. [2020]. Examining the diagram carefully, we see that there are some LASE which are *not* LNE. Such points are potential attractors for the simultaneous gradient algorithm Algorithm 2, and yet are not LNE.

Proposition 4 (Not all attractors of Algorithm 2 are LNE)**.** *Suppose that* $\mathbf{x}$ *is a LASE of (9);* $\mathbf{x}$ *is not necessarily a LNE.*

Proof. We offer a two-player counterexample. Consider a game in which:

$$J_1(x_1, x_2) = \frac{1}{2}ax_1^2 + bx_1x_2$$
$$J_2(x_1, x_2) = \frac{1}{2}dx_2^2 + cx_1x_2 \,.$$

In this game, the only point at which $\omega(\mathbf{x}) = 0$ is the point $\mathbf{x} = (0, 0)$. However,

$$D(\mathbf{x}) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} ,$$

meaning that if $a > 0$ and $d < 0$, the point $\mathbf{x} = (0, 0)$ does not satisfy Proposition 2 and therefore cannot be a LNE.

However, we can find values of the constants a, b, c, d for which this is true, and for which the origin is a LASE. To do so, we idenitfy the eigenvalues of $D(\mathbf{x})$ as having real parts

$$\Re(\lambda_i) = \frac{1}{2}\left(a + d \pm \begin{cases} 0, \beta \leq 0 \\ \sqrt{\beta}, \beta > 0 \end{cases}\right) ,$$

where $\beta = (a + d)^2 - 4(ad - cb)$. Therefore, if we choose constants such that

$$a + d > 0 \text{ and } ad > cb \,,$$

we are assured that all eigenvalues of $D(\mathbf{x})$ have positive real part, and therefore that the point $\mathbf{x} = (0, 0)$ is a LASE. One such choice is: $a = 2, b = -2, c = 2, d = -1$. □

Recall that the generative adversarial network (GAN) introduced of Example 9 was not a general-sum game, but rather had a (two-player) zero-sum structure. Mazumdar et al. [2019] propose the following second-order correction to the simultaneous gradient play in

(9), for which all LNE are LASE and vice versa:

$$\dot{\mathbf{x}} = -\underbrace{\frac{1}{2}\left(\omega(\mathbf{x}) + D(\mathbf{x})^\top D(\mathbf{x})^{-1}\omega(\mathbf{x})\right)}_{h(\mathbf{x})} . \tag{10}$$

First, we show that replacing the gradient update $\omega(\mathbf{x})$ with the vector field $h(\mathbf{x})$ does not change the set of critical points.

Lemma 1 (CPs of Equations (9) and (10) coincide)**.** *Assume that the Jacobian $D(\mathbf{x})$ is nonsingular, and that at all points $\mathbf{x}$, we have*

$$D(\mathbf{x})^\top D(\mathbf{x})^{-1}\omega(\mathbf{x}) \neq -\omega(\mathbf{x}) .$$

Then, $\omega(\mathbf{x}) = 0 \iff h(\mathbf{x}) = 0$.

Proof. Clearly, $\omega(\mathbf{x}) = 0 \implies h(\mathbf{x}) = 0$ holds from the definition of $h(\mathbf{x})$ in (10). To show that $h(\mathbf{x}) = 0 \implies \omega(\mathbf{x}) = 0$, we construct a proof by contradiction.

Suppose that $h(\mathbf{x}) = 0$, but that $\omega(\mathbf{x}) \neq 0$. From $h(\mathbf{x}) = 0$, we know that:

$$\begin{aligned} 0 = h(\mathbf{x}) &= \frac{1}{2}\left(\omega(\mathbf{x}) + D(\mathbf{x})^\top D(\mathbf{x})^{-1}\omega(\mathbf{x})\right) \\ &\implies \underbrace{-\omega(\mathbf{x}) = D(\mathbf{x})^\top D(\mathbf{x})^{-1}\omega(\mathbf{x})}_{\Rightarrow\Leftarrow,\ \text{by assumption}} \end{aligned}$$

Thus, the CPs of Equations (9) and (10) coincide. □

Now, we recognize that the Jacobian D can be decomposed into symmetric (S) and antisymmetric (A) components as follows:

$$D(\mathbf{x}) = S(\mathbf{x}) + A(\mathbf{x})$$

with

$$S(\mathbf{x}) = \begin{bmatrix} \nabla^2_{x_1x_1}J(\mathbf{x}) & 0 \\ 0 & -\nabla^2_{x_2x_2}J(\mathbf{x}) \end{bmatrix}, \ A(\mathbf{x}) = \begin{bmatrix} 0 & \nabla^2_{x_1x_2}J(\mathbf{x}) \\ -\nabla^2_{x_2x_1}J(\mathbf{x}) & 0 \end{bmatrix} .$$

Proposition 5 (Attractors of (10) and LNE are equivalent)**.** *All LASE of (10) are LNE and vice versa.*

Proof. Define $D_h(\mathbf{x}) := \nabla_{\mathbf{x}} h(\mathbf{x})$. Suppose that $\mathbf{x}$ is a CP of (10) (and therefore of (9) as well by Lemma 1). We observe:

$$\begin{aligned} D_h(\mathbf{x}) &= \frac{1}{2}\left(\nabla_{\mathbf{x}}\omega(\mathbf{x}) + \nabla_{\mathbf{x}}\big(D(\mathbf{x})^\top D(\mathbf{x})^{-1}\omega(\mathbf{x})\big)\right) \\ &= \frac{1}{2}\left(D + \nabla_{\mathbf{x}}\big(D(\mathbf{x})^\top D(\mathbf{x})^{-1}\big)\underbrace{\omega(\mathbf{x})}_{=0} + D(\mathbf{x})^\top D(\mathbf{x})^{-1}\underbrace{\nabla_{\mathbf{x}}\omega(\mathbf{x})}_{=D(\mathbf{x})}\right) \\ &= \underbrace{\frac{1}{2}\left(D(\mathbf{x}) + D(\mathbf{x})^\top\right)}_{=S(\mathbf{x})} . \end{aligned}$$

The matrix $S(\mathbf{x})$ is symmetric, and therefore has purely real eigenvalues.

Now, suppose that all eigenvalues of $D_h(\mathbf{x}) = S(\mathbf{x})$ are strictly positive. This condition precisely coincides with the definition of LASE and the necessary and sufficient conditions for LNE from Proposition 2. This implies that the set of LASE of (10) and LNE coincide, and also that non-Nash LASE of (9) *cannot* be LASE of (10). $\square$

Static, Smooth, Constrained Games

Constraints on players' strategies can arise in any number of circumstances. In this chapter, we examine two broad classes of *constrained* static games: finite mixed strategy games, and smooth pure strategy games. To build an understanding of these problems, we will also review the fundamentals of constrained optimization.

Mixed strategies in finite static games

Reconsider the setting of Static, Finite, Pure Strategy Games, in which each of N players has a finite number of actions available. Suppose that we seek a pure-strategy NE in such a problem, i.e., a tuple of actions $(x_i^*)_{i=1}^N$ satisfying Definition 2. Unfortunately, such points do not always exist, as we shall see shortly in Example 14. In these cases, we shall see that a Nash solution *always* exists when players are permitted to play a mixed—or stochastic—strategy, and players optimize for the average outcome.

Example 14 (Rock paper scissors has no NE in pure strategies)**.** *Consider the (two player, zero sum) rock-paper-scissors (RPS) game, in which* P1*'s cost is encoded by matrix*

$$M = \begin{bmatrix} 0 & 1 & -1 \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix}.$$

We saw in Example 11 that IBR does not converge in this game. More fundamentally, however, we may also see that none *of the nine pure strategy pairs $\{(x_1, x_2) : x_i \in \{\mathrm{R}, \mathrm{P}, \mathrm{S}\}\}$ satisfies the Nash conditions from Definition 2. Therefore, we conclude that RPS has no NE in pure strategies.*

Having established the non-existence of NE in pure strategies (at least, in some cases), we instead consider the following relaxation of a pure strategy game:

$$x_i^* \in \underset{x_i \in \mathcal{X}_i}{\operatorname{argmin}} \; J_i(x_i, x_{\neg i}) \qquad \Rightarrow \qquad p_i^* \in \underset{p_i \in \Delta[\mathcal{X}_i]}{\operatorname{argmin}} \; \mathbb{E}_{x_j \sim p_j}[J_i(x_i, x_{\neg i})].$$

Here, we allow each player to choose a *mixed strategy* p_i from the set of probability distributions over the set of pure strategies, $\Delta[\mathcal{X}_i]$.[21] Pi then wishes to minimize the *expected value* of its cost, given that all players select actions $(x_j)_{j=1}^N$ at random from their respective mixed strategies.

[21] Suppose that $\mathcal{X}_i \equiv \{1, 2, \dots, k_i\}$. In this case, $\Delta[\mathcal{X}_i] = \{p \in \mathbb{R}^{k_i} : p \geq 0, \mathbf{1}^\top p = 1\}$ is the k_i-dimensional probability simplex.

Intuitively, we may imagine that mixed strategies are appropriate when playing a game repeatedly, *ad infinitum*. Assuming that each player must select a new (pure) strategy at each round of the game, and that each wishes to minimize its long-run average cost per game, it makes sense to seek a Nash equilibrium in mixed strategies.

Theorem 1 (Nash equilibria always exist in mixed strategy, finite, static games)**.** *Consider an N-player game in which* Pi *has* $|\mathcal{X}_i| < \infty$ *available actions, and take its objective to be:*

$$p_i^* \in \underset{p_i \in \Delta[\mathcal{X}_i]}{\operatorname{argmin}} \ \mathbb{E}_{x_j \sim p_j, \forall j}[J_i(x_i, x_{\neg i})] \,. \tag{11}$$

There is guaranteed to be a point $(p_i^*)_{i=1}^N$ *simultaneously satisfying (11) for all players i. Note that (11) is identical to the conditions of Definition 2, where we understand the expected value in the objective of (11) to take the role of* $J_i(\cdot)$ *in Definition 2. The point* $(p_i^*)_{i=1}^N$ *is called a Nash equilibrium in* mixed strategies*.*

Proof. This result is due to Nash [1950], and is quite famous! The proof follows from a straightforward application of the Kakutani fixed point theorem, and readers are encouraged to consult the original paper for the complete (and very short) proof. □

The expectation in (11) may be expressed compactly in the two player setting. In this case, we may treat $p_i \in \Delta[\mathcal{X}_i] \subset \mathbb{R}^{k_i}$ (with $k_i := |\mathcal{X}_i|$ the number of actions available to Pi), and write

$$\mathbb{E}_{x_1 \sim p_1, x_2 \sim p_2}[J_i(x_i, x_{\neg i})] \equiv p_i^\top M_i p_{\neg i} \,.$$

Reconsider the RPS game of Example 14, in which there were no pure strategy Nash solutions. Theorem 1 clearly implies that RPS will have at least one mixed NE.

Example 15 (Mixed Nash solution for RPS)**.** *The (unique) mixed NE for RPS is given by:*

$$p_1^* = p_2^* = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right)^\top .$$

We verify that this solution satisfies definition of Nash equilibria as follows. For P1, $\mathbb{E}_{x_1 \sim p_1, x_2 \sim p_2^*}[J_1(x_1, x_2)] = 0$ *so long as* $p_2^* = (1/3, 1/3, 1/3)$. P1 *has no incentive to deviate from this strategy* p_1^**. The same argument holds from the perspective of* P2*, by symmetry. Therefore,* (p_1^*, p_2^*) *is a NE in mixed strategies.*

The Nash solution in Example 15 could have been identified by inspection and intuition. In Static, Smooth, Unconstrained Games, we saw that *local* NE are characterized by first- and second-order conditions (Definition 4), which may be identified by iterative algorithms such as IBR and simultaneous gradient play (Algorithms 1 and 2, respectively). However, when players' strategies are restricted by the presence of constraints,[22] we must take care to account for the presence of these constraints in algorithms we develop.

[22] such as the restriction $p_i \in \Delta[\mathcal{X}_i]$

Fundamentals of constrained optimization

Before developing techniques for the N-player setting, we provide a brief synopsis of the ideas underlying the theory of constrained optimization.[23]

[23] This material is drawn from Nocedal and Wright [1999]; interested readers are further directed to Bertsekas [1999].

What are constraints?

Consider an optimization problem in standard form:

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) && \text{(12a)} \\ \text{subject to} \quad & c_i(x) = 0, \quad \forall i \in \mathcal{E} && \text{(12b)} \\ & c_i(x) \geq 0, \quad \forall i \in \mathcal{I}\,. && \text{(12c)} \end{aligned}$$

Here, the function $f : \mathbb{R}^n \to \mathbb{R}$, and the *decision variable* x must satisfy both equality constraints $\{c_i : i \in \mathcal{E}\}$ and inequality constraints $\{c_i : i \in \mathcal{I}\}$. Consider the following examples.

Example 16 (Mixed strategies as constrained optimization)**.** *Recall the definition of mixed strategy Nash equilibria from (11). In a two-player game, holding* P2*'s strategy p_2 fixed, we obtain the following constrained optimization problem for* P1*:*

$$\begin{aligned} \min_{p_1 \in \mathbb{R}^{k_1}} \quad & p_1^\top M_1 p_2 \\ \text{subject to} \quad & \mathbf{1}^\top p_1 = 1 \\ & p_1 \geq 0\,. \end{aligned}$$

This is clearly of the same form as (12). In fact, in this case the objective and both constraints are linear in the decision variable p_1, making this a linear *program.*

Example 17 (Trajectory optimization)**.** *Consider a car navigating an empty street (i.e., with no other moving traffic). We can encode the driver's decision problem in the form of (12), interpreting variables as follows:*

- *x—the collection of all state and control input variables[24] at finite times $t \in \{1, 2, \ldots, T\}$*
- *f—a function penalizing quantities such as the vehicle's distance from a desired lane center, deviation from a preferred speed, acceleration, etc.*

[24] We have expressed the problem in "collocation" form, i.e., treating both vehicle states and control inputs as decision variables. In the literature, it is also common to treat vehicle states as deterministic functions of control inputs (i.e., substituting the equality constraint below within the objective).

- $c_i, i \in \mathcal{E}$—*for example, a function which is zero if and only if the sequence of vehicle states and control inputs is* dynamically feasible *(i.e., physically consistent with the dynamics of the vehicle)*
- $c_i, i \in \mathcal{I}$—*for example, a function which is negative whenever the vehicle is in collision with an obstacle, violates road boundaries, etc.*

Preview of coming attractions. In constrained games, it is possible for one of Pi's constraints, the j^{th} of which we denote as $c^i_j(x_i, x_{\neg i})$, to depend upon other players' decisions $x_{\neg i}$. In this case, other players can choose to act in ways that force Pi to unilaterally enforce constraint c^i_j, and potentially pay a huge price as measured by Pi's objective. For example, consider a two-player variant of the trajectory optimization problem from Example 17, where a car and a pedestrian are navigating the same intersection. At least in the US states where the author has obtained a driver's license (GA, CA, and TX), drivers always bear (sole) legal responsibility for avoiding pedestrians in public roadways. Mathematically, this implies that the car should bear full responsibility for enforcing a collision-avoidance constraint that depends upon the pedestrian's actions. An extremely greedy pedestrian could therefore, theoretically, walk straight across the intersection and force the car to stop and take evasive action.

The feasible set

Recall the standard form in (12). We can always construct the "feasible set"

$$\Omega := \{x \in \mathbb{R}^n : c_i(x) = 0 \ \forall i \in \mathcal{E}, c_i(x) \geq 0 \ \forall i \in \mathcal{I}\}, \tag{13}$$

which comprises the points x which satisfy all constraints jointly. It should be clear that, thus equipped, an equivalent form for the problem in (12) is:

$$\min_{x \in \Omega} f(x). \tag{14}$$

While mathematically identical, a key point to take away from the forthcoming discussion is that *not all descriptions of the constraints c_i are created equal!* In particular, we will see that one can often rewrite the constraints in a form more amenable to numerical optimization, without changing the fundamental geometry of Ω. Consider the following examples.

Example 18 (Eigenvalue problem)**.** *Suppose that $x \in \mathbb{R}^2$, matrix $M \succeq 0$,* *and we wish to solve:*

$$\left.\begin{array}{ll}\min_x & x^\top M x \\ \text{subject to} & \|x\|_2^2 = 1\end{array}\right\} \iff \left\{\begin{array}{ll}\min_x & x^\top M x \\ \text{subject to} & x \in \Omega \equiv \underbrace{S^1}_{\text{unit circle}}\end{array}\right.$$

We recognize the solution to this problem x^* as the eigenvector of M with minimal eigenvalue. Presuming that M has distinct eigenvalues (no duplicates), then this optimizer is unique.

We may be tempted to transform the problem as follows, by observing that $\exists \theta \in \mathbb{R}$ such that $x = (\cos\theta, \sin\theta)^\top \in S^1$:

$$\min_{\theta \in \mathbb{R}} \begin{bmatrix}\cos\theta \\ \sin\theta\end{bmatrix}^\top M \begin{bmatrix}\cos\theta \\ \sin\theta\end{bmatrix}.$$

The problem is mathematically identical; however in this new form there are an infinite number of global minima θ^* which correspond to the same (unique) minimal eigenvector x^*.

Example 19 (Problem smoothness)**.** *Consider the following optimization problem:*

$$\min_{x \in \mathbb{R}} \max(x, x^2).$$

This problem is unconstrained, but has an objective which is non-smooth at the point $x = 0$. Without changing this fundamental geometry, we can transform the problem as follows:

The variable s is known as a "slack" variable.

$$\begin{array}{ll}\min\limits_{x,s \in \mathbb{R}} & s \\ \text{subject to} & s \geq x \\ & s \geq x^2.\end{array}$$

How can we interpret this transformation? Construct the feasible set $\Omega = \{x, s \in \mathbb{R} : s \geq x, s \geq x^2\}$. The boundary of this set, $\partial\Omega := \{x, s \in \mathbb{R} : s = \max(x, x^2)\}$, precisely corresponds to the potential pairs of decision variable and objective value in the original problem. It is clear that the minimum value of s must lie within $\partial\Omega$, which assures us that the transformation above has not changed the fundamental geometry of the problem.

The slack variable transformation in Example 19 evinces a further subtlety: the original problem had a non-smooth objective (at the origin $x = 0$, the derivative of the objective does not exist). However, in the transformed problem, the objective and constraints are all smooth functions of the decision variables.

While we have yet to develop optimality conditions for constrained optimization problems of the form (12), we recall from Proposition 2 that—in the absence of constraints—the first- and second-order conditions for NE (and hence for optimality in the single-player setting) require taking both first- and second-derivatives of players' objectives. Transformations such as that of Example 19 will help us to obtain smooth representations of both optimization problems and games which are amenable to derivative-based solution techniques.

Local constraint geometry

In order to build first- and second-order conditions of optimality which account for constraints, we need to build a formal understanding of the *local* geometry of the feasible set and the constraints which define it. Consider the following example.

Example 20 (Optimizing on the circle). *Suppose that we wish to solve the following problem:*

$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & \overbrace{\mathbf{1}^\top x}^{f(x)} \\ \text{subject to} \quad & \underbrace{1 - \|x\|_2^2}_{c(x)} = 0 \,. \end{aligned}$$

Figure 4 illustrates the geometry of $f, c, \nabla f$, and ∇c. By inspection, we see that the point $x^* = -(\sqrt{2}/2, \sqrt{2}/2)^\top$ is globally optimal. At this point, one may confirm that, as illustrated, the gradients of the objective and constraint are parallel: i.e., $\exists \lambda \in \mathbb{R} : \nabla f(x^*) = \lambda \nabla c(x^*)$.

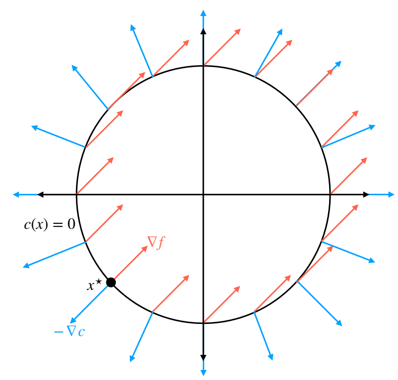

Figure 4: Illustration of the relationship between gradients of f and c.

The proportionality constant λ in Example 20 is known as a *Lagrange multiplier*, and the observation that $\exists \lambda \in \mathbb{R} : \nabla f(x^*) = \lambda \nabla c(x^*)$ holds in general.

Theorem 2 (Lagrange multipliers for equality constraints). *Consider an equality constrained optimization problem of the form*

$$\begin{aligned} \min_{x \in \mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & c_i(x) = 0, \forall i \in \mathcal{E} \,, \end{aligned}$$

with functions f, $(c_i)_{i \in \mathcal{E}}$ differentiable. Under an appropriate constraint qualification *(e.g., the linear independence constraint qualification discussed below), we are guaranteed that:*

$$\exists (\lambda_i)_{i \in \mathcal{E}} : \nabla f(x^*) = \sum_{i \in \mathcal{E}} \lambda_i \nabla c_i(x^*) \,.$$

That is, the gradient of the objective $\nabla f(x^*)$ lies within the span of the gradients of the constraints $\big(\nabla c_i(x^*)\big)_{i \in \mathcal{E}}$.

Proof. Suppose x^* is a local minimizer. Then, it must satisfy $c_i(x^*) = 0$ for each constraint $i \in \mathcal{E}$, and further, the objective f cannot increase along feasible directions d. We will define these feasible directions more carefully later in this section; however, it is straightforward to see that for each constraint c_i, a feasible direction at x^* must be orthogonal to $\nabla c_i(x^*)$, i.e.

$$d^\top \nabla c_i(x^*) = 0, \forall i \in \mathcal{E}\,.$$

If the objective f cannot decrease along feasible directions from x^*, we have $d^\top \nabla f(x^*) \geq 0$, and because there are only (differentiable) equality constraints in the problem, we may additionally conclude that

$$d^\top \nabla f(x^*) = 0\,.$$

Together, these imply that $\nabla f(x^*) \in \operatorname{span}\left((\nabla c_i(x^*))_{i \in \mathcal{E}}\right)$. □

If this were not the case, then the feasible set would locally take the shape of a cone in $\mathbb{R}^n$ with vertex at x^*, rather than a smooth manifold $\Omega := \{x : c_i(x) = 0, i \in \mathcal{E}\}$ with dimension less than n.

Theorem 2 considers equality-constrained problems and assumes that constraints are differentiable, which yields a feasible set Ω characterized by one or more smooth surfaces embedded within $\operatorname{dom}(x) \equiv \mathbb{R}^n$. Of course, when composed with differentiable *inequality* constraints, the feasible set Ω can include the interiors of these surfaces and possess volume, as illustrated in the example below.

Example 21 (Optimizing on the circle, continued)**.** *Suppose that the constraint in Example 20 is given by*

$$c(x) := 1 - \|x\|_2^2 \geq 0\,.$$

Geometrically, the feasible set Ω of this new problem is the boundary and interior of the circle shown in Figure 4. As in Example 20, the global minimizer occurs at $x^* = -(\sqrt{2}/2, \sqrt{2}/2)^\top$. Also as in Example 20, we observe that, at x^*, the gradients of the objective and constraint are parallel and satisfy $\nabla f(x^*) = \lambda \nabla c(x^*)$ for some value of $\lambda > 0$.

Inequality constraints can be *inactive*, *weakly active*, or *strongly active*.

Definition 5 (Active and inactive constraints)**.** *Consider a local optimizer x^* and an inequality constraint c_i, with $i \in \mathcal{I}$. There are three possibilities:*

- $c_i(x^*) \neq 0$*—In this case, we call the constraint* inactive *and observe that its presence does not (locally) affect the solution point x^*. That is, we could remove the constraint without affecting the solution.*
- $c_i(x^*) = 0$ *and* $\lambda > 0$*—In this case, the constraint is satisfied with equality at optimum, and the positive Lagrange multiplier tells us that, if the constraint were not present, we could improve the objective value by*

moving away from the boundary of the feasible set $\partial\Omega := \{x : c_i(x) = 0, i \in \mathcal{I}\}$*. Therefore, the constraint is* strongly active.[25]

- $c_i(x^*) = 0$ *and* $\lambda = 0$*—In this case, the constraint is satisfied with equality but the Lagrange multiplier indicates that, locally, the value of* f *cannot be improved by moving away from* $\partial\Omega$*. Such constraints are only* weakly active.

[25] We will almost always assume that we are operating with *strongly active* constraints, rather than weakly active ones. Handling weakly active constraints requires algorithmic subtlety which is beyond the scope of this monograph.

Recall the problem in Example 21. Because x^* satisfies the inequality constraint with equality (i.e. $c(x^*) = 0$), we call the constraint *active*. Moreover, the value of λ is strictly positive at this point; therefore, the constraint is *strongly* active.

The relationship between local minimizers x^* and Lagrange multipliers λ can be summarized by the following relation:

Beginning here, we will use the notation λ^* to denote a Lagrange multiplier specifically associated to a point x^*. Later, we will see that one may consider other values $\lambda \neq \lambda^*$ at intermediate stages of algorithms designed to find local optima x^*.

$$\forall i \in \mathcal{I} \begin{cases} c_i(x^*) \geq 0 \\ \lambda_i^* \geq 0 \\ \lambda_i^* \cdot c_i(x^*) = 0\,. \end{cases} \tag{15}$$

We refer to (15) as—from top to bottom—*primal feasibility*, *dual feasibility*, and *complementary slackness*. It is common to summarize these conditions with the *complementarity* relation

$$0 \leq c_i(x^*) \perp \lambda_i^* \geq 0\,, \tag{16}$$

which we read as: "both $c_i(x^*) \geq 0$ and $\lambda_i^* \geq 0$, and additionally $c_i(x^*) \perp \lambda_i^*$." Of course, since both c_i and λ_i^* are scalar-valued, orthogonality is equivalent to the bottom condition in (15). Later, we shall define vector-valued constraints by concatenating the $(c_i)_{i\in\mathcal{I}}$; in this case, orthogonality reduces to an elementwise relation due to the other nonnegativity constraints.

Thus equipped, we shall define the *Lagrangian* of problem (12):

Definition 6 (Lagrangian)**.** *Consider problem (12), repeated here for convenience:*

$$\begin{aligned} \min_{x\in\mathbb{R}^n} \quad & f(x) \\ \text{subject to} \quad & c_i(x) = 0, \quad \forall i \in \mathcal{E} \\ & c_i(x) \geq 0, \quad \forall i \in \mathcal{I}\,. \end{aligned}$$

For each $i \in \mathcal{E}$ *introduce scalar* $\lambda_i \in \mathbb{R}$*, and for each* $i \in \mathcal{I}$ *introduce nonnegative scalar* $\lambda_i \geq 0$*. Let the vector* $\lambda := (\lambda_i)_{i\in\mathcal{E}\cup\mathcal{I}} \in \mathbb{R}^{|\mathcal{E}|} \times \mathbb{R}^{|\mathcal{I}|}_{\geq 0}$*. The* Lagrangian *for this problem is the function:*

$$\mathcal{L}(x,\lambda) := f(x) - \sum_{i\in\mathcal{E}} \lambda_i c_i(x) - \sum_{i\in\mathcal{I}} \lambda_i c_i(x)\,. \tag{17}$$

Equipped with the Lagrangian in (17), we see that the relation $\nabla f(x^*) = \sum_i \lambda_i^* c_i(x^*)$ can be recovered from

$$0 = \nabla_x \mathcal{L}(x^*, \lambda^*) \quad (18a)$$
$$= \nabla f(x^*) - \sum_i \lambda_i^* \nabla c_i(x^*)\,. \quad (18b)$$

This condition is sometimes referred to as the "vanishing gradient" condition.

The linear independence constraint qualification

Question 3. *When do Lagrange multipliers exist such that Theorem 2 and (18) hold?*

Answer: *Under an appropriate* constraint qualification.

Consider the following example of where things can go wrong.

Example 22 (Why we need constraint qualifications)**.** *Consider the following problem, illustrated in Figure 5:*

$$\min_{x \in \mathbb{R}^2} \quad \overbrace{\mathbf{1}^\top x}^{f(x)}$$
$$\text{subject to} \quad c_1(x) = \|x\|_2^2 - 1 = 0$$
$$c_2(x) = -1 - x_2 \geq 0\,.$$

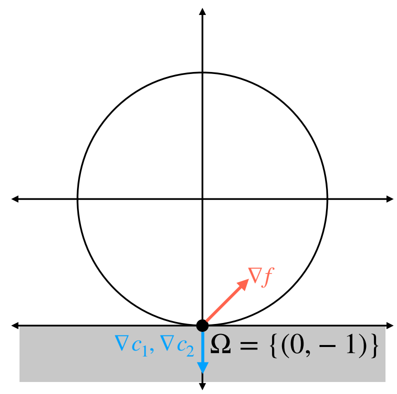

Figure 5: Illustration of objective and constraints in Example 22.

Here, the feasible set contains only a single point at the intersection of the two constraints, i.e. $\Omega = \{(0, -1)\}$, and therefore the optimal solution is $x^* = (0, -1)$. At this point, the gradients of both constraints are collinear and point downward; however, the gradient of the objective is oriented at a $45°$ angle. Clearly, in this case $\nabla f(x^*)$ does not lie in the span of $\{\nabla c_1(x^*), \nabla c_2(x^*)\}$, and there cannot be a set of Lagrange multipliers λ^* such that $\nabla_x \mathcal{L}(x^*, \lambda^*) = 0$.

The fundamental issue in Example 22 is that the gradients of the constraints are not an accurate description of the true constraint geometry near the point x^*. We formalize this idea below.

Definition 7 (Tangent)**.** *A* tangent *direction to the feasible set Ω at a point x^* is a vector d such that there exists a feasible sequence $\{z_k : z_k \in \Omega\}$ with $\lim_{k\uparrow\infty} z_k = x^*$ and a sequence of scalars $t_k > 0$ with $\lim_{k\uparrow\infty} t_k = 0$, which together satisfy the relation*

$$d = \lim_{k\uparrow\infty} \left(\frac{z_k - x^*}{t_k} \right)\,.$$

Definition 8 (Tangent cone)**.** *The set of all tangents to Ω at x^* is called the* tangent cone*. Formally, the tangent cone is defined as*

$$T_\Omega(x^*) := \{d : d \text{ satisfies Definition 7}\}\,.$$

Question 4. *Why is this called a "cone?"*
Answer: *By construction, if $d \in T_\Omega(x^*)$, then $\alpha d \in T_\Omega(x^*)$ as well, for any $\alpha > 0$.*

Equipped with Definition 8, we define local solutions as follows.

Definition 9 (Local solution)**.** *If x^* is a local solution, then*

$$d^\top \nabla f(x^*) \geq 0,\ \forall d \in T_\Omega(x^*)\,,$$

i.e., no feasible perturbation can improve the value of $f(x^*)$.

Recall Definition 5, and let us denote the set of equality and active inequality constraint indices as

$$\mathcal{A}(x^*) := \mathcal{E} \cup \{i : c_i(x^*) = 0, i \in \mathcal{I}\}\,.$$

Now, we recognize the limit in Definition 7 as a close relation to a gradient; ideally, we should therefore expect the tangent cone to resemble the span of the gradients of active constraints. These are called "linearized feasible directions."

Definition 10 (Linearized feasible directions)**.** *We denote the set of linearized feasible directions as*

$$\mathcal{F}(x^*) := \left\{ d : \begin{array}{l} d^\top \nabla c_i(x^*) = 0,\ \forall i \in \mathcal{E} \\ d^\top \nabla c_i(x^*) \geq 0,\ \forall i \in \mathcal{I} \cap \mathcal{A}(x^*) \end{array} \right\}.$$

Constraint qualifications are conditions under which $\mathcal{F}(x^*)$ coincides with $T_\Omega(x^*)$, i.e., conditions under which the gradients of active constraints accurately reflect local constraint geometry. We should expect that such conditions must hold if we are to employ the gradients of constraints when identifying optimal solutions x^* and corresponding Lagrange multipliers λ^*.

Many sets of such conditions exist and interested readers are directed to sources such as Nocedal and Wright [1999], Bertsekas [1999]. For our purposes, the following will suffice:

Definition 11 (Linear independence constraint qualification)**.** *The linear independence constraint qualification (LICQ) holds at x^* if the set $\{\nabla c_i(x^*) : i \in \mathcal{A}(x^*)\}$ is linearly independent.*[26]

[26] Note that this must imply that none of the constraint gradients are zero.

Reconsider the case of Example 22 in which we saw how $\nabla f(x^*) \notin \operatorname{span}(\{\nabla c_i(x^*) : i \in \mathcal{A}(x^*)\})$. We may check that the LICQ does not hold at x^*.

Example 23 (Broken LICQ)**.** *Recall the problem in Example 22. Because the feasible set includes only the point x^*, we trivially identify the tangent cone as*

$$T_\Omega(x^*) = \{0\}\,.$$

We also compute the set of linearized feasible directions as

$$\begin{aligned} \mathcal{F}(x^*) &= \{(a,0) : a \in \mathbb{R}\} \cap \{(a,b) : a \in \mathbb{R}, b \leq 0\} \\ &= \{(a,0) : a \in \mathbb{R}\}\,. \end{aligned}$$

Clearly, $\mathcal{F}(x^*) \neq T_\Omega(x^*)$. Therefore, we do not expect the LICQ to hold. Indeed, we see from Figure 5 that the constraint gradients $\nabla c_1(x^*)$ and $\nabla c_2(x^*)$ are collinear, violating the LICQ.

First- and second-order optimality conditions

Recall the definition of the Lagrangian from Definition 6. The Karush-Kuhn-Tucker (KKT) (necessary) conditions for first-order optimality are given by:

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = 0 \tag{19a}$$
$$c_i(x^*) = 0, \quad \forall i \in \mathcal{E} \tag{19b}$$
$$c_i(x^*) \geq 0, \quad \forall i \in \mathcal{I} \tag{19c}$$
$$\lambda_i^* \geq 0, \quad \forall i \in \mathcal{I} \tag{19d}$$
$$\lambda_i^* c_i(x^*) = 0, \quad \forall i \in \mathcal{I} \tag{19e}$$

The conditions in Equations (19c) to (19e) are collectively known as the "complementarity conditions" and are commonly expressed as

$$0 \leq \lambda_i^* \perp c_i(x^*) \geq 0, \ \forall i \in \mathcal{I}$$

per the discussion surrounding (16). We will often presume that "strict complementarity" holds, i.e., that

$$\lambda_i^* = 0 \text{ OR } c_i(x^*) = 0 \text{ but not both.} \tag{20}$$

Claim 1. *When the LICQ holds at a local solution x^*, there exist a set of Lagrange multipliers λ^* which satisfy the KKT conditions in (19).*

Claim 2. *When the LICQ holds at local solution x^*, the corresponding Lagrange multipliers λ^* satisfying (19) are unique.*

Regarding second-order optimality, a simple (sufficiency) condition is the following: whenever $\nabla^2 f(x^*) \succ 0$, i.e. $d^\top \nabla^2 f(x^*) d > 0$ for all d, and the KKT conditions are satisfied under an appropriate constraint qualification, we can conclude that x^* is a strict local minimizer. However, this is far more restrictive of a condition than is actually required. To build a tighter second order optimality condition, we first introduce a refinement of the tangent cone:

Definition 12 (Critical cone)**.** *The* critical cone $\mathcal{C}(x^*, \lambda^*)$ *is the set of feasible directions which strictly adhere to strongly active inequality constraints. Formally:*

$$\begin{aligned}
\mathcal{C}(x^*, \lambda^*) &:= \left\{ d \in \mathcal{F}(x^*) : d^\top \nabla c_i(x^*) = 0, \ \forall i \in \mathcal{A}(x^*) \cap \mathcal{I}, \lambda_i^* > 0 \right\} \\
&\equiv \left\{ d : \begin{array}{l} d^\top \nabla c_i(x^*) = 0, \ \forall i \in \mathcal{E} \\ d^\top \nabla c_i(x^*) = 0, \ \forall i \in \mathcal{A}(x^*) \cap \mathcal{I} \text{ with } \lambda_i^* > 0 \\ d^\top \nabla c_i(x^*) \geq 0, \ \forall i \in \mathcal{A}(x^*) \cap \mathcal{I} \text{ with } \lambda_i^* = 0 \end{array} \right\}.
\end{aligned}$$

From Definition 12, we see that

$$d \in \mathcal{C}(x^*, \lambda^*) \implies \lambda_i^* d^\top \nabla c_i(x^*) = 0, \; \forall i \in \mathcal{E} \cup \mathcal{I}\,. \tag{21}$$

We also recall that x^* and λ^* satisfy the KKT conditions, and in particular the vanishing gradient condition in (18). Taken together, we have:

$$d \in \mathcal{C}(x^*, \lambda^*) \implies d^\top \underbrace{\nabla f(x^*)}_{\sum_i \lambda_i^* \nabla c_i(x^*)} = 0\,. \tag{22}$$

Therefore, the critical cone contains precisely those feasible directions along which—locally—we cannot tell if $f(x^*)$ is increasing or decreasing. This suggests the following refinement of the second-order criterion above:

Proposition 6 (Second-order conditions)**.** *Suppose the LICQ is satisfied at a point x^* and that together (x^*, λ^*) satisfy the KKT conditions. If*

$$d^\top \nabla^2 f(x^*) d > 0, \; \forall d \in \mathcal{C}(x^*, \lambda^*)\,,$$

we conclude that x^* is a strict local minimizer.

Proof. (Sketch) We need only consider directions $d \in \mathcal{C}(x^*, \lambda^*)$ because along other directions first-order information is all we require to measure increase or decrease in $f(x^*)$. The remainder of the proof follows standard arguments in unconstrained cases. □

Duality

Duality is typically a key component of optimization courses. However, it will not factor into our discussion of smooth (dynamic) games very deeply; what follows is only intended to be a very general introduction.

Every constrained optimization problem of the form in (12) can be re-expressed as follows:

$$\begin{array}{ll} \min_{x \in \mathbb{R}^n} & f(x) \\ \text{subject to} & c(x) \geq 0 \end{array} \iff \min_{x \in \mathbb{R}^n} \max_{\lambda \geq 0} \overbrace{f(x) - \lambda^\top c(x)}^{\mathcal{L}(x,\lambda)}\,. \tag{23}$$

Here, note that we have ignored equality constraints to simplify notation, and concatenated inequality constraints into a vector valued function $c : \mathbb{R}^n \to \mathbb{R}^m$, with Lagrange multipliers $\lambda \in \mathbb{R}^m_{\geq 0}$.

Interpret the right hand side of (23) as follows: the function

$$\psi(x) := \max_{\lambda \geq 0} \mathcal{L}(x, \lambda) \tag{24}$$

will take the value positive infinity if and only if $c(x) \not\geq 0$. However, if $c(x) \geq 0$ and the constraint is satisfied at x, the maximizer in (24) is $\lambda^* = 0$, and we conclude that $\psi(x) = f(x)$.

Now consider a "dual" to (24):

$$\phi(\lambda) := \min_{x \in \mathbb{R}^n} \mathcal{L}(x, \lambda) \,. \tag{25}$$

Problem (23) is called the "primal" problem, and x the "primal" variable. The "dual" problem is given by

$$\max_{\lambda \geq 0} \underbrace{\min_{x} \mathcal{L}(x, \lambda)}_{\phi(\lambda)} \,, \tag{26}$$

and the multipliers λ is known as the "dual" variable. Note the following facts:

Claim 3. *$\phi(\lambda)$ is a concave function of λ, and the feasible set for λ in (26) is convex.*

Claim 4 (Weak duality)**.** *Regardless of the convexity of $f(x)$ or the constraint $c(x)$, we have:*

$$\max_{\lambda \geq 0} \underbrace{\min_{x} \mathcal{L}(x, \lambda)}_{\phi(\lambda)} \leq \min_{x} \underbrace{\max_{\lambda \geq 0} \mathcal{L}(x, \lambda)}_{\psi(x)} \,. \tag{27}$$

Claim 4 is, in fact, a direct corollary of Proposition 1: the upper value of a static game is greater than the lower value. The following is a key result in convex analysis:

Claim 5 (Strong duality)**.** *When $f(x)$ and $c(x)$ are convex functions of x, and the feasible set $\Omega := \{x : c(x) \geq 0\}$ has a nonempty relative interior, the inequality above holds with equality. These criteria are known as* Slater's condition*.*

Note that there is a close relationship between Claim 5 and saddle point equilibria (cf. Definition 1). Under Slater's condition, it is clear that $\mathcal{L}(x, \lambda)$ is convex in x (and it is always concave—in fact, affine—in λ). In zero-sum games with cost of this form, one can show that a saddle point solution exists and that lower and upper values coincide. Proving this result is left as an exercise.

Mixed strategy Nash equilibria, revisited

Equipped with a basic understanding of constrained optimization, let us now revisit the topic of mixed strategy equilibria. In Example 16, we saw that the computation of P1's strategy requires solving a linear, constrained optimization problem for each fixed strategy of P2. We shall return to this (more general) problem shortly, but first let us consider a zero-sum variant.

Example 24 (Mixed strategy saddle points)**.** *Consider a two-player, zero-sum matrix game with players' objectives defined as follows:*

$$\text{(P1):} \quad x_1^* \in \underset{x_1 \in \Delta}{\operatorname{argmin}} \; x_1^\top M x_2 \,, \qquad \text{(P2):} \quad x_2^* \in \underset{x_2 \in \Delta}{\operatorname{argmin}} \; - x_1^\top M x_2 \,. \tag{28}$$

Recall from Definition 1 that (x_1^*, x_2^*) constitute a saddle point solution to this game, if we interpret the x_2 in P1*'s problem as x_2^* and vice versa for* P2*.*

It turns out that, miraculously, we can transform the coupled problems in Example 24 into a single optimization problem.[27] We follow three steps: First, we recall the value of the game

[27] The forthcoming developments follow Başar and Olsder [1998] closely.

$$V(x_1) := \max_{x_2 \in \Delta} x_1^\top M x_2 \,, \tag{29}$$

and rescale P1's strategy accordingly, i.e. $\tilde{x}_1 := x_1 / V(x_1)$. Because $x_1 \in \Delta$, we know that

$$\mathbf{1}^\top \tilde{x}_1 = \frac{\mathbf{1}^\top x_1}{V(x_1)} = \frac{1}{V(x_1)} \,, \tag{30}$$

and, assuming (without loss of generality) that $M > 0$,

$$V(x_1) > 0 \implies \tilde{x}_1 \geq 0 \,. \tag{31}$$

Second, we observe that

$$V(x_1) = \max_{x_2 \in \Delta} x_1^\top M x_2 \geq x_1^\top M x_2, \forall x_2 \in \Delta \tag{32a}$$

$$\mathbf{1} \cdot V(x_1) \geq M^\top x_1 \tag{32b}$$

$$\iff \mathbf{1} \geq M^\top \tilde{x}_1 \,. \tag{32c}$$

In order to understand the conversion of the infinite set of inequalities in (32a) to the finite dimensional inequality in (32b), we recall that $x_2 \in \Delta$, and therefore it suffices to check (32a) at the vertices of the simplex.

Observe: the aforementioned transformation effectively eliminates P2's strategy, x_2, from the problem. Consequently, we solve for x_1 by seeking a security strategy for P1 as follows:

Note that we can construct an equivalent problem for P2.

$$\begin{aligned} \min_{x_1} \quad & V(x_1) & \iff \quad \max_{\tilde{x}_1} \quad & \mathbf{1}^\top \tilde{x}_1 & \text{(33a)} \\ \text{subject to} \quad & x_1 \geq 0 & \text{subject to} \quad & \tilde{x}_1 \geq 0 & \text{(33b)} \\ & \mathbf{1}^\top x_1 = 1 & & M^\top \tilde{x}_1 \leq \mathbf{1} & \text{(33c)} \end{aligned}$$

The problem on the right hand side of (33) is a linear program (LP), and can be solved efficiently via, e.g., a simplex algorithm which iteratively refines an estimate for which constraints are active at optimum.

The case of bimatrix games. In the more general case of bimatrix games, it will no longer be possible to transform the game into a single optimization problem. Consider the example below:

Example 25 (Mixed strategy Nash equilibria in bimatrix games)**.** *Consider now a general-sum variant of the problem in Example 24:*

$$\text{(P1):}\quad x_1^* \in \operatorname*{argmin}_{x_1\in\Delta}\; x_1^\top M_1 x_2\,,\quad \text{(P2):}\quad x_2^* \in \operatorname*{argmin}_{x_2\in\Delta}\; x_1^\top M_2 x_2\,. \tag{34}$$

Recall that the point (x_1^*, x_2^*) is a Nash equilibrium if both players' strategies solve the problems above, jointly.

In general, we must approach this problem and other more general cases via the players' optimality conditions. To that end, we construct their Lagrangians as follows:

$$\text{(P1):}\quad \mathcal{L}_1(x_1, x_2, \lambda_1, \mu_1) = x_1^\top M_1 x_2 - \lambda_1(\mathbf{1}^\top x_1 - 1) - \mu_1^\top x_1 \tag{35a}$$
$$\text{(P2):}\quad \mathcal{L}_2(x_1, x_2, \lambda_2, \mu_2) = x_1^\top M_2 x_2 - \lambda_2(\mathbf{1}^\top x_2 - 1) - \mu_2^\top x_2\,. \tag{35b}$$

Here, λ_i is the Lagrange multiplier for Pi's equality constraint ($\mathbf{1}^\top x_i - 1 = 0$), and μ_i is the Lagrange multiplier for the Pi's inequality constraint ($x_i \geq 0$). The corresponding KKT conditions for both players are:

$$\overbrace{M_1 x_2 - \lambda_1 \mathbf{1} - \mu_1}^{\nabla_{x_1}\mathcal{L}_1} = 0, \qquad \overbrace{M_2^\top x_1 - \lambda_2 \mathbf{1} - \mu_2}^{\nabla_{x_2}\mathcal{L}_2} = 0 \tag{36a}$$
$$\mathbf{1}^\top x_1 - 1 = 0, \qquad \mathbf{1}^\top x_2 - 1 = 0 \tag{36b}$$
$$0 \leq x_1 \perp \mu_1 \geq 0, \qquad 0 \leq x_2 \perp \mu_2 \geq 0 \tag{36c}$$

The conditions in (36) constitute a linear complementarity problem (LCP). LCPs are a class of variational inequality which include both linear programs (LPs) and quadratic programs (QPs).

Definition 13 (Linear complementarity problem)**.** *An linear complementarity problem (LCP) is an expression of the form:*

$$0 \leq z \perp Mz + m \geq 0 \iff \text{find} \quad w, z \tag{37a}$$
$$\text{subject to} \quad 0 \leq z \perp w \geq 0 \tag{37b}$$
$$w = Mz + m \tag{37c}$$

Proposition 7 (Every QP is a LCP)**.** *Consider a QP in standard form:*

$$\min_x \quad \frac{1}{2} x^\top Q x + q^\top x \tag{38a}$$
$$\text{subject to} \quad Ax \geq b \tag{38b}$$
$$x \geq 0\,. \tag{38c}$$

Problem (38) is an instance of a LCP, and can be expressed in the form of (37).

Proof. The Lagrangian of problem (38) is given by

$$\mathcal{L}(x, \lambda, \mu) = \frac{1}{2} x^\top Q x + q^\top x - \lambda^\top (Ax - b) - \mu^\top x\,, \tag{39}$$

and the corresponding KKT conditions are:

$$0 = \nabla_x \mathcal{L} = Qx + q - A^\top \lambda - \mu \tag{40a}$$
$$0 \le \lambda \perp Ax - b \ge 0 \tag{40b}$$
$$0 \le \mu \perp x \ge 0\,. \tag{40c}$$

Consider the following composite variables:

$$M = \begin{bmatrix} Q & -A^\top \\ A & 0 \end{bmatrix}, \quad m = \begin{bmatrix} q \\ -b \end{bmatrix}, \text{ and } z = \begin{bmatrix} x \\ \lambda \end{bmatrix}. \tag{41}$$

The LCP $0 \le z \perp Mz + m \ge 0$ is equivalent to

$$0 \le x \perp Qx - A^\top \lambda + q \ge 0 \tag{42a}$$
$$0 \le \lambda \perp Ax - b \ge 0 \tag{42b}$$

We immediately see that (42b) matches (40b). Examining (42a), we observe that either:

$$x = 0 \text{ and } Qx - A^\top \lambda + q \ge 0, \text{ or} \tag{43a}$$
$$x > 0 \text{ and } Qx - A^\top \lambda + q = 0\,. \tag{43b}$$

Recall that $\nabla_x \mathcal{L} = Qx - A^\top \lambda + q - \mu$. When the constraint $x \ge 0$ is active, the KKT conditions in (40) imply that $Qx - A^\top \lambda + q = \mu \ge 0$, which is equivalent to the condition in (43a). When the constraint $x \ge 0$ is inactive, we likewise conclude that $\mu = 0$ and recover $Qx - A^\top \lambda + q = 0$ as in (43b), Together, therefore, Equations (43a) and (43b) recover the remaining KKT conditions from Equations (40a) and (40c). □

We conclude this section with a discussion of the parametric dependence of mixed Nash equilibria upon the matrices M_1 and M_2 in Example 25. As we shall see, we may understand this relationship by examining the structure of the LCP in (36). The KKT conditions in (36) *implicitly* specify the relationships that primal and dual variables must satisfy, and how those relationships depend upon M_1 and M_2. Consider the following illustrative example.

Example 26 (Parametric RPS)**.** *Recall the rock-paper-scissors (RPS) game from Examples 14 and 15, and consider a situation in which a third party can modify the cost—θ—which* P1 *incurs when it plays "rock" and* P2 *plays "paper." This modification results in a general-sum, bimatrix game with* *cost matrices:*

$$M_1 = \begin{bmatrix} 0 & 1 & \theta \\ -1 & 0 & 1 \\ 1 & -1 & 0 \end{bmatrix} \text{ and } M_2 = \begin{bmatrix} 0 & -1 & 1 \\ 1 & 0 & -1 \\ -1 & 1 & 0 \end{bmatrix}. \tag{44}$$

Examine the KKT conditions for both players in (36). Suppose, for the sake of argument, that for a particular value of θ, the corresponding Nash strategies (x_1^*, x_2^*) are completely mixed, *i.e., they have no zero entries. In this case, we would conclude that $\mu_1, \mu_2 = 0$ and can ignore (36c), leaving only Equations (36a) and (36b) to determine the relationship among $(x_1^*, x_2^*, \lambda_1^*, \lambda_2^*, \theta)$. These equations are linear, and can be expressed as follows:*

$$\begin{bmatrix} 0 & M_1 & -\mathbf{1} & 0 \\ M_2^\top & 0 & 0 & -\mathbf{1} \\ \mathbf{1}^\top & 0 & 0 & 0 \\ 0 & \mathbf{1}^\top & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1^* \\ x_2^* \\ \lambda_1^* \\ \lambda_2^* \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 1 \end{bmatrix} \tag{45}$$

Solving this system of equations reveals that, in the neighborhood of the point $(x_1^*, x_2^*, \lambda_1^*, \lambda_2^*, \theta)$:

$$x_1^* = -M_2^{-\top}\mathbf{1}(\mathbf{1}^\top M_2^{-\top}\mathbf{1})^{-1} \tag{46a}$$

$$x_2^* = -M_1^{-\top}\mathbf{1}(\mathbf{1}^\top M_1^{-\top}\mathbf{1})^{-1}. \tag{46b}$$

From (46a), we see that x_1^* does not depend upon M_1 and therefore is—at least locally—insensitive to variations in the parameter θ.

This phenomenon is very interesting, at least to me. It implies that, for small perturbations to P1's objective, at Nash P2 will act in a way which compensates for any new incentive P1 may have to deviate from x_1^*.

The general case: mixed complementarity problems

So far, we have only seen *linear* complementarity programs (LCPs) arise from players' KKT conditions in a bimatrix game. More generally, when players have nonlinear objectives and/or constraints, the result is a mixed complementarity problem (MCP). In general, these problems take the form:

$$x_1^* \in \quad \operatorname*{argmin}_{x_1} \quad f_1(x_1, x_{\neg 1}) \tag{47a}$$

$$\text{subject to} \quad c_1(x_1, x_{\neg 1}) = 0 \tag{47b}$$

$$h_1(x_1, x_{\neg 1}) \geq 0 \tag{47c}$$

$$\vdots$$

$$x_N^* \in \quad \operatorname*{argmin}_{x_N} \quad f_N(x_N, x_{\neg N}) \tag{47d}$$

$$\text{subject to} \quad c_N(x_N, x_{\neg N}) = 0 \tag{47e}$$

$$h_N(x_N, x_{\neg N}) \geq 0 \tag{47f}$$

$$\text{subject to} \quad c(x_1, \dots, x_N) = 0 \tag{47g}$$

$$h(x_1, \dots, x_N) \geq 0. \tag{47h}$$

Without the constraints, problem (47) is equivalent to a Nash equilibrium problem (Definition 2). However, the presence of the constraints introduces additional subtlety.

First, observe that there are two distinct types of constraints in (47): private and shared. The private constraints in Equations (47b), (47c), (47e) and (47f) are each "owned" by a single player, and will appear in only that player's Lagrangian (and KKT conditions). On the other hand, the constraints in Equations (47g) and (47h) are shared among all N players.[28]

[28] For simplicity, we have assumed that these constraints are shared by *all* players. In practice, of course, one can also consider constraints that are shared by subsets of players. The discussion below will also apply in that setting.

These latter constraints present a challenge: how should we construct each individual player's Lagrangian to incorporate these constraints? As we shall see, each of these constraints need only be assigned a single Lagrange multiplier (rather than a separate multiplier for every player).

Question 5. *What is wrong with assigning every player a separate multiplier for the constraints in Equations (47g) and (47h)?*

Answer: *As we shall see, the KKT conditions for (47) would be underdetermined in that case, indicating that many possible values of the Lagrange multipliers may simultaneously satisfy the necessary conditions.*

Before proceeding to construct players' Lagrangians and the corresponding KKT conditions, we first state the following definitions as direct analogues of Definitions 2 and 4.

Definition 14 (Generalized Nash equilibrium)**.** *A generalized Nash equilibrium (GNE) problem is a game of the form (47). A GNE solution is a point $(x_i^*)_{i=1}^N$ jointly minimizing each player's objective and satisyfing all private and shared constraints.*

Definition 15 (Local generalized Nash equilibrium)**.** *A local GNE is a point $(x_i^*)_{i=1}^N$ in which x_i^* only* locally *minimizes* Pi*'s objective along directions in* Pi*'s critical cone at $(x_i^*)_{i=1}^N$.*

Thus equipped, we introduce multipliers for each constraint in (47)—including λ_{sh} and μ_{sh} for the shared equality and inequality constraints, respectively—and write out each player's Lagrangian

Recall: $\mathbf{x} := (x_i)_{i=1}^N$, and similarly we will denote $\boldsymbol{\lambda} := (\lambda_i)_{i=1}^N, \boldsymbol{\mu} := (\mu_i)_{i=1}^N$.

$$\mathcal{L}_i(\mathbf{x}, \lambda_i, \mu_i, \lambda_{\text{sh}}, \mu_{\text{sh}}) = f_i(\mathbf{x}) - \lambda_i^\top c_i(\mathbf{x}) - \mu_i^\top h_i(\mathbf{x}) - \lambda_{\text{sh}}^\top c(\mathbf{x}) - \mu_{\text{sh}}^\top h(\mathbf{x}) \quad (48)$$

and derive the corresponding KKT conditions:

$$\forall i \in [N] \begin{cases} 0 = \nabla_{x_i} \mathcal{L}_i(\mathbf{x}, \lambda_i, \mu_i, \lambda_{\text{sh}}, \mu_{\text{sh}}) \\ 0 = c_i(\mathbf{x}) \\ 0 \le h_i(\mathbf{x}) \perp \mu_i \ge 0 \end{cases} \quad (49)$$
$$0 = c(\mathbf{x})$$
$$0 \le h(\mathbf{x}) \perp \mu_{\text{sh}} \ge 0\,.$$

Note that, per the discussion in Local constraint geometry, we do require a constraint qualification to hold in order to claim that the conditions in (49) hold at every GNE point satisfying Definition 14.

Observe that these KKT conditions may be concatenated into the following form, for appropriate functions G and H:

$$0 = G(\mathbf{z}, \mathbf{y}) \tag{50a}$$
$$0 \leq H(\mathbf{z}, \mathbf{y}) \perp \mathbf{y} \geq 0\,. \tag{50b}$$

Problem (50) is the standard form for a MCP,[29] and we interpret

$$\mathbf{z} = (\mathbf{x}, \boldsymbol{\lambda}, \lambda_{\text{sh}}), \mathbf{y} = (\boldsymbol{\mu}, \mu_{\text{sh}})\,,$$

and G, H to contain the corresponding expressions in (49).

For completeness, we also note that it is also common to express MCPs via the equivalent expression, in terms of a single vector-valued function F operating on all variables $\mathbf{w} = (\mathbf{z}, \mathbf{y})$:

$$\underline{\mathbf{w}} \leq \mathbf{w} \leq \overline{\mathbf{w}} \perp F(\mathbf{w})\,. \tag{51}$$

This expression is interpreted as follows, elementwise:

- If $w_i = \underline{w}_i$, then $[F(\mathbf{w})]_i \geq 0$.
- If $w_i = \overline{w}_i$, then $[F(\mathbf{w})]_i \leq 0$.
- If $\underline{w}_i < w_i < \overline{w}_i$, then $[F(\mathbf{w})]_i = 0$.

In particular, for problems of the form (49), we can take the i^{th} element of $\mathbf{w}$, w_i, to be either zero or negative infinity (depending upon which variable in (49) it corresponds to). We conclude this chapter with the following example.

Example 27 (A trajectory game)**.** *Consider the following* trajectory game*, in which N agents interact in a physical space over time. The agents—which could represent drivers on the road, for example—may each select a sequence of states and control inputs (i.e., a trajectory). That is,* Pi*'s decision variable x_i consists of sequences of state and control variables for that player, and must satisfy private constraints which enforce, e.g., dynamic feasibility (physics), staying on road, maintaining speed limits, etc.*[30] *A state of the art toolchain for formulating and solving these types of games is publicly available.*[31]

Note: games of this type are more appropriately termed "open-loop, dynamic" (generalized) Nash games. We will study dynamic games in the following chapters, and focus particularly on the role of information structure which will determine when and how we can find equilibria via mixed complementarity methods and when other algorithmic ideas are needed.

[29] Francisco Facchinei and Jong-Shi Pang. *Finite-dimensional variational inequalities and complementarity problems*. Springer, 2003

[30] Collision avoidance is also a good example of a constraint that can appear in these problems. However, note that collision avoidance pertains to pairs of players and therefore presents several options: a single player can be responsible for ensuring that the pair avoids collision, or both can be responsible (and share a multiplier).

[31] Lasse Peters and Xinjie Liu. MCPTrajectoryGames.jl. URL `https://github.com/JuliaGameTheoreticPlanning/MCPTrajectoryGameSolver.jl`. Accessed: September 2023

Dynamic, Finite Games

Now that we have established a foundation for understanding static games, we turn to more interesting, dynamic settings. Besides players' objectives, dynamic games are defined by two additional features:

- *State*—players make decisions sequentially over time in a dynamic game, and these decisions impact the game's underlying state according to rules, *dynamics*.
- *Information structure*—when each player makes a decision in a dynamic game, that decision is based upon a very specific set of available information.

In this chapter, we will introduce these and other related fundamentals in the context of games with finitely many (discrete) states and actions, and provide a cursory overview of search-based algorithms for finding equilibria of these games. This chapter is only intended as a brief introduction to several of the main ideas: future chapters will go into greater detail on some of the more interesting directions.

A brief word on notation. We will use x_t to denote the state of the game at time t, and u_t^i to denote Pi's action/control/input at time t. Additionally, we will use the following shorthand:

$$\mathbf{x} := (x_1, \dots, x_T) \tag{52a}$$
$$\mathbf{u}_t := (u_t^1, \dots, u_t^N) \tag{52b}$$
$$\mathbf{u}^i := (u_1^i, \dots, u_T^i) \tag{52c}$$
$$\mathbf{u} := (\mathbf{u}^1, \dots, \mathbf{u}^N)\,. \tag{52d}$$

Extensive form

As in the case of static games, our first order of business will be to discuss a common representation for dynamic games. This representation is called "extensive form," and is expressed as a tree with nodes corresponding to states and branches corresponding to actions. Consider the following example.

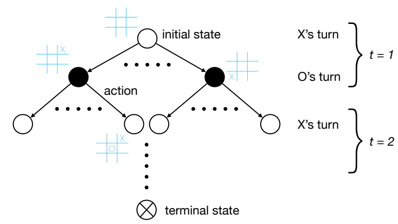

Figure 6: Extensive form of tic tac toe.

Example 28 (Tic tac toe)**.** *Figure 6 illustrates the extensive form corresponding to a game of tic tac toe.*

As shown, the tree is rooted at the initial state, corresponding to an empty board. Each player takes actions sequentially, at alternating levels of the tree—time t only increments after both (all) players act. Eventually, the game will end and we call that state "terminal."

Information patterns

The information pattern (or structure) of a dynamic game determines what each player knows at each time t when it must take an action. In particular, two games with the same extensive form can possess different information patterns, and thus admit different strategies and equilibria. Consider the following simple example.

Example 29 (Information structure of a matrix game)**.** *Suppose* P1*'s cost in a (zero-sum) matrix game is determined by*

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} .$$

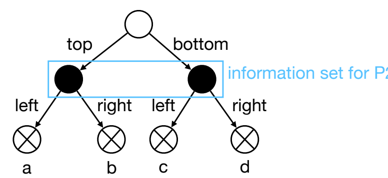

Figure 7: Extensive form for saddle point matrix game.

In this case, there is no state, and only a single time step. Suppose that we wish to find a saddle point (Nash) equilibrium of this game, in which both players must select a strategy simultaneously, i.e. $u^1 \in \{\text{top}, \text{bottom}\}$ *and* $u^2 \in \{\text{left}, \text{right}\}$.

We construct the extensive form of this game as shown in Figure 7. As shown, P1 *first chooses a row, and* then P2 *chooses a column. However, in order to render this game strategically equivalent to a simultaneous (saddle point) problem, we specify that* P2 cannot distinguish *the states within the blue rectangle. Consequently,* P2 *cannot base its strategy upon* P1*'s decision.*

The blue rectangle in Figure 7 illustrates P2's *information set*, defined as follows.

Definition 16 (Information set)**.** *An information set for* Pi *is a set of sets*

$$\mathcal{N}^i := \{\eta_k^i\} ,$$

where η_k^i *is the* k^{th} *set of states* Pi *cannot distinguish between.*

Now, let us reconsider the matrix game of Example 29 under a different information structure for P2.

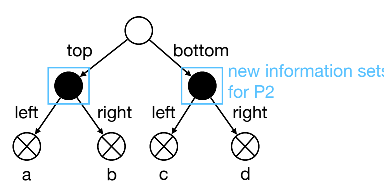

Figure 8: Extensive form with revised information sets for P2.

Example 30 (Information structure of a matrix game, continued)**.** *Suppose that* P2 *can distinguish between the two nodes within the blue box* *in Figure 7, yielding the information structure in Figure 8. There are still two (pure) strategies available to* P1*, i.e.,* $u^1 \in \{\text{top}, \text{bottom}\}$*. However, we will shortly understand that* P2 *can choose among* four *(pure) strategies: for each action of* P1*,* P2 *can select either of its actions in response.*

So far, we have only seen examples of *single-act* games in Examples 29 and 30. In general, of course, we will be interested in *multi-act* games in which all players take actions over multiple time steps $t \in \{1, 2, \dots, T\} \equiv [T]$. Although one can readily imagine a number of information structures in such games, we will focus our attention upon two extreme cases: feedback and open-loop structures.

Definition 17 (Feedback information structure)**.** *A multi-act game is a feedback game if players always know the time step t, each information set of* P1 *is a singleton, and every information set for other players at t includes all the nodes emanating from a single* P1 *information set at t. In short: every player knows the time t and the game state x_t.*

Definition 18 (Open-loop information structure)**.** *A multi-act game is open-loop if each player only has a single information set at each time t. Equivalently: each player cannot observe the game state at any t besides $t = 1$, but is always aware of the time t.*

Strategies

We can now define the concept of a "strategy" more formally, within the context of a dynamic game.

Definition 19 (Strategy)**.** *A (pure) strategy γ^i for* Pi *is a map from that player's information sets $\mathcal{N}^i$ to its action set $\mathcal{U}^i$, i.e.*

$$\gamma^i : \mathcal{N}^i \to \mathcal{U}^i \,.$$

We will also commonly index a player's strategy by time: γ^i_t denotes the aforementioned map when restricted to the t^{th} time step, and we will treat $\gamma^i \equiv (\gamma^i_1, \dots, \gamma^i_T)$ since any information set $\eta^i_k \in \mathcal{N}^i$ implicitly specifies a corresponding time step t.

Let us revisit Example 30 in the context of Definition 19.

Example 31 (Information structure of a matrix game, continued)**.** *Consider the game in Figure 8. The (pure) strategies available to* P2 *are*

$$\gamma^2 \in \{\text{LL}, \text{LR}, \text{RL}, \text{RR}\} \,,$$

where we interpret strategy LL *as the map*

$$\text{LL} \equiv \begin{cases} u^1 = \text{top} \to u^2 = \text{left} \\ u^1 = \text{bottom} \to u^2 = \text{left} \end{cases}$$

and likewise for the other three strategies. Consequently, we may understand this game as a 2×4 *matrix game, with cost matrix*

$$\begin{bmatrix} a & a & b & b \\ c & d & c & d \end{bmatrix}. \tag{53}$$

Each row in this matrix corresponds to one of the strategies (actions, in this case) available to P1*, and each column to one of the strategies for* P2*.*

Thus far we have only discussed pure strategies in dynamic games.

Question 6. *Suppose we found a mixed strategy saddle point in the matrix game (53). Would such a mixed strategy always be a rational choice?*

Answer: *No. For example, suppose that* P2*'s mixed strategy at equilibrium was given by the vector* $(0.5, 0.5, 0, 0)$*, or equivalently, a* $50/50$ *split between* $\{\mathrm{LL}, \mathrm{LR}\}$*. Imagine playing this strategy: suppose that* P1 *chooses* top—P2 *would then end up first randomly choosing between* LL *and* LR*, and then executing that strategy (and in this case, deterministically choosing* left*). This is not really appropriate: what* P2 *should do instead is consider mixing between its available* actions *that emanate from each information set, rather than mixing among the pure strategy maps from Definition 19.*

Definition 20 (Behavioral strategy)**.** *A behavioral strategy is a mixed strategy in a dynamic game, wherein mixing only occurs among the actions which emanate from individual information sets.*

Consider the following example.

Example 32 (Behavioral strategies in Example 31)**.** P1 *could mix between* top *and* bottom *directly.* P2 *should only consider mixing between the actions* left *and* right *within each information set in Figure 8, i.e.* P2 *should choose a strategy of the form*

$$\gamma^2(\overbrace{u^1}^{\equiv \eta^2}) = \begin{cases} (\text{left}, \text{right}) \text{ with probability } (p, 1-p), & u^1 = \text{top} \\ (\text{left}, \text{right}) \text{ with probability } (q, 1-q), & u^1 = \text{bottom} \end{cases}$$

The map γ^2 *is called a* behavioral strategy.

A brute force solution method

Suppose that we wish to solve a multi-act game with known information structure. A natural way to construct a solution will be to work from the bottom of the extensive form to the top, and solve subgames emanating from each information set recursively.

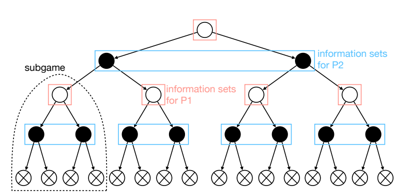

Figure 9: Two-stage finite game, with feedback information structure.

Consider the two-stage game shown in Figure 9, and begin by analyzing the subgame on the far left, as shown. This game (like every other subgame beginning at the second stage) is equivalent to a (bi)matrix game of the form in Example 29. Suppose that we identify a (mixed) Nash equilibrium of this subgame by solving an appropriate LP or LCP, per the discussion in Static, Smooth, Constrained Games. Each such mixed strategy pair can be understood as the evaluation of a behavioral strategy at the root of the corresponding subgame. Furthermore, the expected value of the resulting subgame can be computed for each player, and associated to that root. Finally, let us consider the root of the entire game: once each player selects an action, the state will evolve toward one of the four subgames we solved initially. We can therefore construct a fifth (bi)matrix game by arranging the outcomes of these four subgames in a 2×2 matrix for each player, and find a (mixed) Nash solution by solving the resulting LP or LCP. The resulting mixed strategies are again understood as the evaluation of a behavioral strategy for each player at the root of the entire extensive form.

Ultimately, therefore, this two-stage recursion yields a behavioral strategy for each player which satisfies the Nash equilibrium conditions for each subgame. Because the information pattern for the game in Figure 9 is feedback, we additionally call the resulting equilibrium a "feedback Nash equilibrium."

Remark 1 (Feedback Nash equilibria)**.** *Note that the use of the term "feedback" in the phrase "feedback Nash equilibria" should be interpreted two ways. First, we mean to say that the game being played has a* feedback information pattern*. Second, and more subtly, we also imply that the strategies satisfy a subgame structure, i.e., at time t the players understand that their future decisions at $s > t$ must also be in equilibrium with one another. This latter condition is perhaps easiest to understand in the context of the dynamic programming procedure explained above: if one constructs a solution recursively by working backward from the end of time to the beginning, the result will satisfy this subgame equilibrium condition. We will defer a more detailed treatment of this concept for Smooth Dynamic Games.*

Monte Carlo tree search

Clearly, the brute force recursion described above involves solving a potentially large number of subproblems. The size of each subproblem scales linearly with the number of actions available to each player; the number of subproblems also grows as players have more actions available, but more importantly, it scales exponentially in

Algorithm 3: Monte Carlo tree search (feedback information)

1 **Input:** root x_1, maximum computation time
2 **while** *computation time remains* **do**
3 $\quad x_{\text{new}} \leftarrow x_1$ ▷ Start at the root
4 $\quad$ **while** x_{new} *is not a leaf* **do**
5 $\quad\quad x_{\text{new}} \leftarrow$ outcome of strategies from x_{new} ▷ Fictitious play
6 $\quad$ **if** x_{new} *is not terminal* **then**
7 $\quad\quad x_{\text{new}} \leftarrow$ random child of x_{new} ▷ Expand
8 $\quad\quad$ simulate a game played from x_{new} ▷ Simulate
9 $\quad$ backpropagate outcomes to all ancestors ▷ Backpropagate
10 **return** tree rooted at x_1

the game horizon (number of stages) T. This means that brute force approaches are ill-suited to large, and particularly long-horizon, problems—e.g., chess or go.

Nevertheless, a family of stochastic algorithms known as Monte Carlo tree search (MCTS) are widely used to approximate solutions in these cases, and have been successfully deployed in games such as chess and go (and others!). The most basic version of MCTS requires repeating five steps *ad infinitum* or until a pre-specified amount of computation time has elapsed, cf. Algorithm 3. For simplicity, we will present a variant of MCTS for turn-based games (e.g., chess), which transpire over a variable time horizon, and in which there are only a finite number of outcomes.

Start at the root

Per Line 3, each iteration of MCTS begins at the root of the extensive form tree.[32] Each episode therefore represents a complete (hypothetical) game.

[32] One may naturally wonder whether there are circumstances under which it makes sense to begin iterations from intermediate nodes in the tree.

Fictitious play

Ultimately, the tree which results from Algorithm 3 encodes not just the extensive form of the game, but also (and more importantly) a set of behavioral strategies which are in equilibrium with one another. To this end, each node x records the following information:

- $N^i_j(x)$ is the number of simulated games which have passed through x and had outcome v_j for player Pi
- $N(x) = \sum_{i,j} N^i_j(x)$

- $x^+ = f(x, u^i)$ is the node which arises when Pi takes action u^i from x

There are a variety of approaches to designing strategies for each player given access to this information. A common choice is the upper confidence tree (UCT) rule, wherein Pi's strategy is

$$\gamma^i(x) := \underset{u^i}{\operatorname{argmin}} \left(\overline{V}^i(x^+) - \beta \sqrt{\frac{\log N(x)}{N(x^+)}} \right), \tag{54}$$

where $\overline{V}^i(x^+) = \left(\sum_j v_j N^i_j(x^+)\right) / N(x^+)$ is an empirical estimate of Pi's outcome from x^+. Likewise, the term at the right is an estimate of the uncertainty in this estimate: as the number of simulated games increases, this term diminishes toward 0. The parameter $\beta > 0$ controls the degree to which Pi treats this uncertainty *optimistically*: larger values of β imply that Pi will be more inclined to take risky actions u^i whose expected outcome is less predictable.[33]

[33] Randomized algorithms such as MCTS are generally characterized by an "exploration-exploitation tradeoff" of some kind.

During fictitious play (Line 5), each player selects an action according to a strategy such as the UCT rule (54) sequentially, until reaching a leaf node. A *leaf node* is either (a) terminal, or (b) a node with unexplored potential children.

Expand

Once we have identified a leaf node, if it is not terminal, then choose a successor (child) node by selecting an arbitrary (e.g., uniformly random) action playable from the current leaf node.

Simulate

Simulate (e.g., by selecting playable actions uniformly at random) a game played from the aforementioned child node until termination, and record the outcome v_j for each player Pi.

Backpropagate

For each node in the chain of ancestors linking the child node to the root, increment counters N^i_j according to the simulation outcome.

Accelerating tree search: value function estimation

Reconsider each step of MCTS: the most naïve step in Algorithm 3 is Line 8, in which we simulate the outcome of a random game played from node x_{new}. Often, when playing a game we as humans develop an intuition for which states (i.e., board positions) are more favorable than others. We rely upon this intuition to simplify the process of planning, precisely by removing the need to simulate many moves possible into the future. Algorithmically, it is increasingly common to employ machine learning methods to build an estimator for the value function $\overline{V}^i(x)$, which ultimately serves the same purpose as that intuition. This value function estimation underlies the success of systems such as AlphaGo.[34]

[34] David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017

Interlude: Subtleties of Equilibrium, Information, and Time

So far we have seen several of the key ideas underlying dynamic games, but have yet to see much of the subtlety. This chapter provides a brief glimpse into several of the more interesting of these concepts: refinements to the Nash equilibrium concept, informational inferiority, and time consistency.

The trembling hand equilibrium

While they remain our general focus in this text, Nash equilibria are problematic for many reasons. Consider the following example, which illustrates how a "dominated" Nash point can arise:

Example 33 (Dominated Nash equilibrium)**.** *Imagine a bimatrix game in which each player's cost matrix is given by*

$$\begin{bmatrix} 0 & 1 \\ 1 & 1 \end{bmatrix}.$$

There are two NE: one in the top-left with value 0*, and the other in the bottom-right with value* 1*. Clearly, both players would prefer to operate in the top-left and attain lower cost.*

The "trembling hand" equilibrium concept[35] is an attempt to refine the Nash concept in such a way as to remove these kinds of "dominated" equilibria. That is, all trembling hand equilibria are Nash equilibria, but not the converse.

[35] Reinhard Selten. *Reexamination of the perfectness concept for equilibrium points in extensive games*. Springer, 1988

Definition 21 (Trembling hand equilibrium)**.** *Consider a perturbed game in which, when* Pi *wishes to choose pure strategy* x_i*, it errs with probability* $\epsilon < 1$ *and, in that event, chooses a different action* $\tilde{x}_i \neq x_i$ *uniformly at random. A (pure) Nash equilibrium is a* trembling hand equilibrium *if it remains Nash in the limit as* $\epsilon \downarrow 0$*.*

We now demonstrate that the top-left point in Example 33 is the only trembling hand equilibrium in that game.

Example 34 (Trembling hand computation)**.** *One may readily compute that the cost matrix for each player in the perturbed game is:*

$$\begin{bmatrix} 2\epsilon - \epsilon^2 & \epsilon^2 - \epsilon + 1 \\ \epsilon^2 - \epsilon + 1 & 1 - \epsilon^2 \end{bmatrix}.$$

In the limit as $\epsilon \downarrow 0$, we observe that $\epsilon^2 - \epsilon + 1 < 1 - \epsilon^2$ (and therefore the bottom-right point cannot be a Nash point) and $\epsilon^2 - \epsilon + 1 > 2\epsilon - \epsilon^2$ (indicating that the top-left point remains Nash).

Other equilibrium refinements. There are many, many other refinements of the Nash equilibrium (and others)! This text is by no means complete.

Informational inferiority

As we have seen, information structure plays a pivotal role in determining the form of players' strategies in dynamic games. Consider the following setting:

Definition 22 (Informational inferiority)**.** *Suppose two games,* I *and* II*, are identical in all respects except for their information structure, and that the strategy spaces for* Pi*,* Γ_{I}^i *and* Γ_{II}^i*, satisfy the relations*

$$\Gamma_{\mathrm{I}}^i \subseteq \Gamma_{\mathrm{II}}^i \ \forall i\,, \text{ and } \exists i \text{ s.t. } \Gamma_{\mathrm{I}}^i \subsetneq \Gamma_{\mathrm{II}}^i\,.$$

Then, I *is* informationally inferior *to* II*.*

There are two important properties of Nash equilibria in these settings.

Proposition 8. *Any NE of game* I *is also a NE in game* II*.*

Proof. We offer a proof by contradiction. Suppose that $\gamma_{\mathrm{I}}^* = (\gamma_{\mathrm{I}}^{i*})_{i=1}^N$ (with $\gamma_{\mathrm{I}}^{i*} \in \Gamma_{\mathrm{I}}^i$) is a NE in game I. Without loss of generality, suppose $\exists \gamma_{\mathrm{II}}^N \in \Gamma_{\mathrm{II}}^N \backslash \Gamma_{\mathrm{I}}^N$ with which PN attains lower cost, i.e., $J^N(\gamma_{\mathrm{I}}^*) > J^N(\gamma_{\mathrm{I}}^{1*}, \ldots, \gamma_{\mathrm{I}}^{(N-1)*}, \gamma_{\mathrm{II}}^N)$. This can only occur if the change in strategy for PN results in a different game trajectory.

Denote $\tilde{\eta}_{\mathrm{II}}^N$ the information set of game II corresponding to PN which contains a node on this trajectory at which the strategies yield different actions, and likewise for $\tilde{\eta}_{\mathrm{I}}^N$. There must exist a strategy $\overline{\gamma}_{\mathrm{I}}^N \in \Gamma_{\mathrm{I}}^N$ such that $\overline{\gamma}_{\mathrm{I}}^N(\tilde{\eta}_{\mathrm{I}}^N) = \gamma_{\mathrm{II}}^N(\tilde{\eta}_{\mathrm{II}}^N)$.[36] Therefore, we have that

$$J^N(\gamma_{\mathrm{I}}^{1*}, \ldots, \gamma_{\mathrm{I}}^{(N-1)*}, \gamma_{\mathrm{II}}^N) = J^N(\gamma_{\mathrm{I}}^{1*}, \ldots, \gamma_{\mathrm{I}}^{(N-1)*}, \underbrace{\overline{\gamma}_{\mathrm{I}}^N}_{\in \Gamma_{\mathrm{I}}^N}) < J^N(\gamma_{\mathrm{I}}^*)\,.$$

Question 7. *What does the condition* $\Gamma_{\mathrm{I}}^i \subseteq \Gamma_{\mathrm{II}}^i$ *imply about the information sets in each game?*

Answer: $\Gamma_{\mathrm{I}}^i \subseteq \Gamma_{\mathrm{II}}^i$ *means that every strategy* Pi *can play in* I *is also available in* II*. Because strategies are maps from information sets to actions, this implies that the information sets of* Pi *in game* II *are* subsets *of those in* I*. To see this connection, observe that for a strategy* $\gamma^i \in \Gamma_{\mathrm{I}}^i$*, one may construct an equivalent strategy in* Γ_{II}^i *which—for every information set* η_{I}^i *in game* I *and corresponding action* $\gamma^i(\eta_{\mathrm{I}}^i)$*—maps each information set* $\eta_{\mathrm{II}}^i \subseteq \eta_{\mathrm{I}}^i$ *to the same action.*

[36] This auxiliary strategy $\overline{\gamma}_{\mathrm{I}}^N$ can simply output a constant action which matches $\gamma_{\mathrm{II}}^N(\tilde{\eta}_{\mathrm{II}}^N)$ for every possible input.

This implies that γ_{I}^* could not have been a NE of game I; we have established contraction and the proof is complete. □

It is also possible to establish a similar "converse" result by a similar argument.

Proposition 9. *If $\gamma_{\mathrm{II}}^* = (\gamma_{\mathrm{II}}^{i*})_{i=1}^N$ is a NE in* II *such that $\gamma_{\mathrm{II}}^{i*} \in \Gamma_{\mathrm{I}}^i\ \forall i$, then γ_{II}^* is also a NE in game* I.

Though abstract, these results have very clear implications (and subtle misinterpretations) in the context of feedback and open loop games. Consider the following example.

Example 35 (Informational inferiority in feedback and open-loop games)**.** *Recall Definitions 17 and 18, which define the feedback and open-loop information structures, respectively. From the definitions, it is clear that each information set in a feedback game (e.g., at time t) is a* subset *of an information set in the same game with an open-loop information pattern. Therefore, from Definition 22 we say that open-loop game is informationally inferior to the corresponding feedback game.*

In this context, the result in Proposition 8 indicates that every NE of the open-loop game is also Nash in the feedback game. It is clear that, for every open-loop strategy there is an equivalent feedback strategy which simply disregards its input at each time t and regurgitates whatever action the open-loop strategy would have taken at that time. It is also straightforward to verify that, with all players other than Pi *playing strategies corresponding to an open-loop NE,* Pi *can choose no better feedback strategy than one which recovers the same sequence of actions as* Pi*'s open-loop Nash strategy.*

However, there is an important subtlety here! While one could indeed call such a point a NE in feedback strategies, one should not *call it a "feedback Nash equilibrium." The distinction is that feedback Nash equilibria must additionally satisfy subgame perfection: per Remark 1, every set of feedback Nash strategies defined on times $t \in \{s, \dots, T\}$ and at all states x_t must be a Nash equilibrium when restricted to $t \in \{s', \dots, T\}, \forall s < s' \leq T$ and beginning at arbitrary state $x_{s'}$. This concept is closely related to Time consistency.*

Time consistency

Suppose that $(\gamma^{1*}, \dots, \gamma^{N*})$, with $\gamma^{i*} = (\gamma_1^{i*}, \dots, \gamma_T^{i*})$, solves a particular game. Before discussing time consistency,[37] we first introduce the concept of "strategic representation."

[37] Time consistency is a concept which applies in *any* sequential decision problem, be it single- or multi-player. Likewise, it applies to any solution concept, not only to the Nash concept.

Definition 23 (Strategic representation)**.** *Consider two distinct strategies $\gamma^i \neq \tilde{\gamma}^i$. We say that $\tilde{\gamma}^i$ is a* representation *of γ^i (holding other players' strategies $\gamma^j, j \neq i$ fixed) if:*

- *$\tilde{\gamma}^i$ and γ^i yield the same game trajectory, and*
- *they also share the same open-loop value along that trajectory.*

Example 36 (Open-loop and feedback representations). *Suppose a player's strategy is*

$$\gamma^i(x) = -Px, \text{ and} \tag{55a}$$
$$\tilde{\gamma}^i(x) = \alpha, \text{ where } \alpha = -Px^* \tag{55b}$$

and x^* is understood to represent the state of the game along the trajectory generated by γ from a fixed, common initial condition. In this setting, we call $\tilde{\gamma}^i$ a representation of γ^i.

We will employ the following notation in the subsequent definition and discussion of time consistency. Let $D(\Gamma, [1, T], \mathrm{SOL})$ denote a game with strategy set Γ, time $t \in \{1, \ldots, T\}$, and solution concept SOL. Further, let $\gamma^i_{[s,t]} \in \Gamma^i_{[s,t]}$ and $\gamma_{[s,t]} \in \Gamma_{[s,t]}$ denote strategies for individual and all players (respectively), truncated to the time interval $[s, t]$. Finally, let

$$D^{\beta}_{[s,t]} := D\big(\{\gamma_{[1,s)} = \beta_{[1,s)}, \gamma_{(t,T]} = \beta_{(t,T]}, \gamma_{[s,t]} \in \Gamma_{[s,t]}\}, [1, T], \mathrm{SOL}\big)\,.$$

Definition 24 (Weak time consistency). *A solution γ^* is* weakly time consistent *if*

$$\forall s \in (1, T], \gamma^*_{(s,T]} \text{ solves } D^{\gamma^*}_{(s,T]}\,.$$

If γ^* is not weakly time consistent, it is *time inconsistent.*

Question 8. *Suppose that γ^* solves a (single-player) open-loop optimal control problem with time-additive cost structure. Is γ^* weakly time consistent?*

Answer: *Yes! This can be verified by choosing an arbitrary $s \in [1, T]$, following $\gamma^*_{[1,s)} = (u^*_1, \ldots, u^*_{s-1})$ up until time $s - 1$, and then realizing that the optimal control sequence from the resulting state x^*_s at time s will precisely coincide with $\gamma^*_{[s,T]}$.*

Definition 25 (Strong time consistency). *A strategy γ^* is* strongly time consistent *if*

$$\forall s \in (1, T], \gamma^*_{[s,T]} \text{ solves } D^{\beta}_{[s,T]} \text{ for every } \beta_{[1,s)} \in \Gamma_{[1,s)}\,.$$

Question 9. *Consider the same open-loop strategy, γ^* from Question 8. Is γ^* strongly time consistent?*

Answer: *No! To see this, apply some other (open-loop) sequence of actions $\beta_{[1,s)}$ up until time $s - 1$. This will take the system to a different state $\tilde{x}_s \neq x^*_s$ at time s. Applying the open-loop strategy $\gamma^*_{[s,T]}$ for the remainder of the time horizon is no longer optimal from this different state. Therefore, γ^* does not satisfy Definition 25.*

Proposition 10. *Feedback Nash solutions are always strongly time consistent.*

Proof. The proof is left to the reader, but follows directly from the subgame perfection implied in Remark 1. □

Smooth Dynamic Games

We are now ready to begin our discussion of *smooth dynamic games*. This chapter will focus on games of the following form:

$$
\begin{aligned}
(\mathrm{P}i): \quad & \min_{\mathbf{x},\mathbf{u}^i} && J^i(\mathbf{x},\mathbf{u}) && && \text{(56a)}\\
& \text{subject to} && x_{t+1} = f_t(x_t,\mathbf{u}_t), && \forall t \in \{1,2,\ldots,T-1\} && \text{(56b)}\\
& && c^i_t(x_t,\mathbf{u}_t) = 0, && \forall t \in \{1,2,\ldots,T\} && \text{(56c)}\\
& && h^i_t(x_t,\mathbf{u}_t) \geq 0, && \forall t \in \{1,2,\ldots,T\} && \text{(56d)}
\end{aligned}
$$

with the possible addition of shared constraints $c_t(x_t,\mathbf{u}_t) = 0$ and $h_t(x_t,\mathbf{u}_t) \geq 0$, enforced for all players at all times.

In particular, this chapter introduces algorithmic ideas for solving variants of (56) in two specific cases:

- The open-loop case, in which we seek a NE in open-loop strategies, and
- The feedback case, in which we seek feedback strategies which are in equilibrium when restricted to games played from any state and time into the future (i.e., which are *strongly time consistent*).

In both cases, we will begin by solving a simplified, linear quadratic (LQ) variant of (56) which can be treated in closed form. We will then discuss how to approach more general cases.

The open-loop case

We begin by considering games (56) played in open-loop information structures. As we have seen in Example 27, these problems reduce to mixed complementarity problem (MCP). In this section, we will see how a specific subset of these problems afford a closed-form solution, and how that solution relates to a popular family of algorithms for more general cases.

The linear-quadratic setting

Consider the following simplified variant of (56):

We will assume that each player's problem is convex; because constraints are linear in (57b), this amounts to a positive definiteness condition on the objective when restricted to the tangent space of (57b).

$$(\mathrm{P}i): \quad \min_{\mathbf{x},\mathbf{u}^i} \quad \frac{1}{2}\sum_{t=1}^{T}\left(x_t^\top Q_t^i x_t + \sum_{j=1}^{N} u_t^{j\top} R_t^{ij} u_t^j\right) \tag{57a}$$

$$\text{subject to} \quad x_{t+1} = A_t x_t + \sum_{j=1}^{N} B_t^j u_t^j, \forall t \in \{1,\dots,T-1\}. \tag{57b}$$

Game (57) can be solved analytically, as follows. First, assign each player a *separate* Lagrange multiplier for its variant of constraint (57b), with $\boldsymbol{\lambda}^i := (\lambda_t^i)_{t=1}^{T-1}$ and as usual, $\boldsymbol{\lambda} := (\boldsymbol{\lambda}^i)_{i=1}^{N}$, and express its Lagrangian as

$$\begin{aligned}\mathcal{L}^i(\mathbf{x},\mathbf{u},\boldsymbol{\lambda}) = \frac{1}{2}\sum_{t=1}^{T}\left(x_t^\top Q_t^i x_t + \sum_{j=1}^{N} u_t^{j\top} R_t^{ij} u_t^j\right)& \\ - \sum_{t=1}^{T-1} \lambda_t^{i\top}\left(x_{t+1} - A_t x_t - \sum_{j=1}^{N} B_t^j u_t^j\right)&. \end{aligned}\tag{58}$$

Now, we are ready to derive the KKT conditions, which are both necessary and sufficient in this case, due to the convexity of each player's problem:

$$\begin{aligned}
&0 = \nabla_{u_t^i}\mathcal{L}^i = R_t^{ii}u_t^i + B_t^{i\top}\lambda_t^i, && \forall t \in \{1,2,\dots,T-1\} && (59a)\\
& && \implies u_t^i = -(R_t^{ii})^{-1}B_t^{i\top}\lambda_t^i && \\
&0 = \nabla_{u_T^i}\mathcal{L}^i = R_T^{ii}u_T^i && \implies u_T^i = 0 && (59b)\\
&0 = \nabla_{x_t}\mathcal{L}^i = Q_t^i x_t - \lambda_{t-1}^i + A_t^\top \lambda_t^i, && \forall t \in \{2,3,\dots,T-1\} && (59c)\\
& && \implies \lambda_{t-1}^i = Q_t^i x_t + A_t^\top \lambda_t^i && \\
&0 = \nabla_{x_T}\mathcal{L}^i = Q_T^i x_T - \lambda_{T-1}^i && \implies \lambda_{T-1}^i = Q_T^i x_T && (59d)\\
&0 = x_{t+1} - Ax_t - \sum_{j=1}^{N} B_t^j u_t^j, && \forall t \in \{1,2,\dots,T-1\} && (59e)
\end{aligned}$$

Assuming that second-order conditions hold, solving the equations in (59) will yield a set of open-loop Nash strategies for all players. To do so, we make the following substitutions and manipulations, beginning with the blue implications in (59).

Question 10. *Enumerate the second-order conditions under which a solution to (59) is a (unique) open-loop Nash equilibrium of game (57); i.e., when is* P*i's problem convex in its decision variables?*

Taking (59e) at $t = T-1$, and substituting Equations (59a) and (59d) we have that

$$x_T = A_{T-1}x_{T-1} + \sum_{j=1}^{N} B^j_{T-1}u^j_{T-1} \tag{60a}$$
$$= A_{T-1}x_{T-1} - \sum_{j=1}^{N} B^j_{T-1}(R^{jj}_{T-1})^{-1}B^{j\top}_{T-1}\lambda^j_{T-1} \tag{60b}$$
$$= A_{T-1}x_{T-1} - \sum_{j=1}^{N} B^j_{T-1}(R^{jj}_{T-1})^{-1}B^{j\top}_{T-1}Q^j_T x_T \tag{60c}$$
$$\implies x_T = \underbrace{\left(I + \sum_{j=1}^{N} B^j_{T-1}(R^{jj}_{T-1})^{-1}B^{j\top}_{T-1}Q^j_T\right)}_{\Lambda_{T-1}}^{-1} A_{T-1}x_{T-1}\,. \tag{60d}$$

Taking a step back in time and substituting (59d) and the previous result, we have that

$$\lambda^i_{T-2} = Q^i_{T-1}x_{T-1} + A^\top_{T-1}\lambda^i_{T-1} \tag{61a}$$
$$= Q^i_{T-1}x_{T-1} + A^\top_{T-1}Q^i_T\Lambda^{-1}_{T-1}A_{T-1}x_{T-1} \tag{61b}$$
$$= \big(\underbrace{Q^i_{T-1} + A^\top_{T-1}\overbrace{Q^i_T}^{M^i_T}\Lambda^{-1}_{T-1}A_{T-1}}_{M^i_{T-1}}\big)x_{T-1}\,. \tag{61c}$$

and substituting this result in (59a) we have that

$$0 = R^{ii}_{T-2}u^i_{T-2} + B^{i\top}_{T-2}\lambda^i_{T-2} \tag{62a}$$
$$\implies u^i_{T-2} = -(R^{ii}_{T-2})^{-1}B^{i\top}_{T-2}\lambda^i_{T-2} \tag{62b}$$
$$= (R^{ii}_{T-2})^{-1}B^{i\top}_{T-2}M^i_{T-1}x_{T-1}\,. \tag{62c}$$

Together with (59e), this result yields:

$$x_{T-1} = A_{T-2}x_{T-2} + \sum_{j=1}^{N} B^j_{T-2}u^j_{T-2} \tag{63a}$$
$$= A_{T-2}x_{T-2} + \sum_{j=1}^{N} B^j_{T-2}(R^{jj}_{T-2})^{-1}B^{j\top}_{T-2}M^j_{T-1}x_{T-1} \tag{63b}$$
$$\implies x_{T-1} = \underbrace{\left(I + \sum_{j=1}^{N} B^j_{T-2}(R^{jj}_{T-2})^{-1}B^{j\top}_{T-2}M^j_{T-1}\right)}_{\Lambda_{T-2}}^{-1} A_{T-2}x_{T-2}\,. \tag{63c}$$

Extending these arguments recursively, we find that

$$M^i_t = Q^i_t + A^\top_t M^i_{t+1}\Lambda^{-1}_t A_t, \qquad \forall t \in \{1, 2, \dots, T-1\}, \text{ with } M^i_T = Q^i_T \tag{64a}$$
$$\Lambda_t = I + \sum_{j=1}^{N} B^j_t(R^{jj}_t)^{-1}B^{j\top}_t M^j_{t+1}, \qquad \forall t \in \{1, 2, \dots, T-1\}\,, \tag{64b}$$

and thus,

$$\begin{aligned} x_{t+1} &= \Lambda_t^{-1} A_t x_t && (65a) \\ u_t^i &= -(R_t^{ii})^{-1} B_t^{i\top} M_{t+1}^i x_{t+1} && (65b) \\ \lambda_t^i &= M_{t+1}^i x_{t+1} \,. && (65c) \end{aligned}$$

Question 11. *As an exercise, derive the open-loop Nash solution when players' objectives include affine components, i.e. for the game*

$$(\mathrm{P}i): \quad \min_{\mathbf{x}, \mathbf{u}^i} \quad \frac{1}{2} \sum_{t=1}^{T} \left((x_t^\top Q_t^i + 2 q_t^{i\top}) x_t + \sum_{j=1}^{N} (u_t^{j\top} R_t^{ij} + 2 r_t^{ij\top}) u_t^j \right) \quad (66a)$$

$$\text{s.t.} \quad x_{t+1} = A_t x_t + \sum_{j=1}^{N} B_t^j u_t^j, \forall t \in \{1, \dots, T-1\} \,. \quad (66b)$$

Outlook Equation (65) and the preceding derivation constitute an analytic solution to LQ Nash games played in open-loop strategies. Given the structure from Question 10, the KKT conditions are both necessary and sufficient, i.e., the solution in (65) is unique. Shortly, we shall examine the analogous solution for feedback LQ Nash games; as we shall see, the solution is quite different.

Beyond the linear-quadratic setting

Reconsider the general case from (56), but with a time-additive cost structure:

$$\begin{aligned} (\mathrm{P}i): \quad & \min_{\mathbf{x}, \mathbf{u}^i} && \sum_{t=1}^{T} g_t^i(x_t, \mathbf{u}_t) && && (67a) \\ & \text{subject to} && x_{t+1} = f_t(x_t, \mathbf{u}_t), && \forall t \in \{1, 2, \dots, T-1\} && (67b) \\ & && c_t^i(x_t, \mathbf{u}_t) = 0, && \forall t \in \{1, 2, \dots, T\} && (67c) \\ & && h_t^i(x_t, \mathbf{u}_t) \geq 0, && \forall t \in \{1, 2, \dots, T\} && (67d) \\ (\mathrm{All}): \quad & \text{subject to} && c_t(x_t, \mathbf{u}_t) = 0, && \forall t \in \{1, 2, \dots, T\} && (67e) \\ & && h_t(x_t, \mathbf{u}_t) \geq 0, && \forall t \in \{1, 2, \dots, T\} \,, && (67f) \end{aligned}$$

where the constraints in Equations (67e) and (67f) are shared among all players.

This general problem can be reformulated as a MCP as discussed in The general case: mixed complementarity problems. However, there is a very particular structure here, which efficient solvers can exploit! The key idea is to recognize a relationship between Newton steps on the KKT conditions of (67) and the LQ problem considered in The linear-quadratic setting.

For the sake of simplicity, let us ignore the inequality constraints in Equations (67d) and (67f).[38] Considering this simplified prob-

[38] These inequality constraints can be handled in a number of ways. For example, Dirkse and Ferris [1993] employ a "pivoting" (or "active set") strategy, and more recently Le Cleac'h et al. [2022] employ an augmented Lagrangian approach.

lem, we can immediately introduce Lagrange multipliers for all constraints, form a Lagrangian for each player, and examine the resulting KKT conditions. These will be of the form:

$$\forall i \in \{1,\dots,N\} \begin{cases} 0 = \nabla_{\mathbf{x}} \mathcal{L}^i(\mathbf{x},\mathbf{u},\boldsymbol{\nu},\boldsymbol{\lambda},\boldsymbol{\lambda}_{\text{sh}}) \\ 0 = \nabla_{\mathbf{u}^i} \mathcal{L}^i(\mathbf{x},\mathbf{u},\boldsymbol{\nu},\boldsymbol{\lambda},\boldsymbol{\lambda}_{\text{sh}}) \\ 0 = x_{t+1} - f_t(x_t,\mathbf{u}_t), \forall t \in \{1,\dots,T-1\} \\ 0 = c_t^i(x_t,\mathbf{u}_t), \forall t \in \{1,\dots,T\} \end{cases} \tag{68}$$
$$0 = c(x_t,\mathbf{u}_t), \forall t \in \{1,\dots,T\},$$

where Pi's Lagrangian is

$$\mathcal{L}^i(\mathbf{x},\mathbf{u},\boldsymbol{\nu},\boldsymbol{\lambda},\boldsymbol{\lambda}_{\text{sh}}) = \sum_{t=1}^{T} g_t^i(x_t,\mathbf{u}_t) - \sum_{t=1}^{T-1} \nu_t^{i\top}(x_{t+1} - f_t(x_t,\mathbf{u}_t)) - \sum_{t=1}^{T} \lambda_t^{i\top} c_t^i(x_t,\mathbf{u}_t) - \sum_{t=1}^{T} \lambda_{\text{sh},t}^{\top} c(x_t,\mathbf{u}_t)\,. \tag{69}$$

The KKT conditions in (68) form a nonlinear system of equations in primal and dual variables for all players. Like any smooth system of equations, it is natural to construct a solution via Newton's method.[39] In short, we will let $z := (\mathbf{x},\mathbf{u},\boldsymbol{\nu},\boldsymbol{\lambda},\boldsymbol{\lambda}_{\text{sh}})$ and the right hand side of (68) as the function $G(z)$, and update

$$z \leftarrow z - \delta z\,, \tag{70}$$

where the step direction z satisfies Newton's equation

$$\nabla_z G(z)\delta z = -G(z)\,. \tag{71}$$

Consider for a moment: what terms will be present in the Jacobian $\nabla_z G(z)$? We will see first and second derivatives of the stage cost $g_t^i(\cdot)$ with respect to state x_t and that player's input u_t^i. The same is true of functions in the constraints, $c_t^i(\cdot), c_t(\cdot)$, and $f_t(\cdot)$. In the LQ setting, we did not consider any constraints except those due to dynamics $f_t(\cdot)$; however one can observe that *the only difference between the computation in (71) and that in (65) is that the latter does not account for second derivatives of the map $f_t(\cdot)$.* Two observations are in order.

[39] Chapter 11 of the textbook by Nocedal and Wright [1999] provides an excellent introduction to Newton's method for solving systems of equations.

Note that it is common to modify the step δz, e.g. via a linesearch or trust region strategy. Chapters 3 and 4 of the textbook by Nocedal and Wright [1999] provide an accessible overview of these approaches.

Remark 2. *Neglecting these second order terms in the construction of (71) amounts to the distinction between differential dynamic programming (DDP) and the iterative linear quadratic regulator (ILQR) in the optimal control literature. In that single-player context, it can be shown that the approximation still yields steps δz which descend an appropriate merit function (albeit at a potentially slower rate). However, to the author's knowledge, a similar result has not been shown in the more general case with N players.*

Remark 3. *It is absolutely possible to construct the system of (linear) equations in (71). Upon doing so, one will immediately observe that it is sparse (indeed, banded), and can be solved via a (somewhat messier) variant of the recursions used in the LQ setting.*

The feedback case

In this section, we mirror the development of Nash solutions for open-loop games and illustrate the construction of feedback Nash equilibria, first in the LQ setting and then in more general cases.

First, however, it is time that we provide a more formal definition of what we mean by the term "feedback Nash equilibrium" which makes the discussion of Remark 1 precise.

Definition 26 (Feedback Nash equilibrium)**.** *Consider a variant of (56) in which each* P*i's objective is time-separable,*[40] *i.e.,* $J^i(\mathbf{x}, \mathbf{u}) = J_1^i(\mathbf{x}, \mathbf{u})$ *where*

$$J_t^i\big((x_s)_{s=t}^T, (\mathbf{u}_s)_{s=t}^T\big) = \ell\Big(g_t^i(x_t, \mathbf{u}_t), J^i\big((x_s)_{s=t+1}^T, (\mathbf{u}_s)_{s=t+1}^T\big)\Big)$$
$$\text{and } J_T^i(x_T, \mathbf{u}_T) = g_T^i(x_T, \mathbf{u}_T)\,,$$

where the function $\ell : \mathbb{R} \times \mathbb{R} \to \mathbb{R}$ *is arbitrary.*[41]

Define a set of strategies $\gamma = (\gamma_t^i)_{i=1,t=1}^{N,T}$*, and for brevity interpret* $J_t^i\big(x_t, (\gamma_s^i)_{i=1,s=t}^{N,T}\big) := J_t^i\big((x_s)_{s=t}^T, (\mathbf{u}_s)_{s=t}^T\big)$ *with the states and controls corresponding to those which would arise by executing strategies* $(\gamma_s^i)_{i=1,s=t}^{N,T}$ *from state* x_t*. The strategies* $(\gamma_t^{i*})_{i=1,t=1}^{N,T}$ *constitute a* feedback Nash equilibrium *if the following condition is satisfied for all players* i*, times* t*, and states* x_t*:*

$$J_t^i\big(x_t, (\gamma_s^{j*})_{j=1,s=t}^{N,T}\big) \leq J_t^i\Big(x_t, \big(\gamma_t^i, (\gamma_s^{i*})_{s=t+1}^T\big), (\gamma_s^{j*})_{j\neq i,s=t}^T\Big), \ \forall \gamma_t^i\,.$$

In other words, when restricted to begin at each state x_t *and time* t*, each player's strategy must be unilaterally optimal from that point forward.*

[40] For example, a time-additive objective structure with $J^i(\mathbf{x}, \mathbf{u}) = \sum_t g_t^i(x_t, \mathbf{u}_t)$.

[41] For example, if $\ell(a, b) = a + b$, we obtain $J^i(\mathbf{x}, \mathbf{u}) = \sum_t g_t^i(x_t, \mathbf{u}_t)$.

The linear-quadratic setting

Let us reconsider the LQ game from (57) in the context of Definition 26. We can encode the recursive structure of Definition 26 explicitly, by defining a *value function* $V_t^i(x_t)$ which records the value of the game for Pi when played from state x_t and beginning at time t, i.e.

$$(\text{P}i): \quad V_t^i(x_t) = \min_{x_{t+1}, u_t^i} \quad \frac{1}{2}\Big(x_t^\top Q_t^i x_t + \sum_{j=1}^N u_t^{j\top} R_t^{ij} u_t^j\Big) + V_{t+1}^i(x_{t+1}) \quad (72a)$$

$$\text{s.t.} \qquad x_{t+1} = A_t x_t + \sum_{j=1}^N B_t^j u_t^j\,, \quad (72b)$$

where $V^i_{T+1}(x_{T+1}) = 0$. Presuming that the minimum in (72) is unique, we define the feedback Nash strategy $\gamma^{i*}_t(x_t)$ to return the corresponding control u^{i*}_t.

We construct a solution to (72) via a dynamic program,[42] and begin at the terminal time $t = T$. Here, (assuming that all matrices $R^{ij}_T \succ 0$) we have that

[42] It will be self-evident from this procedure that the resulting strategies satisfy the strong time consistency property of Definition 25.

$$V^i_T(x_T) = \min_{x_{T+1}, u^i_T} \quad \frac{1}{2}\Big(x_T^\top Q^i_T x_T + \sum_{j=1}^{N} u^{j\top}_T R^{ij}_T u^j_T\Big) + \overbrace{V^i_{T+1}(x_{T+1})}^{0} \tag{73a}$$

$$= \frac{1}{2} x_T^\top \underbrace{Q^i_T}_{Z^i_T} x_T \,, \text{ with} \tag{73b}$$

$$\gamma^{i*}_T(x_T) = 0 \,. \tag{73c}$$

Moving backward in time, at $t = T - 1$ we have

$$V^i_{T-1}(x_{T-1}) = \tag{74a}$$

$$\min_{x_T, u^i_{T-1}} \quad \frac{1}{2}\Big(x_t^\top Q^i_{T-1} x_{T-1} + \sum_{j=1}^{N} u^{j\top}_{T-1} R^{ij}_{T-1} u^j_{T-1}\Big) + \overbrace{\frac{1}{2} x_T^\top Z^i_T x_T}^{V^i_{T+1}(x_{T+1})}$$

$$\text{s.t.} \qquad x_T = A_{T-1} x_{T-1} + \sum_{j=1}^{N} B^j_{T-1} u^j_{T-1} \,. \tag{74b}$$

Solving the first-order necessary conditions of (74) for all players, we have

$$0 = R^{ii}_{T-1} u^{i*}_{T-1} + B^{i\top}_{T-1} Z^i_T \big(A_{T-1} x_{T-1} + \sum_{j=1}^{N} B^j_{T-1} u^{j*}_{T-1}\big) \,, \tag{75}$$

which can be rearranged into the following linear system of equations

$$M_{T-1} \begin{bmatrix} u^{1*}_{T-1} \\ u^{2*}_{T-1} \\ \vdots \\ u^{N*}_{T-1} \end{bmatrix} = - \begin{bmatrix} B^{1\top} Z^1_T A_{T-1} x_{T-1} \\ B^{2\top} Z^2_T A_{T-1} x_{T-1} \\ \vdots \\ B^{N\top} Z^N_T A_{T-1} x_{T-1} \end{bmatrix} \tag{76}$$

with M_{T-1} defined as

$$M_{T-1} = \begin{bmatrix} R^{11}_{T-1} + B^{1\top}_{T-1} Z^1_T B^1_{T-1} & \cdot & \cdots & B^{1\top}_{T-1} Z^1_T B^N_{T-1} \\ B^{2\top}_{T-1} Z^2_T B^1_{T-1} & \cdot & \cdots & \cdot \\ \vdots & \cdot & \ddots & \vdots \\ B^{N\top}_{T-1} Z^N_T B^1_{T-1} & \cdot & \cdots & R^{NN}_{T-1} + B^{N\top}_{T-1} Z^N_T B^N_{T-1} \end{bmatrix} \tag{77}$$

Question 12. *When does a solution to (76) exist?*

Answer: *Observe that one condition which ensures the existence of a solution is taking $R^{ii}_{T-1} \gg Z^j_T, \forall i \neq j$. Clearly, the most general condition is that the matrix M_{T-1} has full rank.*

Question 13. *What is the second-order condition under which (76) is also sufficient to specify a unique equilibrium?*

Answer: *We require the Hessian of* Pi*'s problem at $T-1$ to be positive definite within the critical cone of the dynamics constraint. From (75), we see that this amounts to the condition $R^i i_{T-1} + B^{i\top}_{T-1} Z^i_T B^i_{T-1} \succ 0$, or equivalently, that the diagonal of the matrix in (77) is positive definite.*

Observe: a solution to (76) will be of the form

$$\gamma^{i*}_{T-1}(x_{T-1}) = u^{i*}_{T-1} = -P^i_{T-1} x_{T-1}, \tag{78}$$

where the matrices $(P^i_{T-1})_{i=1}^N$ solve the system of equations

$$M_{T-1} \begin{bmatrix} P^1_{T-1} \\ P^2_{T-1} \\ \vdots \\ P^N_{T-1} \end{bmatrix} = \begin{bmatrix} B^{1\top} Z^1_T A_{T-1} \\ B^{2\top} Z^2_T A_{T-1} \\ \vdots \\ B^{N\top} Z^N_T A_{T-1} \end{bmatrix}. \tag{79}$$

Substituting (78) into (74), we find that

$$V^i_{T-1}(x_{T-1}) = \frac{1}{2} x^\top_{T-1} Z^i_{T-1} x_{T-1}, \tag{80}$$

where

$$\begin{aligned} Z^i_{T-1} = Q^i_{T-1} &+ \sum_{j=1}^N P^{j\top}_{T-1} R^{ij}_{T-1} P^j_{T-1} \\ &+ \Big(A_{T-1} - \sum_{j=1}^N B^j_{T-1} P^j_{T-1}\Big)^\top Z^i_T \Big(A_{T-1} - \sum_{j=1}^N B^j_{T-1} P^j_{T-1}\Big). \end{aligned} \tag{81}$$

The preceding steps can be repeated backward in time, inductively, until reaching $t = 1$.[43]

[43] Often, the recursions in (81) is termed a set of "coupled Riccati equations."

Remark 4 (Differences from the open-loop solution)**.** *Apart from the differences in how the solutions were derived, observe the following:*

- *The open-loop equilibrium does not depend upon the matrices $\{R^{ij}_t\}_{i \neq j}$, whereas those matrices do influence the feedback solution via (81).*
- *The Lagrange multipliers $\{\lambda^i_t\}$ played an essential role in deriving the open-loop solution, but were completely ignored in the feedback solution.*
- *It may be tempting to examine (65b) and conclude that the open-loop equilibrium strategy for* Pi *is in fact a state-feedback strategy, because it is expressed as a function of state. This is untrue! The reason is subtle, however: the state x_{t+1} in (65b) is not* any *state—as it is in (78)—but rather it is the* specific *state arising when players follow equilibrium strategies.*

Question 14. *Can you think of a scenario where the terms $\{R^{ij}_t\}_{i \neq j}$ are present and encode something significant?*

Question 15. *What is the proper value for the multipliers for constraints (72b)? Does it match the open-loop value?*

Example 37 (LQ open-loop vs. feedback brain teaser)**.** *Consider a two-player LQ Nash game, with state* $x_t = (x_t^1, x_t^2)$ *and* $x_t^i = (p_{x,t}^i, v_{x,t}^i, p_{y,t}^i, v_{y,t}^i)$ *including horizontal and vertical position and velocity, with dynamics*

Thank you Forrest Laine for this example!

$$x_{t+1}^i = \overbrace{\begin{bmatrix} 1 & \Delta & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta \\ 0 & 0 & 0 & 1 \end{bmatrix}}^{A_t^i} x_t^i + \overbrace{\begin{bmatrix} 0 & 0 \\ \Delta & 0 \\ 0 & 0 \\ 0 & \Delta \end{bmatrix}}^{B_t^i} u_t^i \,, \tag{82}$$

where Δ *is a small number representing the length of a discrete time-step. Suppose that the players' objectives are as follows:*

$$J^1(\mathbf{x}, \mathbf{u}) = \frac{1}{2} \sum_{t=1}^{T} \left(\|x_t^2\|_2^2 + \|u_t^1\|_2^2 \right) \tag{83a}$$

$$J^2(\mathbf{x}, \mathbf{u}) = \frac{1}{2} \sum_{t=1}^{T} \left(\|x_t^1 - x_t^2\|_2^2 + \|u_t^2\|_2^2 \right) . \tag{83b}$$

Which of the plots in Figure 10 represents plausible open-loop or feedback NE outcomes?

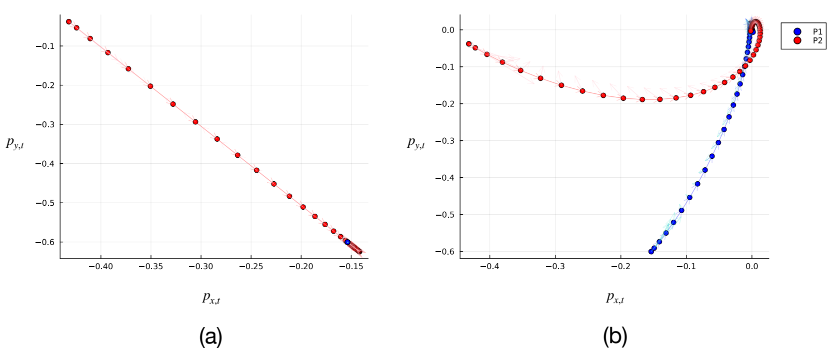

Figure 10: Open-loop and feedback Nash equilibria of the game in Example 37. Which is which?

Beyond the linear-quadratic setting

As in the open-loop setting, there is a close relationship between Newton's method and the (coupled) Riccati solution for feedback Nash LQ games. For further details, please refer to the paper by Laine et al. [2023]. The key idea is to observe that the nesting structure in (72) implies that the corresponding KKT conditions for each player will be far more complicated than in the open-loop setting. To see this, observe that Pi's strategy at time t is characterized by the

KKT conditions for (a generalized version of) problem (72), i.e.

$$\gamma_t^i(x_t) = \underset{u_t^i}{\text{argmin}} \quad g_t^i(x_t, \mathbf{u}_t) + V_{t+1}^i(x_{t+1}) \tag{84a}$$

$$\text{subject to} \quad x_{t+1} = f_t(x_t, \mathbf{u}_t) \tag{84b}$$

$$u_t^j = \gamma_t^j(x_t), \forall j \neq i\,, \tag{84c}$$

where the value function $V_{t+1}^i(\cdot)$ itself depends upon the strategies $(\gamma_s^i)_{i=1,s=t+1}^{N,T}$. That is, the function $\gamma_t^i(\cdot)$ is itself a function of $\nabla_x \gamma_s^j, \forall s > t$ and $j \in \{1, \ldots, N\}$. Unrolling this recursion backward in time, we see that, in fact, $\gamma_t^i(\cdot)$ depends upon the $(s-t)^{\text{th}}$ order gradient of $\gamma_s^j(\cdot)$ with respect to state x_s, for every $s > t$ and player j. And this is just to write down the KKT conditions!

Laine et al. [2023] proposes to simplify this process by treating all strategies as linear and discarding all higher-order gradients when constructing KKT conditions. This approximation renders the problem tractable, and (with the caveats of Remarks 2 and 3 in mind) Newton steps on these modified conditions can be computed by solving the coupled Riccati equations in the previous section. Handling non-dynamic equality and inequality constraints (both private and shared) can proceed accordingly with minimal changes from the open-loop setting. We close by mentioning an important subtlety.

Remark 5. *The KKT conditions for (84) only make sense to discuss in cases where constraint qualifications hold and, critically, the relevant objects are differentiable! This is not always the case; for example, when a player's strategy at time $t+1$ is non-differentiable on the boundary of an active constraint, it becomes impractical to construct the KKT conditions for players' strategies at time t. In practice, one may consider employing interior point strategies to "smoothen" these constraint boundaries, but to the author's knowledge the theoretical implications are not well understood in the context of feedback games.*[44]

[44] Li et al. [2024] develops such a technique for the related problem of finding feedback *Stackelberg* equilibria in constrained settings.

Parting Thoughts, Cautions, and Ideas

Let us close with a few parting thoughts, cautions, and ideas.

First, a reminder: this document is *not* intended as a substitute for any of the references mentioned in its pages. In particular, readers are strongly encouraged to consult Nocedal and Wright [1999] and Bertsekas [1999] for a deeper (and far more complete) introduction to nonlinear programming. Facchinei and Pang [2003], and particularly the first chapter, serves a similar purpose for complementarity programming. Finally, please refer to Başar and Olsder [1998] for further details about dynamic games.

Rather, this document is intended to expose readers to the fundamentals of optimization *in tandem with* those of game theory because they are inextricably linked, and because the connection is not always emphasized or well-explained.

Why the focus on "smooth" games, again?

Smooth (i.e., differentiable) games are, on the face of things, more complicated than finite games, so why focus on them here? The reasons are at least twofold. First, many (most?) games that involve physical quantities (position, velocity, etc.) are naturally modeled in terms of continuous quantities. Second, algorithms to solve smooth games can exploit derivative information which is not present in finite games, and this extra structure can lead to substantial computational acceleration.

Take care! There is no free lunch.

Caution 1 The aforementioned computational acceleration comes at a huge cost: any of the gradient-based algorithms used to solve these games will—at best—find local variants of the desired equilibria. Worse, naive algorithms can even converge to points which only satisfy first-order necessary conditions for all players, but do not satisfy second-order conditions (i.e., points which are not even local equilibria)! Care must be taken to design algorithms which avoid these issues. For example, Mazumdar et al. [2019] and Chinchilla et al. [2023] study zero-sum Nash and Stackelberg cases, respectively.

Caution 2 It should go without saying, but these local equilibria are not necessarily global and they *should not be used for any safety-critical application* without additional structure and corresponding analysis. More precisely: finding a local Stackelberg equilibrium in a zero-sum game does not necessarily yield a security strategy. One may very well employ such an equilibrium strategy and obtain an outcome *less* favorable than that at a local equilibrium.

Caution 3 Feedback games remain intractable to solve, even to local Nash equilibrium. The best we can do, to date, is to find a point which *approximately* satisfies the KKT conditions for all players in each stage of the game. It remains an open problem to provide a firm theoretical bound on this error, although it appears to be suitable for practical applications.

What next?

Progress in optimization and machine learning has opened to way for new and exciting work in games. What follows is a brief outline of several promising directions.

Partial information There is a whole spectrum of information patterns between the open-loop and feedback structures, yet these seem to be the most widely studied, by far. In particular, recent advances in reinforcement and representation learning have shown promise in coping with these limited-information settings, but it remains a challenge to do so with limited computing resources and, more importantly, to characterize the properties of equilibrium solutions which can be found by these algorithms.

Incomplete information Our entire premise in this document has been that all players are aware of each other's presence and objectives. Several theoretical frameworks do exist to remove these assumptions (e.g., Bayesian games, hypergames) but to the best of the author's knowledge, there remains a substantial disconnect between the efficient optimization-based algorithms for complementarity programming and feedback games and those which are studied in incomplete information settings.

Stochastic games Likewise, our focus has been on deterministic problems in this text. The algorithms described, however, can directly apply in highly-structured stochastic games.[45,46] Relaxing some of these structural assumption is an important direction of future inquiry.

[45] Wilko Schwarting, Alyssa Pierson, Sertac Karaman, and Daniela Rus. Stochastic dynamic games in belief space. *IEEE Transactions on Robotics*, 37 (6):2157–2172, 2021

[46] Jingqi Li, Chih-Yuan Chiu, Lasse Peters, Fernando Palafox, Mustafa Karabag, Javier Alonso-Mora, Somayeh Sojoudi, Claire Tomlin, and David Fridovich-Keil. Scenario-Game ADMM: A Parallelized Scenario-Based Solver for Stochastic Noncooperative Games. In *IEEE Conference on Decision and Control (CDC)*, 2023

Operating at scale Even the fastest algorithms discussed in this text have complexity on the order of the number of variables cubed. For games involving large numbers of players (e.g., online marketplaces, air traffic management, power grid management) it will be critical to develop algorithms capable of exploiting parallel and/or decentralized computation to mitigate the inherent complexity.

Tools for planning and policy-making City planners and policy-makers must make decisions based upon forecasts of how others will react to those decisions. This amounts to solving a hierarchical game with one or more leaders and (perhaps many) followers. The algorithms discussed in this text—particularly those for solving smooth feedback games—provide an exciting first step toward building these capabilities.

Bibliography

Tamer Başar and Geert Jan Olsder. *Dynamic noncooperative game theory*. SIAM, 1998.

Dimitri P Bertsekas. *Nonlinear programming*. Athena Scientific, 1999.

Raphael Chinchilla, Guosong Yang, and João P Hespanha. Newton and interior-point methods for (constrained) nonconvex–nonconcave minmax optimization with stability and instability guarantees. *Mathematics of Control, Signals, and Systems*, pages 1–41, 2023.

Vincent Conitzer. On Stackelberg mixed strategies. *Synthese*, 193(3): 689–703, 2016.

Steven P Dirkse and Michael C Ferris. The path solver: A non-monotone stabilization scheme for mixed complementarity problems. Technical report, University of Wisconsin-Madison Department of Computer Sciences, 1993.

Francisco Facchinei and Jong-Shi Pang. *Finite-dimensional variational inequalities and complementarity problems*. Springer, 2003.

David Fridovich-Keil, Ellis Ratner, Lasse Peters, Anca D Dragan, and Claire J Tomlin. Efficient iterative linear-quadratic approximations for nonlinear multi-player general-sum differential games. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1475–1481. IEEE, 2020.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial nets. *Advances in neural information processing systems*, 27, 2014.

Forrest Laine, David Fridovich-Keil, Chih-Yuan Chiu, and Claire Tomlin. The computation of approximate generalized feedback Nash equilibria. *SIAM Journal on Optimization*, 33(1):294–318, 2023.

Simon Le Cleac'h, Mac Schwager, and Zachary Manchester. AL-GAMES: a fast augmented Lagrangian solver for constrained dynamic games. *Autonomous Robots*, 46(1):201–215, 2022.

Jingqi Li, Chih-Yuan Chiu, Lasse Peters, Fernando Palafox, Mustafa Karabag, Javier Alonso-Mora, Somayeh Sojoudi, Claire Tomlin, and David Fridovich-Keil. Scenario-Game ADMM: A Parallelized Scenario-Based Solver for Stochastic Noncooperative Games. In *IEEE Conference on Decision and Control (CDC)*, 2023.

Jingqi Li, Somayeh Sojoudi, Claire Tomlin, and David Fridovich-Keil. The computation of approximate feedback Stackelberg equilibria in multi-player nonlinear constrained dynamic games. *arXiv preprint arXiv:2401.15745*, 2024.

Eric Mazumdar, Lillian J Ratliff, and S Shankar Sastry. On gradient-based learning in continuous games. *SIAM Journal on Mathematics of Data Science*, 2(1):103–131, 2020.

Eric V Mazumdar, Michael I Jordan, and S Shankar Sastry. On finding local nash equilibria (and only local nash equilibria) in zero-sum games. *arXiv preprint arXiv:1901.00838*, 2019.

John F Nash. Equilibrium points in n-person games. *Proceedings of the national academy of sciences*, 36(1):48–49, 1950.

Jorge Nocedal and Stephen J Wright. *Numerical optimization*. Springer, 1999.

Lasse Peters and Xinjie Liu. MCPTrajectoryGames.jl. URL `https://github.com/JuliaGameTheoreticPlanning/MCPTrajectoryGameSolver.jl`. Accessed: September 2023.

Lillian J Ratliff, Samuel A Burden, and S Shankar Sastry. On the characterization of local Nash equilibria in continuous games. *IEEE transactions on automatic control*, 61(8):2301–2307, 2016.

Wilko Schwarting, Alyssa Pierson, Sertac Karaman, and Daniela Rus. Stochastic dynamic games in belief space. *IEEE Transactions on Robotics*, 37(6):2157–2172, 2021.

Reinhard Selten. *Reexamination of the perfectness concept for equilibrium points in extensive games*. Springer, 1988.

David Silver, Julian Schrittwieser, Karen Simonyan, Ioannis Antonoglou, Aja Huang, Arthur Guez, Thomas Hubert, Lucas Baker, Matthew Lai, Adrian Bolton, Yutian Chen, Timothy Lillicrap, Fan Hui, Laurent Sifre, George van den Driessche, Thore Graepel, and Demis Hassabis. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

Heinrich von Stackelberg. Marktform und gleichgewicht. 1934.