Game Theory: Penn State Math 486 Lecture Notes

Version 1.1.1

Christopher Griffin

With Major Contributions By:

James Fam

George Kesidis

Contents

List of Figures

CHAPTER 1

Preface and an Introduction to Game Theory

1. Using These Notes

Stop! This is a set of lecture notes. It is not a book. Go away and come back when you have a real textbook on Game Theory. Okay, do you have a book? Alright, let's move on then. This is a set of lecture notes for Math 486–Penn State's undergraduate Game Theory course. Since I use these notes while I teach, there may be typographical errors that I noticed in class, but did not fix in the notes. If you see a typo, send me an e-mail and I'll add an acknowledgement. There may be many typos, that's why you should have a real textbook.

The lecture notes are loosely based on Luce and Raiffa's *Games and Decisions: Introduction and Critical Survey*. This is the same book Nash used when he taught (or so I've heard). There are elements from Myerson's book on Game Theory (more appropriate for economists) as well as Morris' book on Game Theory. Naturally, I've also included elements from Von Neuman and Morgenstern's classic tome. Most of these books are reasonably good, but each has some thing that I didn't like. Luce and Raiffa is not as rigorous as one would like for a math course; Myerson's book is not written for mathematicians; Morris' book has a host of problems, not the least of which is that it does not include a modern treatment of general sum games; Von Neumann's book is excellent but too thick and frightening for a first course–also it's old. If you choose any collection of books, you can find something wrong with them, I've picked on these only because I had them at hand when writing these notes. I also draw on other books referenced in the bibliography.

This set of notes corrects some of the problems I mention by presenting the material in a format for that can be used easily in an undergraduate mathematics class. Many of the proofs in this set of notes are adapted from the textbooks with some minor additions. One thing that is included in these notes is a treatment of the use of quadratic programs in general sum games two player games. This does not appear in many textbooks.

In order to use these notes successfully, you should have taken a course in: matrix algebra (Math 220 at Penn State), though courses in Linear Programming (Math 484 at Penn State) and Vector Calculus (Math 230/231 at Penn State) wouldn't hurt. I review a substantial amount of the material you will need, but it's always good to have covered prerequisites before you get to a class. That being said, I hope you enjoy using these notes!

2. An Overview of Game Theory

Game Theory is the study of decision making under competition. More specifically, Game Theory is the study of optimal decision making under competition when one individual's decisions affect the outcome of a situation for all other individuals involved. You've naturally encountered this phenomenon in your everyday life: when you play play chess or Halo, chase your baby brother in an attempt to wrestle him into his P.J.'s or even negotiate a price on a car, your decisions and the decisions of those around you will affect the quality of the end result for everyone.

Game Theory is a broad discipline within Applied Mathematics that influences and is itself influenced by Operations Research, Economics, Control Theory, Computer Science, Psychology, Biology and Sociology (to name a few disciplines). If you want to start a fight in bar with a Game Theorist (or an Economist) you might say that Game Theory can be broadly classified into four main sub-categories of study:

(1) Classical Game Theory: Focuses on optimal play in situations where one or more people must make a decision and the impact of that decision and the decisions of those involved is known. Decisions may be made by use of a randomizing device (like flipping a coin). Classical Game Theory has helped people understand everything from the commanders in military engagements to the behavior of the car salesman during negotiations. See [**vNM04**, **LR89**, **Mor94**, **Mye01**, **Dre81**, **PR71**] and Chapter 1 of [**Wei97**] or [**Bra04**] for extensive details on this sub-discipline of Game Theory.

(2) Combinatorial Game Theory: Focuses on optimal play in two-player games in which each player takes turns changing in pre-defined ways. Combinatorial Game Theory does *not* consider games with chance (no randomness). Combinatorial Game Theory is used to investigate games like Chess, Checkers or Go. Of all branches, Combinatorial Game Theory is the least *directly related* to real life scenarios. See[**Con76**] and [**BCG01a**, **BCG01b**, **BCG01c**, **BCG01d**], which are widely regarded as the *bible* of Combinatorial Game Theory.

(3) Dynamic Game Theory: Focuses on the analysis of games in which players must make decisions over time and in which those decisions will affect the outcome at the next moment in time. Dynamic Game Theory often relies on differential equations to model the behavior of players over time. Dynamic Game Theory can help optimize the behavior of unmanned vehicles or it can help you capture your baby sister who has escaped from her playpen. See [**DJLS00**, **BO82**] for a survey on dynamic games. The latter reference is extremely technical.

(4) Other Topics in Game Theory: Game Theory, as noted, is broad. This category captures those topics that are derivative from the three other branches. Examples include, but are not limited to: (i) Evolutionary Game Theory, which attempts to model evolution as competition between species, (ii) Dual games in which players may choose from an infinite number of strategies, but time is not a factor, (iii) Experimental Game Theory, in which people are studied to determine how accurately classical game theoretic models truly explain their behavior. See [**Wei97**, **Bra04**] for examples.

Figure 1.1 summarizes the various types of Game Theory.

In these notes, we focus primarily on Classical Game Theory. This work is relatively young (under 70 years old) and was initiated by Von Neumann and Morgenstern. Major contributors to this field include Nash (of *A Beautiful Mind* fame), and several other Nobel Laureates.

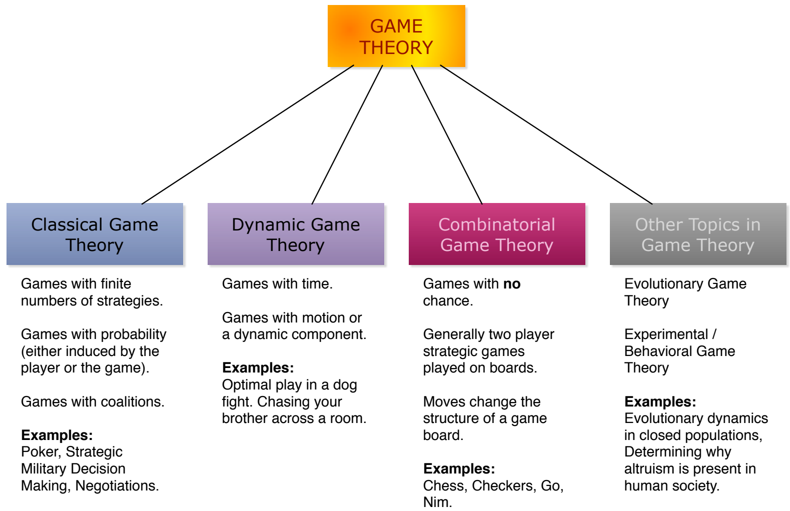


Figure 1.1. There are several sub-disciplines within Game Theory. Each one has its own unique sets of problems and applications. We will study Classical Game Theory, which focuses on questions like, "What is my best decision in a given economic scenario, where a reward function provides a way for me to understand how my decision will impact my result." We may also investigate Combinatorial Game Theory, which is interested in games like Chess or Go. If there's time, we'll study Evolutionary Game Theory, which is interesting in its own right.

CHAPTER 2

Probability Theory and Games Against the House

1. Probability

Our study of Game Theory starts with a characterization of optimal decision making for an individual in the absence of any other players. The *games* we often see on television fall into this category. TV Game Shows (that do not pit players against each other in knowledge tests) often require a single player (who is, in a sense, playing against *The House*) to make a decision that will affect only his life.

EXAMPLE 2.1. Congratulations! You have made it to the very final stage of *Deal or No Deal*. Two suitcases with money remain in play, one contains \$0.01 while the other contains \$1,000,000. The banker has offered you a payoff of \$499,999. Do you accept the banker's safe offer or do you risk it all to try for \$1,000,000. Suppose the banker offers you \$100,000 what about \$500,000 or \$10,000?

Example 2.1 may seem contrived, but it has real world implications and most of the components needed for a serious discussion of decision making under risk. In order to study these concepts formally, we will need a grounding in probability. Unfortunately, a formal study of probability requires a heavy dose of Measure Theory, which is well beyond the scope of an introductory course on Game Theory. Therefore, the following definitions are meant to be intuitive rather than mathematically rigorous.

Let Ω be a finite set of elements describing the outcome of a chance event (a coin toss, a roll of the dice etc.). We will call Ω the *Sample Space*. Each element of Ω is called an *outcome*.

EXAMPLE 2.2. In the case of Example 2.1, the world as we care about it is purely the position of the \$1,000,000 and \$0.01 within the suitcases. In this case Ω consists of two possible outcomes: \$1,000,000 is in suitcase number 1 (while \$0.01 is in suitcase number 2) or \$1,000,000 is in suitcase number 2 (while \$0.01 is in suitcase number 1).

Formally, let us refer to the first outcome as A and the second outcome as B. Then $\Omega = \{A, B\}$.

DEFINITION 2.3 (Event). If Ω is a sample space, then an event is any subset of Ω.

EXAMPLE 2.4. Clearly, the sample space in Example 2.1 consists of precisely four events: $\emptyset$ (the empty event), $\{A\}$, $\{B\}$ and $\{A, B\} = \Omega$. These four sets represent all possible subsets of the set $\Omega = \{A, B\}$.

DEFINITION 2.5 (Union). If $E, F \subseteq \Omega$ are both events, then $E \cup F$ is the *union* of the sets E and F and consists of all outcomes in either E or F. Event $E \cup F$ occurs if either even E or event F occurs.

Example 2.6. Consider the role of a fair six sided dice. The outcomes are 1,. . . ,6. If $E = \{1, 3\}$ and $F = \{2, 4\}$, then $E \cup F = \{1, 2, 3, 4\}$ and will occur as long as we don't roll a 5 or 6.

Definition 2.7 (Intersection). If $E, F \subseteq \Omega$ are both events, then $E \cap F$ is the *intersection* of the sets E and F and consists of all outcomes in both E and F. Event $E \cap F$ occurs if both even E or event F occur.

Example 2.8. Again, consider the role of a fair six sided dice. The outcomes are 1,. . . ,6. If $E = \{1, 2\}$ and $F = \{2, 4\}$, then $E \cap F = \{2\}$ and will occur only if we roll a 2.

Definition 2.9 (Mutual Exclusivity). Two events $E, F \subseteq \Omega$ are said to be *mutually exclusive* if and only if $E \cap F = \emptyset$.

Definition 2.10 (Discrete Probability Distribution (Function)). Given discrete sample space Ω, let $\mathcal{F}$ be the set of all events on Ω. A *discrete probability function* is a mapping from $P : \mathcal{F} \to [0, 1]$ with the properties:

(1) $P(\Omega) = 1$
(2) If $E, F \in \mathcal{F}$ and $E \cap F = \emptyset$, then $P(E \cup F) = P(E) + P(F)$

Remark 2.11 (Power Set). In this definition, we talked about the set $\mathcal{F}$ as the set of all events over a set of outcomes Ω. This is an example of the *power set*: the set of all subsets of a set. We sometimes denote this set as 2^Ω. Thus, if Ω is a set, then 2^Ω is the power set of Ω or the set of all subsets of Ω.

Definition 2.10 is surprisingly technical and probably does not conform to your ordinary sense of what probability is. It's best not to think of probability in this very formal way and instead to think that a probability function assigns a number to an outcome (or event) that tells you the chances of it occurring. Put more simply, suppose we could run an experiment where the result of that experiment will be an outcome in Ω. The the function P simply tells us the proportion of times we will observe an event $E \subset \Omega$ if we ran this experiment an exceedingly large number of times.

Example 2.12. Suppose we could play the *Deal or No Deal* example over and over again and observe where the money ends up. A smart game show would mix the money up so that approximately one-half of the time we observe \$1,000,000 in suitcase 1 and the other half the time we observe this money in suitcase 2.

A probability distribution formalizes this notion and might assign 1/2 to event $\{A\}$ and 1/2 to event $\{B\}$. However to obtain a true probability distribution, we must also assign probabilities to $\emptyset$ and $\{A, B\}$. In the former case, we know that something must happen! Therefore, we can assign 0 to event $\emptyset$. In the latter case, we know that for certain that either outcome A or B must occur and so in this case we assign a value of 1.

Example 2.13. In a fair six sided dice, the probability of rolling any value is 1/6. Formally, $\Omega = \{1, 2, \ldots, 6\}$ any role yields is an event with only one element: $\{\omega\}$ where ω is some value in Ω. If we consider the event $E = \{1, 2, 3\}$ then $P(E)$ gives us the probability that we will roll a 1, 2 or 3. Since $\{1\}$, $\{2\}$ and $\{3\}$ are disjoint sets and $\{1, 2, 3\} = \{1\} \cup \{2\} \cup \{3\}$, we know that:

$$P(E) = \frac{1}{6} + \frac{1}{6} + \frac{1}{6} = \frac{1}{2}$$

Definition 2.14 (Discrete Probability Space). The triple $(\Omega, \mathcal{F}, P)$ is called a *discrete probability space* over Ω.

Lemma 2.15. *Let $(\Omega, \mathcal{F}, P)$ be a discrete probability space. Then $P(\emptyset) = 0$.*

Proof. The set $\Omega \in \mathcal{F}$ and $\emptyset \in \mathcal{F}$ are disjoint (i.e., $\Omega \cap \emptyset = \emptyset$). Thus:

$$P(\Omega \cup \emptyset) = P(\Omega) + P(\emptyset)$$

We know that $\Omega \cup \emptyset = \Omega$. Thus we have:

$$P(\Omega) = P(\Omega) + P(\emptyset) \implies 1 = 1 + P(\emptyset) \implies 0 = P(\emptyset)$$

□

Lemma 2.16. *Let $(\Omega, \mathcal{F}, P)$ be a discrete probability space and let $E, F \in \mathcal{F}$. Then:*

$$P(E \cup F) = P(E) + P(F) - P(E \cap F) \tag{2.1}$$

Proof. If $E \cap F = \emptyset$ then by definition $P(E \cup F) = P(E) + P(F)$ but $P(\emptyset) = 0$, so $P(E \cup F) = P(E) + P(F) - P(E \cap F)$.

Suppose $E \cap F \neq \emptyset$. Then let:

$$E' = \{\omega \in E | \omega \notin F\}$$
$$F' = \{\omega \in F | \omega \notin E\}$$

Then we know:

(1) $E' \cap F' = \emptyset$,
(2) $E' \cap (E \cap F) = \emptyset$,
(3) $F' \cap (E \cap F) = \emptyset$,
(4) $E = E' \cup (E \cap F)$ and
(5) $F = F' \cup (E \cap F)$.

Thus, (by inductive extension of the definition of discrete probability function) we know:

$$P(E \cup F) = P(E' \cup F' \cup (E \cap F)) = P(E') + P(F') + P(E \cap F) \tag{2.2}$$

We also know that:

$$P(E) = P(E') + P(E \cap F) \implies P(E') = P(E) - P(E \cap F) \tag{2.3}$$

and

$$P(F) = P(F') + P(E \cap F) \implies P(F') = P(F) - P(E \cap F) \tag{2.4}$$

Combing these three equations yields:

$$P(E \cup F) = P(E) - P(E \cap F) + P(F) - P(E \cap F) + P(E \cap F) = \\ P(E \cup F) = P(E) + P(F) - P(E \cap F) \tag{2.5}$$

This completes the proof. □

Exercise 1. A fair 4 sided die is rolled. Assume the sample space of interest is the number appearing on the die and the numbers run from 1 to 4. Identify the space Ω precisely and all the possible outcomes and events within the space. What is the (logical) fair probability distribution in this case. [Hint: See Example 2.13.]

Exercise 2. Prove the following: Let $E \subseteq \Omega$ and define E^c to be the set of elements of Ω *not* in E (this is called the complement of E). Suppose $(\Omega, \mathcal{F}, P)$ is a discrete probability space. Show that $P(E^c) = 1 - P(E)$.

Lemma 2.17. *Let $(\Omega, \mathcal{F}, P)$ be a discrete probability space and let $E, F \in \mathcal{F}$. Then:*

$$P(E) = P(E \cap F) + P(E \cap F^c) \tag{2.6}$$

Exercise 3. Prove Lemma 2.17. [Hint: Show that $E \cap F$ and $E \cap F^c$ are mutually exclusive events. Then show that $E = (E \cap F) \cup (E \cap F^c)$.]

The following lemma is provided without proof. The exercise to prove it is somewhat challenging.

Lemma 2.18. *Let $(\Omega, \mathcal{F}, P)$ be a probability space and suppose that $E, F_1, \dots, F_n$ are subsets of Ω. Then:*

$$E \cap \bigcup_{i=1}^{n} F_i = \bigcup_{i=1}^{n} (E \cap F_i) \tag{2.7}$$

That is, intersection distributes over union.

Exercise 4. Prove Lemma 2.18. [Hint: Use induction. Begin by showing that if $n = 1$, then the statement is clearly true. Then show that if the statement holds for $F_1, \dots, F_k$ $k \leq n$, then it must hold for $n+1$ using the fact that union and intersection are associative.]

Theorem 2.19. *Let $(\Omega, \mathcal{F}, P)$ be a discrete probability space and let $E \in \mathcal{F}$. Let $F_1, \dots, F_n$ be any pairwise disjoint collection of sets that partition Ω. That is, assume:*

$$\Omega = \bigcup_{i=1}^{n} F_i \tag{2.8}$$

and $F_i \cap F_j = \emptyset$ if $i \neq j$. Then:

$$P(E) = \sum_{i=1}^{n} P(E \cap F_i) \tag{2.9}$$

Proof. We proceed by induction on n. If $n = 1$, then $F_1 = \Omega$ and we know that $P(E) = P(E \cap \Omega)$ by necessity. Therefore, suppose the statement is true for $k \leq n$. We show that the statement is true for $n + 1$.

Let $F_1, \dots, F_{n+1}$ be pairwise disjoint subsets satisfying Equation 2.8. Consider:

$$F = \bigcup_{i=1}^{n} F_i \tag{2.10}$$

Clearly if $x \in F$, then $x \notin F_{n+1}$ since $F_{n+1} \cap F_i = \emptyset$ for $i = 1, \dots, n$. Also, if $x \notin F$, then $x \in F_{n+1}$ since from Equation 2.8 we must have $F \cup F_{n+1} = \Omega$. Thus $F^c = F_{n+1}$ and we can conclude inductively that:

$$P(E) = P(E \cap F) + P(E \cap F_{n+1}) \tag{2.11}$$

We may apply Lemma 2.18 to show that:

$$E \cap F = E \cap \bigcup_{i=1}^{n} F_i = \bigcup_{i=1}^{n} (E \cap F_i) \tag{2.12}$$

Note that if $i \neq j$ then $(E \cap F_i) \cap (E \cap F_j) = \emptyset$ because $F_i \cap F_j = \emptyset$ and therefore:

$$P(E \cap F) = P\left(\bigcup_{i=1}^{n} (E \cap F_i)\right) = \sum_{i=1}^{n} P(E \cap F_i) \tag{2.13}$$

Thus, we may write:

$$P(E) = \sum_{i=1}^{n} P(E \cap F_i) + P(E \cap F_{n+1}) = \sum_{i=1}^{n+1} P(E \cap F_i) \tag{2.14}$$

This completes the proof. □

EXAMPLE 2.20. Welcome to Vegas! We're playing craps. In craps we roll two dice and winning combinations are determined by the sum of the values on the dice. An ideal first craps roll is 7. The sample space Ω in which we are interested has elements 36 elements, one each for the possible values the dice will show (the related set of sums can be easily obtained).

Suppose that the dice are colored blue and red (so they can be distinguished), and let's call the blue die number 1 and the red die number two. Let's suppose we are interested in the event that we roll a 1 on die number 1 and that the pair of values obtained sums to 7. There is only one way this can occur–namely we roll a 1 on die number one and a 6 on die number two. Thus the probability of this occurring is 1/36. In this case, event E is the event that we roll a 7 in our craps game and event F_1 is the event that die number one shows a 1. We could also consider event F_2 that die number one shows a 2. By similar reasoning, we know that the probability of both E and F_2 occurring is 1/36. In fact, if F_i is the event that one of the dice shows value i ($i = 1, \dots, 6$), then we know that:

$$P(E \cap F_i) = \frac{1}{36}$$

Clearly the events F_i ($i = 1, \dots, 6$) are pairwise disjoint (you can't have both a 1 and a 2 on the same die). Furthermore, $\Omega = F_1 \cup F_2 \cup \cdots \cup F_6$. (After all, some number has to appear on die number one!) Thus, we can compute:

$$P(E) = \sum_{i=1}^{6} P(E \cap F_i) = \frac{6}{36} = 16$$

EXERCISE 5. Suppose that I change the definition of F_i to read: value i appears on either die, while keeping the definition of event E the same. Do we still have:

$$P(E) = \sum_{i=1}^{6} P(E \cap F_i)$$

If so, show the computation. If not, explain why.

2. Random Variables and Expected Values

The concept of a random variable can be made extremely mathematically specific. A good intuitive understanding of a random variable is a variable X whose value is not known a priori and which is determined according to some probability distribution P that is a part of a probability space $(\Omega, \mathcal{F}, P)$.

Example 2.21. Suppose that we consider flipping a fair coin. Then the probability of seeing *heads* (or *tails*) should be 1/2. If we let X be a random variable that provides the outcome of the flip, then it will take on values *heads* or *tails* and it will take each value exactly 50% of the time.

The problem with allowing a random variable to take on arbitrary values (like *heads* or *tails*) is that it makes it difficult to use random variables in formulas involving numbers. There is a *very* technical definition of random variable that arises in formal probability theory. However, it is well beyond the scope of this class. We can, however, get a flavor for this definition in the following restricted form that is appropriate for this class:

Definition 2.22. Let $(\Omega, \mathcal{F}, P)$ be a discrete probability space. Let $D \subseteq \mathbb{R}$ be a finite discrete subset of real numbers. A random variable X is a function that maps each element of Ω to an element of D. Formally $X : \Omega \to D$.

Remark 2.23. Clearly, if $S \subseteq D$, then $X^{-1}(S) = \{\omega \in \Omega | X(\omega) \in S\} \in \mathcal{F}$. We can think of the probability of X taking on a value in $S \subseteq D$ is precisely $P(X^{-1}(S))$.

Using this observation, if $(\Omega, \mathcal{F}, P)$ is a discrete probability distribution function and $X : \Omega \to D$ is a random variable and $x \in D$ then let $P(x) = P(X^{-1}(\{x\})$. That is, the probability of X taking value x is the probability of the element in Ω corresponding to x.

Definition 2.22 still is a bit complex, so it's easiest to give a few examples.

Example 2.24. Consider our coin flipping random variable. Instead of having X take values *heads* or *tails*, we can instead let X take on values 1 if the coin comes up *heads* and 0 if the coin comes up *tails*. Thus if $\Omega = \{\textit{heads}, \textit{tails}\}$, then $X(\textit{heads}) = 1$ and $X(\textit{tails}) = 0$.

Example 2.25. When Ω (in probability space $(\Omega, \mathcal{F}, P)$) is already a subset of $\mathbb{R}$, then defining random variables is very easy. The random variable can just be the obvious mapping from Ω into itself. For example, if we consider rolling a fair die, then $\Omega = \{1, \dots, 6\}$ and any random variable defined on $(\Omega, \mathcal{F}, P)$ will take on values $1, \dots, 6$.

Definition 2.26. Let $(\Omega, \mathcal{F}, P)$ be a discrete probability distribution and let $X : \Omega \to D$ be a random variable. Then the *expected value* of X is:

$$\mathbb{E}(X) = \sum_{x \in D} xP(x) \tag{2.15}$$

Example 2.27. Let's play a die rolling game. You put up your own money. Even numbers lose \$10 times the number rolled, while odd numbers win \$12 times the number rolled. What is the expected amount of money you'll win in this game?

Let $\Omega = \{1, \dots, 6\}$. Then $D = \{12, -20, 36, -40, 60, -60\}$: these are the dollar values you will win for various rolls of the dice. Then the expected value of X is:

$$\mathbb{E}(X) = 12\left(\frac{1}{6}\right) + (-20)\left(\frac{1}{6}\right) + 36\left(\frac{1}{6}\right) + (-40)\left(\frac{1}{6}\right) + 60\left(\frac{1}{6}\right) + (-60)\left(\frac{1}{6}\right) = -2 \tag{2.16}$$

Would you still want to play this game considering the expected payoff is $-\$2$?

3. Conditional Probability

Suppose we are given a discrete probability space $(\Omega, \mathcal{F}, P)$ and we are told that an event E has occurred. We now wish to compute the probability that some other event F has occurred. This value is called the conditional probability of event F given event E and is written $P(F|E)$.

EXAMPLE 2.28. Suppose we roll a fair 6 sided die twice. The sample space in this case is the set $\Omega = \{(x, y)|x = 1, \dots, 6,\ y = 1, \dots, 6\}$. Suppose I roll a 2 on the first try. I want to know what the probability of rolling a combined score of 8 is. That is, given that I've rolled a 2, I wish to determine the conditional probability of rolling a 6.

Since the die is fair, the probability of rolling any pair of values $(x, y) \in \Omega$ is equally likely. There are 36 elements in Ω and so each is assigned a probability of $1/36$. That is, $(\Omega, \mathcal{F}, P)$ is defined so that $P((x, y)) = 1/36$ for each $(x, y) \in \Omega$.

Let E be the event that we roll a 2 on the first try. We wish to assign a new set of probabilities to the elements of Ω to reflect this information. We know that our final outcome must have the form $(2, y)$ where $y \in \{1, \dots, 6\}$. **In essence, E becomes our new sample space.** Further, we know that each of these outcomes is equally likely because the die is fair. Thus, we may assign $P((2, y)|E) = 1/6$ for each $y \in \{1, \dots, 6\}$ and $P((x, y)|E) = 0$ just in case $x \neq 2$, so $(x, y) \notin E$. This last definition occurs because we know that we've already observed a 2 on the first roll, so it's impossible to see another first number not equal to 2.

At last, we can answer the question we originally posed. The only way to obtain a sum equal to 8 is to roll a six on the second attempt. Thus, the probability of rolling a combined score of 8 given a 2 on the first roll is $1/6$.

LEMMA 2.29. *Let $(\Omega, \mathcal{F}, P)$ be a discrete probability space and suppose that event $E \subseteq \Omega$. Then $(E, \mathcal{F}_E, P_E)$ is a discrete probability space when:*

$$P_E(F) = \frac{P(F)}{P(E)} \tag{2.17}$$

for all $F \subseteq E$ and $P_E(\omega) = 0$ for any $\omega \notin E$.

PROOF. Our objective is to construct a new probability space $(E, \mathcal{F}_E, P_E)$.

If $\omega \notin E$, then we can assign $P_E(\omega) = 0$. Suppose that $\omega \in E$. For $(E, \mathcal{F}_E, P_E)$ to be a discrete probability space, we must have: $P_E(E) = 1$ or:

$$P_E(E) = \sum_{\omega \in E} P_E(\omega) = 1 \tag{2.18}$$

We know from the Defintion 2.10 that

$$P(E) = \sum_{\omega \in E} P(\omega)$$

Thus, if we assign $P_E(\omega) = P(\omega)/P(E)$ for all $\omega \in E$, then Equation 2.18 will be satisfied automatically. Since for any $F \subseteq E$ we know that:

$$P(F) = \sum_{\omega \in F} P(\omega)$$

it follows at once that $P_E(F) = P(F)/P(E)$. Finally, if $F_1, F_2 \subseteq E$ and $F_1 \cap F_2 = \emptyset$, then the fact that $P_E(F_1 \cup F_2) = P_E(F_1) + P_E(F_2)$ follows from the properties of the original probability space $(\Omega, \mathcal{F}, P)$. Thus $(E, \mathcal{F}_E, P_E)$ is a discrete probability space. □

REMARK 2.30. The previous lemma gives us a direct way to construct $P(F|E)$ for arbitrary $F \subseteq \Omega$. Clearly if $F \subseteq E$, then

$$P(F|E) = P_E(F) = \frac{P(F)}{P(E)}$$

Now suppose that F is not a subset of E but that $F \cap E \neq \emptyset$. Then clearly, the only possible events that can occur in F, given that E has occurred are the ones that are also in E. Thus, $P_E(F) = P_E(E \cap F)$. More to the point, we have:

$$P(F|E) = P_E(F \cap E) = \frac{P(F \cap E)}{P(E)} \tag{2.19}$$

DEFINITION 2.31 (Conditional Probability). Given a discrete probability space $(\Omega, \mathcal{F}, P)$ and an event $E \in \mathcal{F}$, the conditional probability of event $F \in \mathcal{F}$ given event E is:

$$P(F|E) = \frac{P(F \cap E)}{P(E)} \tag{2.20}$$

EXAMPLE 2.32 (Simple Blackjack). Blackjack is a game in which decisions can be made entirely based on conditional probabilities. The chances of a card appearing are based entirely on whether or not you have seen that card already since cards are discarded as the dealer works her way through the deck.

Consider a simple game of Blackjack played with only the cards A, 2, 3, 4, 5, 6, 7, 8, 9, 10, J, Q, K. In this game, the dealer deals two cards to the player and two to herself. The objective is to obtain a score as close to 21 as possible without going over. Face cards are worth 10, A is worth 1 or 11 all other cards are worth their face value. We'll assume that the dealer must hit (take a new card) on 16 and below and will stand on 17 and above.

The complete sample space in this case is very complex; it consists of all possible valid hands that could be dealt over the course of a standard play of the game. We can however consider a simplified sample space of hands after the initial deal. In this case, the sample space has the form:

$$\Omega = \{(\langle x, y\rangle, \langle s, t\rangle)\}$$

Here x, y, s, t are cards without repeats. The total size of the sample space is

$$13 \times 12 \times 11 \times 10 = 17,160$$

This can be seen by noting that the player can receive any of the 13 cards as first card and any of the remaining 12 cards for the second card. The dealer then receives 1 of the 11 remaining cards and then 1 of the 11 remaining cards.

Let's suppose that the player is dealt 10 and 6 for a score of 16 while the dealer receives a 4 and 5 for a total of 9. If we suppose that the player decides to hit, then the large sample space (Ω) becomes:

$$\Omega = \{(\langle x, y, z\rangle, \langle s, t\rangle)\}$$

which has size:

$$13 \times 12 \times 11 \times 10 \times 9 = 154,440$$

while the event is:

$$E = \{(\langle 10, 6, z\rangle, \langle 4, 5\rangle)\}$$

There are 9 possible values for z and thus $P(E) = 9/154,440$.

Let us now consider the probability of busting on our first hit. This is event F and is given as:

$$F = \{(\langle x, y, z\rangle, \langle s, t\rangle) : x + y + z > 21\}$$

(Here we take some liberty by assuming that we can add card values like digits.)

The set F is very complex, but we can see immediately that:

$$E \cap F = \{(\langle 10, 6, z\rangle, \langle 4, 5\rangle) : z \in \{7, 8, 9, J, Q, K\}\}$$

because these are the hands that will cause us to bust. Thus we can easily compute:

$$P(F|E) = \frac{P(E \cap F)}{P(E)} = \frac{6/154,440}{9/154,440} = \frac{6}{9} = \frac{2}{3} \tag{2.21}$$

Thus the probability of not busting given the hand we have drawn must be 1/3. We can see at once that our odds when taking a hit are not very good. Depending on the probabilities associated with the dealer busting, it may be smarter for us to not take a hit and see what happens to the dealer, however in order to be sure we'd have to work out the chances of the dealer busting (since we know she will continue to hit until she busts or exceeds our value of 16).

Unfortunately, this computation is quite tedious and we will not include it here.

REMARK 2.33. The complexity associated with blackjack makes knowing exact probabilities difficult, if not impossible. Thus most card counting strategies use heuristics to attempt to understand approximately what the probabilities are for winning given the history of observed hands. To do this, simple numeric values are assigned to cards, generally a +1 to cards with low values (2,3, 4 etc.) a 0 to cards with mid-range values (7, 8, 9) and negative values for face cards (10, J, Q, K). As the count gets *high* there are more face cards in the deck and thus the chances of the dealer busting or the player drawing blackjack increase. If the count is low, there are fewer face cards in the deck and the chance of the dealer drawing a sufficient number of cards without busting is higher. Thus, players favor tables with high counts.

The chief roadblock to card counters is knowing the count before sitting at the table. The MIT card counting team (featured in the movie *21*) used a *big player team* strategy. In this strategy, card counters would sit at a table and make safe bets winning or losing very little over the course of time. They would keep the card count and signal *big players* from their team who would arrive at the table and make large bets when the count was high (in their favor). The big players would leave once signaled that the count had dropped. Using this strategy, the MIT players cleared millions from the casinos using basic probability theory.

EXERCISE 6. Use Definition 2.31 to compute the probability of obtaining a sum of 8 in two rolls of a die given that in the first roll a 1 or 2 appears. [Hint: The space of outcomes is still $\Omega = \{(x, y)|x = 1, \ldots, 6,\ y = 1, \ldots, 6\}$. First identify the event E within this space.

How many elements within this set will enable you to obtain an 8 in two rolls? This is the set $E \cap F$ What is the probability of $E \cap F$? What is the probability of E? Use the formula in Defintion 2.31. It might help to write out the space Ω.]

Example 2.34 (The Monty Hall Problem). Congratulations! You are a contestant on *Let's Make a Deal* and you are playing for *The Big Deal of the Day*! You must choose between Door Number 1, Door Number 2 and Door Number 3. Behind one of these doors is a fabulous prize! Behind the other two doors, are goats. Once you choose your door, Monty Hall (or Wayne Brady, you pick) will reveal a door that did not have the big deal. At this point you can decide if you want to keep the original door you chose or switch doors. When the time comes, what do you do?

It is tempting at first to suppose that it doesn't matter whether you switch or not. You have a 1/3 chance of choosing the correct door on your first try, so why would that change after you are given information about an incorrect door? It turns out–it does matter.

To solve this problem, it helps to understand the set of potential outcomes. There are really three possible pieces of information that determine an outcome:

(1) Which door the producer chooses for the big deal,
(2) Which door you choose first, and
(3) Whether you switch or not.

For the first decision, there are three possibilities (three doors). For the second decision, there are again three possibilities (again three doors). For the third decision there are two possibilities (either you switch, or not). Thus, there are $3 \times 3 \times 2 = 18$ possible outcomes. These outcomes can be visualized in the order in which the decisions are made (more or less) this is shown in Figure 2.1. The first step (where the producers choose a door to hide the prize) is not observable by the contestant, so we adorn this part of the diagram with a box. We'll get into what this box means when we discuss *game trees*.

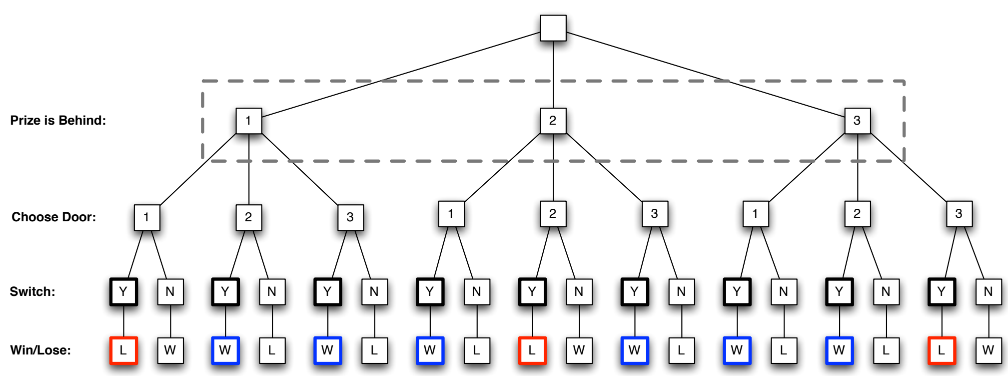


Figure 2.1. The Monty Hall Problem is a multi-stage decision problem whose solution relies on conditional probability. The stages of decision making are shown in the diagram. We assume that the prizes are randomly assigned to the doors. We can't see this step–so we've adorned this decision with a square box. We'll discuss these boxes more when we talk about *game trees*. You the player must first choose a door. Lastly, you must decide whether or not to switch doors having been shown a door that is incorrect.

The next to the last row (labeled "Switch") of Figure 2.1 illustrates the 18 elements of the probability space. We could assume that they are all equally likely (i.e., that you randomly choose a door and that you randomly decide to switch and that the producers of the show randomly choose a door for hiding the prize). In this case, the probability of any outcome is 1/18. Now, let's focus exclusively on the outcomes in which we decide to switch. In the figure, these appear with bold, colored borders. This is our event set E. Suppose event set F consists of those outcomes in which the contestant wins. (This is shown in the bottom row of the diagram with a W.) We are now interesting in $P(F|E)$. That is, what are our chances of winning, given we actively choose to switch?

Within E, there are precisely 6 outcomes in which we win. If each of these mutually exclusive outcomes has probability 1/18:

$$P(E \cap F) = 6\left(\frac{1}{18}\right) = \frac{1}{3}$$

Obviously, we switch in 9 of the possible 18 outcomes, so:

$$P(E) = 9\left(\frac{1}{18}\right) = \frac{1}{2}$$

Thus we can compute:

$$P(F|E) = \frac{P(E \cap F)}{P(E)} = \frac{1/3}{1/2} = \frac{2}{3}$$

Thus if we switch, there is a 2/3 chance we will win the prize. If we don't switch, there is only a 1/3 chance we win the prize. Thus, switching is better than not switching.

If this reasoning doesn't appeal to you, there's another way to see that the chance of winning given switching is 2/3: In the case of switching we're making a conscious decision; there is no probabilistic voodoo that is affecting this part of the outcome. So just consider the outcomes in which we switch. Notice there are 9 outcomes in which we switch from our original door to a door we did not pick first. In 6 of these 9 we win the prize, while in 3 we fail to win the prize. Thus, the chances of winning the prize when we switch is 6/9 or 2/3.

Exercise 7. Show (in anyway you like) that the probability of winning given that you *do not* switch doors is 1/3.

Exercise 8. In the little known *Lost Episodes of Let's Make a Deal*, Monty (or Wayne) introduces a fourth door. Suppose that you choose a door and then are shown two incorrect doors and given the chance to switch. Should you switch? Why? [Hint: Build a figure like Figure 2.1. It will be a bit large. Use the same reasoning we used to compute the probability of successfully winning the prize in the previous example.

Remark 2.35. The *Monty Hall Problem* first appeared in 1975 in the American Statistician (if you believe Wikipedia–http://en.wikipedia.org/wiki/Monty_Hall_problem). It's one of those great problems that seems so obvious until you start drawing diagrams with probability spaces. Speaking of Wikipedia, the referenced article is accessible, but contains more advanced material. We'll cover some of it later. On a related note, this example takes us into our first real topic in game theory, *Optimal Decision Making Under Uncertainty*. As we remarked in the example, the choice of whether to switch is really not a probabilistic thing; it's a decision that you must make in order to improve your happiness. This, at the core, is what decision science, optimization theory and game theory is all about. Making a good decision given all the information (stochastic or not) to improve your happiness.

Definition 2.36 (Independence). Let $(\Omega, \mathcal{F}, P)$ be a discrete probability space. Two events $E, F \in \mathcal{F}$ are called *independent* if $P(E|F) = P(E)$ and $P(F|E) = P(F)$.

Theorem 2.37. *Let $(\Omega, \mathcal{F}, P)$ be a discrete probability space. If $E, F \in \mathcal{F}$ are independent events, then $P(E \cap F) = P(E)P(F)$.*

Proof. We know that:

$$P(E|F) = \frac{P(E \cap F)}{P(F)} = P(E)$$

Multiplying by $P(F)$ we obtain $P(E \cap F) = P(E)P(F)$. This completes the proof. $\square$

Example 2.38. Consider rolling a fair die twice in a row. Let Ω be the sample space of pairs of die results that will occur. Thus $\Omega = \{(x, y)|x = 1, \dots, 6,\ y = 1, \dots, 6\}$. Let E be the event that says we obtain a 6 on the first roll. Then $E = \{(6, y) : y = 1, \dots, 6\}$ and let F be the event that says we obtain a 6 on the second roll. Then $F = \{(x, 6) : x = 1, \dots, 6\}$. Obviously these two events are independent. The first roll *cannot* affect the outcome of the second roll, thus $P(F|E) = P(F)$. We know that $P(E) = P(F) = 1/6$. That is, there is a 1 in 6 chance of observing a 6. Thus the chance of rolling double sixes in two rolls is precisely the probability of both events E and F occurring. Using our result on independent events we can see that: $P(E \cap F) = P(E)P(F) = (1/6)^2 = 1/36$; just as we expect it to be.

Example 2.39. Suppose we're interested in the probability of rolling at least one six in two rolls of a die. Again, the rolls are independent. Let's consider the probability of not rolling a six at all. Let E be the event that we *do not* roll a 6 in the first roll. Then $P(E) = 5/6$ (there are 5 ways to not roll a 6). If F is the event that we do not roll a 6 on the second roll, then again $P(F) = 5/6$. Since theses events are independent (as before) we can compute $P(E \cap F) = (5/6)(5/6) = 25/36$. This is the probability of not rolling a 6 on the first roll and not rolling a 6 on the second roll. We are interested in rolling at least one 6. Thus, if G is the event of not rolling a six at all, then G^c must be the event of rolling at least one 6. Thus $P(G^c) = 1 - P(G) = 1 - 25/36 = 11/36$.

Exercise 9. Compute the probability of rolling a double 6 in 24 rolls of a pair of dice. [Hint: Each roll is independent of the last roll. Let E be the event that you *do not* roll a double 6 on a given roll. The probability of E is 35/36 (there are 35 other ways the dice could come out other than double 6). Now, compute the probability of not seeing a double six in all 24 rolls using independence. (You will have a power of 24.) Let this probability be p. Finally, note that the probability of a double 6 occurring is precisely $1 - p$. To see this note that p is the probability of the event that a double six does not occur. Thus, the probability of the event that a double 6 does occur must be $1 - p$.]

4. Bayes Rule

Bayes rule (or theorem) is a useful little theorem that allows us to compute certain conditional probabilities given other conditional probabilities and a bit of information on the probability space in question.

Lemma 2.40 (Bayes Theorem 1). *Let $(\Omega, \mathcal{F}, P)$ be a discrete probability space and suppose that $E, F \in \mathcal{F}$, then:*

$$P(F|E) = \frac{P(E|F)P(F)}{P(E)} \tag{2.22}$$

Exercise 10. Prove Bayes Theorem 1. [Hint: Use Definition 2.31.]

We can generalize this theorem when we have a collection of sets $F_1, \dots, F_n \in \mathcal{F}$ that completely partition Ω and are pairwise disjoint.

Theorem 2.41 (Bayes Theorem 2). *Let $(\Omega, \mathcal{F}, P)$ be a discrete probability space and suppose that $E, F_1, \dots, F_n \in \mathcal{F}$ with $F_1, \dots, F_n$ being pairwise disjoint and*

$$\Omega = \bigcup_{i=1}^{n} F_i$$

Then:

$$P(F_i|E) = \frac{P(E|F_i)P(F_i)}{\sum_{j=1}^{n} P(E|F_j)P(F_j)} \tag{2.23}$$

Proof. Consider:

$$\sum_{j=1}^{n} P(E|F_j)P(F_j) = \sum_{j=1}^{n} \left(\frac{P(E \cap F_j)}{P(F_j)} P(F_j) \right) = \sum_{j=1}^{n} P(E \cap F_j) = P(E)$$

by Theorem 2.19. From Lemma 2.40, we conclude that:

$$\frac{P(E|F_i)P(F_i)}{\sum_{j=1}^{n} P(E|F_j)P(F_j)} = \frac{P(E|F_i)P(F_i)}{P(E)} = P(F_i|E)$$

This completes the proof. □

Example 2.42. Here's a rather morbid example: suppose that a specific disease occurs with a probability 1 in 1,000,000. A test exists to determine whether or not an individual has this disease. When an individual has the disease, the test will detect it 99 times out of 100. The test also has a false positive rate of 1 in 1,000 (that is there is a 0.001 probability of misdiagnosis). The treatment for this disease is costly and unpleasant. You have just tested positive. What do you do?

We need to understand the events that are in play here:

(1) The event of having the disease (F)
(2) The event of testing positive (E)

We are interested in knowing the following:

$$P(F|E) = \text{The probability of having the disease given a positive test.}$$

We know the following information:

(1) $P(F) = 1 \times 10^{-6}$: There is a 1 in 1,000,000 chance of having this disease.
(2) $P(E|F) = 0.99$: The probability of testing positive given that you have the disease is 0.99.

(3) $P(E|F^c) = 0.001$: The probability of testing positive given that you do not have the disease is 1 in 1,000.

We can apply Bayes Theorem:

$$P(F|E) = \frac{P(E|F)P(F)}{P(E|F)P(F) + P(E|F^c)P(F^c)} = \frac{(0.99)(1 \times 10^{-6})}{(0.99)(1 \times 10^{-6}) + (0.001)(1 - 1 \times 10^{-6})} = 0.00098 \tag{2.24}$$

Thus the probability of having the disease given the positive test is less than 1 in 1,000. You should probably get a few more tests done before getting the unpleasant treatment.

Exercise 11. In the previous example, for what probability of having the disease is there a 1 in 100 chance of having the disease given that you've tested positive? [Hint: I'm asking for what value of $P(F)$ is the value of $P(F|E)$ 1 in 100. Draw a graph of $P(F|E)$ and use your calculator.

Exercise 12. Use Bayes Rule to show that the probability of winning in the Monty Hall Problem is 2/3.

CHAPTER 3

Utility Theory

1. Decision Making Under Certainty

In the example 2.42 we began looking at the problem of making decisions under uncertainty. In this section, we explore this topic and develop an axiomatic treatment of this subject. This topic represents one of the fundamental building blocks of modern decision theory. Suppose we are presented with a set of prizes denoted $A_1, \dots, A_n$.

EXAMPLE 3.1. In *Deal or No Deal*, the prizes are monetary in nature. In shows like *Let's Make a Deal* or *The Price is Right* the prizes may be monetary in nature or they may be tangible goods.

DEFINITION 3.2 (Lottery). A *lottery* $L = \langle\{A_1, \dots, A_n\}, P\rangle$ is a collection of prizes (or rewards, or costs) $\{A_1, \dots, A_n\}$ along with a discrete probability distribution P with the sample space $\{A_1, \dots, A_n\}$. We denote the set of all lotteries over $A_1, \dots, A_n$ by $\mathcal{L}$.

REMARK 3.3. To simplify notation, we will say that $L = \langle(A_1, p_1), \dots, (A_n, p_n)\rangle$ is the lottery consisting of prizes A_1 through A_n where you receive prize A_1 with probability p_1, prize A_2 with probability p_2 etc.

REMARK 3.4. The lottery in which we win prize A_i with probability 1 and all other prizes with probability 0 will be denoted as A_i as well. Thus, the prize A_i can be thought of as being equivalent to a lottery in which one always wins prize A_i.

EXAMPLE 3.5. Congratulations! You are on *The Price is Right*! You are going to play *Temptation*. In this game, you are offered four prizes and given their dollar value. From the dollar values you must then construct the price of a car. Once you are shown all the prizes (and constructed a guess for the price of the car) you must make a choice between taking the prizes and leaving or hoping that your have chosen the right numbers in the price of the car.

In this example, there are two lotteries: the prize option and the car option. The prize option contains a single prize consisting of the various items you've seen, denote this A_1. This lottery is (A_1, P_1) where $P_1(A_1) = 1$. The car option contains two prizes: the car A_2, and the null prize A_0 (where you leave with nothing). Depending up the dynamics of the game, this lottery has form: $\langle\{A_0, A_2\}, P_2\rangle$ where $P_2(A_0) = p$ and $P_2(A_2) = 1 - p$ and $p \in (0, 1)$ and depends on the nature of the prices of the prizes in A_1, which were used to construct the guess for the price of the car.

EXERCISE 13. First watch the full excerpt from *Temptation* at `http://www.youtube.com/watch?v=rQ06urOTxE0`. Assume you have *no* knowledge on the price of the car. Compute the value of p in the probability distribution on the lottery containing the car. [Hint: Suppose I tell you that a model car you could win as a value between $10 and $19. I show you an alternate prize worth 46¢. You must choose either the 4 or the 6 for the value of the second digit in the price of the model car. What is the probability you choose the correct value?

Remark 3.6. In a lottery (of this type) we do not assume that we will determine the probability distribution P as a result of repeated exposure. (This is not like *The Pennsylvania Lottery*.) Instead, the probability is given *ab initio* and is constant.

Definition 3.7 (Preference). Let L_1 and L_2 be lotteries. We write $L_1 \succeq L_2$ to indicate that an individual *prefers* lottery L_1 to lottery L_2. If both $L_1 \succeq L_2$ and $L_2 \succeq L_1$, then $L_1 \sim L_2$ and L_1 and L_2 are considered equivalent to the individual.

Remark 3.8. The axiomatic treatment of utility theory rests on certain assumptions about an individual's behavior when they are confronted with a choice of two or more lotteries. We have already seen this type of scenario in Example 3.5. We assume these choices are governed by preference. Preference can vary from individual to individual.

Remark 3.9. **For the remainder of this section** we will assume that every lottery consists of prizes $A_1, \dots, A_n$ and that there prizes are preferred in order:

$$A_1 \succeq A_2 \succeq \cdots \succeq A_n \tag{3.1}$$

Assumption 1. Let L_1, L_2 and L_3 be lotteries:

1. Either $L_1 \succeq L_2$ or $L_2 \succeq L_1$ or $L_1 \sim L_2$.
2. If $L_1 \preceq L_2$ and $L_2 \preceq L_3$, then $L_1 \preceq L_3$.
3. If $L_1 \sim L_2$ and $L_2 \sim L_3$, then $L_1 \sim L_3$.
4. If $L_1 \succeq L_2$ and $L_2 \succeq L_1$, then $L_1 \sim L_2$.

Remark 3.10. Item 1 of Assumption 1 states that the ordering $\preceq$ is a total ordering on the set of all lotteries with which an individual may be presented. That is, we can compare any two lotteries two each other and always be able to decide which one is preferred or whether they are equivalent.

Item 2 of Assumption 1 states that this ordering is *transitive*.

It should be noted that these assumptions rarely work out in real-life. The idea that everyone has in their mind a total ranking of all possible lotteries (or could construct one) is difficult to believe. Ignoring that however, problems often arise more often with the assumption of transitivity.

Remark 3.11. Assumption 1 asserts that preference is *transitive* over the set of all lotteries. Since it is clear that preference should be *reflexive* (i.e., $L_1 \sim L_1$ for all lotteries L_1) and *symmetric* ($L_1 \sim L_2$ if and only if $L_2 \sim L_1$ for all lotteries L_1 and L_2) preferential equivalence is an *equivalence relation* over the set of all lotteries.

Example 3.12 (Problem with transitivity). For this example, you must use your imagination and think like a pre-schooler (probably a boy pre-schooler).

Suppose I present a pre-schooler with the following choices (lotteries with only one item): a ball, a stick and a crayon (and paper). If I present the choice of the stick and crayon, the child may choose the crayon (crayons are fun to use when you have lot's of imagination). In presenting the stick and the ball, the child may choose the stick (a stick can be made into anything using imagination). On the other hand, suppose I present the crayon and the ball. If the child chooses the ball, then transitivity is violated. Why might the child choose the ball? Suppose that the ball is not *a ball* but the ultimate key to the galaxy's last energy source! The child's preferences will change depending upon the current requirements of his/her imagination. Thus leading to a simple example of an intransitive ordering on the items he is presented. This is evident only when presenting the items in pairs.

Definition 3.13 (Compound Lottery). Let $L_1, \dots, L_n$ be a set of lotteries and suppose that the probability of being presented with lottery i ($i = 1, \dots, n$) is q_i. A lottery $Q = \langle (L_1, q_1), \dots, (L_n, q_n) \rangle$ is called a *compound lottery.*

Example 3.14. Two contestants are playing a new game called *Flip of a Coin!* in which "Your life can change on the flip of a coin!" The contestants first enter a round in which they choose heads or tails. A coin is flipped and the winner is offered a choice of a sure $\$1,000$ or a 10% chance of winning a car. The loser is presented with a lottery in which they can leave with nothing (and stay dry) or choose a lottery in which there is a 10% chance they will win $1,000 and 90% they will fall into a tank of water dyed blue.

The coin flip stage is a compound lottery composed of the lotteries the contestants will be offered later in the show.

Assumption 2. Let $L_1, \dots, L_n$ be a compound lottery with probabilities $q_1, \dots, q_n$ and suppose each L_i is composed of prizes $A_1, \dots, A_m$ with probabilities p_{ij} ($j = 1, \dots, m$). Then this compound lottery is equivalent to a simply lottery in which the probability of prize A_j is:

$$r_j = q_1 p_{1j} + q_2 p_{2j} + \cdots + q_n p_{nj}$$

Remark 3.15. All Assumption 2 is saying is that compound lotteries can be transformed into equivalent simple lotteries. Note further that the probability of prize j (A_j) is actually:

$$P(A_j) = \sum_{i=1}^{n} P(A_j | L_j) P(L_j) \tag{3.2}$$

This statement should be very clear from Theorem 2.19, when we define our probability space in the right way.

Assumption 3. For each prize (or lottery) A_i there is a number $u_i \in [0, 1]$ so that the prize A_i (or lottery L_i) is preferentially equivalent to the lottery in which you win prize A_1 with probability u_i and A_n with probability $1 - u_i$ and all other prizes with probability 0. This lottery will be denoted $\tilde{A}_i$.

Remark 3.16. Assumption 3 is a strange assumption often called the *continuity* assumption. It assumes that for any ordered set of prizes $(A_1, \dots, A_n)$ that a person would view winning any specific prize (A_i) as equivalent to playing a game of chance in which either the worst or best prize could be obtained.

This assumption is clearly not valid in all cases. Suppose that the best prize was a new car, while the worst prize is spending 10 years in jail. If the prize in question (A_i) is that you receive \$100, is there a game of chance you would play involving a new car or 10 years in jail that would be equal to receiving \$100?

Assumption 4. If $L = \langle (A_1, p_1), \dots, (A_i, p_i), \dots, (A_n, p_n) \rangle$ is a lottery, then L is preferentially equivalent to the lottery $\langle (A_1, p_1), \dots, (\tilde{A}_i, p_i), \dots, (A_n, p_n) \rangle$

Remark 3.17. Assumption 4 only asserts that we can substitute any equivalent lottery in for a prize and not change the individuals preferential ordering. It is up to you to evaluate the veracity of this claim in real life.

Assumption 5. A lottery L in which A_1 is obtained with probability p and A_n is obtained with probability $(1 - p)$ is always preferred or equivalent to a lottery in which A_1 is obtained with probability p' and A_n is obtained with probability $(1 - p')$ if and only if $p \geq p'$.

Remark 3.18. Our last assumption, Assumption 5 states that we would prefer (or be indifferent) to win A_1 with a higher probability and A_n with lower probability. This assumption is reasonable when we have the case $A_1 \succeq A_n$, however as [**LR89**] point out, there are psychological reasons why this assumption may be violated.

At last we've reached the fundamental theorem in our study of utility.

Theorem 3.19 (Expected Utility Theorem). *Let $\succeq$ be a preference relation satisfying Assumptions 1 - 5 over the set of all lotteries $\mathcal{L}$ defined over prizes $A_1, \dots, A_n$. Furthermore, assume that:*

$$A_1 \succeq A_2 \succeq \cdots \succeq A_n$$

Then there is a function $u : \mathcal{L} \to [0, 1]$ with the property that:

$$u(L_1) \geq u(L_2) \iff L_1 \succeq L_2 \tag{3.3}$$

Proof. The trick to this proof is to define the utility function and then show the if and only if statement. We will define the utility function as follows:

(1) Define $u(A_1) = 1$. Recall that A_1 is not only prize A_1 but also the lottery in which we receive A_1 with probability 1. That is the lottery in which $p_1 = 1$ and $p_2 \dots, p_n = 0$.
(2) Define $u(A_n) = 0$. Again, recall that A_n is also the lottery in which we receive A_n with probability 1.
(3) By Assumption 3, for lottery A_i ($i \neq 1$ and $i \neq n$) there is a u_i so that A_i is equivalent to $\tilde{A}_i$: the lottery in which you win prize A_1 with probability u_i and A_n with probability $1 - u_i$ and all other prizes with probability 0. Define $u(A_i) = u_i$.
(4) Let $L \in \mathcal{L}$ be a lottery in which we win prize A_i with probability p_i. Then

$$u(L) = p_1 u_1 + p_2 u_2 + \cdots + p_n u_n \tag{3.4}$$

Here $u_1 \equiv 1$ and $u_n \equiv 0$.

We now show that this utility function satisfies Expression 3.3.

($\Leftarrow$) Let $L_1, L_2 \in \mathcal{L}$ and suppose that $L_1 \succeq L_2$. Suppose:

$$L_1 = \langle (A_1, p_1), (A_2, p_2), \dots, (A_n, p_n) \rangle$$
$$L_2 = \langle (A_1, q_1), (A_2, q_2), \dots, (A_n, q_n) \rangle$$

By Assimption 3, for each A_i, $(i \neq 1, i \neq n)$, we know that $A_i \sim \tilde{A}_i$ with $\tilde{A}_i \equiv \langle (A_1, u_i), (A_n, 1 - u_i) \rangle$. Then by Assumption 4 we know:

$$L_1 \sim \langle (A_1, p_1), (\tilde{A}_2, p_2), \dots, (\tilde{A}_{n-1}, p_{n-1}), (A_n, p_n) \rangle$$
$$L_2 \sim \langle (A_1, q_1), (\tilde{A}_2, q_2), \dots, (\tilde{A}_{n-1}, q_{n-1}), (A_n, q_n) \rangle$$

These are compound lotteries and we can expand them as:

$$\begin{aligned} L_1 \sim \langle (A_1, p_1), (\langle (A_1, u_2), (A_n, (1 - u_2)) \rangle, p_2), \dots, \\ (\langle (A_1, u_{n-1}), (A_n, (1 - u_{n-1})) \rangle, p_{n-1}), (A_n, p_n) \rangle \end{aligned} \tag{3.5}$$

$$\begin{aligned} L_1 \sim \langle (A_1, q_1), (\langle (A_1, u_2), (A_n, (1 - u_2)) \rangle, q_2), \dots, \\ (\langle (A_1, u_{n-1}), (A_n, (1 - u_{n-1})) \rangle, q_{n-1}), (A_n, q_n) \rangle \end{aligned} \tag{3.6}$$

We may apply Assumption 2 to transform these compound lotteries into simple lotteries by combing the like prizes:

$$L_1 \sim \langle (A_1, p_1 + u_2 p_2 + \dots + u_{n-1} p_{n-1}), (A_n, (1 - u_2) p_2 + \dots + (1 - u_{n-1}) p_{n-1} + p_n) \rangle$$
$$L_2 \sim \langle (A_1, q_1 + u_2 q_2 + \dots + u_{n-1} q_{n-1}), (A_n, (1 - u_2) q_2 + \dots + (1 - u_{n-1}) q_{n-1} + q_n) \rangle$$

Let

$$\tilde{L}_1 \equiv \langle (A_1, p_1 + u_2 p_2 + \dots + u_{n-1} p_{n-1}), (A_n, (1 - u_2) p_2 + \dots + (1 - u_{n-1}) p_{n-1} + p_n) \rangle$$
$$\tilde{L}_2 \equiv \langle (A_1, q_1 + u_2 q_2 + \dots + u_{n-1} q_{n-1}), (A_n, (1 - u_2) q_2 + \dots + (1 - u_{n-1}) q_{n-1} + q_n) \rangle$$

We can apply Assumption 1 to see: $L_1 \sim \tilde{L}_1$ and $L_2 \sim \tilde{L}_2$ and $L_1 \succeq L_2$ implies that $\tilde{L}_1 \succeq \tilde{L}_2$. We can now apply Assumption 5 to conclude that:

$$p_1 + u_2 p_2 + \dots + u_{n-1} p_{n-1} \geq q_1 + u_2 q_2 + \dots + u_{n-1} q_{n-1} \tag{3.7}$$

Note, however, that

$$u(L_1) = p_1 + u_2 p_2 + \dots + u_{n-1} p_{n-1} \tag{3.8}$$
$$u(L_2) = q_1 + u_2 q_2 + \dots + u_{n-1} q_{n-1} \tag{3.9}$$

Thus we have $u(L_1) \geq u(L_2)$.

($\Rightarrow$) Suppose now that $L_1, L_2 \in \mathcal{L}$ and that $u(L_1) \geq u(L_2)$. Then we know that:

$$\begin{aligned} u(L_1) \equiv u_1 p_1 + u_2 p_2 + \dots + u_{n-1} p_{n-1} + u_n p_n \geq \\ u_1 q_1 + u_2 q_2 + \dots + u_{n-1} q_{n-1} + u_n q_n \equiv u(L_2) \end{aligned} \tag{3.10}$$

As before, we may note that $L_1 \sim \tilde{L}_1$ and $L_2 \sim \tilde{L}_2$. We may further note that $u(L_1) = u(\tilde{L}_1)$ and $u(L_2) = u(\tilde{L}_2)$. To see this, note that in L_1, the probability associated to prize A_1 is:

$$p_1 + u_2 p_2 + \dots + u_{n-1} p_{n-1}$$

Thus, (since $u_1 \equiv 1$ and $u_n \equiv 0$) we know that:

$$u(\tilde{L}_1) = u_1\left(p_1 + u_2p_2 + \cdots + u_{n-1}p_{n-1}\right) = u_1p_1 + u_2p_2 + \cdots + u_{n-1}p_{n-1} + u_np_n$$

A similar statement holds for $\tilde{L}_2$ and thus we can conclude that:

$$p_1 + u_2p_2 + \cdots + u_{n-1}p_{n-1} \geq q_1 + u_2q_2 + \cdots + u_{n-1}q_{n-1} \tag{3.11}$$

We can now apply Assumption 5 (which is an if and only if statement) to see that:

$$\tilde{L}_1 \succeq \tilde{L}_2$$

We can now conclude from Assumption 1 that since $L_1 \sim \tilde{L}_1$ and $L_2 \sim \tilde{L}_2$ and $\tilde{L}_1 \succeq \tilde{L}_2$ that $L_1 \succeq L_2$. This completes the proof. □

REMARK 3.20. This theorem is called the Expected Utility Theorem because the utility for any lottery is really the expected utility from any of the prizes. That is, let U be the random variable that takes value u_i if prize A_i is received. Then:

$$\mathbb{E}(U) = \sum_{i=1}^{n} u_ip(A_i) = u_1p_1 + u_2p_2 + \cdots + u_np_n \tag{3.12}$$

This is just the utility of the lottery in which prize i is received with probability p_i.

EXAMPLE 3.21. Congratulations! You're on Let's Make a Deal. The following prizes are up for grabs:

(1) A_1: A new car (worth \$15,000)
(2) A_2: A gift card (worth \$1,000) to Best Buy
(3) A_3: A new iPad (worth \$800)
(4) A_4: A Donkey (technically worth \$500, but somewhat challenging)

We'll assume that you prefer these prizes in the order in which they appear. Wayne Brady offers you the following deal you can compete in either of the following games (lotteries):

(1) $L_1 = \langle(A_1, 0.25), (A_2, 0.25), (A_3, 0.25), (A_4, 0.25)\rangle$
(2) $L_2 = \langle(A_1, 0.15), (A_2, 0.4), (A_3, 0.4), (A_4, 0.05)\rangle$

Which games should you choose to make you the most happy? The problem here is actually valuing the prizes. Maybe you really really need a new car (or you just bought a new car). The car may be worth more than it's dollar value. Alternatively, suppose you actually want a donkey? Suppose you know that donkeys are expensive to own and the "retail" \$450 value is false. Maybe there's not a Best Buy near you and it would be hard to use the gift card.

For the sake of argument, let's suppose that you determine that the donkey is worth nothing to you. You might say that:

(1) $A_2 \sim \langle(A_1, 0.1), (A_4, 0.9)\rangle$
(2) $A_3 \sim \langle(A_1, 0.05), (A_4, 0.95)\rangle$

The numbers really don't make any difference, you can supply any values you want for 0.1 and 0.05 as long as the other numbers enforce Assumption 3. Then we can write:

(1) $L_1 \sim \langle(A_1, 0.25), (\langle(A_1, 0.1), (A_4, 0.9)\rangle, 0.25), (\langle(A_1, 0.05), (A_4, 0.95)\rangle, 0.25), (A_4, 0.25)\rangle$
(2) $L_2 \sim \langle(A_1, 0.15), (\langle(A_1, 0.1), (A_4, 0.9)\rangle, 0.4), (\langle(A_1, 0.05), (A_4, 0.95)\rangle, 0.4), (A_4, 0.05)\rangle$

We can now simplify this by expanding these compound lotteries into simple lotteries in terms of A_1 and A_4:

To see how we do this, let's consider just Lottery 1: Lottery 1 is a compound lottery that contains the following sub-lotteries:

(1) S_1: A_1 with probability 0.25
(2) S_2: $\langle (A_1, 0.1), (A_4, 0.9)\rangle$ with probability 0.25
(3) S_3: $\langle (A_1, 0.05), (A_4, 0.95)\rangle$ with probability 0.25
(4) S_4: A_4 with probability 0.25

To convert this lottery into a simpler lottery, we apply Assumption 2. The probability of winning prize A_1 is just the probability of winning prize A_1 in one of the lotteries that make up the compound lottery multiplied by the probability of playing in that lottery. Or:

$$P(A_1) = P(A_1|S_1)P(S_1) + P(A_1|S_2)P(S_2) + P(A_1|S_3)P(S_3) + P(A_1|S_4)P(S_4)$$

This can be computed as:

$$P(A_1) = (1)(0.25) + (0.1)(0.25) + (0.05)(0.25) + (0)(0.25) = 0.2875$$

Similarly:

$$P(A_4) = (0)(0.25) + (0.9)(0.25) + (0.95)(0.25) + (1)(0.25) = 0.71250$$

Thus $L_1 \sim \langle (A_1, 0.2875), (A_4, 0.71250)\rangle$. We can perform a similar calculation for L_2 to obtain: $L_2 \sim \langle (A_1, 0.21), (A_4, 0.79)\rangle$

Thus, even though there is less of a chance of winning the donkey in Lottery (Game) 2, you should prefer Lottery (Game) 1. Thus, you tell Wayne that you'd like to play that game instead. Given the information provided, we know $u_2 = 0.1$ and $u_3 = 0.05$. Thus, we can compute the utility of the two games as:

$$u(L_1) = (0.25)(1) + (0.25)(0.1) + (0.25)(0.05) + (0.25)(0) = 0.2875 \tag{3.13}$$

$$u(L_2) = (0.15)(1) + (0.4)(0.1) + (0.4)(0.05) + (0.05)(0) = 0.21 \tag{3.14}$$

Exercise 14. Make up an example of a game with four prizes and perform the same calculation that we did in Example 3.21. Explain what happens to your computation if you replace the "donkey prize" with something more severe like being imprisoned for 10 years. Does a penalty that is difficult to compare to prizes make it difficult to believe that the u_i values actually exist in all cases?

Definition 3.22 (Linear Utility Function). We say that a utility function $u : \mathcal{L} \to \mathbb{R}$ is *linear* if given any lotteries $L_1, L_2 \in \mathcal{L}$ and some $q \in [0, 1]$, then:

$$u\left(\langle (L_1, q), (L_2, (1-q))\rangle\right) = qu(L_1) + (1-q)u(L_2) \tag{3.15}$$

Here $\langle (L_1, q), (L_2, (1-q))\rangle$ is the compound lottery made up of lotteries L_1 and L_2 each having probabilities q and $(1-q)$ respectively.

Lemma 3.23. *Let $\mathcal{L}$ be the collection of lotteries defined over prizes $A_1, \dots, A_n$ with $A_1 \succeq A_2 \succeq \cdots \succeq A_n$. Let $u : \mathcal{L} \to [0, 1]$ be the utility function defined in Theorem 3.19. Then $L_1 \sim L_2$ if and only if $u(L_1) = u(L_2)$.*

EXERCISE 15. Prove Lemma 3.23. [Hint: We know $L_1 \succeq L_2$ and $L_2 \succeq L_1$ if and only if $L_1 \sim L_2$. We also know $L_1 \succeq L_2$ if and only if $u(L_1) \geq u(L_2)$. What, then do we know is true about $u(L_1)$ and $u(L_2)$ when $L_2 \succeq L_1$? Use this, along with the rules of ordering in the real numbers to prove the lemma.]

THEOREM 3.24. *The utility function $u : \mathcal{L} \to [0, 1]$ in Theorem 3.19 is linear.*

PROOF. Let:

$$L_1 = \langle (A_1, p_1), (A_2, p_2), \dots, (A_n, p_n) \rangle$$
$$L_2 = \langle (A_1, r_1), (A_2, r_2), \dots, (A_n, r_n) \rangle$$

Thus we know that:

$$u(L_1) = \sum_{i=1}^{n} p_i u_i$$
$$u(L_2) = \sum_{i=1}^{n} r_i u_i$$

Choose $q \in [0, 1]$. The lottery $L = \langle (L_1, q), (L_2, (1 - q)) \rangle$ is equivalent to a lottery in which prize A_i is obtained with probability:

$$\Pr(A_i) = qp_i + (1 - q)r_i$$

Thus, applying Assumption 2 we have:

$$\tilde{L} = \langle (A_1, [qp_1 + (1 - q)r_1]), \dots, (A_n, [qp_1 + (1 - q)r_1]) \rangle \sim L$$

Applying Lemma 3.23, we can compute:

$$u(L) = u(\tilde{L}) = \sum_{i=1}^{n} [qp_i + (1 - q)r_i] u_i = \sum_{i=1}^{n} qp_i u_i + \sum_{i=1}^{n} (1 - q) r_i u_i = q\left(\sum_{i=1}^{n} p_i u_i\right) + (1 - q)\left(\sum_{i=1}^{n} r_i u_i\right) = qu(L_1) = (1 - q)u(L_2) \tag{3.16}$$

Thus u is linear. This completes the proof. □

THEOREM 3.25. *Suppose that $a, b \in \mathbb{R}$ with $a > 0$. Then the function: $u' : \mathcal{L} \to \mathbb{R}$ given by:*

$$u'(L) = au(L) + b \tag{3.17}$$

also has the property that $u'(L_1) \geq u'(L_2)$ if and only if $L_1 \succeq L_2$, where u is the utility function given in Theorem 3.19. Furthermore, this utility function is linear.

REMARK 3.26. A generalization of Theorem 3.25 simply shows that the class of linear utility functions is closed under a subset of affine transforms. That means that given one linear utility function we can construct another by multiplying by a positive constant and adding another constant.

EXERCISE 16. Prove Theorem 3.25. [Hint: Verify the claim using the fact that it holds for u.]

2. Advanced Decision Making under Uncertainty

If you study Game Theory in an Economics context and use Myerson's classic book [**Mye01**], you will see a more complex (and messier) treatment of the Expected Utililty Theorem. We will not prove the more general theorem given in Myerson's book, but we will discuss the conditions under which the theorem is constructed.

Let Ω be a set of outcomes. We will assume that the set Ω gives us information about the real world as it is. Let $X = \{A_1, \dots, A_n\}$ be the set of prizes.

Definition 3.27. Define $\Delta(X)$ as the set of all possible probability functions over the set X. Formally, if $\mathcal{P} \in \Delta(X)$, then $\mathcal{P} = (X, \mathcal{F}_X, P_X)$ is a probability space over X with probability function P_X and we can associate the element $\mathcal{P}$ with P_X.

In this more complex case, the lotteries are composed not just of probability distributions over prizes (i.e., elements of $\Delta(X)$) but these probabilities are *conditioned* on the state of the world $\omega \in \Omega$.

Definition 3.28 (Lottery). A lottery is a mapping $f : \Omega \to \Delta(X)$. The set of all such lotteries is still named $\mathcal{L}$.

Remark 3.29. In this situation, we assume that the lotteries can change depending upon the state of the world, which is provided by an event in $S \subseteq \Omega$.

Example 3.30. In this world, suppose that the set of outcomes is the days of the week. A game show might go something like this: on Tuesday and Thursday a contestant has a 50% chance of winning a car and a 50% chance of winning a donkey. On Monday, Wednesday and Friday, there is a 20% chance of winning \$100 and an 80% chance of winning \$2,000. On Saturday and Sunday there is a 100% chance of winning nothing (because the game show does not tape on the weekend).

Under these conditions, the Assumptions 1 through 5 must be modified to deal with the state of the world. This is done by making the preference relation $\succeq$ *dependent* on any given subset $S \subseteq \Omega$. Thus we end up with a collection of orderings $\succeq_S$ for any given subset $S \subseteq \Omega$ ($S \neq \emptyset$).

The transformation of our assumptions into assumptions for the more complex case is beyond the scope of our course. If you are interested, see Myerson's Book (Chapter 1) for a complete discussion. For those students interesting in studying graduate economics, this is a worthwhile activity. The proof of the Generalized Expected Utility Theorem is substantially more complex than our proof here. It is worth the effort to work through if you are properly motivated.

CHAPTER 4

Game Trees, Extensive Form, Normal Form and Strategic Form

The purpose of this chapter is to create a formal and visual representation for a certain class of games. This representation will be called *extensive form*, which we will define formally as we proceed. We will proceed with our study of games under the following assumptions:

(1) There are a finite set of Players: $\mathbf{P} = \{P_1, \dots, P_N\}$
(2) Each player has a knowledge of the *rules of the game* (the rules under which the game state evolves) and the rules are fixed.
(3) At any time $t \in \mathbb{R}_+$ during game play, the player has a finite set of *moves* or choices to make. These choices will affect the evolution of the game. The set of all available moves will be denoted $\mathcal{S}$.
(4) The game ends after some finite number of moves.
(5) At the end of the game, each player receives a prize. (Using the results in the previous section, we assume that these prizes can be ordered according to preference and that a utility function exists to assign numerical values to these prizes.)

In addition to these assumptions, some games may incorporate two other components:

(1) At certain points, there may be chance moves which advance the game in a non-deterministic way. This only occurs in games of chance. (This occurs, e.g., in poker when the cards are dealt.)
(2) In some games the players will know the *entire* history of moves that have been made at all times. (This occurs, e.g., in Tic-Tac-Toe and Chess, but not e.g., in Poker.)

1. Graphs and Trees

In order to formalize game play, we must first understand the notion of graphs and trees, which are used to model the sequence of moves in any game.

Definition 4.1 (Graph). A *digraph* (directed graph) is a pair $G = (V, E)$ where V is a finite set of vertexes and $E \subseteq V \times V$ is a finite set of directed edges composed of ordered two element subsets of V. By convention, we assume that $(v, v) \notin E$ for all $v \in V$.

Example 4.2. There are $2^6 = 64$ possible digraphs on 3 vertices. This can be computed by considering the number of permutations of 2 elements chosen from a 3 element set. This yields 6 possible ordered pairs of vertices (directed edges). For each of these edges, there are 2 possibilities: either the edge is in the edge set or not. Thus, the total number of digraphs on three edges is $2^6 = 64$.

Exercise 17. Compute the number of directed graphs on four vertices. [Hint: How many different pairs of vertices are there?]

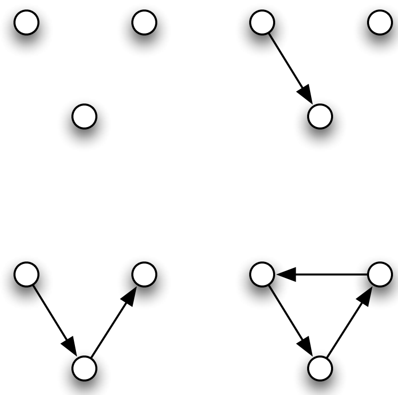


Figure 4.1. Digraphs on 3 Vertices: There are $64 = 2^6$ distinct graphs on three vertices. The increased number of edges graphs is caused by the fact that the edges are now directed.

DEFINITION 4.3 (Path). Let $G = (V, E)$ be a digraph. Then a *path* in G is a sequence of vertices $\langle v_0, v_1, \ldots, v_n \rangle$ so that $(v_i, v_{i+1}) \in E$ for each $i = 0, \ldots, n-1$. We say that the path goes from vertex v_0 to vertex v_n. The number of vertices in a path is called its *length*.

EXAMPLE 4.4. We illustrate both a path and a cycle in Figure 4.2. There are not many paths in a graph with only three vertices.

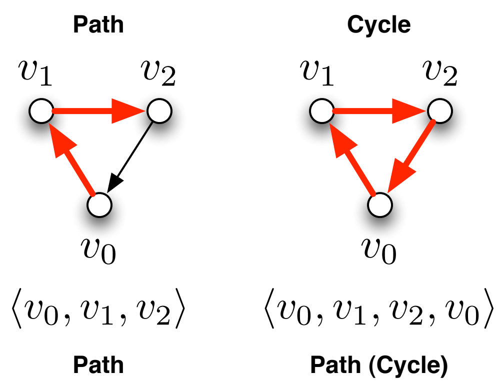


Figure 4.2. Two Paths: We illustrate two paths in a digraph on three vertices.

DEFINITION 4.5 (Directed Tree). A digraph $G = (V, E)$ that posses a unique vertex $r \in V$ called the *root* so that (i) there is a unique path from r to every vertex $v \in V$ and (ii) there is no $v \in V$ so that $(v, r) \in E$ is called a *directed tree*.

EXAMPLE 4.6. Figure 4.3 illustrates a simple directed tree. Note that there is a (directed) path connecting the root to every other vertex in the tree.

DEFINITION 4.7 (Descendants). If $T = (V, E)$ is a directed tree and $v, u \in V$ with $(v, u) \in E$, then u is called a *child* of v and v is called the *parent* of u. If there is a path from v to u in the T, then u is called a *descendent* of v and v is called an *ancestor* of u.

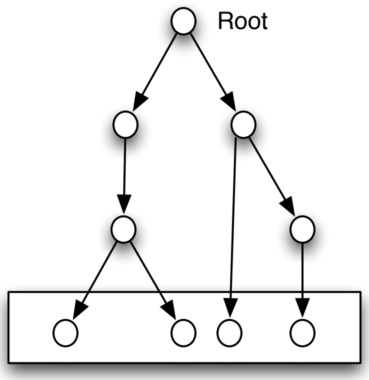


Figure 4.3. Directed Tree: We illustrate a directed tree. Every directed tree has a unique vertex called the *root*. The root is connected by a directed path to every other vertex in the directed tree.

Definition 4.8 (Out-Edges). If $T = (V, E)$ is a directed tree and $v \in V$, then we will denote the *out-edges* of vertex v by $E_o(v)$. These are edges that connect v to its children. Thus,

$$E_o(v) = \{(v, u) \in V : (v, u) \in E\}$$

Definition 4.9 (Terminating Vertex). If $T = (V, E)$ is a directed tree and $v \in V$ so that v has no descendants, then v is called a *terminal* vertex. All vertices that are not terminal are *non-terminal* or *intermediate*.

Definition 4.10 (Tree Height). Let $T = (V, E)$ be a tree. The *height* of the tree is the length of the longest path in T.

Example 4.11. The height of the tree shown in Figure 4.3 is 4. There are three paths of length 4 in the tree that start at the root of the tree and lead to three of the four terminal vertices.

Lemma 4.12. *Let $T = (V, E)$ be a directed tree. If v is a vertex of v and u is a descendent of v, then there is no path from u to v.*

Proof. Let r be the root of the tree. Clearly if $v = r$, then the theorem is proved. Suppose not. Let $\langle w_0, w_1, \dots, w_n\rangle$ be a path from u to v with $w_0 = u$ and $w_n = v$. Let $\langle x_0, x_1, \dots, x_m\rangle$ be the path from the root of the tree to the node v (thus $x_0 = r$ and $x_m = v$). Let $\langle y_0, y_1, \dots, y_k\rangle$ be the path leading from the r to u (thus $y_0 = r$ and $y_k = u$. Then we can construct a new path:

$$\langle r = y_0, y_1, \dots, y_k = u = w_0, w_1, \dots, w_n = v\rangle$$

from r (the root) to the vertex v. Thus there are two paths leading from the root to vertex v, contradicting our assertion that T was a tree. $\square$

Theorem 4.13. *Let $T = (V, E)$ be a tree. Suppose $u \in V$ is a vertex and let:*

$$V(u) = \{v \in V : v = u \text{ or } v \text{ is a descendent of } u\}$$

Let $E(u)$ be the set of all edges defined in paths connecting u to a vertex in $V(u)$. Then the graph $T_u = (V(u), E(u))$ is a tree with root u and is called the sub-tree of T descended from u.

Example 4.14. A sub-tree of the tree shown in Example 4.6 is shown in Figure 4.4. Sub-trees can be useful in analyzing decisions in games.

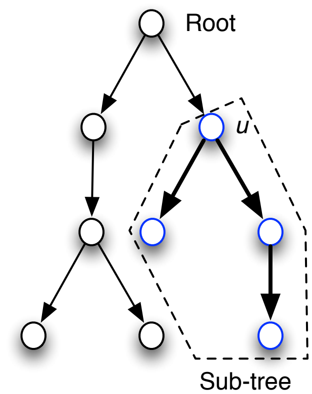


Figure 4.4. Sub Tree: We illustrate a sub-tree. This tree is the collection of all nodes that are descended from a vertex u.

Proof. If u is the root of T, then the statement is clear. There is a unique path from u (the root) to every vertex in T, by definition. Thus, T_u is the whole tree.

Suppose that u is not the root of T. The set $V(u)$ consists of all descendants of u and u itself. Thus between u and each $v \in V(u)$ there is a path $p = \langle v_0, v_1, \dots, v_n \rangle$ where $v_0 = u$ and $v_n = v$. To see this path must be unique, suppose that it is not, then there is at least one other distinct path $\langle w_0, w_1, \dots, w_m \rangle$ with $w_0 = u$ and $w_m = v$. But if that's so, we know there is a unique path $\langle x_0, \dots, x_k \rangle$ with x_0 being the root of T and $x_k = u$. It follows that there are two paths:

$$\langle x_0, \dots, x_k = v_0 = u, v_1, \dots, v_n = v \rangle$$
$$\langle x_0, \dots, x_k = w_0 = u, w_1, \dots, w_m = v \rangle$$

between the root x_0 and the vertex v. This is a contradiction of our assumption that T was a directed tree.

To see that there is no path leading from any element in $V(u)$ back to u, we apply Lemma 4.12. Since, by definition, every edge in the paths connecting u with its descendants are in $E(u)$ it follows that T_u is a directed tree and u is the root since there is a unique path from u to each element of $V(u)$ and there is no path leading from any element of $V(u)$ back to u. This completes the proof. □

2. Game Trees with Complete Information and No Chance

In this section, we define what we mean by a Game Tree with *perfect information* and *no chance moves*. Essentially, we will begin with some directed tree T. Each non-terminal vertex of T will be controlled by a player who will make a *move* at the vertices she owns. If v is a vertex controlled by Player P, then out-edges from v will correspond to the possible moves Player P can take. The terminal vertices will represent end-game conditions (e.g., check-mate in chess). Each terminal vertex will be assigned a payoff (score or prize) amount for each player of the game. In this case, there will be *no chance moves* (all moves will be deliberately made by players) and all players will know precisely who is moving and what their move is.

Definition 4.15 (Player Vertex Assignment). If $T = (V, E)$ is a directed tree, let $F \subseteq V$ be the terminal vertices and let $D = V \setminus F$ be the intermediate (or decision) vertices. A *assignment of players to vertices* is an onto function $\nu : D = V \setminus F \to \mathbf{P}$ that assigns to each non-terminal vertex $v \in V \setminus F$ a player $\nu(v) \in \mathbf{P}$. Then Player $\nu(v)$ is said to *own* or *control* vertex v.

Definition 4.16 (Move Assignment). If $T = (V, E)$ is a directed tree, then a move assignment function is a mapping $\mu : E \to \mathcal{S}$ where $\mathcal{S}$ is a finite set of player moves. So that if $v, u_1, u_2 \in V$ and $(v, u_1) \in E$ and $(v, u_2) \in E$, then $\mu(v, u_1) = \mu(v, u_2)$ if and only if $u_1 = u_2$.

Definition 4.17 (Payoff Function). If $T = (V, E)$ is a directed tree, let $F \subseteq V$ be the terminal vertices. A payoff function is a mapping $\pi : F \to \mathbb{R}^N$ that assigns to each terminal vertex of T a numerical payoff for each player in $\mathbf{P}$.

Remark 4.18. It is possible, of course, that the payoffs from a game may not be real valued, but instead tangible assets, prizes or penalties. We will assume that the assumptions of the expected utility theorem are in force and therefore there a linear utility function can be defined that provides the real values required for the definition of the payoff function π.

Definition 4.19 (Game Tree with Complete Information and No Chance Moves). A *game tree with complete information and no chance* is a quadruple $\mathcal{G} = (T, \mathbf{P}, \mathcal{S}, \nu, \mu, \pi)$ such that T is a directed tree, ν is a player vertex assignment on intermediate vertices of T, μ is a move assignment on the edges of T and π is a payoff function on T.

Example 4.20 (Rock-Paper-Scissors). Consider an odd version of rock-paper-scissors played between two people in which the first player plays first and then the second player plays. If we assume that the winner receives $+1$ points and the loser receives -1 points (and in ties both players win 0 points), then the game tree for this scenario is visualized in Figure 4.5: You may think this game is not entirely *fair*, which is not mathematically defined, because it looks like Player 2 has an advantage in knowing Player 1's move before making his own move. Irrespective of feelings, this is a valid game tree.

Definition 4.21 (Strategy–Perfect Information). Let $\mathcal{G} = (T, \mathbf{P}, \mathcal{S}, \nu, \mu, \pi)$ be a game tree with complete information and no chance, with $T = (V, E)$. A *pure strategy* for Player P_i (in a perfect information game) is a mapping $\sigma_i : V_i \to \mathcal{S}$ with the property that if $v \in V_i$ and $\sigma_i(v) = s$, then there is some $y \in V$ so that $(x, y) \in E$ and $\mu(x, y) = s$. (Thus σ_i will only choose a move that labels an edge leaving v.)

Remark 4.22 (Rationality). A strategy tells a player how to play in a specific game at any moment in time. We assume that players are *rational* and that at any time they know the *entire* game tree and that Player i will attempt to maximize her payoff at the end of the game by choosing a strategy function σ_i appropriately.

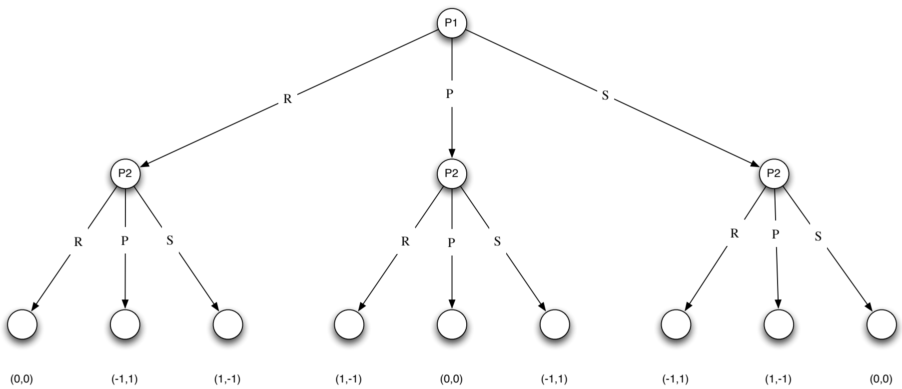


Figure 4.5. Rock-Paper-Scissors with Perfect Information: Player 1 moves first and holds up a symbol for either rock, paper or scissors. This is illustrated by the three edges leaving the root node, which is assigned to Player 1. Player 2 then holds up a symbol for either rock, paper or scissors. Payoffs are assigned to Player 1 and 2 at terminal nodes. The index of the payoff vector corresponds to the players.

EXAMPLE 4.23 (The Battle of the Bismark Sea). Games can be used to illustrate the importance of intelligence in combat. In February 1943, the battle for New Guinea had reached a critical juncture in World War 2. The Allies controlled the southern half of New Guinea and the Japanese the northern half. Reports indicated that the Japanese were

Figure 4.6. New Guinea is located in the south pacific and was a major region of contention during World War II. The northern half was controlled by Japan through 1943, while the southern half was controlled by the Allies. (Image created from Wikipedia (http://en.wikipedia.org/wiki/File:LocationNewGuinea.svg), originally sourced from http://commons.wikimedia.org/wiki/File:LocationPapuaNewGuinea.svg.

massing troops to reinforce their army on New Guinea in an attempt to control the entire island. These troops had to be delivered by naval convoy. The Japanese had a choice of sailing either north of New Britain, where rain and poor visibility was expected or south of

New Britain, where the weather was expected to be good. Either route required the same amount of sailing time.

General Kenney, the Allied Forces Commander in the Southwest Pacific had been ordered to do as much damage to the Japanese convoy fleet as possible. He had reconnaissance aircraft to detect the Japanese fleet, but had to determine whether to concentrate his search planes on the northern or southern route.

The following game tree summarizes the choice for the Japanese (J) and American (A) commanders (players), with payoffs given as the *number of days available for bombing of the Japanese fleet*. (Since the Japanese cannot benefit, there payoff is reported as the negative of these values.) The moves for each player are *sail north* or *sail south* for the Japanese and *search north* or *search south* for the Americans.

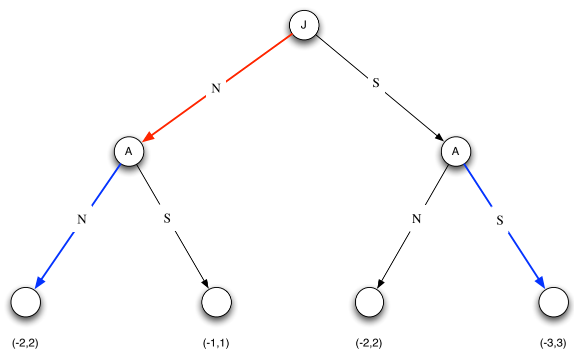


Figure 4.7. The game tree for the Battle of the Bismark Sea. The Japanese could choose to sail either north or south of New Britain. The Americans (Allies) could choose to concentrate their search efforts on either the northern or southern routes. Given this game tree, the Americans would always choose to search the North if they *knew* the Japanese had chosen to sail on the north side of New Britain; alternatively, they would search the south route, if they knew the Japanese had taken that. Assuming the Americans have perfect intelligence, the Japanese would always choose to sail the northern route as in this instance they would expose themselves to only 2 days of bombing as opposed to 3 with the southern route.

This example illustrates the *importance of intelligence* in warfare. In this game tree, we assume perfect information. Thus, the Americans *know* (through backchannels) which route the Japanese will sail. In knowing this, they can make an optimal choice for each contingency. If the Japanese sail north, then the Americans search north and will be able to bomb the Japanese fleet for 2 days. Similarly, if the Japanese sail south, the Americans will search south and be able to bomb the Japanese fleet for 3 days.

The Japanese, however, also have access to this game tree and reasoning that the Americans are payoff maximizers, will chose a path to minimize their exposure to attack. They *must* choose to go north and accept 2 days of bombing. If they choose to go south, then they know they will be exposed to 3 days of bombing. Thus, their optimal strategy is to sail north.

Naturally, the Allies did not know which route the Japanese would take and there was no backchannel intelligence. We will come back to this case later. However, this example serves to show how important intelligence is in warfare since it can help commanders make optimal decisions.

Exercise 18. Using the approach from Example 4.23 derive a strategy for Player 2 in the Rock-Paper-Scissors game (Example 4.20) assuming she will attempt to maximize her payoff. Similarly, show that it doesn't matter whether Player 1 chooses Rock, Paper or Scissors in this game and thus any strategy for Player 1 is equally good (or bad).

Remark 4.24. The complexity of a game (especially one with perfect information and no chance moves) can often be measured by how many nodes are in its game tree. A computer that wishes to play a game often attempts to explore the game tree in order to make its moves. Certain games, like Chess and Go, have *huge* game trees. Another measure of complexity is the length of the longest path in the game tree.

In our odd version Rock-Paper-Scissors, the length of the longest path in the game tree is 3 nodes. This reflects the fact that there are only two moves in the game: first Player 1 moves and then Player 2 moves.

Exercise 19. Consider a simplified game of tic-tac-toe where the objective is to fill in a board shown in Figure 4.8

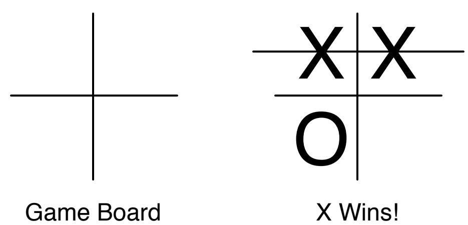


Figure 4.8. Simple tic-tac-toe: Players in this case try to get two in a row.

Assuming that X goes first. Construct the game tree for this game by assuming that the winner receives $+1$ while the loser receives -1 and draws result in 0 for both players. Compute the depth of the longest path in the game tree. Show that there is a strategy so that the first player always wins. [Hint: You will need to consider each position in the board as one of the moves that can be made.]

Exercise 20. In a standard 3×3 tic-tac-toe board, compute the length of the longest path in the game tree. [Hint: Assume you draw in this game.]

3. Game Trees with Incomplete Information

Remark 4.25 (Power Set and Partitions). Recall from Remark 2.11 that, if X is a set, then 2^X is the power set of X or the set of all subsets of X. Any parition of X is a set $\mathcal{I} \subseteq 2^X$ so that: **For all $x \in X$ there is exactly one element $I \in \mathcal{I}$ so that $x \in I$.** (Remember, I is a subset of X and as such, $I \in \mathcal{I} \subseteq 2^X$.

Definition 4.26 (Information Sets). If $T = (V, E)$ is a tree and $D \subset V$ are the intermediate (decision) nodes of the tree, ν is a player assignment function and μ is a move assignment, then *information sets* are a set $\mathcal{I} \subset 2^D$, satisfying the following:

(1) For all $v \in D$ there is exactly one set $I_v \in \mathcal{I}$ so that $v \in I_v$. This is the information set of the vertex v.

(2) If $v_1, v_2 \in I_v$, then $\nu(v_1) = \nu(v_2)$.
(3) If $(v_1, v) \in E$ and $\mu(v_1, v) = m$, and $v_2 \in I_{v_1}$ (that is, v_1 and v_2 are in the same information set), then there is some $w \in V$ so that $(v_2, w) \in E$ and $\mu(v_2, w) = m$

Thus $\mathcal{I}$ is a partition of D.

REMARK 4.27. Definition 4.26 says that every vertex in a game tree is assigned a single information set. It also says that if two vertices are in the same information set, then they must both be controlled by the same player. Finally, the definition says that two vertices can be in the same information set only if the moves from these vertices are indistinguishable.

An information set is used to capture the notion that a player doesn't know what vertex of the game tree he is at; i.e., that he cannot distinguish between two nodes in the game tree. All that is known is that the same moves are available at all vertices in a given information set.

In a case like this, it is possible that the player doesn't know which vertex in the game tree will come next as a result of choosing a move, but he can certainly limit the possible vertices.

REMARK 4.28. We can also think of the information set as being a mapping $\xi : V \to \mathcal{I}$ where $\mathcal{I}$ is a finite set of information labels and the labels satisfy requirements like those in Definition 4.26. This is the approach that Myerson [**Mye01**] takes.

EXERCISE 21. Consider the information sets a set of labels $\mathcal{I}$ and let $\xi : V \to \mathcal{I}$. Write down the constraints that ξ must satisfy so that this definition of information set is analogous to Definition 4.26.

DEFINITION 4.29 (Game Tree with Incomplete Information and No Chance Moves). A *game tree with incomplete information and no chance* is a tuple $\mathcal{G} = (T, \mathbf{P}, \mathcal{S}, \nu, \mu, \pi, \mathcal{I})$ such that T is a directed tree, ν is a player vertex assignment on intermediate vertices of T, μ is a move assignment on the edges of T and π is a payoff function on T and $\mathcal{I}$ are information sets.

DEFINITION 4.30 (Strategy–Imperfect Information). Let $\mathcal{G} = (T, \mathbf{P}, \mathcal{S}, \nu, \mu, \pi, \mathcal{I})$ be a game tree with incomplete information and no chance moves, with $T = (V, E)$. Let $\mathcal{I}_i$ be the information sets controlled by Player i. A *pure strategy* for Player P_i is a mapping $\sigma_i : \mathcal{I}_i \to \mathcal{S}$ with the property that if $I \in \mathcal{I}_i$ and $\sigma_i(I) = s$, then for every $v \in I$ there is some edge $(v, w) \in E$ so that $\mu(v, w) = s$.

PROPOSITION 4.31. *If* $\mathcal{G} = (T, \mathbf{P}, \mathcal{S}, \nu, \mu, \pi, \mathcal{I})$ *and* $\mathcal{I}$ *consists of only singleton sets, then* $\mathcal{G}$ *is equivalent to a game with complete information.*

PROOF. The information sets are used only in defining strategies. Since each $I \in \mathcal{I}$ is a singleton, we know that for each $I \in \mathcal{I}$ we have $I = \{v\}$ where $v \in D$. (Here D is the set of decision nodes in V with $T = (V, E)$.) Thus any strategy $\sigma_i : \mathcal{I}_i \to E$ can easily be converted into $\sigma_i : V_i \to E$ by stating that $\sigma_i(v) = \sigma_i(\{v\})$ for all $v \in V_i$. This completes the proof. $\square$

EXAMPLE 4.32 (The Battle of the Bismark Sea (Part 2)). Obviously, General Kenney did *not* know a priori which route the Japanese would take. This can be modeled using information sets. In this game, the two nodes that are owned by the Allies in the game tree

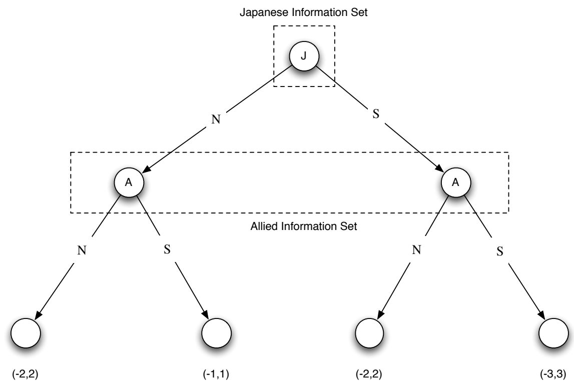


Figure 4.9. The game tree for the Battle of the Bismark Sea with incomplete information. Obviously Kenney could not have known *a priori* which path the Japanese would choose to sail. He could have reasoned (as they might) that there best plan was to sail north, but he wouldn't really *know*. We can capture this fact by showing that when Kenney chooses his move, he cannot distinguish between the two intermediate nodes that belong to the Allies.

are in the same information set. General Kenney doesn't know whether the Japanese will sail north or south. He could (in theory) have reasoned that they should sail north, but he doesn't know. The information set for the Japanese is likewise shown in the diagram.

In determining a strategy, the Allies and Japanese must think a little differently. The Japanese could choose to go south. If the Allies are lucky and choose to search south, the Japanese will be in for three days worth of attacks. If the allies are unlucky and choose to go north, the Japanese will still face two days of bombing. On the other hand, if the Japanese choose to go north, then they may be unlucky and the Allies will choose to search north in which case they will again take 2 days of bombing. If however, the allies are unlucky, the japanese will face only 1 day of bombing.

From the perspective of the Japanese, since the routes will take the same amount of time, the northern route is more favorable. To see this note Table 1: If the Japanese sail

	Sail North		**Sail South**
Search North	Bombed for 2 days	$\leq$	Bombed for 2 Days
Search South	Bombed for 1 days	$\leq$	Bombed for 3 Days

Table 1. Various Strategies and Payoffs for the Battle of the Bismark Sea. The northern route is favored by the Japanese who will always do no worse in taking it then they do the southern route.

north, then the worst they will suffer is 2 days of bombing and the best they will suffer is one day of bombing. If the Japanese sail south, the worse they will suffer is 3 days of bombing and the best they will suffer is 2 days of bombing. Thus, the northern route should be preferable as the cost to taking it is never worse than taking the southern route. We say that the northern route strategy *dominates* the southern route strategy. If General Kenney could reason this, then he might choose to commit his reconnaissance forces to searching the north, even without being able to determine whether the Japanese sailed north or south.

Exercise 22. Identify the information sets for Rock-Paper-Scissors and draw the game tree to illustrate the incomplete information. Do *not* worry about trying to identify an optimal strategy for either player.

4. Games of Chance

In games of chance, there is always a point in the game where a chance move is made. In card games, the initial deal is *one* of these points. To accommodate chance moves, we assume the existence of a *Player 0* who is sometimes called *Nature*. When dealing with games of chance, we assume that the player vertex assignment function assigns some vertices the label P_0.

Definition 4.33 (Moves of Player 0). Let $T = (V, E)$ and let ν be a player vertex assignment function. For all $v \in D$ such that $\nu(v) = P_0$ here is a probability assignment function $p_v : E_o(v) \to [0, 1]$ satisfying:

$$\sum_{e \in E_o(v)} p_v(e) = 1 \tag{4.1}$$

Remark 4.34. The probability function(s) p_v in Definition 4.33 essentially defines an roll of the dice. When game play reaches a vertex owned by P_0, Nature (or Player 0 or Chance) probabilistically advances the game by moving along an randomly chosen edge. The fact that Equation 4.1 holds simply asserts that the chance moves of Nature form a probability space at that point, whose outcomes are all the possible chance moves.

Definition 4.35 (Game Tree). Let $T = (V, E)$ be a directed tree, let $F \subseteq V$ be the terminal vertices and let $D = V \setminus F$ be the intermediate (or decision) vertices. Let $\mathbf{P} = \{P_0, P_1, \ldots, P_n\}$ be a set of players including P_0 the chance player. Let $\mathcal{S}$ be a set of moves for the players. Let $\nu : D \to \mathbf{P}$ be a player vertex assignment function and $\mu : E \to \mathcal{S}$ be a move assignment function. Let

$$\mathcal{P} = \{p_v : \nu(v) = P_0 \text{ and } p_v \text{ is the moves of Player 0}\}$$

Let $\pi : F \to \mathbb{R}^n$ be a payoff function. Let $\mathcal{I} \subseteq 2^D$ be the set of information sets.

A *game tree* is a tuple $\mathcal{G} = (T, \mathbf{P}, \mathcal{S}, \nu, \mu, \pi, \mathcal{I}, \mathcal{P})$. In this form, the game defined by the game tree $\mathcal{G}$ is said to be in *extensive* form.

Remark 4.36. A strategy for Player i in a game tree like the one in Definition 4.35 is the same as that in Definition 4.30

Example 4.37 (Red-Black Poker). This example is taken from Chapter 2 of [Mye01]. At the beginning of this game, each player antes up \$1 into a common pot. Player 1 takes a card from a randomized (shuffled) deck. After looking at the card, Player 1 will decide whether to raise or fold.

(1) If Player 1 folds, he shows the card to Player 2: If the card is red, then Player 1 wins the pot and Player 2 loses the pot. If the card is black, then Player 1 loses the pot and Player 2 wins the pot.
(2) If Player 1 raises, then Player 1 adds another dollar to the pot and Player 2 must decide whether to call or fold.

(a) If Player 2 folds, then the game ends and Player 1 takes the money irrespective of his card.
(b) If Player 2 calls, then he adds \$1 to the pot. Player 1 shows his card. If his card is red, then he wins the pot (\$2) and Player 2 loses the pot. If Player 1's card is black, then he loses the pot and Player 2 wins the pot (\$2).

The game tree for this game is shown in Figure 4.10 The root node of the game tree is

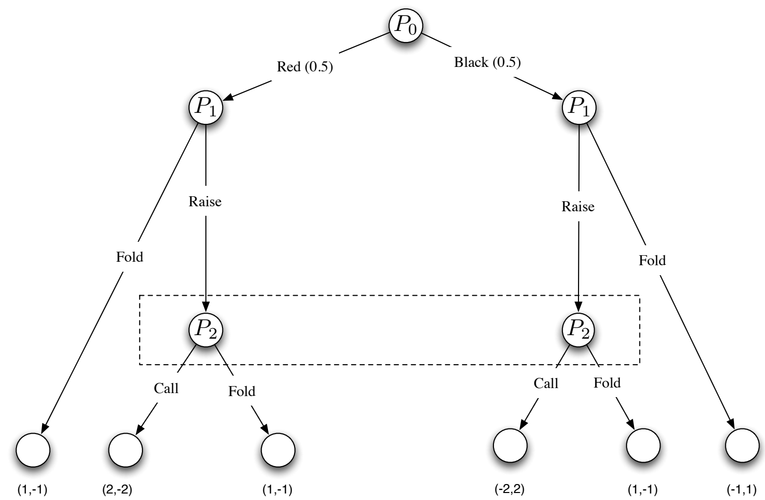


Figure 4.10. Poker: The root node of the game tree is controlled by Nature. At this node, a single random card is dealt to Player 1. Player 1 can then decide whether to end the game by folding (and thus receiving a payoff or not) or continuing the game by raising. At this point, Player 2 can then decide whether to call or fold, thus potentially receiving a payoff.

controlled by Nature (Player 0). This corresponds to the initial draw of Player 1, which is random and will result in a red card 50% of the time and a black card 50% of the time.

Notice that the nodes controlled by P_2 are in the same information set. This is because it is *impossible* for Player 2 to know whether or not Player 1 has a red card or a black card.

The payoffs shown on the terminal nodes are determined by how much each player will win or loose.

Exercise 23. Draw a game tree for the following game: At the beginning of this game, each player antes up \$1 into a common pot. Player 1 takes a card from a randomized (shuffled) deck. After looking at the card, Player 1 will decide whether to raise or fold.

(1) If Player 1 folds, he shows the card to Player 2: If the card is red, then Player 1 wins the pot and Player 2 loses the pot. If the card is black, then Player 1 loses the pot and Player 2 wins the pot.
(2) If Player 1 raises, then Player 1 adds another dollar to the pot and Player 2 picks a card and must decide whether to call or fold.
    (a) If Player 2 folds, then the game ends and Player 1 takes the money irrespective of any cards drawn.
    (b) If Player 2 calls, then he adds \$1 to the pot. Both players show their cards. If both cards of the same suit, then Player 1 wins the pot (\$2) and Player 2

loses the pot. If the cards are of opposite suits, then Player 2 wins the pot and Player 1 loses.

5. Pay-off Functions and Equilibria

THEOREM 4.38. *Let $\mathcal{G} = (T, \mathbf{P}, \mathcal{S}, \nu, \mu, \pi, \mathcal{I}, \mathcal{P})$ be a game tree and let $u \in D$, where D is the set of non-terminal vertices of T. Then the following is a game tree:*

$$\mathcal{G}' = (T_u, \mathbf{P}, \mathcal{S}, \nu|_{T_u}, \mu|_{T_u}, \pi|_{T_u}, \mathcal{I}|_{T_u}, \mathcal{P}|_{T_u})$$

where $\mathcal{I}|_{T_u} = \mathcal{I} \cap 2^{V(T_u)}$, with $V(T_u)$ being the vertex set of T_u, and $\mathcal{P}|_{T_u}$ is the set of probability assignment functions in $\mathcal{P}$ restricted only to the edges in T_u.

PROOF. By Theorem 4.13 we know that T_u is a sub-tree of T. Restricting the domains of the function ν, μ and π to the vertices and edges of this sub-tree does not invalidate these functions.

Let v be a descendant of u controlled by Chance. Since all descendants of u are included in T_u, it follows that all descendants of v are contained in T_u. Thus:

$$\sum_{e \in E_o(v)} p_v|_{T_u}(e) = 1$$

as required. Thus $\mathcal{P}|_{T_u}$ is an appropriate set of probability functions.

Finally, since $\mathcal{I}$ is a partition of T_u, we may compute $\mathcal{I}|_{T_u}$ by simply removing the vertices in the subsets of $\mathcal{I}$ that are not in T_u. This set $\mathcal{I}_{T_u}$ is a partition of T_u and necessarily satisfied the requirements set forth in Definition 4.26 because all the descendents of u are elements of $V(T_u)$. □

EXAMPLE 4.39. If we consider the game in Example 4.37, but suppose that Player 1 is known to have been dealt a red card, then the new game tree is derived by considering only the sub-tree in which Player 1 is dealt a red card. This is shown in Figure 4.11 It is worth

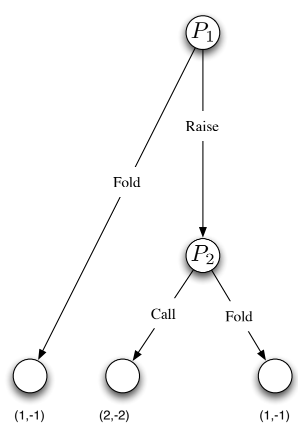


Figure 4.11. Reduced Red Black Poker: We are told that Player 1 receives a red card. The resulting game tree is substantially simpler. Because the information set on Player 2 controlled nodes indicated a lack of knowledge of Player 1's card, we can see that this sub-game is now a complete information game.

noting that when we restrict our attention to this sub-tree, a game that was originally an incomplete information game becomes a complete information game. That is, each vertex is now the sole member in its information set. Additionally, we have removed chance from the game.

Exercise 24. Continuing from Exercise 23 draw the game tree when we know that Player 1 is dealt a red card. Illustrate in your drawing how it is a sub-tree of the tree you drew in Exercise 23. Determine whether this game is still (i) a game of chance and (ii) whether it is a complete information game or not.

Theorem 4.40. *Let $\mathcal{G} = (T, \mathbf{P}, \mathcal{S}, \nu, \mu, \pi, \mathcal{I})$ be a game with no chance. Let $\sigma_1, \dots, \sigma_N$ be set of strategies for Players 1 through n. Then these strategies determine a unique path through the game tree.*

Proof. To see this, suppose we begin at the root node r. If this node is controlled by Player i, then node r exists in information set $I_r \in \mathcal{I}_i$. Then $\sigma_i(I_r) = s \in \mathcal{S}$ and there is some edge $(r, u) \in E$ so that $\mu(r, u) = s$. The next vertex determined by the strategy σ_i is u. In either case, we have a two vertex path (r, u).

Consider the game tree $\mathcal{G}'$ constructed from sub-tree T_u and determined as in Theorem 4.38. This game tree has root u. We can apply the same argument to construct a two vertex path (u, u'), which when joined with the initial path forms the three node path (r, u, u'). Repeating this argument inductively will yield a path through the game tree that is determined by the strategy functions of the players. Since the number of vertices in the tree is finite, this process must stop, producing the desired path. Uniqueness of the path is ensured by the fact that at the strategies are functions and thus at any information set, exactly one move will be chosen by the player in control. $\square$

Example 4.41. In the Battle of the Bismark Sea, the strategy we defined in Example 4.23 clearly defines a unique path through the tree: Since each player determines a priori

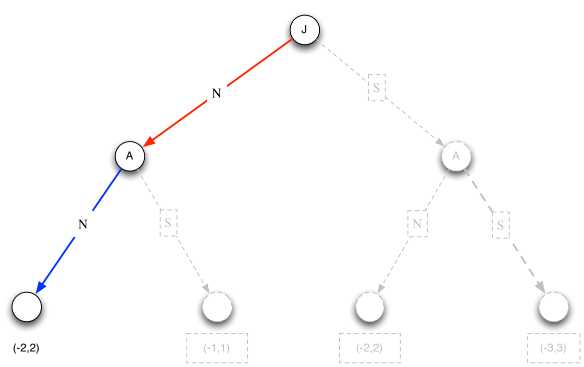


Figure 4.12. A unique path through the game tree of the Battle of the Bismark Sea. Since each player determines a priori the unique edge he/she will select when confronted with a specific information set, a path through the tree can be determined from these selections.

the unique edge he/she will select when confronted with a specific information set, a path through the tree can be determined from these selections. This is illustrated in Figure 4.12.

EXERCISE 25. Define a strategy for Rock-Paper-Scissors and show the unique path through the tree in Figure 4.5 determined by this strategy. Do the same for the game tree describing the Battle of the Bismark Sea with incomplete information.

THEOREM 4.42. *Let $\mathcal{G} = (T, \mathbf{P}, \mathcal{S}, \nu, \mu, \pi, \mathcal{I}, \mathcal{P})$. Let $\sigma_1, \ldots, \sigma_N$ be a collection of strategies for Players 1 through n. Then these strategies determine a discrete probability space $(\Omega, \mathcal{F}, P)$ where Ω is a set of paths leading from the root of the tree to a subset of the terminal nodes and if $\omega \in \Omega$, then $P(\omega)$ is the product of the probabilities of the chance moves defined by the path ω.*

PROOF. We will proceed inductively on the height of the tree T. Suppose the tree T has a height of 1. Then there is only one decision vertex (the root). If that decision vertex is controlled by a player other than chance, then applying Theorem 4.40 we know that the strategies $\sigma_1, \ldots, \sigma_N$ defined a unique path through the tree. The only paths in a tree of height 1 have the form $\langle r, u\rangle$ where r is the root of T and u is a terminal vertex. Thus, Ω is the singleton consisting of only the path $\langle r, u\rangle$ determined by the strategies and it is assigned a probability of 1.

If chance controls the root vertex, then we can define:

$$\Omega = \{\langle r, u\rangle : u \in F\}$$

here F is the set of terminal nodes in V. The probability assigned to path $\langle r, u\rangle$–$P(\langle r, u\rangle)$–is simply the probability $p_r(r, u)$–the probability that chance (Player P_0) selects edge $(r, u) \in E$. The fact that:

$$\sum_{u \in F} p_r(r, u) = 1$$

ensures that we can define the probability space $(\Omega, \mathcal{F}, P)$. Thus we have shown that the theorem is true for game trees of height 1.

Suppose the statement is true for game trees with height up to $k \geq 1$. We will show that the theorem is true for game trees of height $k + 1$. Let r be the root of tree T and consider the set of children of $U = \{u \in V : (r, u) \in E\}$. For each $u \in U$, we can define a game tree of height k with tree T_u by Theorem 4.38. The fact that this tree has height k implies that we can define a probability space $(\Omega_u, \mathcal{F}_u, P_u)$ with Ω_u composed of paths from u to the terminal vertices of T_u.

Suppose that vertex r is controlled by Player P_j ($j \neq 0$). Then the strategy σ_j determines a unique move that will be made by Player j at vertex r. Suppose that move m is determined by σ_j at vertex r and $\mu(r, u) = m$ for edge $(r, u) \in E$ with $u \in U$ (that is edge (r, u) is labeld m). We can define the new event set Ω of paths in the tree T from root r to a terminal vertex. The probability function on paths can then be defined as:

$$P(\langle r, v_1, \ldots, v_k\rangle) = \begin{cases} P_u(\langle v_1, \ldots, v_k\rangle) & \langle v_1, \ldots, v_k\rangle \in \Omega_u \\ 0 & \text{else} \end{cases}$$

The fact that P_u is a properly defined probability function over Ω_u implies that P is a properly defined probability function over Ω and thus $(\Omega, \mathcal{F}, P)$ is a probability space over the paths in T.

Now suppose that chance (Player P_0) controls r in the game tree. Again, Ω is the set of paths leading from r to a terminal vertex of T. The probability function on paths can then be defined as:

$$P(\langle r, v_1, \dots, v_k \rangle) = p_r(r, v_1)P_{v_1}(\langle r, v_1, \dots, v_k \rangle)$$

Here $v_1 \in U$ and $\langle r, v_1, \dots, v_k \rangle \in \Omega_{v_1}$, the set of paths leading from v_1 to a terminal vertex in tree T_{v_1} and $p(r, v_1)$ is the probability chance assigns to edge $(r, v_1) \in E$.

To see that this is a properly defined probability function, suppose that $\omega \in \Omega_u$ that is, ω is a path in tree T_u leading from u to a terminal vertex of T_u. Then a path in Ω is constructed by joining the path that leads from vertex r to vertex u and then following a path $\omega \in \Omega_u$. Let $\langle r, \omega \rangle$ denote such a path. Then we know:

$$(4.2) \qquad \sum_{u \in U} \sum_{\omega \in \Omega_u} P(\langle r, \omega \rangle) = \sum_{u \in U} \sum_{\omega \in \Omega_u} p(r, u) P_u(\omega) =$$

$$\sum_{u \in U} p(r, u) \left(\sum_{\omega \in \Omega_u} P_u(\omega) \right) = \sum_{u \in U} p(r, u) = 1$$

This is because $\sum_{\omega \in \Omega_u} P_u(\omega) = 1$. Since clearly $P(\langle r, \omega \rangle) \in [0, 1]$ and the paths through the game tree are independent, it follows that $(\Omega, \mathcal{F}, P)$ is a properly defined probability space. Thus the theorem follows by induction. This completes the proof. □

Example 4.43. Consider the simple game of poker we defined in Example 4.37. Suppose we fix strategies in which Player 1 always raises and Player 2 always calls. Then the resulting probability distribution defined as in Theorem 4.42 contains two paths (one when a red card is dealt and another when a black card is dealt. This is shown in Figure 4.13. The sample

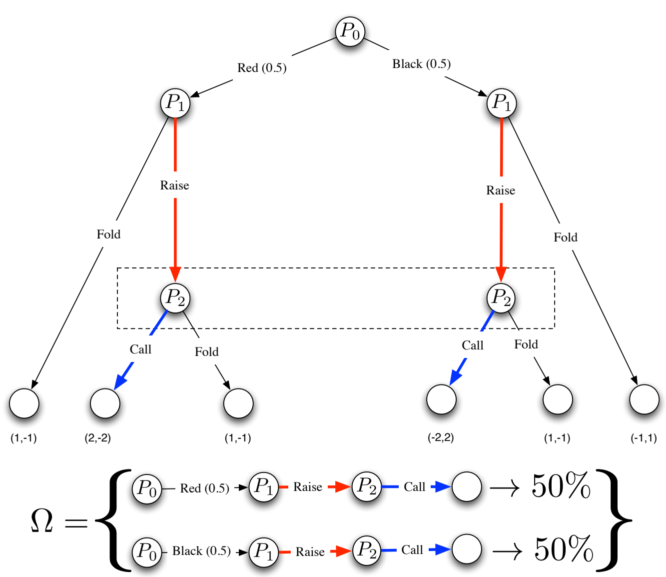


Figure 4.13. The probability space constructed from fixed player strategies in a game of chance. The strategy space is constructed from the unique choices determined by the strategy of the players and the independent random events that are determined by the chance moves.

space consists of the possible paths through the game tree. Notice that as in Theorem 4.40

the paths through the game tree are completely specified (and therefore unique) when the non-chance players are determining the moves. The only time probabilistic moves occur is when chance is causes the game to progress..

Example 4.44. Suppose we play a game in which Players 1 and 2 ante \$1 each. One card each is dealt to Player 1 and Player 2. Player 1 can choose to raise (and add a \$1 to the pot) or fold (and lose the pot). Player 2 can then choose to call (adding \$1) or fold (and loose the pot). Player 1 wins if both cards are black. Player 2 wins if both cards are red. The pot is split if the cards have opposite color. Suppose that Player 1 always chooses to raise and Player 2 always chooses to call. Then the game tree and strategies are shown in Figure 4.14. The sample space in this case consists of 4 distinct paths each with probability

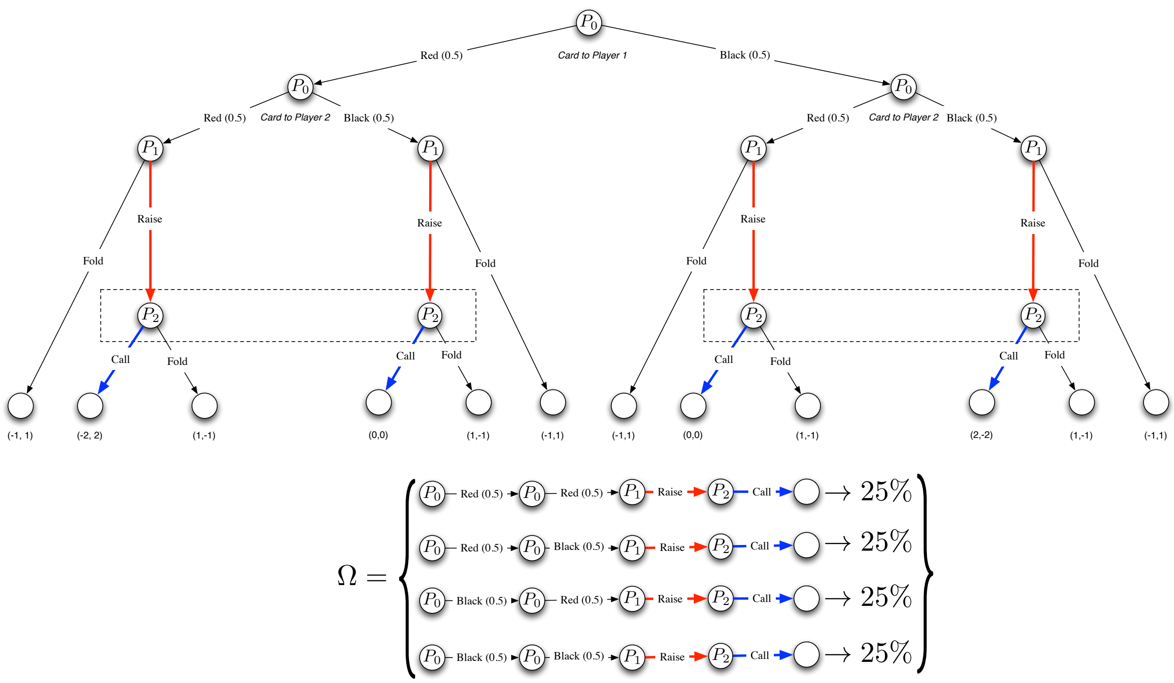


Figure 4.14. The probability space constructed from fixed player strategies in a game of chance. The strategy space is constructed from the unique choices determined by the strategy of the players and the independent random events that are determined by the chance moves. Note in this example that constructing the probabilities of the various events requires *multiplying* the probabilities of the chance moves in each path.

1/4, assuming that the cards are dealt with equal probability. Note in this example that constructing the probabilities of the various events requires *multiplying* the probabilities of the chance moves in each path. This is illustrated in the theorem when we write:

$$P(\langle r, v_1, \dots, v_k \rangle) = p_r(r, v_1) P_{v_1}(\langle r, v_1, \dots, v_k \rangle)$$

Exercise 26. Suppose that players always raise and call in the game defined in Exercise 23. Compute the probability space defined by these strategies in the game tree you developed.

Definition 4.45 (Strategy Space). Let Σ_i be the set of all strategies for Player i in a game tree $\mathcal{G}$. Then the entire *strategy space* is $\Sigma = \Sigma_1 \times \Sigma_2 \times \cdots \times \Sigma_n$.

Definition 4.46 (Strategy Payoff Function). Let $\mathcal{G}$ be a game tree with *no chance moves*. The *strategy payoff function* is a mapping $\pi : \Sigma \to \mathbb{R}^n$. If $\sigma_1, \dots, \sigma_N$ are strategies for Players 1 through n, then $\pi(\sigma_1, \dots, \sigma_N)$ is the vector of payoffs assigned to the terminal node of the path determined by the strategies $\sigma_1, \dots, \sigma_N$ in game tree $\mathcal{G}$. For each $i = 1, \dots, N$ $\pi_i(\sigma_1, \dots, \sigma_N)$ is the payoff to Player i in $\pi_i(\sigma_1, \dots, \sigma_N)$.

Example 4.47. Consider the Battle of the Bismark Sea game from Example 4.32. Then there are four distinct strategies in Σ with the following payoffs:

$$\begin{aligned}
&\pi\,(\text{Sail North, Search North}) = (-2, 2)\\
&\pi\,(\text{Sail South, Search North}) = (-2, 2)\\
&\pi\,(\text{Sail North, Search South}) = (-1, 1)\\
&\pi\,(\text{Sail South, Search South}) = (-3, 3)
\end{aligned}$$

Definition 4.48 (Expected Strategy Payoff Function). Let $\mathcal{G}$ be a game tree *with chance moves*. The *expected strategy payoff function* is a mapping $\pi : \Sigma \to \mathbb{R}^n$ defined as follows: If $\sigma_1, \dots, \sigma_N$ are strategies for Players 1 through n, then let $(\Omega, \mathcal{F}, P)$ be the probability space over the paths constructed by these strategies as given in Theorem 4.42. Let Π_i be a random variable that maps $\omega \in \Omega$ to the payoff for Player i at the terminal node in path ω. Let:

$$\pi_i(\sigma_1, \dots, \sigma_N) = \mathbb{E}(\Pi_i)$$

Then:

$$\pi(\sigma_1, \dots, \sigma_N) = \langle \pi_1(\sigma_1, \dots, \sigma_N), \dots, \pi_N(\sigma_1, \dots, \sigma_N)\rangle$$

As before, $\pi_i(\sigma_1, \dots, \sigma_N)$ is the expected payoff to Player i in $\pi(\sigma_1, \dots, \sigma_N)$.

Example 4.49. Consider Example 4.37. There are 4 distinct strategies in Σ:

$$\begin{cases}
\text{(Fold, Call)}\\
\text{(Fold, Fold)}\\
\text{(Raise, Call)}\\
\text{(Raise, Fold)}
\end{cases}$$

Let's focus on the strategy (Fold, Call). Then the resulting paths in the graph defined by these strategies are shown in Figure 4.15. There are two paths and we note that the decision

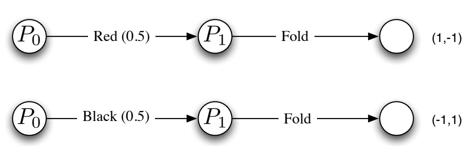


Figure 4.15. Game tree paths derived from the Simple Poker Game as a result of the strategy (Fold, Fold). The probability of each of these paths is 1/2.

made by Player 2 makes no difference in this case because Player 1 folds. Each path has probability 1/2. Our random variable Π_1 will map the top path (in Figure 4.15) to a \$1 payoff for Player 1 and will map the bottom path (in Figure 4.15) to a payoff of $-\$1$ for Player 1. Thus we can compute:

$$\pi_1\,(\text{Fold, Fold}) = \frac{1}{2}(1) + \frac{1}{2}(-1) = 0$$

Likewise,

$$\pi_2\,(\text{Fold, Fold}) = \frac{1}{2}(-1) + \frac{1}{2}(1) = 0$$

Thus we compute:

$$\pi\,(\text{Fold, Fold}) = (0, 0)$$

Using this approach, we can compute the expected payoff function to be:

$$\pi\,(\text{Fold, Call}) = (0, 0)$$
$$\pi\,(\text{Fold, Fold}) = (0, 0)$$
$$\pi\,(\text{Raise, Call}) = (0, 0)$$
$$\pi\,(\text{Raise, Fold}) = (1, -1)$$

Exercise 27. Explicitly show that the expected payoff function for Simple Poker is the one given in the previous example.

Definition 4.50 (Equilibrium). A strategy $(\sigma_1^*, \dots, \sigma_N^*) \in \Sigma$ is an equilibrium if for all i.

$$\pi_i(\sigma_1^*, \dots, \sigma_i^*, \dots, \sigma_N^*) \geq \pi_i(\sigma_1^*, \dots, \sigma_i, \dots, \sigma_N^*)$$

where $\sigma_i \in \Sigma_i$.

Example 4.51. Consider the Battle of the Bismark Sea. We can show that (Sail North, Search North) is an equilibrium strategy. Recall that:

$$\pi\,(\text{Sail North, Search North}) = (-2, 2)$$

Now, suppose that the Japanese deviate from this strategy and decide to sail south. Then the new payoff is:

$$\pi\,(\text{Sail South, Search North}) = (-2, 2)$$

Thus:

$$\pi_1\,(\text{Sail North, Search North}) \geq \pi_1\,(\text{Sail South, Search North})$$

Now suppose that the Allies deviate from the strategy and decide to search south. Then the new payoff is:

$$\pi\,(\text{Sail North, Search South}) = (-1, 1)$$

Thus:

$$\pi_2\,(\text{Sail North, Search North}) > \pi_2\,(\text{Sail North, Search South})$$

Exercise 28. Show that the strategy (Raise, Call) is an equilibrium strategy in Simple Poker.

Theorem 4.52. *Let $\mathcal{G} = (T, \mathbf{P}, \mathcal{S}, \nu, \mu, \pi, \mathcal{I}, \mathcal{P})$ be a game tree with complete information. Then there is an equilibrium strategy $(\sigma_1^*, \dots, \sigma_N^*) \in \Sigma$.*

Proof. We will apply induction on the height of the game tree $T = (V, E)$. Before proceeding to the proof, recall that a game with complete information is one in which if $v \in V$ and $I_v \in \mathcal{I}$ is the information set of vertex v, then $I_v = \{v\}$. Thus we can think of a strategy σ_i for player P_i as being as being a mapping from V to $\mathcal{S}$ as in Definition 4.21. We now proceed to the proof.

Suppose the height of the tree is 1. Then the tree consists of a root node r and a collection of terminal nodes F so that if $u \in F$ then $(r, u) \in E$. If chance controls r, then there is no strategy for any of the players, they are randomly assigned a payoff. Thus we can think of the empty strategy as the equilibrium strategy. On the other hand, if player P_i controls r, then we let $\sigma_i(r) = m \in \mathcal{S}$ so that if $\mu(r, u) = m$ for some $u \in F$ then $\pi_i(u) \geq \pi_i(v)$ for all other $v \in U$. That is, the vertex reached by making move m has a payoff for Player i that is greater than or equal to any other payoff Player i might receive at another vertex. All other players are assigned empty strategies (as they never make a move). Thus it is easy to see that this is an equilibrium strategy since no player can improve their payoff by changing strategies. Thus we have proved that there is an equilibrium strategy in this case.

Now suppose that the theorem is true for game trees $\mathcal{G}$ with complete information of height some $k \geq 1$. We will show that the statement holds for a game tree of height $k + 1$. Let r be the root of the tree and let $U = \{u \in V : (r, u) \in E\}$ be the set of children of r in T. If r is controlled by chance, then the first move of the game is controlled by chance. For each $u \in U$, we can construct a game tree with tree T_u by Theorem 4.38. By the induction hypothesis, we know there is some equilibrium strategy $(\sigma_1^{u^*}, \dots, \sigma_N^{u^*})$. Let $\pi_i^{u^*}$ be the payoff associated with using this strategy for Player P_i. Now consider any alternative strategy $(\sigma_1^{u^*}, \dots, \sigma_{i-1}^{u^*}, \sigma_i^u, \sigma_{i+1}^{u^*} \dots, \sigma_N^{u^*})$. Let π_i^u be the payoff to Player P_i that results from using this new strategy in the game with game tree T_u. It must be that

$$(4.3) \qquad \pi_i^{u^*} \geq \pi_i^u \quad \forall i \in \{1, \dots, N\}, u \in U$$

Thus we construct a new strategy for Player P_i so that if chance causes the game to transition to vertex u in the first step, then Player P_i will use strategy $\sigma_i^{u^*}$. Equation 4.3 ensures that Player i will never have a motivation to deviate from this strategy as the assumption of complete information assures us that Player i will know for certain to which $u \in U$ the game has transitioned.

Alternatively, suppose that the root is controlled by Player P_j. Let U and $\pi_i^{u^*}$ be as above. Then let $\sigma_j(r) = m \in \mathcal{S}$ so that if $\mu(r, u) = m$ then:

$$(4.4) \qquad \pi_j^{u^*} \geq \pi_j^{v^*}$$

for all $v \in U$. That is, Player P_j chooses a move that will yield a new game tree T_u that has the greatest terminal payoff using the equilibrium strategy $(\sigma_1^{u^*}, \dots, \sigma_N^{u^*})$ in that game tree. We can now define a new strategy:

1. At vertex r, $\sigma_j(r) = m$.
2. Every move in tree T_u is governed by $(\sigma_1^{u^*}, \dots, \sigma_N^{u^*})$
3. If $v \neq r$ and $v \notin T_u$ and $\nu(v) = i$, then $\sigma_i(v)$ may be chosen at random from $\mathcal{S}$ (because this vertex will never be reached during game play).

We can show that this is an equilibrium strategy. To see this, consider any other strategy. If Player $i \neq j$ deviates, then we know that this player will receive payoff π_i^u (as above) because Player j will force the game into the tree T_u after the first move. We know further that $\pi_i^{u^*} \geq \pi_i^u$. Thus, there is no incentive for Player P_i to deviate from the given strategy. He must play $(\sigma_1^{u^*}, \dots, \sigma_N^{u^*})$ in T_u. If Player j deviates at some vertex in T_u, then we know Player j will receive payoff $\pi_j^u \leq \pi_j^{u^*}$. Thus, once game play takes place inside tree T_u there is no reason to deviate from the given strategy. If Player j deviates on the first move and chooses a move m' so that $\mu(r, v) = m'$, then there are two possibilities:

(1) $\pi_j^{v^*} = \pi_j^{u^*}$
(2) $\pi_j^{v^*} < \pi_j^{u^*}$

In the first case, we can construct a strategy as before in which Player P_j will still receive the same payoff as if he played the strategy in which $\sigma_j(r) = m$ (instead of $\sigma_j(r) = m'$). In the second case, the best payoff Player P_j can obtain is $\pi_j^{v^*} < \pi_j^{u^*}$, so there is certainly no reason for Player P_j to deviate and chose to define $\sigma_j(r) = m'$. Thus, we have shown that this new strategy is an equilibrium. Thus there is an equilibrium strategy for this tree of height $k + 1$ and the proof follows by induction. □

Example 4.53. We can illustrate the construction in the theorem with the Battle of the Bismark Sea. In fact, you have already seen this construction once. Consider the game tree in Figure 4.12: We construct the equilibrium solution from the bottom of the tree up.

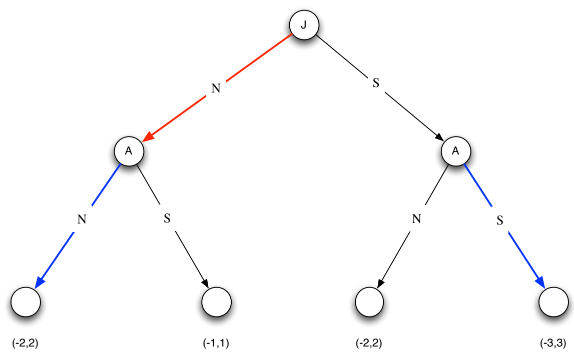


Figure 4.16. The game tree for the Battle of the Bismark Sea. If the Japanese sail north, the best move for the Allies is to search north. If the Japanese sail south, then the best move for the Allies is to search south. The Japanese, observing the payoffs, note that given these best strategies for the Allies, there best course of action is to sail North.

Consider the vertex controlled by the Allies in which the Japanese sail north. In the sub-tree below this node, the best move for the Allies is to search north (they receive the highest payoff). This is highlighted in blue. Now consider the vertex controlled by the Allies where the Japanese sail south. The best move for the Allies is to search south. Now, consider the root node controlled by the Japanese. The Japanese can examine the two sub-trees below this node and determine that the payoffs resulting from the equilibrium solutions in these trees are -2 (from sailing north) and -3 (from sailing south). Naturally, the Japanese will choose to so make the move of sailing north as this is the highest payoff they can achieve. Thus the equilibrium strategy is shown in red and blue in the tree in Figure 4.16.

Exercise 29. Show that in Rock-Paper-Scissors with perfect information, there are three equilibrium strategies.

Corollary 4.54 (Zermelo's Theorem). *Let $\mathcal{G} = (T, \mathbf{P}, \mathcal{S}, \nu, \mu, \pi)$ be a two-player game with complete information and no chance. Assume that the payoff is such that:*

1. *The only payoffs are $+1$ (win), -1 (lose).*
2. *Player 1 wins $+1$ if and only if Player 2 wins -1.*
3. *Player 2 wins $+1$ if and only if Player 1 wins -1.*

Finally, assume that the players alternate turns. Then one of the two players must have a strategy to obtain $+1$.

Exercise 30. Prove Zermelo's Theorem. Can you illustrate a game of this type?[Hint: Use Theorems 4.52 and 4.40. There are many games of this type.]

CHAPTER 5

Normal and Strategic Form Games and Matrices

1. Normal and Strategic Form

Let $\mathbf{P} = \{P_1, \dots, P_N\}$ be players in a game. In this section, we will assume that $\Sigma = \Sigma_1 \times, \Sigma_N$ is a discrete strategy space. That is, to each player $P_i \in \mathbf{P}$ we may ascribe a certain discrete set of strategies Σ_i. Certain types of game theory consider the case when Σ_i is not discrete; we will **not** consider this case in this section.

Definition 5.1 (Normal Form). Let $\mathbf{P}$ be a set of players, $\Sigma = \Sigma_1 \times \Sigma_2 \times \cdots \times \Sigma_N$ be a strategy space and let $\pi : \Sigma \to \mathbb{R}^N$ be a strategy payoff function. Then the triple: $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ is a game in *normal form*.

Remark 5.2. If $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ is a normal form game, then the function $\pi_i : \Sigma \to \mathbb{R}$ is the payoff function for Player P_i and returns the i^{th} component of the function π.

Definition 5.3 (Constant / General Sum Game). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ be a game in normal form. If there is a constant $C \in \mathbb{R}$ so that for all tuples $(\sigma_1, \dots, \sigma_N) \in \Sigma$ we have:

$$\sum_{i=1}^{N} \pi_i(\sigma_1, \dots, \sigma_N) = C \tag{5.1}$$

then $\mathcal{G}$ is called a *constant sum game*. If $C = 0$, then $\mathcal{G}$ is called a *zero sum game*. Any game that is *not* constant sum is called *general sum*.

Example 5.4. This example comes from http://www.advancednflstats.com/2008/06/game-theory-and-runpass-balance.html. A football play (in which the score does not change) is an example of a zero-sum game when the payoff is measured by yards gained or lost. In a football game, there are two players: the Offense (P_1) and the Defense (P_2) . The Offense may choose between two strategies:

$$\Sigma_1 = \{\text{Pass}, \text{Run}\} \tag{5.2}$$

The Defense may choose between three strategies:

$$\Sigma_2 = \{\text{Pass Defense}, \text{Run Defense}, \text{Blitz}\} \tag{5.3}$$

The yards gained by the Offense are lost by the Defense. Suppose the following payoff function (in terms of yards gained or lost by each player) π is defined:

$$\pi(\text{Pass}, \text{Pass Defense}) = (-3, 3)$$
$$\pi(\text{Pass}, \text{Run Defense}) = (9, -9)$$
$$\pi(\text{Pass}, \text{Blitz}) = (-5, 5)$$
$$\pi(\text{Run}, \text{Pass Defense}) = (4, -4)$$

$$\pi(\text{Run}, \text{Run Defense}) = (-3, 3)$$
$$\pi(\text{Run}, \text{Blitz}) = (6, -6)$$

If $\mathbf{P} = \{P_1, P_2\}$ and $\Sigma = \Sigma_1 \times \Sigma_2$, then the tuple $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ is a zero-sum game in normal form. Note that each pair in the definition of the payoff function sums to zero.

Remark 5.5. Just as in a game in extensive form, we can define an equilibrium. This definition is identical to the definition we gave in Chapter 4.50.

Definition 5.6 (Equilibrium). A strategy $(\sigma_1^*, \ldots, \sigma_N^*) \in \Sigma$ is an equilibrium if for all i.

$$\pi_i(\sigma_1^*, \ldots, \sigma_i^*, \ldots, \sigma_N^*) \geq \pi_i(\sigma_1^*, \ldots, \sigma_i, \ldots, \sigma_N^*)$$

where $\sigma_i \in \Sigma_i$.

2. Strategic Form Games

Recall an $m \times n$ matrix is a rectangular array of numbers, usually drawn from a field such as $\mathbb{R}$. We write an $m \times n$ matrix with values in $\mathbb{R}$ as $\mathbf{A} \in \mathbb{R}^{m \times n}$. The matrix consists of m rows and n columns. The element in the i^{th} row and j^{th} column of $\mathbf{A}$ is written as $\mathbf{A}_{ij}$. The j^{th} column of $\mathbf{A}$ can be written as $\mathbf{A}_{\cdot j}$, where the $\cdot$ is interpreted as ranging over every value of i (from 1 to m). Similarly, the i^{th} row of $\mathbf{A}$ can be written as $\mathbf{A}_{i\cdot}$. When $m = n$, then the matrix $\mathbf{A}$ is called *square*.

Definition 5.7 (Strategic Form–2 Player Games). $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ be a normal form game with $\mathbf{P} = \{P_1, P_2\}$ and $\Sigma = \Sigma_1 \times \Sigma_2$. If the strategies in Σ_i $(i = 1, 2)$ are ordered so that $\Sigma_i = \{\sigma_1^i, \ldots, \sigma_{n_i}^i\}$ $(i = 1, 2)$. Then for each player there is a matrix $\mathbf{A}_i \in \mathbb{R}^{n_1 \times n_2}$ so that element (r, c) of $\mathbf{A}_i$ is given by $\pi_i(\sigma_r^1, \sigma_c^2)$. Then the tuple $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}_1, \mathbf{A}_2)$ is a two-player game in *strategic form*.

Remark 5.8. Games with two players given in strategic form are also sometimes called *matrix games* because they are defined completely by matrices. Note also that by convention, Player P_1's strategies correspond to the rows of the matrices, while Player P_2's strategies correspond to the columns of the matrices.

Example 5.9. Consider the two-player game defined in the Battle of the Bismark Sea. If we assume that the strategies for the players are:

$$\Sigma_1 = \{\text{Sail North}, \text{Sail South}\}$$
$$\Sigma_2 = \{\text{Search North}, \text{Search South}\}$$

Then the payoff matrices for the two players are:

$$\mathbf{A}_1 = \begin{bmatrix} -2 & -1 \\ -2 & -3 \end{bmatrix}$$
$$\mathbf{A}_2 = \begin{bmatrix} 2 & 1 \\ 2 & 3 \end{bmatrix}$$

Here, the *rows* represent the different strategies of Player 1 and the *columns* represent the strategies of Player 2. Thus the $(1, 1)$ entry in matrix A_1 is the payoff to Player 1 when the strategy pair (Sail North, Search North) is played. The $(2, 1)$ entry in matrix A_2 is the payoff to Player 1 when the strategy pair (Sail South, Search North) is played etc. Notice in this case that $\mathbf{A}_1 = -\mathbf{A}_2$. This is because the Battle of the Bismark Sea is a zero-sum game.

Exercise 31. Compute the payoff matrices for Example 5.4.

Example 5.10 (Chicken). Consider the following two-player game: Two cars face each other and begin driving (quickly) toward each other. (See Figure 5.1.) The player who swerves first loses 1 point, the other player wins 1 point. If both players swerve, then each receives 0 points. If neither player swerves, a very bad crash occurs and both players lose 10 points. Assuming that the strategies for Player 1 are in the rows, while the strategies for

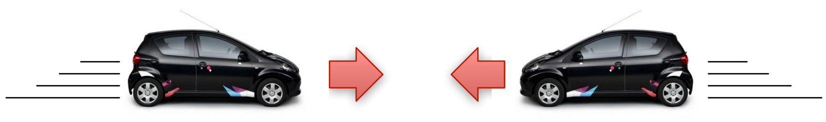


Figure 5.1. In Chicken, two cars drive toward one another. The player who swerves first loses 1 point, the other player wins 1 point. If both players swerve, then each receives 0 points. If neither player swerves, a very bad crash occurs and both players lose 10 points.

Player 2 are in the columns, then the two matrices for the players are:

	Swerve	Don't Swerve
Swerve	0	-1
Don't Swerve	1	-10

	Swerve	Don't Swerve
Swerve	0	1
Don't Swerve	-1	-10

From this we can see the matrices are:

$$\mathbf{A}_1 = \begin{bmatrix} 0 & -1 \\ 1 & -10 \end{bmatrix}$$

$$\mathbf{A}_2 = \begin{bmatrix} 0 & 1 \\ -1 & -10 \end{bmatrix}$$

Note that the Game of Chicken is **not** a zero-sum game, i.e. it is a general sum game.

Exercise 32. Construct payoff matrices for Rock-Paper-Scissors. Also construct the normal form of the game.

Remark 5.11. Definition 5.7 can be extended to N player games. However, we no longer have matrices with payoff values for various strategies. Instead, we construct N N-dimensional arrays (or tensors). So a game with 3 players yields 3 arrays with dimension 3. This is illustrated in Figure 5.2 Multidimensional arrays are easy to represent in computers, but hard to represent on the page. They have multiple indices, instead of just 1 index like a vector or 2 indices like a matrix. The elements of the array for Player i store the various payoffs for Player i under different strategy combinations of the different players. If there are three players, then there will be three different arrays, one for each player.

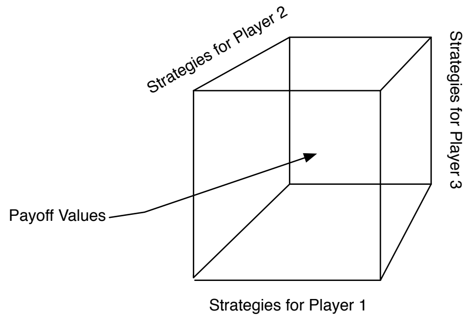


Figure 5.2. A three dimensional array is like a matrix with an extra dimension. They are difficult to capture on a page. The elements of the array for Player i store the various payoffs for Player i under different strategy combinations of the different players. If there are three players, then there will be three different arrays.

REMARK 5.12. The normal form of a (two-player) game is essentially the recipe for transforming a game in extensive form into a game in strategic form. Any game in extensive form can be transformed in this way and the strategic form can be analyzed. Reasons for doing this include the fact that the strategic form is substantially more compact. However, it can be complex to compute if the size of the game tree in extensive form is very large.

EXERCISE 33. Compute the strategic form of the two-player Simple Poker game using the expected payoff function defined in Example 4.49

3. Review of Basic Matrix Properties

DEFINITION 5.13 (Dot Product). Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ be two vectors. If:

$$\begin{aligned}\mathbf{x} &= (x_1, x_2, \dots, x_n)\\ \mathbf{y} &= (y_1, y_2, \dots, y_n)\end{aligned}$$

Then the *dot product* of these vectors is:

$$\mathbf{x} \cdot \mathbf{y} = x_1y_1 + x_2y_2 + \cdots + x_ny_n \tag{5.4}$$

REMARK 5.14. We can apply Definition 5.13 to the case when $\mathbf{x}$ and $\mathbf{y}$ are column or row vectors in the obvious way.

DEFINITION 5.15 (Matrix Addition). If $\mathbf{A}$ and $\mathbf{B}$ are both in $\mathbb{R}^{m \times n}$, then $\mathbf{C} = \mathbf{A} + \mathbf{B}$ is the matrix sum of $\mathbf{A}$ and $\mathbf{B}$ and

$$\mathbf{C}_{ij} = \mathbf{A}_{ij} + \mathbf{B}_{ij} \text{ for } i = 1, \dots, m \text{ and } j = 1, \dots, n \tag{5.5}$$

EXAMPLE 5.16.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} + \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1+5 & 2+6 \\ 3+7 & 4+8 \end{bmatrix} = \begin{bmatrix} 6 & 8 \\ 10 & 12 \end{bmatrix} \tag{5.6}$$

Definition 5.17 (Row/Column Vector). A $1 \times n$ matrix is called a *row vector*, and a $m \times 1$ matrix is called a *column vector*. For the remainder of these notes, every vector will be thought of **column vector** unless otherwise noted.

It should be clear that any row of matrix $\mathbf{A}$ could be considered a row vector in $\mathbb{R}^n$ and any column of $\mathbf{A}$ could be considered a column vector in $\mathbb{R}^m$.

Definition 5.18 (Matrix Multiplication). If $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $B \in \mathbb{R}^{n\times p}$, then $\mathbf{C} = \mathbf{AB}$ is the *matrix product* of $\mathbf{A}$ and $\mathbf{B}$ and

$$\mathbf{C}_{ij} = \mathbf{A}_{i\cdot} \cdot \mathbf{B}_{\cdot j} \tag{5.7}$$

Note, $\mathbf{A}_{i\cdot} \in \mathbb{R}^{1\times n}$ (an n-dimensional vector) and $\mathbf{B}_{\cdot j} \in \mathbb{R}^{n\times 1}$ (another n-dimensional vector), thus making the dot product meaningful.

Example 5.19.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}\begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix} = \begin{bmatrix} 1(5)+2(7) & 1(6)+2(8) \\ 3(5)+4(7) & 3(6)+4(8) \end{bmatrix} = \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix} \tag{5.8}$$

Definition 5.20 (Matrix Transpose). If $\mathbf{A} \in \mathbb{R}^{m\times n}$ is a $m \times n$ matrix, then the *transpose* of $\mathbf{A}$ dented $\mathbf{A}^T$ is an $m \times n$ matrix defined as:

$$\mathbf{A}^T_{ij} = \mathbf{A}_{ji} \tag{5.9}$$

Example 5.21.

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}^T = \begin{bmatrix} 1 & 3 \\ 2 & 4 \end{bmatrix} \tag{5.10}$$

The matrix transpose is a particularly useful operation and makes it easy to transform column vectors into row vectors, which enables multiplication. For example, suppose $\mathbf{x}$ is an $n \times 1$ column vector (i.e., $\mathbf{x}$ is a vector in $\mathbb{R}^n$) and suppose $\mathbf{y}$ is an $n \times 1$ column vector. Then:

$$\mathbf{x} \cdot \mathbf{y} = \mathbf{x}^T\mathbf{y} \tag{5.11}$$

Exercise 34. Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$. Use the definitions of matrix addition and transpose to prove that:

$$(\mathbf{A} + \mathbf{B})^T = \mathbf{A}^T + \mathbf{B}^T \tag{5.12}$$

[Hint: If $\mathbf{C} = \mathbf{A} + \mathbf{B}$, then $\mathbf{C}_{ij} = \mathbf{A}_{ij} + \mathbf{B}_{ij}$, the element in the (i, j) position of matrix $\mathbf{C}$. This element moves to the (j, i) position in the transpose. The (j, i) position of $\mathbf{A}^T + \mathbf{B}^T$ is $\mathbf{A}^T_{ji} + \mathbf{B}^T_{ji}$, but $\mathbf{A}^T_{ji} = \mathbf{A}_{ij}$. Reason from this point.]

Exercise 35. Let $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$. Prove by example that $\mathbf{AB} \neq \mathbf{BA}$; that is, matrix multiplication is *not commutative*. [Hint: Almost any pair of matrices you pick (that can be multiplied) will not commute.]

Exercise 36. Let $\mathbf{A} \in \mathbb{R}^{m\times n}$ and let, $\mathbf{B} \in \mathbb{R}^{n\times p}$. Use the definitions of matrix multiplication and transpose to prove that:

$$(\mathbf{AB})^T = \mathbf{B}^T\mathbf{A}^T \tag{5.13}$$

[Hint: Use similar reasoning to the hint in Exercise 34. But this time, note that $\mathbf{C}_{ij} = \mathbf{A}_{i\cdot} \cdot \mathbf{B}_{\cdot j}$, which moves to the (j, i) position. Now figure out what is in the (j, i) position of $\mathbf{B}^T\mathbf{A}^T$.]

Let $\mathbf{A}$ and $\mathbf{B}$ be two matrices with the same number of rows (so $\mathbf{A} \in \mathbb{R}^{m\times n}$ and $\mathbf{B} \in \mathbb{R}^{m\times p}$). Then the augmented matrix $[\mathbf{A}|\mathbf{B}]$ is:

$$\left[\begin{array}{cccc|cccc} a_{11} & a_{12} & \dots & a_{1n} & b_{11} & b_{12} & \dots & b_{1p} \\ a_{21} & a_{22} & \dots & a_{2n} & b_{21} & b_{22} & \dots & b_{2p} \\ \vdots & & \ddots & \vdots & \vdots & & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} & b_{m1} & b_{m2} & \dots & b_{mp} \end{array}\right] \tag{5.14}$$

Thus, $[\mathbf{A}|\mathbf{B}]$ is a matrix in $\mathbb{R}^{m\times(n+p)}$.

EXAMPLE 5.22. Consider the following matrices:

$$\mathbf{A} = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} 7 \\ 8 \end{bmatrix}$$

Then $[\mathbf{A}|\mathbf{B}]$ is:

$$[\mathbf{A}|\mathbf{B}] = \left[\begin{array}{cc|c} 1 & 2 & 7 \\ 3 & 4 & 8 \end{array}\right]$$

EXERCISE 37. By analogy define the augmented matrix $\left[\frac{\mathbf{A}}{\mathbf{B}}\right]$. Note, this is **not** a fraction. In your definition, identify the appropriate requirements on the relationship between the number of rows and columns that the matrices must have. [Hint: Unlike $[\mathbf{A}|\mathbf{B}]$, the number of rows don't have to be the same, since your concatenating on the rows, not columns. There should be a relation between the numbers of columns though.]

4. Special Matrices and Vectors

DEFINITION 5.23 (Identify Matrix). The $n \times n$ *identify matrix* is:

$$\mathbf{I}_n = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & 1 \end{bmatrix} \tag{5.15}$$

When it is clear from context, we may simply write $\mathbf{I}$ and omit the subscript n.

EXERCISE 38. Let $\mathbf{A} \in \mathbb{R}^{n\times n}$. Show that $\mathbf{A}\mathbf{I}_n = \mathbf{I}_n\mathbf{A} = \mathbf{A}$. Hence, $\mathbf{I}$ is an identify for the matrix multiplication operation on square matrices. [Hint: Do the multiplication out long hand.]

DEFINITION 5.24 (Standard Basis Vector). The standard basis vector $\mathbf{e}_i \in \mathbb{R}^n$ is:

$$\mathbf{e}_i = \Bigg(\underbrace{0, 0, \dots,}_{i-1} 1, \underbrace{0, \dots, 0}_{n-i-1}\Bigg)$$

Note, this definition is only valid for $n \geq i$. Further the standard basis vector $\mathbf{e}_i$ is also the i^{th} row or column of $\mathbf{I}_n$.

DEFINITION 5.25 (Unit and Zero Vectors). The vector $\mathbf{e} \in \mathbb{R}^n$ is the *one vector* $\mathbf{e} = (1, 1, \dots, 1)$. Similarly, the *zero vector* $\mathbf{0} = (0, 0, \dots, 0) \in \mathbb{R}^n$. We assume that the length of $\mathbf{e}$ and $\mathbf{0}$ will be determined from context.

Exercise 39. Let $\mathbf{x} \in \mathbb{R}^n$, considered as a column vector (our standard assumption). Define:

$$\mathbf{y} = \frac{\mathbf{x}}{\mathbf{e}^T\mathbf{x}}$$

Show that $\mathbf{e}^T\mathbf{y} = \mathbf{y}^T\mathbf{e} = 1$. [Hint: First remember that $\mathbf{e}^T\mathbf{x}$ is a scalar value (it's $\mathbf{e} \cdot \mathbf{x}$). Second, remember that a scalar times a vector is just a new vector with each term multiplied by the scalar. Last, use these two pieces of information to write the product $\mathbf{e}^T\mathbf{y}$ as a sum of fractions.]

5. Strategy Vectors and Matrix Games

Consider a two-player game in strategic form $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}_1, \mathbf{A}_2)$. When only two players are involved, we usually write $\mathbf{A}_1 = \mathbf{A}$ and $\mathbf{A}_2 = \mathbf{B}$. This removes unnecessary subscripts.

Furthermore, in a zero-sum game, we know that $\mathbf{A} = -\mathbf{B}$. Since we can easily deduce $\mathbf{B}$ from $\mathbf{A}$ we can write $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A})$ for a zero-sum game. In this case, we will understand that this is a zero sum-game with $\mathbf{B} = -\mathbf{A}$.

We can use standard basis vectors to compute the payoff to Player P_i when a specific set of strategies are used.

Remark 5.26. Our next proposition relates the strategy set Σ to pairs of standard basis vectors and reduces computing the payoff function to simple matrix multiplication.

Proposition 5.27. *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a two-player game in strategic form with $\Sigma_1 = \{\sigma_1^1, \dots, \sigma_m^1\}$ and $\Sigma_2 = \{\sigma_1^2, \dots, \sigma_n^2\}$. If Player P_1 chooses strategy σ_r^1 and Player P_2 chooses strategy σ_c^2, then:*

$$\pi_1(\sigma_r^1, \sigma_c^2) = \mathbf{e}_r^T\mathbf{A}\mathbf{e}_c \tag{5.16}$$

$$\pi_2(\sigma_r^1, \sigma_c^2) = \mathbf{e}_r^T\mathbf{B}\mathbf{e}_c \tag{5.17}$$

Proof. For any matrix $\mathbf{A} \in \mathbb{R}^{m \times n}$, $\mathbf{A}\mathbf{e}_c$ returns column c of matrix $\mathbf{A}$, that is, $\mathbf{A}_{\cdot c}$. Likewise $\mathbf{e}_r^T\mathbf{A}_{\cdot c}$ is the r^{th} element of this vector. Thus, $\mathbf{e}_r^T\mathbf{A}\mathbf{e}_c$ is the $(r, c)^{\text{th}}$ element of the matrix $\mathbf{A}$. By definition, this must be the payoff for the strategy pair (σ_r^1, σ_c^2) for Player P_1. A similar argument follows for Player P_2 and matrix $\mathbf{B}$. □

Remark 5.28. What Proposition 5.27 says is that for two-player matrix games, we can relate any choice of strategy that Player P_i makes with a unit vector. Thus, we can actually define the payoff function in terms of vector and matrix multiplication. We will see that this can be generalized to cases when the strategies of the players are *not* represented by standard basis vectors.

Example 5.29. Consider the game of Chicken. Suppose Player P_1 decides to swerve, while Player P_2 decides not to swerve. Then we can represent the strategy of Player P_1 by the vector:

$$\mathbf{e}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

while the strategy of Player P_2 is represented by the vector:

$$\mathbf{e}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$$

Recall the payoff matrices for this game:

$$\mathbf{A} = \begin{bmatrix} 0 & -1 \\ 1 & -10 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} 0 & 1 \\ -1 & -10 \end{bmatrix}$$

Then we can compute:

$$\pi_1(\text{Swerve}, \text{Don't Swerve}) = \mathbf{e}_1^T \mathbf{A} \mathbf{e}_2 = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & -1 \\ 1 & -10 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = -1$$

$$\pi_2(\text{Swerve}, \text{Don't Swerve}) = \mathbf{e}_1^T \mathbf{B} \mathbf{e}_2 = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \\ -1 & -10 \end{bmatrix} \cdot \begin{bmatrix} 0 \\ 1 \end{bmatrix} = 1$$

We can also consider the case when both players swerve. Then we can represent the strategies of both Players by $\mathbf{e}_1$. In this case we have:

$$\pi_1(\text{Swerve}, \text{Swerve}) = \mathbf{e}_1^T \mathbf{A} \mathbf{e}_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & -1 \\ 1 & -10 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 0$$

$$\pi_2(\text{Swerve}, \text{Swerve}) = \mathbf{e}_1^T \mathbf{B} \mathbf{e}_1 = \begin{bmatrix} 1 & 0 \end{bmatrix} \cdot \begin{bmatrix} 0 & 1 \\ -1 & -10 \end{bmatrix} \cdot \begin{bmatrix} 1 \\ 0 \end{bmatrix} = 0$$

Definition 5.30 (Symmetric Game). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$. If $\mathbf{A} = \mathbf{B}^T$ then $\mathcal{G}$ is called a *symmetric game*.

Remark 5.31. We will not consider symmetric games until later. We simply present the definition in order to observe some of the interesting relationships between matrix operations and games.

Remark 5.32. Our last proposition relates the definition of Equilibria (Definition 5.6) and the properties of matrix games and strategies.

Proposition 5.33 (Equilibrium). *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a two-player game in strategic form with $\Sigma = \Sigma_1 \times \Sigma_2$. The expressions*

$$\mathbf{e}_i^T \mathbf{A} \mathbf{e}_j \geq \mathbf{e}_k^T \mathbf{A} \mathbf{e}_j \quad \forall k \neq i \tag{5.18}$$

and

$$\mathbf{e}_i^T \mathbf{B} \mathbf{e}_j \geq \mathbf{e}_i^T \mathbf{B} \mathbf{e}_l \quad \forall l \neq j \tag{5.19}$$

hold if and only if $(\sigma_i^1, \sigma_j^2) \in \Sigma_1 \times \Sigma_2$ is an equilibrium strategy.

Proof. From Proposition 5.27, we know that:

$$\pi_1(\sigma_i^1, \sigma_j^2) = \mathbf{e}_i^T \mathbf{A} \mathbf{e}_j \tag{5.20}$$

$$\pi_2(\sigma_i^1, \sigma_j^2) = \mathbf{e}_i^T \mathbf{B} \mathbf{e}_j \tag{5.21}$$

From Equation 5.18 we know that for all $k \neq i$:

$$\pi_1(\sigma_i^1, \sigma_j^2) \geq \pi_1(\sigma_k^1, \sigma_j^2) \tag{5.22}$$

From Equation 5.19 we know that for all $l \neq j$:

$$\pi_2(\sigma_i^1, \sigma_j^2) \geq \pi_2(\sigma_i^1, \sigma_l^2) \tag{5.23}$$

Thus from Definition 5.6, it is clear that $(\sigma_i^1, \sigma_j^2) \in \Sigma$ is an equilibrium strategy. The converse is clear from this as well. $\square$

REMARK 5.34. We can now think of relating a strategy choice for player i, $\sigma_k^i \in \Sigma_i$ with the unit vector $\mathbf{e}_k$. From context, we will be able to identify to which player's strategy vector $\mathbf{e}_k$ corresponds.

CHAPTER 6

Saddle Points, Mixed Strategies and the Minimax Theorem

Let us return to the notion of an equilibrium point for a two-player zero sum game. For the remainder of this section, we will assume that $\Sigma = \Sigma_1 \times \Sigma_2$ and $\Sigma_1 = \{\sigma_1^1, \dots, \sigma_m^1\}$ and $\Sigma_2 = \{\sigma_1^2, \dots, \sigma_n^2\}$. Then any two-player zero-sum game in strategic form will be a tuple: $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A})$ with $\mathbf{A} \in \mathbb{R}^{m\times n}$.

1. Saddle Points

THEOREM 6.1. *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A})$ be a zero-sum two player game. A strategy pair $(\mathbf{e}_i, \mathbf{e}_j)$ is an equilibrium strategy if and only if:*

$$\mathbf{e}_i^T\mathbf{A}\mathbf{e}_j = \max_{k\in\{1,\dots,m\}} \min_{l\in\{1,\dots,n\}} \mathbf{A}_{kl} = \min_{l\in\{1,\dots,n\}} \max_{k\in\{1,\dots,m\}} \mathbf{A}_{kl} \tag{6.1}$$

EXAMPLE 6.2. Before we prove Theorem 6.1, let's first consider an example. This example comes from [**WV02**] (Chapter 12). Two network corporations believe there are $100,000,000$ viewers to be had during Thursday night, prime-time (8pm - 9pm). The corporations must decide which type of programming to run: Science Fiction, Drama or Comedy. If the two networks initially split the $100,000,000$ viewers evenly, we can think of the payoff matrix as determining how many excess viewers the networks' strategies will yield over $50,000,000$: The payoff matrix (in millions) for Network 1 is shown in Expression 6.2:

$$\mathbf{A} = \begin{bmatrix} -15 & -35 & 10 \\ -5 & 8 & 0 \\ -12 & -36 & 20 \end{bmatrix} \tag{6.2}$$

The expression:

$$\min_{l\in\{1,\dots,n\}} \max_{k\in\{1,\dots,m\}} \mathbf{A}_{kl}$$

asks us to compute the maximum value in each **column** to create the set:

$$C_{\max} = \{c_l^* = \max\{\mathbf{A}_{kl} : k \in \{1,\dots,m\}\} : l \in \{1,\dots,n\}\}$$

and then choose the smallest value in this case. If we look at this matrix, the column maximums are:

$$\begin{bmatrix} -5 & 8 & 20 \end{bmatrix}$$

We then choose the minimum value in this case and it is -5. This value occurs at position $(2, 1)$.

The expression

$$\max_{k\in\{1,\dots,m\}} \min_{l\in\{1,\dots,n\}} \mathbf{A}_{kl}$$

asks us to compute the minimum value in each **row** to create the set:

$$R_{\min} = \{r_k^* = \min\{\mathbf{A}_{kl} : l \in \{1,\dots,n\}\} : k \in \{1,\dots,m\}\}$$

and then choose the largest value in this case. Again, if we look at the matrix in Expression 6.2 we see that the minimum values in the rows are:

$$\begin{bmatrix} -35 \\ -5 \\ -36 \end{bmatrix}$$

The largest value in this case is -5. Again, this value occurs at position $(2,1)$.

Putting this all together, we get Figure 6.1:

Payoff Matrix			**Row Min**
-15	-35	10	-35
$\boxed{-5}$	8	0	$\boxed{-5}$
-12	-36	20	-36
$\boxed{-5}$	8	20	maxmin = -5
Column Max			minmax = -5

Figure 6.1. The minimax analysis of the game of competing networks. The row player knows that Player 2 (the column player) is trying to maximize her [Player 2's] payoff. Thus, Player 1 asks: "What is the worst possible outcome I could see if I played a strategy corresponding to this row?" Having obtained these *worst possible scenarios* he chooses the row with the highest value. Player 2 does something similar in columns.

Let's try and understand why we would do this. The row player (Player 1) *knows* that Player 2 (the column player) is trying to maximize her [Player 2's] payoff. Since this is a zero-sum game, any increase to Player 2's payoff will come at the expense of Player 1. So Player 1 looks at each *row* independently (since his strategy comes down to choosing a row) and asks, "What is the worst possible outcome I could see if I played a strategy corresponding to this row?" Having obtained these *worst possible scenarios* he chooses the row with the highest value.

Player 2 faces a similar problem. She knows that Player 1 wishes to maximize his payoff and that any gain will come at her expense. So Player 2 looks across each column of matrix **A** and asks what is the best possible score Player 1 can achieve if I [Player 2] choose to play the strategy corresponding to the given column. Remember, the negation of this value will be Player 2's payoff in this case. Having done that, Player 2 then chooses the column that minimizes this value and thus *maximizes* her payoff.

If these two values are equal, then the theorem claims that the resulting strategy pair is an equilibrium.

Exercise 40. Show that the strategy $(\mathbf{e}_2, \mathbf{e}_1)$ is an equilibrium for the game in Example 6.2. That is, show that the strategy (Drama, Science Fiction) is an equilibrium strategy for the networks.

Exercise 41. Show that (Sail North, Search North) is an equilibrium solution for the Battle of the Bismark Sea using the approach from Example 6.2 and Theorem 6.1.

PROOF OF THEOREM 6.1. ($\Rightarrow$) Suppose that $(\mathbf{e}_i, \mathbf{e}_j)$ is an equilibrium solution. Then we know that:

$$\mathbf{e}_i^T \mathbf{A} \mathbf{e}_j \geq \mathbf{e}_k^T \mathbf{A} \mathbf{e}_j$$
$$\mathbf{e}_i^T (-\mathbf{A}) \mathbf{e}_j \geq \mathbf{e}_i^T (-\mathbf{A}) \mathbf{e}_l$$

for all $k \in \{1, \dots, m\}$ and $l \in \{1, \dots, n\}$. We can obviously write this as:

$$\mathbf{e}_i^T \mathbf{A} \mathbf{e}_j \geq \mathbf{e}_k^T \mathbf{A} \mathbf{e}_j \tag{6.3}$$

and

$$\mathbf{e}_i^T \mathbf{A} \mathbf{e}_j \leq \mathbf{e}_i^T \mathbf{A} \mathbf{e}_l \tag{6.4}$$

We know that $\mathbf{e}_i^T \mathbf{A} \mathbf{e}_j = \mathbf{A}_{ij}$ and that Equation 6.3 holds if and only if:

$$\mathbf{A}_{ij} \geq \mathbf{A}_{kj} \tag{6.5}$$

for all $k \in \{1, \dots, m\}$. From this we deduce that element i must be a maximal element in column $\mathbf{A}_{\cdot j}$. Based on this, we know that for each row $k \in \{1, \dots, m\}$:

$$\mathbf{A}_{ij} \geq \min\{\mathbf{A}_{kl} : l \in \{1, \dots, n\}\} \tag{6.6}$$

To see this, note that for a fixed row $k \in \{1, \dots, m\}$:

$$\mathbf{A}_{kj} \geq \min\{\mathbf{A}_{kl} : l \in \{1, \dots, n\}\}$$

This means that if we compute the minimum value in a row k, then the value in column j, $\mathbf{A}_{kj}$ must be at least as large as that minimal value. But, Expression 6.6 implies that:

$$\mathbf{e}_i^T \mathbf{A} \mathbf{e}_j = \mathbf{A}_{ij} = \max_{k \in \{1,\dots,m\}} \min_{l \in \{1,\dots,n\}} A_{kl} \tag{6.7}$$

Likewise, Equation 6.4 holds if and only if

$$\mathbf{A}_{ij} \leq \mathbf{A}_{il} \tag{6.8}$$

for all $l \in \{1, \dots, n\}$. From this we deduce that element j must be a minimal element in row $\mathbf{A}_{i\cdot}$. Based on this, we know that for each column $l \in \{1, \dots, n\}$:

$$\mathbf{A}_{ij} \leq \max\{\mathbf{A}_{kl} : k \in \{1, \dots, m\}\} \tag{6.9}$$

To see this, note that for a fixed column $l \in \{1, \dots, n\}$:

$$\mathbf{A}_{il} \leq \max\{\mathbf{A}_{kl} : k \in \{1, \dots, m\}\}$$

This means that if we compute the maximum value in a column l, then the value in row i, $\mathbf{A}_{il}$ must not exceed that maximal value. But Expression 6.9 implies that:

$$\mathbf{e}_i^T \mathbf{A} \mathbf{e}_j = \mathbf{A}_{ij} = \min_{l \in \{1,\dots,n\}} \max_{k \in \{1,\dots,m\}} \mathbf{A}_{kl} \tag{6.10}$$

Thus it follows that:

$$\mathbf{A}_{ij} = \mathbf{e}_i^T \mathbf{A} \mathbf{e}_j = \max_{k \in \{1,\dots,m\}} \min_{l \in \{1,\dots,n\}} \mathbf{A}_{ij} = \min_{l \in \{1,\dots,n\}} \max_{k \in \{1,\dots,m\}} \mathbf{A}_{kl}$$

($\Leftarrow$) To prove the converse, suppose that:

$$\mathbf{e}_i^T \mathbf{A} \mathbf{e}_j = \max_{k \in \{1,\dots,m\}} \min_{l \in \{1,\dots,n\}} \mathbf{A}_{kl} = \min_{l \in \{1,\dots,n\}} \max_{k \in \{1,\dots,m\}} \mathbf{A}_{kl}$$

Consider:

$$\mathbf{e}_k^T \mathbf{A} \mathbf{e}_j = \mathbf{A}_{kj}$$

The fact that:

$$\mathbf{A}_{ij} = \max_{k \in \{1,\dots,m\}} \min_{l \in \{1,\dots,n\}} \mathbf{A}_{kl}$$

implies that $\mathbf{A}_{ij} \geq \mathbf{A}_{kj}$ for any $k \in \{1, \dots, m\}$. To see this remember:

$$C_{\max} = \{c_l^* = \max\{\mathbf{A}_{kl} : k \in \{1, \dots, m\}\} : l \in \{1, \dots, n\}\} \tag{6.11}$$

and $\mathbf{A}_{ij} \in C_{\max}$ by construction. Thus it follows that:

$$\mathbf{e}_i^T \mathbf{A} \mathbf{e}_j \geq \mathbf{e}_k^T \mathbf{A} \mathbf{e}_j$$

for any $k \in \{1, \dots, m\}$. By a similar argument we know that:

$$\mathbf{A}_{ij} = \min_{l \in \{1,\dots,m\}} \max_{k \in \{1,\dots,n\}} \mathbf{A}_{kl}$$

implies that $\mathbf{A}_{ij} \leq \mathbf{A}_{il}$ for any $l \in \{1, \dots, n\}$. To see this remember:

$$R_{\min} = \{r_k^* = \min\{\mathbf{A}_{kl} : l \in \{1, \dots, n\}\} : k \in \{1, \dots, m\}\}$$

and $\mathbf{A}_{ij} \in R_{\min}$ by construction. Thus it follows that:

$$\mathbf{e}_i^T \mathbf{A} \mathbf{e}_j \leq \mathbf{e}_i^T \mathbf{A} \mathbf{e}_l$$

for any $l \in \{1, \dots, n\}$. Thus $(\mathbf{e}_i, \mathbf{e}_j)$ is an equilibrium solution. This completes the proof. □

THEOREM 6.3. *Suppose that $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A})$ be a zero-sum two player game. Let $(\mathbf{e}_i, \mathbf{e}_j)$ be an equilibrium strategy pair for this game. Show that if $(\mathbf{e}_k, \mathbf{e}_l)$ is a second equilibrium strategy pair, then*

$$\mathbf{A}_{ij} = \mathbf{A}_{kl} = \mathbf{A}_{il} = \mathbf{A}_{kj}$$

EXERCISE 42. Prove Theorem 6.3. [Hint: This proof is in Morris, Page 36.]

DEFINITION 6.4 (Saddle Point). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A})$ be a zero-sum two player game. If $(\mathbf{e}_i, \mathbf{e}_j)$ is an equilibrium, then it is called a *saddle point*.

2. Zero-Sum Games without Saddle Points

REMARK 6.5. It is important to realize that not all games have saddle points of the kind found in Example 6.2. The easiest way to show this is true is to illustrate it with an example.

EXAMPLE 6.6. In August 1944 after the invasion of Normandy, the Allies broke out of their beachhead at Avranches, France and headed into the main part of the country (see Figure 6.2). The German General von Kluge, commander of the ninth army, faced two options:

(1) Stay and attack the advancing Allied armies.
(2) Withdraw into the mainland and regroup.

Simultaneously, General Bradley, commander of the Allied ground forces faced a similar set of options regarding the German ninth army:

(1) Reinforce the gap created by troop movements at Avranches

Figure 6.2. In August 1944, the allies broke out of their beachhead at Avranches and started heading in toward the mainland of France. At this time, General Bradley was in command of the Allied forces. He faced General von Kluge of the German ninth army. Each commander faced several troop movement choices. These choices can be modeled as a game.

(2) Send his forces east to cut-off a German retreat
(3) Do nothing and wait a day to see what the adversary did.

We can see that the player set can be written as $\mathbf{P} = \{\text{Bradley}, \text{von Kluge}\}$. The strategy sets are:

$$\Sigma_1 = \{\text{Reinforce the gap}, \text{Send forces east}, \text{Wait}\}$$
$$\Sigma_2 = \{\text{Attack}, \text{Retreat}\}$$

In real life, there were no pay-off values (as there were in the Battle of the Bismark Sea), however General Bradley's diary indicates the scenarios he preferred in order. There are six possible scenarios; i.e., there are six elements in $\Sigma = \Sigma_1 \times \Sigma_2$. Bradley ordered them from most to least preferable and using this ranking, we can construct the game matrix shown in Figure 6.3. Notice that the maximin value of the rows is *not* equal to the minimax value of the columns. This is indicative of the fact that there is not a pair of strategies that form an equilibrium for this game.

To see this, suppose that von Kluge plays his minimax strategy to retreat then Bradley would do better *not* play his maximin strategy (wait) and instead move east, cutting of von Kluge's retreat, thus obtaining a payoff of $(5, -5)$. But von Kluge would realize this and deduce that he should attack, which would yield a payoff of $(1, -1)$. However, Bradley could deduce this as well and would know to play his maximin strategy (wait), which yields payoff $(6, -6)$. However, von Kluge would realize that this would occur in which case he would decide to retreat yielding a payoff of $(4, -4)$. The cycle then repeats. This is illustrated in Figure 6.4.

	von Kluge's Strategies		**Row Min**
Bradley's Strategy	**Attack**	**Retreat**	—
Reinforce Gap	2	3	2
Move East	1	$\boxed{5}$	1
Wait	6	$\boxed{4}$	$\boxed{4}$
Column Max	6	$\boxed{5}$	maxmin = 4
			minmax = 5

Figure 6.3. At the battle of Avranches General Bradley and General von Kluge faced off over the advancing Allied Army. Each had decisions to make. This game matrix shows that this game has *no* saddle point solution. There is no position in the matrix where an element is simultaneously the maximum value in its column and the minimum value in its row.

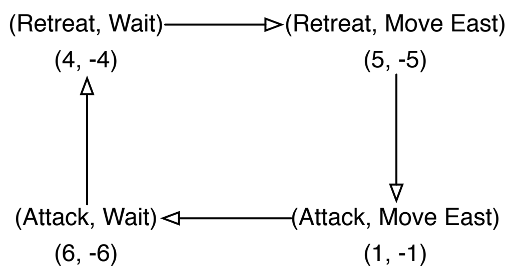


Figure 6.4. When von Kluge chooses to retreat, Bradley can benefit by playing a strategy different from his maximin strategy and he moves east. When Bradley does this, von Kluge realizes he could benefit by attacking and not playing his maximin strategy. Bradley realizes this and realizes he should play his maximin strategy and wait. This causes von Kluge to realize that he should retreat, causing this cycle to repeat.

Definition 6.7 (Game Value). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A})$ be a zero-sum game. If there exists a strategy pair $(\mathbf{e}_i, \mathbf{e}_j)$ so that:

$$\max_{k \in \{1,\dots,m\}} \min_{l \in \{1,\dots,n\}} \mathbf{A}_{kl} = \min_{l \in \{1,\dots,n\}} \max_{k \in \{1,\dots,m\}} \mathbf{A}_{kl}$$

then:

$$V_{\mathcal{G}} = \mathbf{e}_i^T \mathbf{A} \mathbf{e}_j \tag{6.12}$$

is the *value of the game*.

Remark 6.8. We will see that we can define the value of a zero-sum game even when there is no equilibrium point in strategies in Σ. Using Theorem 6.3 we can see that this value is unique, that is any equilibrium pair for a game will yield the same value for a zero-sum game. This is *not* the case in a general-sum game.

Exercise 43. Show that Rock-Paper-Scissors does not have a saddle-point strategy.

3. Mixed Strategies

Heretofore we have assumed that Player P_i will *deterministically* choose a strategy in Σ_i. It's possible, however, that Player P_i might choose a strategy at random. In this case, we assign probability to each strategy in Σ_i.

Definition 6.9 (Mixed Strategy). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ be a game in normal form with $\mathbf{P} = \{P_1, \dots, P_N\}$. A mixed strategy for Player $P_i \in \mathbf{P}$ is a discrete probability distribution function ρ^i defined over the sample space Σ. That is, we can define a discrete probability space $(\Sigma_i, \mathcal{F}_{\Sigma_i}, \rho^i)$ where Σ_i is the discrete sample space, $\mathcal{F}_{\Sigma_i}$ is the power set of Σ_i and ρ^i is the discrete probability function that assigns probabilities to events in $\mathcal{F}_{\Sigma_i}$.

Remark 6.10. We assume that players choose their mixed strategies independently. Thus we can compute the probability of a strategy element $(\sigma^1, \dots, \sigma^N) \in \Sigma$ as:

$$\rho(\sigma^1, \dots, \sigma^N) = \rho^1(\sigma^1)\rho^2(\sigma^2)\cdots\rho^N(\sigma^n) \tag{6.13}$$

Using this, we can define a discrete probability distribution over the sample space Σ as: $(\Sigma, \mathcal{F}_\Sigma, \rho)$. Define Π_i as a random variable that maps Σ into $\mathbb{R}$ so that Π_i returns the payoff to Player P_i as a result of the random outcome $(\sigma^1, \dots, \sigma^N)$. Therefore, the expected payoff for Player P_i for a given mixed strategy $(\rho^1, \dots, \rho^N)$ is given as:

$$\mathbb{E}(\Pi_i) = \sum_{\sigma^1\in\Sigma_1}\sum_{\sigma^2\in\Sigma_2}\cdots\sum_{\sigma^N\in\Sigma_N} \pi_i(\sigma^1, \dots, \sigma^n)\rho^1(\sigma^1)\rho^2(\sigma^2)\cdots\rho^N(\sigma^N)$$

Example 6.11. Consider the Rock-Paper-Scissors Game. The payoff matrix for Player 1 is given in Figure 6.5: Suppose that each strategy is chosen with probability $\frac{1}{3}$ by each

	Rock	**Paper**	**Scissors**
Rock	0	-1	1
Paper	1	0	-1
Scissors	-1	1	0

Figure 6.5. The payoff matrix for Player P_1 in Rock-Paper-Scissors. This payoff matrix can be derived from Figure 4.5.

player. Then the expected payoff to Player P_1 with this strategy is:

$$\begin{aligned}\mathbb{E}(\pi_1) = &\left(\frac{1}{3}\right)\left(\frac{1}{3}\right)\pi_1(\text{Rock}, \text{Rock}) + \left(\frac{1}{3}\right)\left(\frac{1}{3}\right)\pi_1(\text{Rock}, \text{Paper}) + \\ &\left(\frac{1}{3}\right)\left(\frac{1}{3}\right)\pi_1(\text{Rock}, \text{Scissors}) + \left(\frac{1}{3}\right)\left(\frac{1}{3}\right)\pi_1(\text{Paper}, \text{Rock}) + \\ &\left(\frac{1}{3}\right)\left(\frac{1}{3}\right)\pi_1(\text{Paper}, \text{Paper}) + \left(\frac{1}{3}\right)\left(\frac{1}{3}\right)\pi_1(\text{Paper}, \text{Scissors}) + \\ &\left(\frac{1}{3}\right)\left(\frac{1}{3}\right)\pi_1(\text{Scissors}, \text{Rock}) + \left(\frac{1}{3}\right)\left(\frac{1}{3}\right)\pi_1(\text{Scissors}, \text{Paper}) + \\ &\left(\frac{1}{3}\right)\left(\frac{1}{3}\right)\pi_1(\text{Scissors}, \text{Scissors}) = 0\end{aligned}$$

We can likewise compute the same value for $\mathbb{E}(\pi_2)$ for Player P_2.

3.1. Mixed Strategy Vectors.

Definition 6.12 (Mixed Strategy Vector). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ be a game in normal form with $\mathbf{P} = \{P_1, \dots, P_N\}$. Let $\Sigma_i = \{\sigma^i_1, \dots, \sigma^i_{n_i}\}$. To any mixed strategy for Player P_i we may associate a vector $\mathbf{x}^i = [x^i_1, \dots, x^i_{n_i}]^T$ provided that it satisfies the properties:

(1) $x^i_j \geq 0$ for $j = 1, \dots, n_i$
(2) $\sum_{j=1}^{n_i} x^i_j = 1$

These two properties ensure we are defining a mathematically correct probability distribution over the strategies set Σ_i.

Definition 6.13 (Player Mixed Strategy Space). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ be a game in normal form with $\mathbf{P} = \{P_1, \dots, P_N\}$. Let $\Sigma_i = \{\sigma^i_1, \dots, \sigma^i_{n_i}\}$. Then the set:

$$\Delta_{n_i} = \left\{ [x_1, \dots, x_{n_i}]^T \in \mathbb{R}^{n \times 1} : \sum_{i=1}^{n_i} x_i = 1; x_i \geq 0, i = 1, \dots, n_i \right\} \tag{6.14}$$

is the *mixed strategy space* in n_i dimensions for Player P_i.

Remark 6.14. There is a pleasant geometry to the space Δ_n (sometimes called a *simplex*). In three dimensions, for example, the space is the face of a tetrahedron. (See Figure 6.6.)

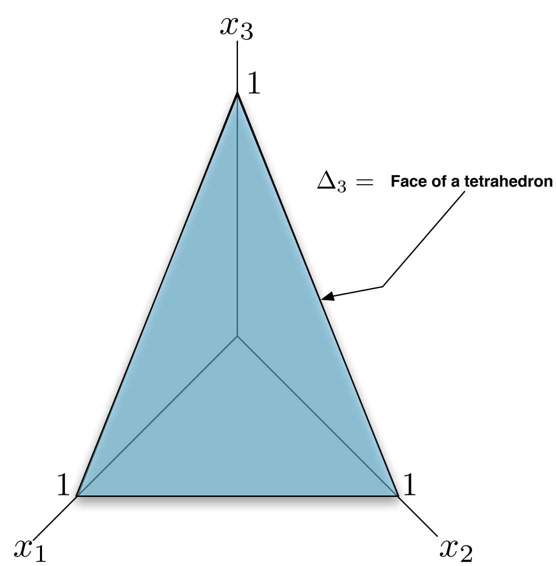


Figure 6.6. In three dimensional space Δ_3 is the face of a tetrahedron. In four dimensional space, it would be a tetrahedron, which would itself be the face of a four dimensional object.

Definition 6.15 (Pure Strategy). Let Σ_i be the strategy set for Player P_i in a game. If $\Sigma_i = \{\sigma^i_1, \dots, \sigma^i_{n_i}\}$, then $\mathbf{e}_j \in \Delta_{n_i}$ (for $j = 1, \dots, n_i$). These standard basis vectors are the *pure strategies* in Δ_{n_i} and $\mathbf{e}_j$ corresponds to a pure strategy choice $\sigma^i_j \in \Sigma_i$.

Definition 6.16 (Mixed Strategy Space). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ be a game in normal form with $\mathbf{P} = \{P_1, \dots, P_N\}$. Let $\Sigma_i = \{\sigma_1^i, \dots, \sigma_{n_i}^i\}$. Then the *mixed strategy space* for the game $\mathcal{G}$ is the set:

$$\Delta = \Delta_{n_1} \times \Delta_{n_2} \times \cdots \times \Delta_{n_N} \tag{6.15}$$

Definition 6.17 (Mixed Strategy Payoff Function). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ be a game in normal form with $\mathbf{P} = \{P_1, \dots, P_N\}$. Let $\Sigma_i = \{\sigma_1^i, \dots, \sigma_{n_i}^i\}$. The expected payoff can be written in terms of a tuple of mixed strategy vectors $(\mathbf{x}^1, \dots, \mathbf{x}^N)$:

$$u_i(\mathbf{x}^1, \dots, \mathbf{x}^N) = \sum_{i_1=1}^{n_1} \sum_{i_2=1}^{n_2} \cdots \sum_{i_N=1}^{n_N} \pi_i(\sigma_{i_1}^1, \dots, \sigma_{i_N}^n)\mathbf{x}_{i_1}^1 \mathbf{x}_{i_2}^2 \cdots \mathbf{x}_{i_N}^N \tag{6.16}$$

Here $\mathbf{x}_i^j$ is the i^{th} element of vector $\mathbf{x}^j$. The function $u_i : \Delta \to \mathbb{R}$ defined in Equation 6.16 is the *mixed strategy payoff function* for Player P_i. (Note: This notation is adapted from [**Wei97**].)

Example 6.18. For Rock-Paper-Scissors, since each player has 3 strategies, $n = 3$ and Δ_3 consists of those vectors $[x_1, x_2, x_3]^T$ so that $x_1, x_2, x_3 \geq 0$ and $x_1 + x_2 + x_3 = 1$. For example, the vectors:

$$\mathbf{x} = \mathbf{y} = \begin{bmatrix} \frac{1}{3} \\ \frac{1}{3} \\ \frac{1}{3} \end{bmatrix}$$

are mixed strategies for Players 1 and 2 respectively that instruct the players to play rock 1/3 of the time, paper 1/3 of the time and scissors 1/3 of the time.

Definition 6.19 (Nash Equilibrium). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ be a game in normal form with $\mathbf{P} = \{P_1, \dots, P_N\}$. Let $\Sigma_i = \{\sigma_1^i, \dots, \sigma_{n_i}^i\}$. A *Nash equilibrium* is a tuple of mixed strategies $(\mathbf{x}^{1^*}, \dots, \mathbf{x}^{N^*}) \in \Delta$ so that for all $i = 1, \dots, N$:

$$u_i(\mathbf{x}^{1^*}, \dots, \mathbf{x}^{i^*}, \dots, \mathbf{x}^{N^*}) \geq u_i(\mathbf{x}^{1^*}, \dots, \mathbf{x}^{i}, \dots, \mathbf{x}^{N^*}) \tag{6.17}$$

for all $\mathbf{x}^i \in \Delta_{n_i}$

Remark 6.20. What Definition 6.19 says is that a tuple of mixed strategies $(\mathbf{x}^{1^*}, \dots, \mathbf{x}^{N^*})$ is a Nash equilibrium if *no player* has any reason to deviate *unilaterally* from her mixed strategy.

Remark 6.21 (Notational Remark). In many texts, it becomes cumbersome in N player games to denote the mixed strategy tuple $(\mathbf{x}^1, \dots, \mathbf{x}^N)$ especially when (as in Definition 6.19) you are only interested in one player (Player P_i). To deal with this, textbooks sometimes adopt the notation $(\mathbf{x}^i, \mathbf{x}^{-i})$. Here $\mathbf{x}^i$ is the mixed strategy for Player P_i) while $\mathbf{x}^{-i}$ denotes the mixed strategy tuple for the other Players (who are not Player P_i). When expressed this way, Equation 6.17 is written as:

$$u_i(\mathbf{x}^{i^*}, \mathbf{x}^{-i^*}) \geq u_i(\mathbf{x}^i, \mathbf{x}^{-i^*})$$

for all $i = 1, \dots, N$. While notationally convenient, we will restrict our attention to two player games, so this will generally not be necessary.

4. Mixed Strategies in Matrix Games

PROPOSITION 6.22. *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a two-player matrix game. Let $\Sigma = \Sigma_1 \times \Sigma_2$ where $\Sigma_1 = \{\sigma_1^1, \dots, \sigma_m^1\}$ and $\Sigma_2 = \{\sigma_1^2, \dots, \sigma_n^2\}$. Let $\mathbf{x} \in \Delta_m$ and $\mathbf{y} \in \Delta_n$ be mixed strategies for Players 1 and 2 respectively. Then:*

$$u_1(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{A}\mathbf{y} \tag{6.18}$$

$$u_2(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{B}\mathbf{y} \tag{6.19}$$

PROOF. For simplicity, let $\mathbf{x} = [x_1, \dots, x_m]^T$ and $\mathbf{y} = [y_1, \dots, y_n]^T$. We know that $\pi_1(\sigma_i^1, \sigma_j^2) = \mathbf{A}_{ij}$. Simple matrix multiplication yields:

$$\mathbf{x}^T\mathbf{A} = \begin{bmatrix}\mathbf{x}^T\mathbf{A}_{\cdot 1} & \cdots & \mathbf{x}^T\mathbf{A}_{\cdot n}\end{bmatrix}$$

That is, $\mathbf{x}^T\mathbf{A}$ is a row vector whose j^{th} element is $\mathbf{x}^T\mathbf{A}_{\cdot j}$. For fixed j we have:

$$\mathbf{x}^T\mathbf{A}_{\cdot j} = x_1\mathbf{A}_{1j} + x_2\mathbf{A}_{2j} + \cdots + x_m\mathbf{A}_{mj} = \sum_{i=1}^{m} \pi_1(\sigma_i^1, \sigma_j^2)x_i$$

From this we can conclude that:

$$\mathbf{x}^T\mathbf{A}\mathbf{y} = \begin{bmatrix}\mathbf{x}^T\mathbf{A}_{\cdot 1} & \cdots & \mathbf{x}^T\mathbf{A}_{\cdot n}\end{bmatrix}\begin{bmatrix}y_1 \\ y_2 \\ \vdots \\ y_n\end{bmatrix}$$

This simplifies to:

$$\begin{aligned}\mathbf{x}^T\mathbf{A}_{\cdot 1}y_1 + \cdots + \mathbf{x}^T\mathbf{A}_{\cdot n}y_n = \\ (x_1\mathbf{A}_{11} + x_2\mathbf{A}_{21} + \cdots + x_m\mathbf{A}_{m1})\,y_1 + \cdots + (x_1\mathbf{A}_{1n} + x_2\mathbf{A}_{2n} + \cdots + x_m\mathbf{A}_{mn})\,y_m\end{aligned} \tag{6.20}$$

Distributing multiplication through, we can simplify Equation 6.20 as:

$$\mathbf{x}^T\mathbf{A}\mathbf{y} = \sum_{i=1}^{m}\sum_{j=1}^{n} A_{ij}x_iy_j = \sum_{i=1}^{m}\sum_{j=1}^{n} \pi_1(\sigma_i^1, \sigma_j^2)x_iy_j = u_1(\mathbf{x}, \mathbf{y}) \tag{6.21}$$

A similar argument shows that $u_2(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{B}\mathbf{y}$. This completes the proof. □

EXERCISE 44. Show explicitly that $u_2(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{B}\mathbf{y}$ as we did in the previous proof.

5. Dominated Strategies and Nash Equilibria

DEFINITION 6.23 (Weak Dominance). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ be a game in normal form with $\mathbf{P} = \{P_1, \dots, P_N\}$. Let $\Sigma_i = \{\sigma_1^i, \dots, \sigma_{n_i}^i\}$. A mixed strategy $\mathbf{x}^i \in \Delta_{n_i}$ for Player P_i *weakly dominates* another strategy $\mathbf{y}^i \in \Delta_{n_i}$ for Player P_i if *for all* mixed strategies $\mathbf{z}^{-i}$ we have:

$$u_i(\mathbf{x}^i, \mathbf{z}^{-i}) \geq u_i(\mathbf{y}^i, \mathbf{z}^{-i}) \tag{6.22}$$

and for *at least one* $\mathbf{z}^{-i}$ the inequality in Equation 6.22 is strict.

DEFINITION 6.24 (Strict Dominance). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ be a game in normal form with $\mathbf{P} = \{P_1, \dots, P_N\}$. Let $\Sigma_i = \{\sigma_1^i, \dots, \sigma_{n_i}^i\}$. A mixed strategy $\mathbf{x}^i \in \Delta_{n_i}$ for Player P_i *stictly dominates* another strategy $\mathbf{y}^i \in \Delta_{n_i}$ for Player P_i if *for all* mixed strategies $\mathbf{z}^{-\mathbf{i}}$ we have:

$$u_i(\mathbf{x}^i, \mathbf{z}^{-i}) > u_i(\mathbf{y}^i, \mathbf{z}^{-i}) \tag{6.23}$$

DEFINITION 6.25 (Dominated Strategy). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ be a game in normal form with $\mathbf{P} = \{P_1, \dots, P_N\}$. Let $\Sigma_i = \{\sigma_1^i, \dots, \sigma_{n_i}^i\}$. A strategy $\mathbf{x}^i \in \Delta_{n_i}$ for Player P_i is said to be *weakly (strictly) dominated* if there is a strategy $\mathbf{y}^i \in \Delta_{n_i}$ that weakly (strictly) dominates $\mathbf{x}^i$.

REMARK 6.26. In a two player matrix game $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$ a mixed strategy $\mathbf{x} \in \Delta_m$ for Player 1 weakly dominates a strategy $\mathbf{y} \in \Delta_m$ if for all $\mathbf{z} \in \Delta_n$ (mixed strategies for Player 2) we have:

$$\mathbf{x}^T\mathbf{A}\mathbf{z} \geq \mathbf{y}^T\mathbf{A}\mathbf{z} \tag{6.24}$$

and the inequality is strict for at least one $\mathbf{z} \in \Delta_n$. If $\mathbf{x}$ strictly dominates $\mathbf{y}$ then we have:

$$\mathbf{x}^T\mathbf{A}\mathbf{z} > \mathbf{y}^T\mathbf{A}\mathbf{z} \tag{6.25}$$

for all $\mathbf{z} \in \Delta_n$.

EXERCISE 45. For a two player matrix game, write what it means for a strategy $\mathbf{y} \in \Delta_n$ for Player 2 to weakly dominate a strategy $\mathbf{x}$. Also write what it means if $\mathbf{y}$ strictly dominates $\mathbf{x}$. [Hint: Remember, Player 2 multiplies on the right hand side of the payoff matrix. Also, you'll need to use $\mathbf{B}$.]

EXAMPLE 6.27 (Prisoner's Dilemma). The following example is called *Prisoner's Dilemma* and is a classic example in Game Theory. Two prisoner's Bonnie and Clyde commit a bank robbery. They stash the cash and are driving around wondering what to do next when they are pulled over and arrested for a weapons violation. The police suspect Bonnie and Clyde of the bank robbery, but do not have any hard evidence. They separate the prisoners and offer them the following options to Bonnie:

(1) If neither Bonnie nor Clyde confess, they will go to prison for 1 year on the weapons violation.
(2) If Bonnie confesses, but Clyde does not, then Bonnie can go free while Clyde will go to jail for 10 years.
(3) If Clyde confesses and Bonnie does not, then Bonnie will go to jail for 10 years while Clyde will go free.
(4) If both Bonnie and Clyde confess, then they will go to jail for 5 years.

A similar offer is made to Clyde. The following two-player matrix game describes the scenario: $\mathbf{P} = \{\text{Bonnie}, \text{Clyde}\}$; $\Sigma_1 = \Sigma_2 = \{\text{Don't Confess}, \text{Confess}\}$. The matrices for this game are given below:

$$\mathbf{A} = \begin{bmatrix} -1 & -10 \\ 0 & -5 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} -1 & 0 \\ -10 & -5 \end{bmatrix}$$

Here payoffs are given in negative years (for years lost to prison). Bonnie's matrix is $\mathbf{A}$ and Clyde's matrix is $\mathbf{B}$. The rows (columns) correspond to the strategies Don't Confess and Confess. Thus, we see that if Bonnie does not confess and Clyde does (row 1, column 2), then Bonnie loses 10 years and Clyde loses 0 years.

We can show that the strategy Confess dominates Don't Confess for Bonnie. Pure strategies correspond to standard basis vectors. Thus we're claiming that $\mathbf{e}_2$ strictly dominates $\mathbf{e}_1$ for Bonnie. We can use remark 6.26 to see that we must show:

$$\mathbf{e}_2^T\mathbf{A}\mathbf{z} > \mathbf{e}_1^T\mathbf{A}\mathbf{z} \tag{6.26}$$

We know that $\mathbf{z}$ is a mixed strategy. That means that:

$$\mathbf{z} = \begin{bmatrix} z_1 \\ z_2 \end{bmatrix}$$

and $z_1 + z_2 = 1$ and $z_1, z_2 \geq 0$. For simplicity, let's define:

$$\mathbf{z} = \begin{bmatrix} z \\ (1-z) \end{bmatrix}$$

with $z \geq 0$. We know that:

$$\mathbf{e}_2^T\mathbf{A} = \begin{bmatrix} 0 & 1 \end{bmatrix} \begin{bmatrix} -1 & -10 \\ 0 & -5 \end{bmatrix} = \begin{bmatrix} 0 & -5 \end{bmatrix}$$

$$\mathbf{e}_1^T\mathbf{A} = \begin{bmatrix} 1 & 0 \end{bmatrix} \begin{bmatrix} -1 & -10 \\ 0 & -5 \end{bmatrix} = \begin{bmatrix} -1 & -10 \end{bmatrix}$$

Then:

$$\mathbf{e}_2^T\mathbf{A}\mathbf{z} = \begin{bmatrix} 0 & -5 \end{bmatrix} \begin{bmatrix} z \\ (1-z) \end{bmatrix} = -5(1-z) = 5z - 5$$

$$\mathbf{e}_1^T\mathbf{A}\mathbf{z} = \begin{bmatrix} -1 & -10 \end{bmatrix} \begin{bmatrix} z \\ (1-z) \end{bmatrix} = -z - 10(1-z) = 9z - 10$$

There are many ways to show that when $z \in [0,1]$ that $5z - 5 > 9z - 10$, but the easiest way is to plot the two functions. This is shown in Figure 6.7. Another method is solving the inequalities.

Exercise 46. Show that Confess strictly dominates Don't Confess for Clyde in Example 6.27.

Remark 6.28. Strict dominance can be extremely useful for identifying pure Nash equilibria. This is especially true in matrix games. This is summarized in the following two theorems.

Theorem 6.29. *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a two player matrix game with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$. If*

$$\mathbf{e}_i^T\mathbf{A}\mathbf{e}_k > \mathbf{e}_j^T\mathbf{A}\mathbf{e}_k \tag{6.27}$$

for $k = 1, \dots, n$, then $\mathbf{e}_i$ strictly dominates $\mathbf{e}_j$ for Player 1.

Remark 6.30. We know that $\mathbf{e}_i^T\mathbf{A}$ is the i^{th} row of $\mathbf{A}$. Theorem 6.29 says: if every element in $\mathbf{A}_{i\cdot}$ (the i^{th} row of $\mathbf{A}$) is greater than its corresponding element in $\mathbf{A}_{j\cdot}$, (the j^{th} row of $\mathbf{A}$), then Player 1's i^{th} strategy strictly dominates Player 1's j^{th} strategy.

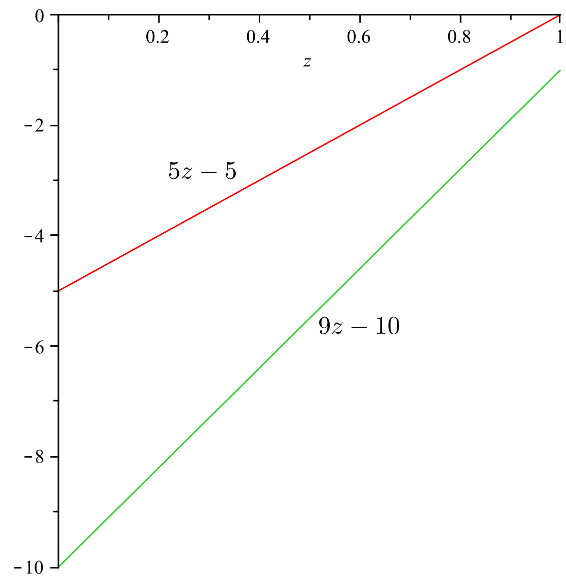


Figure 6.7. To show that Confess dominates over Don't Confess in Prisoner's dilemma for Bonnie, we can compute $\mathbf{e_1}^T\mathbf{Az}$ and $\mathbf{e}_2\mathbf{Az}$ for any arbitrary mixed strategy $\mathbf{z}$ for Clyde. The resulting payoff to Bonnie is $5z-5$ when she confesses and $9z-10$ when she doesn't confess. Here z is the probability that Clyde will not confess. The fact that $5z-5$ is greater than $9z-10$ at every point in the domain $z \in [0,1]$ demonstrates that Confess dominates Don't Confess for Bonnie.

Proof. For all $k = 1, \dots, n$ we know that:

$$\mathbf{e}_i^T\mathbf{A}\mathbf{e}_k > \mathbf{e}_j^T\mathbf{A}\mathbf{e}_k$$

Suppose that $z_1, \dots, z_n \in [0,1]$ with $z_1 + \dots + z_n = 1$. Then for each z_k we know that:

$$\mathbf{e}_i^T\mathbf{A}\mathbf{e}_k z_k > \mathbf{e}_j^T\mathbf{A}\mathbf{e}_k z_k$$

for $k = 1, \dots, n$. This implies that:

$$\mathbf{e}_i^T\mathbf{A}\mathbf{e}_1 z_1 + \dots + \mathbf{e}_i^T\mathbf{A}\mathbf{e}_n z_n > \mathbf{e}_j^T\mathbf{A}\mathbf{e}_1 z_1 + \dots + \mathbf{e}_j^T\mathbf{A}\mathbf{e}_n z_n$$

Factoring we have:

$$\mathbf{e}_i^T\mathbf{A}\left(z_1\mathbf{e}_1 + \dots + z_n\mathbf{e}_n\right) > \mathbf{e}_j^T\mathbf{A}\left(z_1\mathbf{e}_1 + \dots + z_n\mathbf{e}_n\right)$$

Define:

$$\mathbf{z} = z_1\mathbf{e}_1 + \dots + z_n\mathbf{e}_n = \begin{bmatrix} z_1 \\ \vdots \\ z_n \end{bmatrix}$$

Since the original $z_1, \dots, z_n$ where chosen arbitrarily from $[0,1]$ so that $z_1 + \dots z_n = 1$, we know that:

$$\mathbf{e}_i^T\mathbf{Az} > \mathbf{e}_j^T\mathbf{Az}$$

for all $z \in \Delta_n$. Thus $\mathbf{e}_i$ strictly dominates $\mathbf{e}_j$ by Definition 6.24. $\square$

REMARK 6.31. There is an analogous theorem for Player 2 which states that if each element of a column $\mathbf{B}_{\cdot i}$ is greater than the corresponding element in column $\mathbf{B}_{\cdot j}$, then $\mathbf{e}_i$ strictly dominates strategy $\mathbf{e}_j$ for Player 2.

EXERCISE 47. Using Theorem 6.29, state and prove an analogous theorem for Player 2.

REMARK 6.32. Theorem 6.29 can be generalized to N players. Unfortunately, the notation becomes complex and is outside the scope of this set of notes. It is worth knowing, however, that this is the case.

THEOREM 6.33. *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a two player matrix game. Suppose pure strategy $\mathbf{e}_j \in \Delta_m$ for Player 1 is strictly dominated by pure strategy $\mathbf{e}_i \in \Delta_m$. If $(\mathbf{x}^*, \mathbf{y}^*)$ is a Nash equilibrium, then $\mathbf{x}^*_j = 0$. Similarly, if pure strategy $\mathbf{e}_j \in \Delta_n$ for Player 2 is strictly dominated by pure strategy $\mathbf{e}_i \in \Delta_n$, then $\mathbf{y}^*_j = 0$*

PROOF. We will prove the theorem for Player 1; the proof for Player 2 is completely analogous. We will proceed by contradiction. Suppose that $\mathbf{x}^*_j > 0$. We know:

$$\mathbf{e}_i^T \mathbf{A}\mathbf{y}^* > \mathbf{e}_j^* \mathbf{A}\mathbf{y}^*$$

because $\mathbf{e}_i$ strictly dominates $\mathbf{e}_j$. We can express:

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{y} = \left(\mathbf{x}_1^*\mathbf{e}_1^T + \cdots + \mathbf{x}_i^*\mathbf{e}_i^T + \cdots + \mathbf{x}_j^*\mathbf{e}_j^T + \cdots + \mathbf{x}_m^*\mathbf{e}_m^T\right)\mathbf{A}\mathbf{y}^* \tag{6.28}$$

Here $\mathbf{x}_i^*$ is the i^{th} element of vector $\mathbf{x}^*$. Since $\mathbf{x}_j^* > 0$ we know that:

$$\mathbf{x}_j^*\mathbf{e}_i^T\mathbf{A}\mathbf{y}^* > \mathbf{x}_j^*\mathbf{e}_j^*\mathbf{A}\mathbf{y}^*$$

Thus we can conclude that:

$$\begin{multline}\left(\mathbf{x}_1^*\mathbf{e}_1^T + \cdots + \mathbf{x}_i^*\mathbf{e}_i^T + \cdots + \mathbf{x}_j^*\mathbf{e}_i^T + \cdots + \mathbf{x}_m^*\mathbf{e}_m^T\right)\mathbf{A}\mathbf{y}^* > \\ \left(\mathbf{x}_1^*\mathbf{e}_1^T + \cdots + \mathbf{x}_i^*\mathbf{e}_i^T + \cdots + \mathbf{x}_j^*\mathbf{e}_j^T + \cdots + \mathbf{x}_m^*\mathbf{e}_m^T\right)\mathbf{A}\mathbf{y}^*\end{multline} \tag{6.29}$$

If we define $\mathbf{z} \in \Delta_m$ so that:

$$\mathbf{z}_k = \begin{cases} \mathbf{x}_i^* + \mathbf{x}_j^* & k = i \\ 0 & k = j \\ \mathbf{x}_k & \text{else} \end{cases} \tag{6.30}$$

Then Equation 6.29 implies:

$$\mathbf{z}^T\mathbf{A}\mathbf{y}^* > \mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* \tag{6.31}$$

Thus, $(\mathbf{x}^*, \mathbf{y}^*)$ could not have been a Nash equilibrium. This completes the proof. □

EXAMPLE 6.34. We can use the two previous theorems to our advantage. Consider the Prisoner's Dilemma (Example 6.27). The payoff matrices (again) are:

$$\mathbf{A} = \begin{bmatrix} -1 & -10 \\ 0 & -5 \end{bmatrix}$$

$$\mathbf{B} = \begin{bmatrix} -1 & 0 \\ -10 & -5 \end{bmatrix}$$

For Bonnie Row (Strategy) 1 is strictly dominated by Row (Strategy) 2. Thus Bonnie will never play Strategy 1 (Don't Confess) in a Nash equilibrium. That is:

$$\mathbf{A}_{1\cdot} < \mathbf{A}_{2\cdot} \equiv \begin{bmatrix} -1 & -10 \end{bmatrix} < \begin{bmatrix} 0 & -5 \end{bmatrix}$$

Thus, we can consider a new game in which we remove this strategy for Bonnie (since Bonnie will never play this strategy). The new game has $\mathbf{P} = \{\text{Bonnie}, \text{Clyde}\}$, $\Sigma_1 = \{\text{Confess}\}$, $\Sigma_2 = \{\text{Don't Confess}, \text{Confess}\}$. The new game matrices are:

$$\mathbf{A}' = \begin{bmatrix} 0 & -5 \end{bmatrix}$$
$$\mathbf{B}' = \begin{bmatrix} -10 & -5 \end{bmatrix}$$

In this new game, we note that for Clyde (Player 2) Column (Strategy) 2 strictly dominates Column (Strategy 1). That is:

$$\mathbf{B}'_{\cdot 1} < \mathbf{B}'_{\cdot 2} \equiv -10 < -5$$

Clyde will never play Strategy 1 (Don't Confess) in a Nash equilibrium. We can construct a new game with $\mathbf{P} = \{\text{Bonnie}, \text{Clyde}\}$, $\Sigma_1 = \{\text{Confess}\}$, $\Sigma_2 = \{\text{Confess}\}$ and (trivial) payoff matrices:

$$\mathbf{A}'' = -5$$
$$\mathbf{B}'' = -5$$

In this game, there is only one Nash equilibrium in which both players confess. And this equilibrium is the Nash equilibrium of the original game.

Remark 6.35 (Iterative Dominance). A game whose Nash equilibrium is computed using the method from Example 6.34 in which strictly dominated are iteratively eliminated for the two players is said to be *solved by iterative dominance*. A game that can be analyzed in this way is said to be *strictly dominance solvable*.

Exercise 48. Consider the game matrix (matrices) 6.2. Show that this game is strictly dominance solvable. Recall that the game matrix is:

$$\mathbf{A} = \begin{bmatrix} -15 & -35 & 10 \\ -5 & 8 & 0 \\ -12 & -36 & 20 \end{bmatrix}$$

[Hint: Start with Player 2 (the Column Player) instead of Player 1. Note that Column 3 is *strictly dominated* by Column 1, so you can remove Column 3. Go from there. You can eliminate two rows (or columns) at a time if you want.]

6. The Minimax Theorem

In this section we come full circle back to zero-sum games. We show that there is a Nash equilibrium for every zero-sum game. The proof of this fact rests on three theorems.

Remark 6.36. Before proceeding, we'll recall the definition of a Nash equilibrium as it applies to a zero-sum game. A mixed strategy $(\mathbf{x}^*, \mathbf{y}^*) \in \Delta$ is a Nash equilibrium for a zero-sum game $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A})$ with $\mathbf{A} \in \mathbb{R}^{m \times n}$ if we have:

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* \geq \mathbf{x}^T\mathbf{A}\mathbf{y}^*$$

for all $\mathbf{x} \in \Delta_m$ and

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* \leq \mathbf{x}^{*T}\mathbf{A}\mathbf{y}$$

for all $\mathbf{y} \in \Delta_n$.

REMARK 6.37. Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A})$ be a zero-sum game with $\mathbf{A} \in \mathbb{R}^{m\times n}$. We can define a function $v_1 : \Delta_m \to \mathbb{R}$ as:

$$v_1(\mathbf{x}) = \min_{\mathbf{y}\in\Delta_n} \mathbf{x}^T\mathbf{A}\mathbf{y} = \min_{\mathbf{y}\in\Delta_n} \mathbf{x}^T\mathbf{A}_{\cdot 1}\mathbf{y}_1 + \cdots + \mathbf{x}^T\mathbf{A}_{\cdot n}\mathbf{y}_n \tag{6.32}$$

That is, given $\mathbf{x} \in \Delta_m$, we choose a vector $\mathbf{y}$ that *minimizes* $\mathbf{x}^T\mathbf{A}\mathbf{y}$. This value is the *best possible result* Player 1 can expect if he announces to Player 2 that he will play strategy $\mathbf{x}$. Player 1 then faces the problem that he would like to maximize this value by choosing $\mathbf{x}$ appropriately. That is, Player 1 hopes to solve the problem:

$$\max_{\mathbf{x}\in\Delta_m} v_1(\mathbf{x}) \tag{6.33}$$

Thus we have:

$$\max_{\mathbf{x}\in\Delta_m} v_1(\mathbf{x}) = \max_{\mathbf{x}} \min_{\mathbf{y}} \mathbf{x}^T\mathbf{A}\mathbf{y} \tag{6.34}$$

By a similar argument we can define a function $v_2 : \Delta_n \to \mathbb{R}$ as:

$$v_2(\mathbf{y}) = \max_{\mathbf{x}\in\Delta_m} \mathbf{x}^T\mathbf{A}\mathbf{y} = \max_{\mathbf{x}\in\Delta_m} \mathbf{x}_1\mathbf{A}_{1\cdot}\mathbf{y} + \cdots + \mathbf{x}_m\mathbf{A}_{m\cdot}\mathbf{y} \tag{6.35}$$

That is, given $\mathbf{y} \in \Delta_n$, we choose a vector $\mathbf{x}$ that *maximizes* $\mathbf{x}^T\mathbf{A}\mathbf{y}$. This value is the *best possible result* that Player 2 can expect if she announces to Player 1 that she will play strategy $\mathbf{y}$. Player 2 then faces the problem that she would like to minimize this value by choosing $\mathbf{y}$ appropriately. That is, Player 2 hopes to solve the problem:

$$\min_{\mathbf{y}\in\Delta_n} v_2(\mathbf{y}) \tag{6.36}$$

Thus we have:

$$\min_{\mathbf{y}\in\Delta_n} v_2(\mathbf{y}) = \min_{\mathbf{y}} \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y} \tag{6.37}$$

Note that this is the precise analogy in mixed strategies to the concept of a saddle point. The functions v_1 and v_2 are called the value functions for Player 1 and 2 respectively. The main problem we must tackle now is to determine whether these maximization and minimization problems can be solved.

LEMMA 6.38. *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A})$ be a zero-sum game with $\mathbf{A} \in \mathbb{R}^{m\times n}$. Then:*

$$\max_{\mathbf{x}\in\Delta_m} v_1(\mathbf{x}) \leq \min_{\mathbf{y}\in\Delta_n} v_2(\mathbf{y}) \tag{6.38}$$

EXERCISE 49. Prove Lemma 6.38. [Hint: Argue that for all $\mathbf{x} \in \Delta_m$ and for all $\mathbf{y} \in \Delta_n$ we know that $v_1(\mathbf{x}) \leq v_2(\mathbf{y})$ by showing that $v_2(\mathbf{y}) \geq \mathbf{x}^T\mathbf{A}\mathbf{y} \geq v_1(\mathbf{x})$. From this conclude that $\min_{\mathbf{y}} v_2(\mathbf{y}) \geq \max_{\mathbf{x}} v_1(\mathbf{x})$.]

THEOREM 6.39. *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A})$ be a zero-sum game with $\mathbf{A} \in \mathbb{R}^{m\times n}$. Then the following are equivalent:*

(1) *There is a Nash equilibrium $(\mathbf{x}^*, \mathbf{y}^*)$ for $\mathcal{G}$*

(2) *The following equation holds:*

$$v_1 = \max_{\mathbf{x}} \min_{\mathbf{y}} \mathbf{x}^T\mathbf{A}\mathbf{y} = \min_{\mathbf{y}} \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y} = v_2 \tag{6.39}$$

(3) *There exists a real number v and $\mathbf{x}^* \in \Delta_m$ and $\mathbf{y}^* \in \Delta_n$ so that:*
  (a) $\sum_i \mathbf{A}_{ij}\mathbf{x}_i^* \geq v$ *for* $j = 1, \dots, n$ *and*
  (b) $\sum_j \mathbf{A}_{ij}\mathbf{y}_j^* \leq v$ *for* $i = 1, \dots, m$

PROOF. (A version of this proof is given in [LR89], Appendix 2.)

($1 \implies 2$): Suppose that $(\mathbf{x}^*, \mathbf{y}^*) \in \Delta$ is an equilibrium pair. Let $v_2 = \min_{\mathbf{y}} \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y}$. By the definition of a minimum we know that:

$$v_2 = \min_{\mathbf{y}} \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y} \leq \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y}^*$$

The fact that for all $\mathbf{x} \in \Delta_m$:

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* \geq \mathbf{x}^T\mathbf{A}\mathbf{y}^*$$

implies that:

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* = \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y}^*$$

Thus we have:

$$v_2 = \min_{\mathbf{y}} \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y} \leq \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y}^* = \mathbf{x}^{*T}\mathbf{A}\mathbf{y}^*$$

Again, the fact that for all $\mathbf{y} \in \Delta_n$:

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* \leq \mathbf{x}^{*T}\mathbf{A}\mathbf{y}$$

implies that:

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* = \min_{\mathbf{y}} \mathbf{x}^{*T}\mathbf{A}\mathbf{y}$$

Thus:

$$v_2 = \min_{\mathbf{y}} \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y} \leq \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y}^* = \mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* = \min_{\mathbf{y}} \mathbf{x}^{*T}\mathbf{A}\mathbf{y}$$

Finally, by the definition of maximum we know that:

$$v_2 = \min_{\mathbf{y}} \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y} \leq \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y}^* = \mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* = \min_{\mathbf{y}} \mathbf{x}^{*T}\mathbf{A}\mathbf{y} \leq \max_{\mathbf{x}} \min_{\mathbf{y}} \mathbf{x}^{*T}\mathbf{A}\mathbf{y} = v_1 \tag{6.40}$$

when we let $v_1 = \max_{\mathbf{x}} \min_{\mathbf{y}} \mathbf{x}^{*T}\mathbf{A}\mathbf{y}$. By Lemma 6.38 we know that $v_1 \leq v_2$. Thus we have $v_2 \leq v_1$ and $v_1 \leq v_2$ so $v_1 = v_2$ as required.

($2 \implies 3$): Let $v = v_1 = v_2$ and let $\mathbf{x}^*$ be the vector that solves $\max_{\mathbf{x}} v_1(\mathbf{x})$ and $\mathbf{y}^*$ be the vector that solves $\min_{\mathbf{y}} v_2(\mathbf{y})$. For fixed j we know:

$$\sum_i \mathbf{A}_{ij}\mathbf{x}_i^* = \mathbf{x}^{*T}\mathbf{A}\mathbf{e}_j$$

By definition of minimum we know that:

$$\sum_i \mathbf{A}_{ij}\mathbf{x}_i^* = \mathbf{x}^{*T}\mathbf{A}\mathbf{e}_j \geq \min_{\mathbf{y}} \mathbf{x}^{*T}\mathbf{A}\mathbf{y}$$

We defined $\mathbf{x}^*$ so that it is the maximin value and thus:

$$\sum_i \mathbf{A}_{ij}\mathbf{x}_i^* = \mathbf{x}^{*T}\mathbf{A}\mathbf{e}_j \geq \min_{\mathbf{y}} \mathbf{x}^{*T}\mathbf{A}\mathbf{y} = \max_{\mathbf{x}} \min_{\mathbf{y}} \mathbf{x}^T\mathbf{A}\mathbf{y} = v = \min_{\mathbf{y}} \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y}$$

By a similar argument, we defined $\mathbf{y}^*$ so that it is the minimax value and thus:

$$\sum_i \mathbf{A}_{ij}\mathbf{x}_i^* = \mathbf{x}^{*T}\mathbf{A}\mathbf{e}_j \geq \min_{\mathbf{y}} \mathbf{x}^{*T}\mathbf{A}\mathbf{y} = \max_{\mathbf{x}} \min_{\mathbf{y}} \mathbf{x}^T\mathbf{A}\mathbf{y} = v =$$
$$\min_{\mathbf{y}} \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y} = \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y}^*$$

Finally, for fixed i we know that:

$$\sum_j \mathbf{A}_{ij}\mathbf{y}_j^* = \mathbf{e}_i^T\mathbf{A}\mathbf{y}^*$$

and thus we conclude:

$$\sum_i \mathbf{A}_{ij}\mathbf{x}_i^* = \mathbf{x}^{*T}\mathbf{A}\mathbf{e}_j \geq \min_{\mathbf{y}} \mathbf{x}^{*T}\mathbf{A}\mathbf{y} = \max_{\mathbf{x}} \min_{\mathbf{y}} \mathbf{x}^T\mathbf{A}\mathbf{y} = v =$$
$$\min_{\mathbf{y}} \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y} = \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y}^* \geq \mathbf{e}_i^T\mathbf{A}\mathbf{y}^* = \sum_j \mathbf{A}_{ij}\mathbf{y}_j^* \tag{6.41}$$

($3 \implies 1$): For any fixed j we know that:

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{e}_j \geq v$$

Thus if $y_1, \dots, y_n \in [0,1]$ and $y_1 + \dots + y_n = 1$ for each $j = 1, \dots, n$ we know that :

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{e}_j y_j \geq v y_j$$

Thus we can conclude that:

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{e}_1 y_1 + \dots + \mathbf{x}^{*T}\mathbf{A}\mathbf{e}_n y_n = \mathbf{x}^{*T}\mathbf{A}\left(\mathbf{e}_1 y_1 + \dots + \mathbf{e}_n y_n\right) \geq v$$

If

$$\mathbf{y} = \begin{bmatrix} y_1 \\ \vdots \\ y_n \end{bmatrix}$$

we can conclude that:

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{y} \geq v \tag{6.42}$$

for any $\mathbf{y} \in \Delta_n$. By a similar argument we know that:

$$\mathbf{x}^T\mathbf{A}\mathbf{y}^* \leq v \tag{6.43}$$

for all $\mathbf{x} \in \Delta_m$. From Equation 6.43 we conclude that:

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* \leq v \tag{6.44}$$

and from Equation 6.42 we conclude that:

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* \geq v \tag{6.45}$$

Thus $v = \mathbf{x}^{*T}\mathbf{A}\mathbf{y}^*$ and we know for all $\mathbf{x}$ and $\mathbf{y}$:

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* \geq \mathbf{x}^T\mathbf{A}\mathbf{y}^*$$
$$\mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* \leq \mathbf{x}^{*T}\mathbf{A}\mathbf{y}$$

Thus $(\mathbf{x}^*, \mathbf{y}^*)$ is a Nash equilibrium. This completes the proof. □

Remark 6.40. Theorem 6.39 does not assert the existence of a Nash equilibrium, it just provides insight into what happens if one exists. In particular, we know that the game has a unique value:

$$v = \max_{\mathbf{x}} \min_{\mathbf{y}} \mathbf{x}^T\mathbf{A}\mathbf{y} = \min_{\mathbf{y}} \max_{\mathbf{x}} \mathbf{x}^T\mathbf{A}\mathbf{y} \tag{6.46}$$

Proving the existence of a Nash equilibrium can be accomplished in several ways, the oldest of which uses a topological argument, which we present next. We can also use a linear programming based argument, which we will explore in the next chapter.

Lemma 6.41 (Brouwer Fixed Point Theorem). *Let Δ be the mixed strategy space of a two-player zero sum game. If $T : \Delta \to \Delta$ is continuous, then there exists a pair of strategies $(\mathbf{x}^*, \mathbf{y}^*)$ so that $T(\mathbf{x}^*, \mathbf{y}^*) = (\mathbf{x}^*, \mathbf{y}^*)$. That is $(\mathbf{x}^*, \mathbf{y}^*)$ is a **fixed point** of the mapping T.*

Remark 6.42. The proof of Brouwer's Fixed Point Theorem is well outside the scope of these notes. It is a deep theorem in topology. The interested reader should consult [**Mun00**] (Page 351 - 353).

Theorem 6.43 (Minimax Theorem). *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A})$ be a zero-sum game with $\mathbf{A} \in \mathbb{R}^{m\times n}$. Then there is a Nash equilibrium $(\mathbf{x}^*, \mathbf{y}^*)$.*

Nash's Proof. (A version of this proof is given in [**LR89**], Appendix 2.) Let $(\mathbf{x}, \mathbf{y}) \in \Delta$ be mixed strategies for Players 1 and 2. Define the following:

$$c_i(\mathbf{x}, \mathbf{y}) = \begin{cases} \mathbf{e}_i^T\mathbf{A}\mathbf{y} - \mathbf{x}^T\mathbf{A}\mathbf{y} & \text{if this quantity is positive} \\ 0 & \text{else} \end{cases} \tag{6.47}$$

$$d_j(\mathbf{x}, \mathbf{y}) = \begin{cases} \mathbf{x}^T\mathbf{A}\mathbf{y} - \mathbf{x}^T\mathbf{A}\mathbf{e}_j & \text{if this quantity is positive} \\ 0 & \text{else} \end{cases} \tag{6.48}$$

Let $T : \Delta \to \Delta$ where $T(\mathbf{x}, \mathbf{y}) = (\mathbf{x}', \mathbf{y}')$ so that for $i = 1, \dots, m$ we have:

$$\mathbf{x}_i' = \frac{\mathbf{x}_i + c_i(\mathbf{x}, \mathbf{y})}{1 + \sum_{k=1}^m c_k(\mathbf{x}, \mathbf{y})} \tag{6.49}$$

and for $j = 1, \dots, n$ we have:

$$\mathbf{y}_j' = \frac{\mathbf{y}_j + d_j(\mathbf{x}, \mathbf{y})}{1 + \sum_{k=1}^n d_k(\mathbf{x}, \mathbf{y})} \tag{6.50}$$

Since $\sum_i \mathbf{x}_i = 1$ we know that:

$$\mathbf{x}_1' + \cdots + \mathbf{x}_m' = \frac{\mathbf{x}_1 + \cdots + \mathbf{x}_m + \sum_{k=1}^m c_k(\mathbf{x}, \mathbf{y})}{1 + \sum_{k=1}^m c_k(\mathbf{x}, \mathbf{y})} = 1 \tag{6.51}$$

It is also clear that since $\mathbf{x}_i \geq 0$ for $i = 1, \dots, m$ we have $\mathbf{x}_i' \geq 0$. A similar argument shows that $\mathbf{y}_j' \geq 0$ for $j = 1, \dots, n$ and $\sum_j \mathbf{y}_j' = 1$. Thus T is a proper map from Δ to Δ. The fact that T is continuous follows from the continuity of the payoff function (See Exercise 51). We now show that $(\mathbf{x}, \mathbf{y})$ is a Nash equilibrium if and only if it is a fixed point of T.

To see this note that $c_i(\mathbf{x}, \mathbf{y})$ measures the amount that the pure strategy $\mathbf{e}_i$ is better than $\mathbf{x}$ as a response to $\mathbf{y}$. That is, if Player 2 decides to play strategy $\mathbf{y}$ then $c_i(\mathbf{x}, \mathbf{y})$ tells us if and how much playing pure strategy $\mathbf{e}_i$ is better than playing $\mathbf{x} \in \Delta_m$. Similarly, $d_j(\mathbf{x}, \mathbf{y})$ measures how much better $\mathbf{e}_j$ is as a response to Player 1's strategy $\mathbf{x}$ than strategy $\mathbf{y}$ for Player 2. Suppose that $(\mathbf{x}, \mathbf{y})$ is a Nash equilibrim. Then $c_i(\mathbf{x}, \mathbf{y}) = 0 = d_j(\mathbf{x}, \mathbf{y})$ for $i = 1, \dots, m$ and $j = 1, \dots, n$ by the definition of equilibrium. Thus $\mathbf{x}'_i = \mathbf{x}_i$ for $i = 1, \dots, m$ and $\mathbf{y}'_j = \mathbf{y}_j$ for $j = 1, \dots, n$ and thus $(\mathbf{x}, \mathbf{y})$ is a fixed point of T.

To show the converse, suppose that $(\mathbf{x}, \mathbf{y})$ is a fixed point of T. It suffices to show that there is at least one i so that $\mathbf{x}_i > 0$ and $c_i(\mathbf{x}, \mathbf{y}) = 0$. Clearly there is at least one i for which $\mathbf{x}_i > 0$. Note that:

$$\mathbf{x}^T\mathbf{A}\mathbf{y} = \sum_{i=1}^{m} \mathbf{x}_i \mathbf{e}_i^T \mathbf{A}\mathbf{y}$$

Thus, $\mathbf{x}^T\mathbf{A}\mathbf{y} < \mathbf{e}_i^T\mathbf{A}\mathbf{y}$ cannot hold for all $i = 1, \dots, m$ with $\mathbf{x}_i > 0$ (otherwise the previous equation would not hold). Thus for at least one i with $\mathbf{x}_i > 0$ we must have $c_i(\mathbf{x}, \mathbf{y}) = 0$. But for this i, the fact that $(\mathbf{x}, \mathbf{y})$ is a fixed point implies that:

$$\mathbf{x}_i = \frac{\mathbf{x}_i}{1 + \sum_{k=1}^{m} c_k(\mathbf{x}, \mathbf{y})} \tag{6.52}$$

This implies that $\sum_{k=1}^{m} c_k(\mathbf{x}, \mathbf{y}) = 0$. The fact that $c_k(\mathbf{x}, \mathbf{y}) \geq 0$ for all $k = 1, \dots, m$ implies that $c_k(\mathbf{x}, \mathbf{y}) = 0$. A similar argument can be shown for $\mathbf{y}$. Thus we know that $c_i(\mathbf{x}, \mathbf{y}) = 0 = d_j(\mathbf{x}, \mathbf{y})$ for $i = 1, \dots, m$ and $j = 1, \dots, n$ and thus $\mathbf{x}$ is at least as good a strategy for Player 1 responding to $\mathbf{y}$ as any $\mathbf{e}_i \in \Delta_m$; likewise $\mathbf{y}$ is at least as good a strategy for Player 2 responding to $\mathbf{x}$ as any $\mathbf{e}_j \in \Delta_n$. This fact implies that $(\mathbf{x}, \mathbf{y})$ is an equilibrium (see Exercise 50) for details).

Applying Lemma 6.41 (Brouwer's Fixed Point Theorem) we see that T must have a fixed point and thus every two player zero sum game has a Nash equilibrium. This completes the proof. $\square$

Exercise 50. Prove the following: $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A})$ be a zero-sum game with $\mathbf{A} \in \mathbb{R}^{m \times n}$. Let $\mathbf{x}^* \in \Delta_m$ and $\mathbf{y}^* \in \Delta_n$. If:

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* \geq \mathbf{e}_i^T\mathbf{A}\mathbf{y}^*$$

for all $i = 1, \dots, m$ and

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* \leq \mathbf{x}^{*T}\mathbf{A}\mathbf{e}_j$$

for all $j = 1, \dots, n$, then $(\mathbf{x}^*, \mathbf{y}^*)$ is an equilibrium.

Exercise 51. Verify that the function T in Theorem 6.43 is continuous.

7. Finding Nash Equilibria in Simple Games

It is relatively straightforward to find a Nash equilibrium in 2×2 zero-sum games, assuming that a saddle-point cannot be identified using the approach from Example 6.2. We illustrate the approach using *The Battle of Avranches*.

Example 6.44. Consider the Battle of Avranches (Example 6.6). The payoff matrix is:

$$\mathbf{A} = \begin{bmatrix} 2 & 3 \\ 1 & 5 \\ 6 & 4 \end{bmatrix}$$

Note first that Row 1 (Bradley' first strategy) is strictly dominated by Row 3 (Bradley's third strategy) and thus we can reduce the payoff matrix to:

$$\mathbf{A} = \begin{bmatrix} 1 & 5 \\ 6 & 4 \end{bmatrix}$$

Let's suppose that Bradley chooses a strategy:

$$\mathbf{x} = \begin{bmatrix} x \\ 1-x \end{bmatrix}$$

with $x \in [0,1]$. If Von Kluge chooses to Attack (Column 1), then Bradley's expected payoff will be:

$$\mathbf{x}^T\mathbf{A}\mathbf{e}_1 = \begin{bmatrix} x & 1-x \end{bmatrix} \begin{bmatrix} 1 & 5 \\ 6 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 0 \end{bmatrix} = x + 6(1-x) = -5x + 6$$

A similar argument shows that if Von Kluge chooses to Retreat (Column 2), then Bradley's expected payoff will be:

$$\mathbf{x}^T\mathbf{A}\mathbf{e}_2 = 5x + 4(1-x) = x + 4$$

We can visualize these strategies by plotting them (see Figure 6.8, left)). Plotting the expected payoff to Bradley by playing a mixed strategy $[x \;\; (1-x)]^T$ when Von Kluge plays pure strategies shows which strategy Von Kluge should pick. When $x \leq 1/3$, Von Kluge does better if he retreats because $x+4$ is *below* $-5x+6$. That is, the *best* Bradley can hope to get is $-5x+6$ if he announced to Von Kluge that he was playing $x \leq 1/3$.

On the other hand, if $x \geq 1/3$, then Von Kluge does better if he attacks because $-5x+6$ is below $x+4$. That is, the *best* Bradley can hope to get is $x+4$ if he tells Von Kluge that he is playing $x \geq 1/3$. Remember, Von Kluge wants to *minimize* the payoff to Bradley. The point at which Bradley does *best* (i.e., maximizes his expected payoff) comes at $x = 1/3$.

By a similar argument, we can compute the expected payoff to Von Kluge when he plays mixed strategy $[y \;\; (1-y)]^T$ and Bradley plays pure strategies. The expected payoff to Von Kluge when Bradley plays Row 1, is:

$$\mathbf{e}_1^T(-\mathbf{A})\mathbf{y} = -y - 5(1-y) = 4y - 5$$

When Bradley plays Row 2, the expected payoff to Von Kluge is:

$$\mathbf{e}_2^T(-\mathbf{A})\mathbf{y} = -6y - 4(1-y) = -2y - 4$$

We can plot these expressions (see Figure 6.8, right). When $y \leq 1/6$, Bradley does better if he choose Row 1 (Move East) while when $y \geq 1/6$, Bradley does best when he waits. Remember, Bradley is *minimizing* Von Kluge's payoff (since we are working with $-\mathbf{A}$). We know that Bradley cannot do any better than when he plays $\mathbf{x}^* = [1/3 \;\; 2/3]^T$. Similarly, Von Kluge cannot do any better than when he plays $\mathbf{y}^* = [1/6 \;\; 5/6]^T$. The pair $(\mathbf{x}^*, \mathbf{y}^*)$ is the Nash equilibrium for this problem.

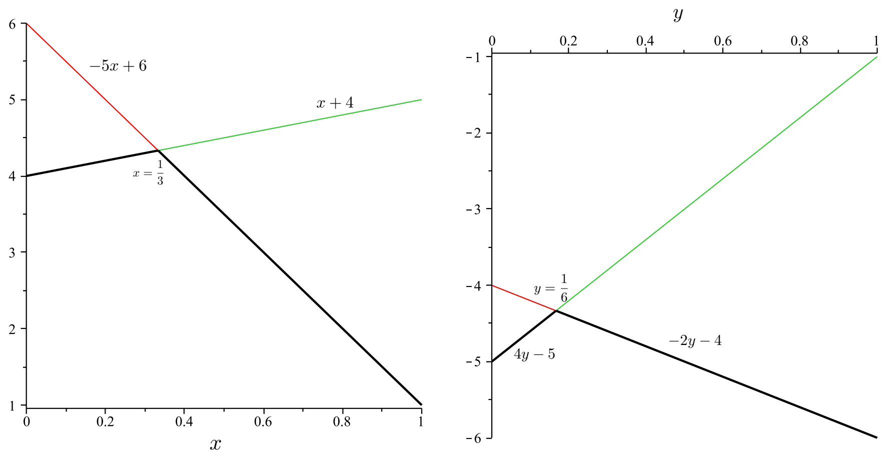


Figure 6.8. Plotting the expected payoff to Bradley by playing a mixed strategy $[x \;\; (1-x)]^T$ when Von Kluge plays pure strategies shows which strategy Von Kluge should pick. When $x \leq 1/3$, Von Kluge does better if he retreats because $x+4$ is *below* $-5x+6$. On the other hand, if $x \geq 1/3$, then Von Kluge does better if he attacks because $-5x+6$ is below $x+4$. Remember, Von Kluge wants to *minimize* the payoff to Bradley. The point at which Bradley does *best* (i.e., maximizes his expected payoff) comes at $x = 1/3$. By a similar argument, when $y \leq 1/6$, Bradley does better if he choose Row 1 (Move East) while when $y \geq 1/6$, Bradley does best when he waits. Remember, Bradley is *minimizing* Von Kluge's payoff (since we are working with $-\mathbf{A}$).

Often, any Nash equilibrium for a zero-sum game is called a saddle-point. To see why we called these points *saddle points*, consider Figure 6.9. This figure shows the payoff function for Player 1 as a function of x and y (from the example). This function is:

$$\begin{bmatrix} x & 1-x \end{bmatrix} \begin{bmatrix} 1 & 5 \\ 6 & 4 \end{bmatrix} \begin{bmatrix} y \\ 1-y \end{bmatrix} = -6yx + 2y + x + 4 \tag{6.53}$$

The figure is a hyperbolic saddle. In 3D space, it looks like a twisted combination of an upside down parabola (like the plot of $y = -x^2$ from high school algebra) and a right-side up parabola (like $y = x^2$ from high school algebra). Note that the maximum of one parabola and minimum of another parabola occur precisely at the point $(x, y) = (1/3, 1/5)$, the point in 2D space corresponding to this Nash equilibrium.

Exercise 52. Consider the following football game in Example 5.4. Ignoring the Blitz option for the defense, compute the Nash equilibrium strategy in terms of Running Plays, Passing Plays, Running Defense and Passing Defense.

Remark 6.45. The techniques discussed in Example 6.44 can be extended to cases when one player has 2 strategies and another player has more than 2 strategies, but these methods are not efficient for finding Nash equilibria in general. In the next chapter we will show how to find Nash equilibria for games by finding solving a specific simple optimization problem. This technique will work for general two player zero-sum games. We will also discuss the problem of finding Nash equilibria in two player general sum matrix games.

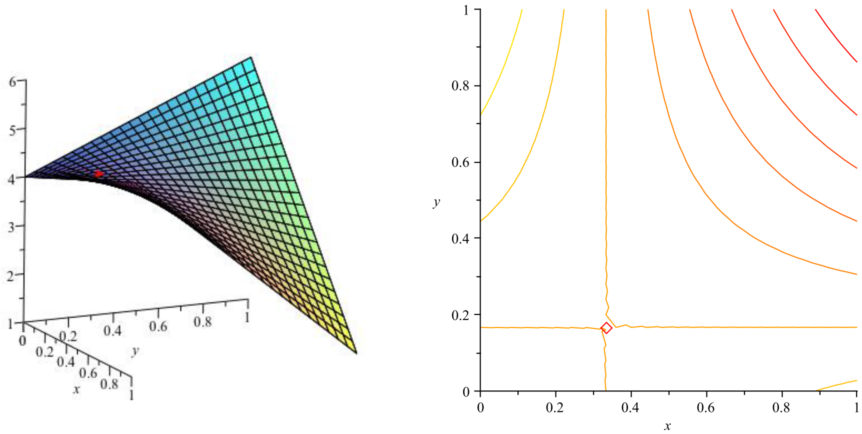


Figure 6.9. The payoff function for Player 1 as a function of x and y. Notice that the Nash equilibrium does in fact occur at a saddle point.

8. A Note on Nash Equilibria in General

Remark 6.46. The functions v_1 and v_2 defined in Remark 6.37 and used in the proof of Theorem 6.39 can be generalized to N player general sum games. The strategies that produce the values in these functions are called *best replies* and are used in proving the existence of Nash equilibria for general sum N player games.

Definition 6.47 (Player Best Response). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ be an N player game in normal form with $\Sigma_i = \{\sigma_1^i, \dots, \sigma_{n_i}^i\}$ and let Δ be the mixed strategy space for this game. If $\mathbf{y} \in \Delta$ is a mixed strategy for all players, then the *best reply* for Player P_i is the set:

$$B_i(\mathbf{y}) = \left\{\mathbf{x}^i \in \Delta_{n_i} : u_i(\mathbf{x}^i, \mathbf{y}^{-i}) \geq u_i(\mathbf{z}^i, \mathbf{y}^{-i}) \quad \forall \mathbf{z}^i \in \Delta_{n_i}\right\} \tag{6.54}$$

Recall $\mathbf{y}^{-i} = (\mathbf{y}^1, \dots, \mathbf{y}^{i-1}, \mathbf{y}^{i+1}, \dots, \mathbf{y}^N)$.

Remark 6.48. Thus if a Player P_i is confronted by some collection of strategies $\mathbf{y}^{-i}$, then the best thing he can do is to choose some strategy $\in B_i(\mathbf{y})$. (Here we assume that $\mathbf{y}$ is composed of $\mathbf{y}^{-i}$ and some arbitrary initial strategy $\mathbf{y}^i$ for Player P_i.) Clearly, $B_i : \Delta \to 2^{\Delta_{n_i}}$

Definition 6.49 (Best Response). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ be an N player game in normal form with $\Sigma_i = \{\sigma_1^i, \dots, \sigma_{n_i}^i\}$ and let Δ be the mixed strategy space for this game. The mapping $B : \Delta \to 2^{\Delta}$ given by:

$$B(\mathbf{x}) = B_1(\mathbf{x}) \times B_2(\mathbf{x}) \cdots \times B_N(\mathbf{x}) \tag{6.55}$$

is called the *best response mapping*.

Theorem 6.50. *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ be an N player game in normal form with $\Sigma_i = \{\sigma_1^i, \dots, \sigma_{n_i}^i\}$ and let Δ be the mixed strategy space for this game. The strategy $\mathbf{x}^* \in \Delta$ is a Nash equilibrium for $\mathcal{G}$ if and only if $\mathbf{x}^* \in B(\mathbf{x}^*)$.*

Proof. Suppose that $\mathbf{x}$ is a Nash equilibrium. Then for all $i = 1 \dots, N$:

$$u_i(\mathbf{x}^{i*}, \mathbf{x}^{-i*}) \geq u_i(\mathbf{z}, \mathbf{x}^{-i*})$$

for every $\mathbf{z} \in \Delta_{n_i}$. Thus:

$$\mathbf{x}^{i^*} \in \left\{\mathbf{x}^i \in \Delta_{n_i} : u_i(\mathbf{x}^i, \mathbf{x}^{-i}) \geq u_i(\mathbf{z}, \mathbf{y}^{-i}) \quad \forall \mathbf{z} \in \Delta_{n_i}\right\}$$

Thus $\mathbf{x}^{i^*} \in B_i(\mathbf{x}^{i^*})$. Since this holds for each $i = 1, \ldots, N$ it follows that $\mathbf{x}^* \in B(\mathbf{x}^*)$.

To prove the converse, suppose that $\mathbf{x}^* \in B(\mathbf{x}^*)$. Then for all $i = 1, \ldots, N$:

$$\mathbf{x}^{i^*} \in \left\{\mathbf{x}^i \in \Delta_{n_i} : u_i(\mathbf{x}^i, \mathbf{x}^{-i}) \geq u_i(\mathbf{z}, \mathbf{y}^{-i}) \quad \forall \mathbf{z} \in \Delta_{n_i}\right\}$$

But this implies that: for all $i = 1 \ldots, N$:

$$u_i(\mathbf{x}^{i^*}, \mathbf{x}^{-i^*}) \geq u_i(\mathbf{z}, \mathbf{x}^{-i^*})$$

for every $\mathbf{z} \in \Delta_{n_i}$. Thus it follows that $\mathbf{x}^{*i}$ is a Nash equilibrium. This completes the proof. □

Remark 6.51. What Theorem 6.50 shows is the in the N player general sum game setting, every Nash equilibrium is a kind of fixed point of the mapping $B : \Delta \to 2^{\Delta}$. This fact along with a more general topological *fixed point theorem* called Kakutani's Fixed Point Theorem is sufficient to show that there exists a Nash equilibrium for any general sum game. This was Nash's original proof for the following theorem:

Theorem 6.52 (Existence of Nash Equilibria). *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ be an N player game in normal form. Then $\mathcal{G}$ has at least one Nash equilibrium.*

Remark 6.53. The proof based on Kakutani's Fixed Point Theorem is neither useful nor satisfying. Nash realized this and constructed an alternate proof using Brouwer's Fixed Point theorem following the same steps we used to prove Theorem 6.43. We can generalize the proof of Theorem 6.43 by defining:

$$J_k^i(\mathbf{x}) = \max\left\{0, u_i(\mathbf{e}_k, \mathbf{x}^{-i}) - u_i(\mathbf{x}^i, \mathbf{x}^{-i})\right\} \tag{6.56}$$

The function $J_k^i(\mathbf{x})$ measures the benefit of changing to the pure strategy $\mathbf{e}_k$ for Player P_i when all other players hold their strategy fixed at $\mathbf{x}^{-i}$.

We can now define:

$$\mathbf{x}_j^{i\prime} = \frac{\mathbf{x}_j^i + J_j^i(\mathbf{x})}{1 + \sum_{k=1}^{n_i} J_k^i(\mathbf{x})} \tag{6.57}$$

Using this equation, we can construct a mapping $T : \Delta \to \Delta$ and show that every fixed point is a Nash Equilibrium. Using the Brouwer fixed point theorem, it then follows that a Nash equilibrium exists. Unfortunately, this is still not a very useful way to construct a Nash equilibrium.

In the next chapter we will explore this problem in depth for two player zero-sum games and then go on to explore the problem for two player general sum-games. The story of computing Nash equilibria takes on a life of its own and is an important study within *computational game theory* that has had a substantial impact on the literature in mathematical programming (optimization), computer science, and economics.

CHAPTER 7

An Introduction to Optimization and the Karush-Kuhn-Tucker Conditions

In this chapter we're going to take a detour into optimization theory. We'll need many of these results and definitions later when we tackle methods for solving two player zero and general sum games. Optimization is an exciting sub-discipline within applied mathematics! Optimization is all about making things better; this could mean helping a company make better decisions to maximize profit; helping a factory make products with less environmental impact; or helping a zoologist improve the diet of an animal. When we talk about optimization, we often use terms like *better* or *improvement*. It's important to remember that words like better can mean *more of something* (as in the case of profit) or *less of something* as in the case of waste. As we study linear programming, we'll quantify these terms in a mathematically precise way. For the time being, let's agree that when we optimize something we are trying to make some decisions that will make it better.

Example 7.1. Let's recall a simple optimization problem from differential calculus (Math 140): Goats are an environmentally friendly and inexpensive way to control a lawn when there are lots of rocks or lots of hills. (Seriously, both Google and some U.S. Navy bases use goats on rocky hills instead of paying lawn mowers!)

Suppose I wish to build a pen to keep some goats. I have 100 meters of fencing and I wish to build the pen in a rectangle with the largest possible area. How long should the sides of the rectangle be? In this case, making the pen *better* means making it have the largest possible area.

The problem is illustrated in Figure 7.1. Clearly, we know that:

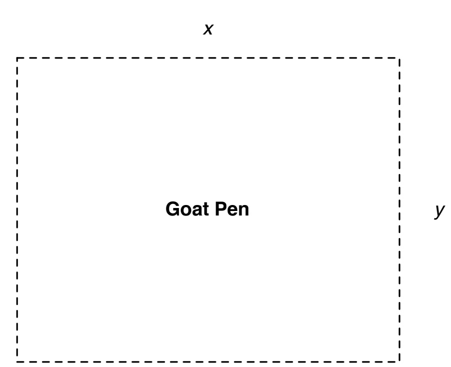


Figure 7.1. Goat pen with unknown side lengths. The objective is to identify the values of x and y that maximize the area of the pen (and thus the number of goats that can be kept).

$$2x + 2y = 100 \tag{7.1}$$

because $2x + 2y$ is the perimeter of the pen and I have 100 meters of fencing to build my pen. The area of the pen is $A(x, y) = xy$. We can use Equation 7.1 to solve for x in terms of y. Thus we have:

$$y = 50 - x \tag{7.2}$$

and $A(x) = x(50 - x)$. To maximize $A(x)$, recall we take the first derivative of $A(x)$ with respect to x, set this derivative to zero and solve for x:

$$\frac{dA}{dx} = 50 - 2x = 0; \tag{7.3}$$

Thus, $x = 25$ and $y = 50 - x = 25$. We further recall from basic calculus how to confirm that this is a maximum; note:

$$\left.\frac{d^2A}{dx^2}\right|_{x=25} = -2 < 0 \tag{7.4}$$

Which implies that $x = 25$ is a *local maximum* for this function. Another way of seeing this is to note that $A(x) = 50x - x^2$ is an "upside-down" parabola. As we could have guessed, a square will maximize the area available for holding goats.

Exercise 53. A canning company is producing canned corn for the holidays. They have determined that each family prefers to purchase their corn in units of 12 fluid ounces. Assuming that metal costs 1 cent per square inch and 1 fluid ounce is about 1.8 cubic inches, compute the ideal height and radius for a can of corn assuming that cost is to be minimized. [Hint: Suppose that our can has radius r and height h. The formula for the surface area of a can is $2\pi rh + 2\pi r^2$. Since metal is priced by the square inch, the cost is a function of the surface area. The volume of the can is $\pi r^2 h$ and is constrained. Use the same trick we did in the example to find the values of r and h that minimize cost.

1. A General Maximization Formulation

Let's take a more general look at the goat pen example. The area function is a mapping from $\mathbb{R}^2$ to $\mathbb{R}$, written $A : \mathbb{R}^2 \to \mathbb{R}$. The domain of A is the two dimensional space $\mathbb{R}^2$ and its range is $\mathbb{R}$.

Our objective in Example 7.1 is to maximize the function A by choosing values for x and y. In optimization theory, the function we are trying to maximize (or minimize) is called the *objective function*. In general, an objective function is a mapping $z : D \subseteq \mathbb{R}^n \to \mathbb{R}$. Here D is the domain of the function z.

Definition 7.2. Let $z : D \subseteq \mathbb{R}^n \to \mathbb{R}$. The point $\mathbf{x}^*$ is a *global maximum* for z if for all $\mathbf{x} \in D$, $z(\mathbf{x}^*) \geq z(\mathbf{x})$. A point $\mathbf{x}^* \in D$ is a *local maximum* for z if there is a set $S \subseteq D$ with $\mathbf{x}^* \in S$ so that for all $\mathbf{x} \in S$, $z(\mathbf{x}^*) \geq z(\mathbf{x})$.

Exercise 54. Using analogous reasoning write a definition for a global and local minimum. [Hint: Think about what a minimum means and find the correct direction for the $\geq$ sign in the definition above.]

In Example 7.1, we are constrained in our choice of x and y by the fact that $2x+2y = 100$. This is called a *constraint* of the optimization problem. More specifically, it's called an *equality constraint*. If we did not need to use all the fencing, then we could write the constraint as $2x+2y \leq 100$, which is called an *inequality constraint*. In complex optimization problems, we can have many constraints. The set of all points in $\mathbb{R}^n$ for which the constraints are true is called the *feasible set* (or feasible region). Our problem is to *decide* the best values of x and y to maximize the area $A(x,y)$. The variables x and y are called *decision variables*.

Let $z : D \subseteq \mathbb{R}^n \to \mathbb{R}$; for $i = 1, \dots, m$, $g_i : D \subseteq \mathbb{R}^n \to \mathbb{R}$; and for $j = 1, \dots, l$ $h_j : D \subseteq \mathbb{R}^n \to \mathbb{R}$ be functions. Then the general maximization problem with objective function $z(x_1, \dots, x_n)$ and *inequality constraints* $g_i(x_1, \dots, x_n) \leq b_i$ $(i = 1, \dots, m)$ and *equality constraints* $h_j(x_1, \dots, x_n) = r_j$ $(j = 1, \dots, s)$is written as:

$$
\left\{\begin{array}{rl}
\max & z(x_1, \dots, x_n) \\
s.t. & g_1(x_1, \dots, x_n) \leq b_1 \\
& \vdots \\
& g_m(x_1, \dots, x_n) \leq b_m \\
& h_1(x_1, \dots, x_n) = r_1 \\
& \vdots \\
& h_l(x_1, \dots, x_n) = r_l
\end{array}\right. \tag{7.5}
$$

Expression 7.5 is also called a *mathematical programming problem.* Naturally when constraints are involved we define the global and local maxima for the objective function $z(x_1, \dots, x_n)$ in terms of the feasible region instead of the entire domain of z, since we are only concerned with values of $x_1, \dots, x_n$ that satisfy our constraints.

Example 7.3 (Continuation of Example 7.1). We can re-write the problem in Example 7.1:

$$
\left\{\begin{array}{rl}
\max & A(x,y) = xy \\
s.t. & 2x + 2y = 100 \\
& x \geq 0 \\
& y \geq 0
\end{array}\right. \tag{7.6}
$$

Note we've added two inequality constraints $x \geq 0$ and $y \geq 0$ because it doesn't really make any sense to have negative lengths. We can re-write these constraints as $-x \leq 0$ and $-y \leq 0$ where $g_1(x,y) = -x$ and $g_2(x,y) = -y$ to make Expression 7.6 look like Expression 7.5.

We have formulated the general *maximization* problem in Proble 7.5. Suppose that we are interested in finding a value that minimizes an objective function $z(x_1, \dots, x_n)$ subject to certain constraints. Then we can write Problem 7.5 replacing max with min.

Exercise 55. Write the problem from Exercise 53 as a general minimization problem. Add any appropriate non-negativity constraints. [Hint: You must change max to min.]

An alternative way of dealing with minimization is to transform a minimization problem into a maximization problem. If we want to minimize $z(x_1, \dots, x_n)$, we can maximize $-z(x_1, \dots, x_n)$. In maximizing the negation of the objective function, we are actually finding a value that minimizes $z(x_1, \dots, x_n)$.

Exercise 56. Prove the following statement: *Consider Problem 7.5 with the objective function $z(x_1, \dots, x_n)$ replaced by $-z(x_1, \dots, x_n)$. Then the solution to this new problem minimizes $z(x_1, \dots, x_n)$ subject to the constraints of Problem 7.5.*[Hint: Use the definition of global maximum and a multiplication by -1. Be careful with the direction of the inequality when you multiply by -1.]

2. Some Geometry for Optimization

A critical part of optimization theory is understanding the geometry of Euclidean space. To that end, we're going to review some critical concepts from Vector Calculus. Throughout this section, we'll use vectors. We'll assume that there vectors are $n \times 1$

Recall the *dot product* from Definition 5.13. If $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{n\times 1}$

$$\mathbf{x} = [x_1, x_2, \dots, x_n]^T$$
$$\mathbf{y} = [y_1, y_2, \dots, y_n]^T$$

Then the *dot product* of these vectors is:

$$\mathbf{x} \cdot \mathbf{y} = x_1y_1 + x_2y_2 + \cdots + x_ny_n = \mathbf{x}^T\mathbf{y}$$

An alternative and useful definition for the dot product is given by the following formula. Let θ be the angle between the vectors $\mathbf{x}$ and $\mathbf{y}$. Then the dot product of $\mathbf{x}$ and $\mathbf{y}$ may be alternatively written as:

$$\mathbf{x} \cdot \mathbf{y} = ||\mathbf{x}||||\mathbf{y}|| \cos\theta \tag{7.7}$$

Here:

$$||\mathbf{x}|| = \left(x_1^2 + x_2^2 + \cdots + x_n^2\right)^{\frac{1}{2}} \tag{7.8}$$

This fact can be proved using the *law of cosines* from trigonometry. As a result, we have the following small lemma (which is proved as Theorem 1 of [**MT03**]):

Lemma 7.4. *Let $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$. Then the following hold:*

1. *The angle between $\mathbf{x}$ and $\mathbf{y}$ is less than $\pi/2$ (i.e., acute) iff $\mathbf{x} \cdot \mathbf{y} > 0$.*
2. *The angle between $\mathbf{x}$ and $\mathbf{y}$ is exactly $\pi/2$ (i.e., the vectors are orthogonal) iff $\mathbf{x} \cdot \mathbf{y} = 0$.*
3. *The angle between $\mathbf{x}$ and $\mathbf{y}$ is greater than $\pi/2$ (i.e., obtuse) iff $\mathbf{x} \cdot \mathbf{y} < 0$.*

Exercise 57. Use the value of the cosine function and the fact that $\mathbf{x} \cdot \mathbf{y} = ||\mathbf{x}||||\mathbf{y}|| \cos\theta$ to prove the lemma. [Hint: For what values of θ is $\cos\theta > 0$.]

Definition 7.5 (Graph). Let $z : D \subseteq \mathbb{R}^n \to \mathbb{R}$ be function, then the *graph* of z is the set of $n + 1$ tuples:

$$\{(\mathbf{x}, z(\mathbf{x})) \in \mathbb{R}^{n+1} | \mathbf{x} \in D\} \tag{7.9}$$

When $z : D \subseteq \mathbb{R} \to \mathbb{R}$, the graph is precisely what you'd expect. It's the set of pairs $(x, y) \in \mathbb{R}^2$ so that $y = z(x)$. This is the graph that you learned about back in Algebra 1.

Definition 7.6 (Level Set). Let $z : \mathbb{R}^n \rightarrow \mathbb{R}$ be a function and let $c \in \mathbb{R}$. Then the *level set of value c for function z* is the set:

$$\{\mathbf{x} = (x_1, \ldots, x_n) \in \mathbb{R}^n | z(\mathbf{x}) = c\} \subseteq \mathbb{R}^n \tag{7.10}$$

Example 7.7. Consider the function $z = x^2 + y^2$. The level set of z at 4 is the set of points $(x, y) \in \mathbb{R}^2$ such that:

$$x^2 + y^2 = 4 \tag{7.11}$$

You will recognize this as the equation for a circle with radius 4. We illustrate this in the following two figures. Figure 7.2 shows the level sets of z as they sit on the 3D plot of the function, while Figure 7.3 shows the level sets of z in $\mathbb{R}^2$. The plot in Figure 7.3 is called a *contour plot*.

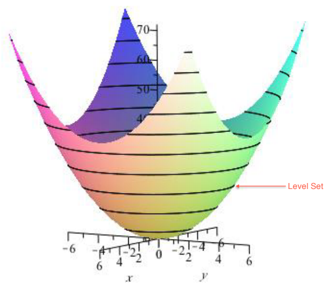


Figure 7.2. Plot with Level Sets Projected on the Graph of z. The level sets existing in $\mathbb{R}^2$ while the graph of z existing $\mathbb{R}^3$. The level sets have been projected onto their appropriate heights on the graph.

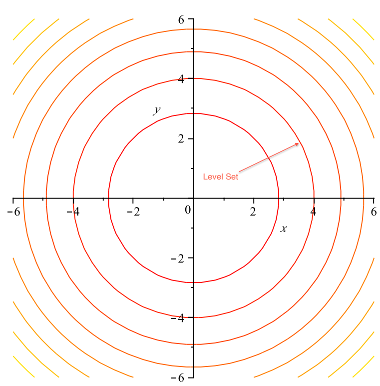


Figure 7.3. Contour Plot of $z = x^2 + y^2$. The circles in $\mathbb{R}^2$ are the level sets of the function. The lighter the circle hue, the higher the value of c that defines the level set.

Definition 7.8. (Line) Let $\mathbf{x}_0, \mathbf{v} \in \mathbb{R}^n$. Then the *line* defined by vectors $\mathbf{x}_0$ and $\mathbf{v}$ is the function $\mathbf{l}(t) = \mathbf{x}_0 + t\mathbf{v}$. Clearly $l : \mathbb{R} \rightarrow \mathbb{R}^n$. The vector $\mathbf{v}$ is called the direction of the line.

EXAMPLE 7.9. Let $\mathbf{x_0} = (2, 1)$ and let $\mathbf{v} = (2, 2)$. Then the line defined by $\mathbf{x}_0$ and $\mathbf{v}$ is shown in Figure 7.4. The set of points on this line is the set $L = \{(x, y) \in \mathbb{R}^2 : x = 2 + 2t, y = 1 + 2t, t \in \mathbb{R}\}$.

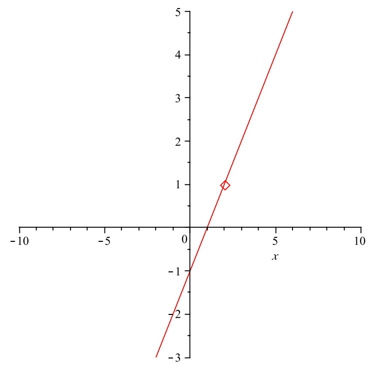


Figure 7.4. A Line Function: The points in the graph shown in this figure are in the set produced using the expression $\mathbf{x}_0 + \mathbf{v}t$ where $\mathbf{x_0} = (2, 1)$ and let $\mathbf{v} = (2, 2)$.

DEFINITION 7.10 (Directional Derivative). Let $z : \mathbb{R}^n \to \mathbb{R}$ and let $\mathbf{v} \in \mathbb{R}^n$ be a vector (direction) in n-dimensional space. Then the directional derivative of z at point $\mathbf{x}_0 \in \mathbb{R}^n$ in the direction of $\mathbf{v}$ is

$$\frac{d}{dt} z(\mathbf{x}_0 + t\mathbf{v})\bigg|_{t=0} \tag{7.12}$$

when this derivative exists.

PROPOSITION 7.11. *The directional derivative of z at $\mathbf{x}_0$ in the direction $\mathbf{v}$ is equal to:*

$$\lim_{h \to 0} \frac{z(\mathbf{x}_0 + h\mathbf{v}) - z(\mathbf{x}_0)}{h} \tag{7.13}$$

EXERCISE 58. Prove Proposition 7.11. [Hint: Use the definition of derivative for a univariate function and apply it to the definition of directional derivative and evaluate $t = 0$.]

DEFINITION 7.12 (Gradient). Let $z : \mathbb{R}^n \to \mathbb{R}$ be function and let $\mathbf{x}_0 \in \mathbb{R}^n$. Then the *gradient* of z at $\mathbf{x}_0$ is the vector in $\mathbb{R}^n$ given by:

$$\nabla z(\mathbf{x}_0) = \left(\frac{\partial z}{\partial x_1}(\mathbf{x}_0), \dots, \frac{\partial z}{\partial x_n}(\mathbf{x}_0)\right) \tag{7.14}$$

Gradients are extremely important concepts in optimization (and vector calculus in general). Gradients have many useful properties that can be exploited. The relationship between the directional derivative and the gradient is of critical importance.

THEOREM 7.13. *If $z : \mathbb{R}^n \to \mathbb{R}$ is differentiable, then all directional derivatives exist. Furthermore, the directional derivative of z at $\mathbf{x}_0$ in the direction of $\mathbf{v}$ is given by:*

$$\nabla z(\mathbf{x}_0) \cdot \mathbf{v} \tag{7.15}$$

where $\cdot$ denotes the dot product of two vectors.

Proof. Let $\mathbf{l}(t) = \mathbf{x}_0 + \mathbf{v}t$. Then $\mathbf{l}(t) = (l_1(t), \dots, l_n(t))$; that is, $\mathbf{l}(t)$ is a vector function whose i^{th} component is given by $l_i(t) = \mathbf{x}_{0_i} + \mathbf{v}_i t$.

Apply the chain rule:

$$\frac{dz(\mathbf{l}(t))}{dt} = \frac{\partial z}{\partial l_1}\frac{dl_1}{dt} + \cdots + \frac{\partial z}{\partial l_n}\frac{dl_n}{dt} \tag{7.16}$$

Thus:

$$\frac{d}{dt} z(\mathbf{l}(t)) = \nabla z \cdot \frac{d\mathbf{l}}{dt} \tag{7.17}$$

Clearly $d\mathbf{l}/dt = \mathbf{v}$. We have $\mathbf{l}(0) = \mathbf{x}_0$. Thus:

$$\left.\frac{d}{dt} z(\mathbf{x}_0 + t\mathbf{v})\right|_{t=0} = \nabla z(\mathbf{x}_0) \cdot \mathbf{v} \tag{7.18}$$

□

We now come to the two most important results about gradients, (i) the fact that they always point in the direction of steepest ascent with respect to the level curves of a function and (ii) that they are perpendicular (normal) to the level curves of a function. We can exploit this fact as we seek to maximize (or minimize) functions.

Theorem 7.14. *Let $z : \mathbb{R}^n \to \mathbb{R}$ be differentiable and let $\mathbf{x}_0 \in \mathbb{R}^n$. If $\nabla z(\mathbf{x}_0) \neq 0$, then $\nabla z(\mathbf{x}_0)$ points in the direction in which z is increasing fastest.*

Proof. Recall $\nabla z(\mathbf{x}_0) \cdot \mathbf{n}$ is the directional derivative of z in direction $\mathbf{n}$ at $\mathbf{x}_0$. Assume that $\mathbf{n}$ is a unit vector. We know that:

$$\nabla z(\mathbf{x}_0) \cdot \mathbf{n} = ||\nabla z(\mathbf{x}_0)|| \cos\theta \tag{7.19}$$

where θ is the angle between the vectors $\nabla z(\mathbf{x}_0)$ and $\mathbf{n}$. The function $\cos\theta$ is largest when $\theta = 0$, that is when $\mathbf{n}$ and $\nabla z(\mathbf{x}_0)$ are parallel vectors. (If $\nabla z(\mathbf{x}_0) = 0$, then the directional derivative is zero in all directions.) □

Theorem 7.15. *Let $z : \mathbb{R}^n \to \mathbb{R}$ be differentiable and let $\mathbf{x}_0$ lie in the level set S defined by $z(\mathbf{x}) = k$ for fixed $k \in \mathbb{R}$. Then $\nabla z(\mathbf{x}_0)$ is normal to the set S in the sense that if $\mathbf{v}$ is a tangent vector at $t = 0$ of a path $\mathbf{c}(t)$ contained entirely in S with $\mathbf{c}(0) = \mathbf{x}_0$, then $\nabla z(\mathbf{x}_0) \cdot v = 0$.*

Before giving the proof, we illustrate this theorem in Figure 7.5. The function is $z(x, y) = x^4 + y^2 + 2xy$ and $\mathbf{x}_0 = (1, 1)$. At this point $\nabla z(\mathbf{x}_0) = (6, 4)$.

Proof. As stated, let $\mathbf{c}(t)$ be a curve in S. Then $\mathbf{c} : \mathbb{R} \to \mathbb{R}^n$ and $z(\mathbf{c}(t)) = k$ for all $t \in \mathbb{R}$. Let $\mathbf{v}$ be the tangent vector to $\mathbf{c}$ at $t = 0$; that is:

$$\left.\frac{d\mathbf{c}(t)}{dt}\right|_{t=0} = \mathbf{v} \tag{7.20}$$

Differentiating $z(\mathbf{c}(t))$ with respect to t using the chain rule and evaluating at $t = 0$ yields:

$$\left.\frac{d}{dt} z(\mathbf{c}(t))\right|_{t=0} = \nabla z(\mathbf{c}(0)) \cdot \mathbf{v} = \nabla z(\mathbf{x}_0) \cdot \mathbf{v} = 0 \tag{7.21}$$

Thus $\nabla z(\mathbf{x}_0)$ is perpendicular to $\mathbf{v}$ and thus normal to the set S as required. □

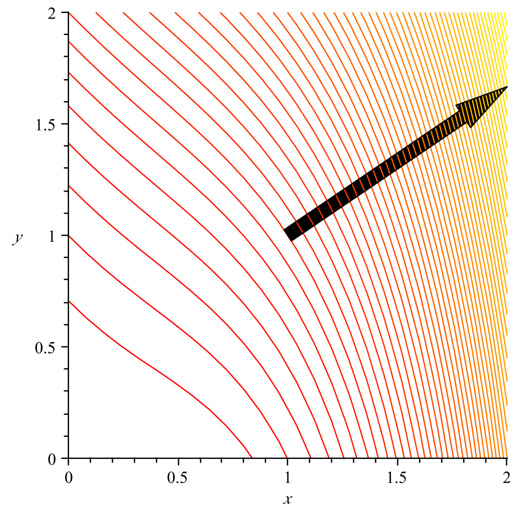


Figure 7.5. A Level Curve Plot with Gradient Vector: We've scaled the gradient vector in this case to make the picture understandable. Note that the gradient is perpendicular to the level set curve at the point $(1, 1)$, where the gradient was evaluated. You can also note that the gradient is pointing in the direction of steepest ascent of $z(x, y)$.

EXERCISE 59. In this exercise you will use elementary calculus (and a little bit of vector algebra) to show that the gradient of a simple function is perpendicular to its level sets:

(a): Plot the level sets of $z(x, y) = x^2 + y^2$. Draw the gradient at the point $(x, y) = (2, 0)$. Convince yourself that it is normal to the level set $x^2 + y^2 = 4$.

(b): Now, choose any level set $x^2 + y^2 = k$. Use implicit differentiation to find dy/dx. This is the slope of a tangent line to the circle $x^2 + y^2 = k$. Let (x_0, y_0) be a point on this circle.

(c): Find an expression for a vector parallel to the tangent line at (x_0, y_0) [Hint: you can use the slope you just found.]

(d): Compute the gradient of z at (x_0, y_0) and use it and the vector expression you just computed to show that two vectors are perpendicular. [Hint: use the dot product.]

3. Gradients, Constraints and Optimization

Since we're talking about optimization (i.e., minimizing or maximizing a certain function subject to some constraints), it follows that we should be interested in the gradient, which indicates the direction of greatest increase in a function. This information will be used in maximizing a function. Logically, the negation of the gradient will point in the direction of greatest decrease and can be used in minimization. We'll formalize these notions in the study of linear programming. We make one more definition:

DEFINITION 7.16 (Binding Constraint). Let $g(\mathbf{x}) \leq b$ be a constraint in an optimization problem. If at point $\mathbf{x}_0 \in \mathbb{R}^n$ we have $g(\mathbf{x}_0) = b$, then the constraint is said to be *binding*. Clearly equality constraints $h(\mathbf{x}) = r$ are always binding.

Example 7.17 (Continuation of Example 7.1). Let's look at the level curves of the objective function and their relationship to the constraints at the point of optimality $(x, y) = (25, 25)$. In Figure 7.6 we see the level curves of the objective function (the hyperbolas) and the feasible region shown as shaded. The elements in the feasible regions are all values for x and y for which $2x + 2y \leq 100$ and $x, y \geq 0$. You'll note that at the point of optimality the level curve $xy = 625$ is tangent to the equation $2x + 2y = 100$; i.e., the level curve of the objective function is tangent to the binding constraint.

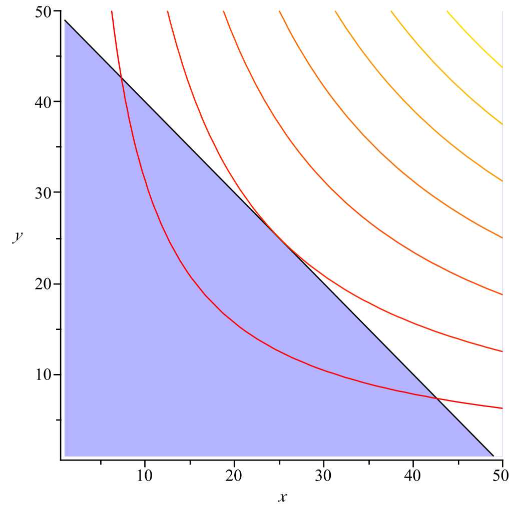


Figure 7.6. Level Curves and Feasible Region: At optimality the level curve of the objective function is tangent to the binding constraints.

If you look at the gradient of $A(x, y)$ at this point it has value $(25, 25)$. We see that it is pointing in the direction of increase for the function $A(x, y)$ (as should be expected) but more importantly let's look at the gradient of the function $2x + 2y$. It's gradient is $(2, 2)$, which is just a scaled version of the gradient of the objective function. Thus the gradient of the objective function is just a dilation of gradient of the binding constraint. This is illustrated in Figure 7.7.

The elements illustrated in the previous example are true in general. You may have discussed a simple example of these when you talked about *Lagrange Multipliers* in Vector Calculus (Math 230/231). We'll revisit these concepts when we discuss 7.31.

Exercise 60. Plot the level sets of the objective function and the feasible region in Exercise 53. At the point of optimality you identified, show that the gradient of the objective function is a scaled version of the gradient (linear combination) of the binding constraints.

4. Convex Sets and Combinations

Definition 7.18 (Convex Set). Let $X \subseteq \mathbb{R}^n$. Then the set X is convex if and only if for all pairs $\mathbf{x}_1, \mathbf{x}_2 \in X$ we have $\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2 \in X$ for all $\lambda \in [0, 1]$.

The definition of convexity seems complex, but it is easy to understand. First recall that if $\lambda \in [0, 1]$, then the point $\lambda\mathbf{x}_1 + (1 - \lambda)\mathbf{x}_2$ is on the line segment connecting $\mathbf{x}_1$ and $\mathbf{x}_2$ in $\mathbb{R}^n$.

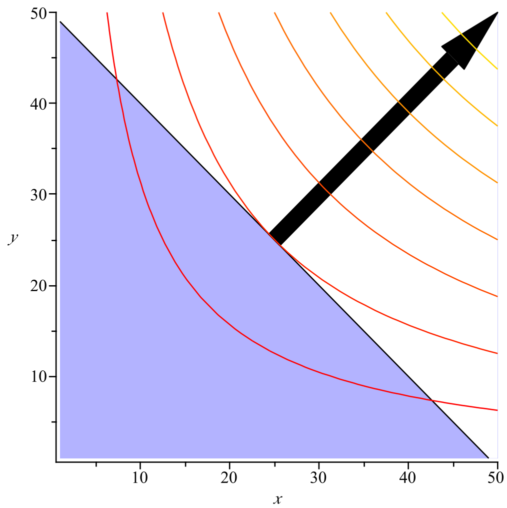


Figure 7.7. Gradients of the Binding Constraint and Objective: At optimality the gradient of the binding constraints and the objective function are *scaled versions of each other*.

For example, when $\lambda = 1/2$, then the point $\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$ is the midpoint between $\mathbf{x}_1$ and $\mathbf{x}_2$. In fact, for every point $\mathbf{x}$ on the line connecting $\mathbf{x}_1$ and $\mathbf{x}_2$ we can find a value $\lambda \in [0,1]$ so that $\mathbf{x} = \lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$. Then we can see that, convexity asserts that if $\mathbf{x}_1, \mathbf{x}_2 \in X$, then every point on the line connecting $\mathbf{x}_1$ and $\mathbf{x}_2$ is also in the set X.

Definition 7.19. Let $\mathbf{x}_1, \ldots, \mathbf{x}_m$ be vectors in $\in \mathbb{R}^n$ and let $\alpha_1, \ldots, \alpha_m \in \mathbb{R}$ be scalars. Then

$$\alpha_1\mathbf{x}_1 + \cdots + \alpha_m\mathbf{x}_m \tag{7.22}$$

is a *linear combination* of the vectors $\mathbf{x}_1, \ldots, \mathbf{x}_m$.

Definition 7.20 (Positive Combination). Let $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^n$. If $\lambda_1, \ldots, \lambda_m > 0$ and then

$$\mathbf{x} = \sum_{i=1}^{m} \lambda_i \mathbf{x}_i \tag{7.23}$$

is called a *positive combination* of $\mathbf{x}_1, \ldots, \mathbf{x}_m$.

Definition 7.21 (Convex Combination). Let $\mathbf{x}_1, \ldots, \mathbf{x}_m \in \mathbb{R}^n$. If $\lambda_1, \ldots, \lambda_m \in [0,1]$ and

$$\sum_{i=1}^{m} \lambda_i = 1$$

then

$$\mathbf{x} = \sum_{i=1}^{m} \lambda_i \mathbf{x}_i \tag{7.24}$$

is called a *convex combination* of $\mathbf{x}_1, \ldots, \mathbf{x}_m$. If $\lambda_i < 1$ for all $i = 1, \ldots, m$, then Equation 7.24 is called a *strict convex combination*.

Remark 7.22. We can see that we move from the very general to the very specific as we go from linear combinations to positive combinations to convex combinations. A linear combination of points or vectors allowed us to choose any real values for the coefficients. A positive combination restricts us to positive values, while a convex combination asserts that those values must be non-negative and sum to 1.

Example 7.23. Figure 7.8 illustrates a convex and non-convex set. Non-convex sets

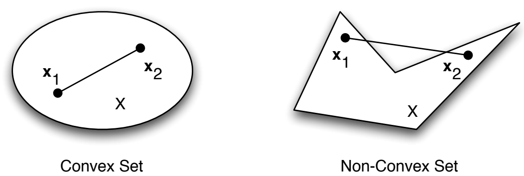


Figure 7.8. Examples of Convex Sets: The set on the left (an ellipse and its interior) is a convex set; every pair of points inside the ellipse can be connected by a line contained entirely in the ellipse. The set on the right is clearly not convex as we've illustrated two points whose connecting line is not contained inside the set.

have some resemblance to crescent shapes or have components that look like crescents.

Theorem 7.24. *The intersection of a finite number of convex sets in $\mathbb{R}^n$ is convex.*

Proof. Let $C_1, \dots, C_n \subseteq \mathbb{R}^n$ be a finite collection of convex sets. Let

$$C = \bigcap_{i=1}^{n} C_i \tag{7.25}$$

be the set formed from the intersection of these sets. Choose $\mathbf{x}_1, \mathbf{x}_2 \in C$ and $\lambda \in [0, 1]$. Consider $\mathbf{x} = \lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2$. We know that $\mathbf{x}_1, \mathbf{x}_2 \in C_1, \dots, C_n$ by definition of C. By convexity, we know that $\mathbf{x} \in C_1, \dots, C_n$ by convexity of each set. Therefore, $\mathbf{x} \in C$. Thus C is a convex set. $\square$

5. Convex and Concave Functions

Definition 7.25 (Convex Function). A function $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function if it satisfies:

$$f(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1-\lambda) f(\mathbf{x}_2) \tag{7.26}$$

for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ and for all $\lambda \in [0, 1]$.

This definition is illustrated in Figure 7.9. When f is a univariate function, this definition can be shown to be equivalent to the definition you learned in Calculus I (Math 140) using first and second derivatives.

Definition 7.26 (Concave Function). A function $f : \mathbb{R}^n \to \mathbb{R}$ is a convex function if it satisfies:

$$f(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \geq \lambda f(\mathbf{x}_1) + (1-\lambda) f(\mathbf{x}_2) \tag{7.27}$$

for all $\mathbf{x}_1, \mathbf{x}_2 \in \mathbb{R}^n$ and for all $\lambda \in [0, 1]$.

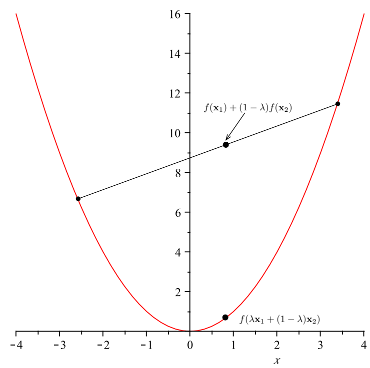


Figure 7.9. A convex function: A convex function satisfies the expression $f(\lambda\mathbf{x}_1 + (1-\lambda)\mathbf{x}_2) \leq \lambda f(\mathbf{x}_1) + (1-\lambda)f(\mathbf{x}_2)$ for all $\mathbf{x}_1$ and $\mathbf{x}_2$ and $\lambda \in [0, 1]$.

To visualize this definition, simply flip Figure 7.9 upside down. The following theorem is a powerful tool that can be used to show sets are convex. It's proof is outside the scope of the class, but relatively easy.

Theorem 7.27. *Let $f : \mathbb{R}^n \to \mathbb{R}$ be a convex function. Then the set $C = \{\mathbf{x} \in \mathbb{R}^n : f(x) \leq c\}$, where $c \in \mathbb{R}$, is a convex set.*

Exercise 61. Prove the Theorem 7.27.

Definition 7.28 (Linear Function). A function $z : \mathbb{R}^n \to \mathbb{R}$ is *linear* if there are constants $c_1, \dots, c_n \in \mathbb{R}$ so that:

$$z(x_1, \dots, x_n) = c_1 x_1 + \cdots + c_n x_n \tag{7.28}$$

Example 7.29. We have had experience with many linear functions already. The left-hand-side of the constraint $2x + 2y \leq 100$ is a linear function. That is the function $z(x, y) = 2x + 2y$ is a linear function of x and y.

Definition 7.30 (Affine Function). A function $z : \mathbb{R}^n \to \mathbb{R}$ is *affine* if $z(\mathbf{x}) = l(\mathbf{x}) + b$ where $l : \mathbb{R}^n \to \mathbb{R}$ is a linear function and $b \in \mathbb{R}$.

Exercise 62. Prove that every affine function is both convex and concave.

6. Kurush-Kuhn-Tucker Conditions

It turns out there is a very powerful theorem that discusses when a point $\mathbf{x}^* \in \mathbb{R}^n$ will maximize a function. The following is the Kuhn-Karush-Tucker theorem, which we will state, but not prove.

Theorem 7.31. *Let $z : \mathbb{R}^n \to \mathbb{R}$ be a differentiable objective function, $g_i : \mathbb{R}^n \to \mathbb{R}$ be differentiable constraint functions for $i = 1, \dots, m$ and $h_j : \mathbb{R}^n \to \mathbb{R}$ be differentiable constraint functions for $j = 1, \dots, l$. If $\mathbf{x}^* \in \mathbb{R}^n$ is an optimal point satisfying an appropriate* *regularity condition for the following optimization problem:*

$$P \begin{cases} \max & z(x_1, \dots, x_n) \\ s.t. & g_1(x_1, \dots, x_n) \le 0 \\ & \qquad \vdots \\ & g_m(x_1, \dots, x_n) \le 0 \\ & h_1(x_1, \dots, x_n) = 0 \\ & \qquad \vdots \\ & h_l(x_1, \dots, x_n) = 0 \end{cases}$$

then there exists $\lambda_1, \dots, \lambda_m \in \mathbb{R}$ *and* $\mu_1, \dots \mu_l \in \mathbb{R}$ *so that:*

$$\begin{aligned} &\textit{Primal Feasibility}: \begin{cases} g_i(\mathbf{x}^*) \le 0 & \textit{for } i = 1, \dots, m \\ h_j(\mathbf{x}^*) = 0 & \textit{for } j = 1, \dots, l \end{cases} \\ &\textit{Dual Feasibility}: \begin{cases} \nabla z(\mathbf{x}^*) - \displaystyle\sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}^*) - \sum_{j=1}^{l} \mu_j \nabla h_j(\mathbf{x}^*) = \mathbf{0} \\ \lambda_i \ge 0 \quad \textit{for } i = 1, \dots, m \\ \mu_j \in \mathbb{R} \quad \textit{for } j = 1, \dots, l \end{cases} \\ &\textit{Complementary Slackness}: \left\{ \lambda_i g_i(\mathbf{x}^*) = 0 \quad \textit{for } i = 1, \dots, m \right. \end{aligned}$$

THEOREM 7.32. *Let* $z : \mathbb{R}^n \to \mathbb{R}$ *be a differentiable concave function,* $g_i : \mathbb{R}^n \to \mathbb{R}$ *be differentiable convex functions for* $i = 1, \dots, m$ *and* $h_j : \mathbb{R}^n \to \mathbb{R}$ *be affine functions for* $j = 1, \dots, l$*. Suppose there are* $\lambda_1, \dots, \lambda_m \in \mathbb{R}$ *and* $\mu_1, \dots \mu_l \in \mathbb{R}$ *so that:*

$$\begin{aligned} &\textit{Primal Feasibility}: \begin{cases} g_i(\mathbf{x}^*) \le 0 & \textit{for } i = 1, \dots, m \\ h_j(\mathbf{x}^*) = 0 & \textit{for } j = 1, \dots, l \end{cases} \\ &\textit{Dual Feasibility}: \begin{cases} \nabla z(\mathbf{x}^*) - \displaystyle\sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}^*) - \sum_{j=1}^{l} \mu_j \nabla h_j(\mathbf{x}^*) = \mathbf{0} \\ \lambda_i \ge 0 \quad \textit{for } i = 1, \dots, m \\ \mu_j \in \mathbb{R} \quad \textit{for } j = 1, \dots, l \end{cases} \\ &\textit{Complementary Slackness}: \left\{ \lambda_i g_i(\mathbf{x}^*) = 0 \quad \textit{for } i = 1, \dots, m \right. \end{aligned}$$

then $\mathbf{x}^*$ *is a global maximizer for*

$$P \begin{cases} \max & z(x_1, \dots, x_n) \\ s.t. & g_1(x_1, \dots, x_n) \leq 0 \\ & \vdots \\ & g_m(x_1, \dots, x_n) \leq 0 \\ & h_1(x_1, \dots, x_n) = 0 \\ & \vdots \\ & h_l(x_1, \dots, x_n) = 0 \end{cases}$$

REMARK 7.33. The values $\lambda_1, \dots, \lambda_m$ and $\mu_1, \dots, \mu_l$ are sometimes called *Lagrange multipliers* and sometimes called *dual variables*. Primal Feasibility, Dual Feasibility and Complementary Slackness are called the *Karush-Kuhn-Tucker* (KKT) conditions.

REMARK 7.34. The regularity condition mentioned in Theorem 7.31 is sometimes called a constraint qualification. A common one is that the gradients of the binding constraints are all linearly independent at $\mathbf{x}^*$. There are many variations of constraint qualifications. We will not deal with these in these notes. Suffice it to say, all the problems we consider will automatically satisfy a constraint qualification, meaning the KKT theorem holds.

REMARK 7.35. This theorem holds as a necessary condition even if $z(\mathbf{x})$ is not concave or the functions $g_i(\mathbf{x})$ $(i = 1, \dots, m)$ are not convex or the functions $h_j(\mathbf{x})$ $(j = 1, \dots, l)$ are not linear. In this case though, the fact that a triple: $(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) \in \mathbb{R}^n \times \mathbb{R}^m \times \mathbb{R}^l$ does not ensure that this is an optimal solution for Problem P.

REMARK 7.36. Looking more closely at the dual feasibility conditions, we see something interesting. Suppose that there are *no* equality constraints (i.e., not constraints of the form $h_j(\mathbf{x}) = 0$). Then the statements:

$$\nabla z(\mathbf{x}^*) - \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}^*) - \sum_{j=1}^{l} \mu_j \nabla h_j(\mathbf{x}^*) = \mathbf{0}$$
$$\lambda_i \geq 0 \quad \text{for } i = 1, \dots, m$$

imply that:

$$\nabla z(\mathbf{x}^*) = \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}^*)$$
$$\lambda_i \geq 0 \quad \text{for } i = 1, \dots, m$$

Specifically, this says that the *gradient of* z *at* $\mathbf{x}^*$ *is a positive combination of the gradients of the constraints at* $\mathbf{x}^*$. But more importantly, since we also have *complementary slackness*, we know that if $\mathbf{g}_i(\mathbf{x}^*) \neq 0$, then $\lambda_i = 0$ because $\lambda_i g_i(\mathbf{x}^*) = 0$ for $i = 1, \dots, m$. Thus, what dual feasibility is really saying is that *gradient of* z *at* $\mathbf{x}^*$ *is a positive combination of the gradients of the* ***binding*** *constraints at* $\mathbf{x}^*$. Remember, a constraint is binding if $g_i(\mathbf{x}^*) = 0$, in which case $\lambda_i \geq 0$.

Remark 7.37. Continuing from the previous remark, in the general case when we have some equality constraints, then dual feasibility says:

$$\begin{aligned}
\nabla z(\mathbf{x}^*) &= \sum_{i=1}^{m} \lambda_i \nabla g_i(\mathbf{x}^*) + \sum_{j=1}^{l} \mu_j \nabla h_j(\mathbf{x}^*) \\
\lambda_i &\geq 0 \quad \text{for } i = 1, \dots, m \\
\mu_j &\in \mathbb{R} \quad \text{for } j = 1, \dots, l
\end{aligned}$$

Since equality constraints are *always binding* this says that the *gradient of z at $\mathbf{x}^*$ is a linear combination of the gradients of the **binding** constraints at $\mathbf{x}^*$.*

Example 7.38. We'll finish the example we started with Example 7.1. Let's rephrase this optimization problem in the form we saw in the theorem: We'll have:

$$(7.29) \qquad \begin{cases} \max \;\; A(x, y) = xy \\ \;\; s.t. \;\; 2x + 2y - 100 = 0 \\ \qquad\;\; -x \leq 0 \\ \qquad\;\; -y \leq 0 \end{cases}$$

Note that the greater-than inequalities $x \geq 0$ and $y \geq 0$ in Expression 7.6 have been changes to less-than inequalities by multiplying by -1. The constraints $2x + 2y = 100$ has simply been transformed to $2x + 2y - 100 = 0$. Thus, if $h(x, y) = 2x + 2y - 100$, we can see $h(x, y) = 0$ is our constraint. We can let $g_1(x, y) = -x$ and $g_2(x, y) = -y$. Then we have $g_1(x, y) \leq 0$ and $g_2(x, y) \leq 0$ as our inequality constraints. We already know that $x = y = 25$ is our optimal solution. Thus we know that there must be Lagrange multipliers μ, λ_1 and λ_2 corresponding to the constraints $h(x, y) =$, $g_1(x, y) \leq 0$ and $g_2(x, y) \leq 0$ that satisfy the KKT conditions.

Let's investigate the three components of the KKT conditions.

Primal Feasibility: If $x = y = 25$, then $h(x, y) = 2x + 2y - 100$ and clearly $h(25, 25) = 0$. Further $g_1(x, y) = -x$ and $g_2(x, y) = -y$ then $g_1(25, 25) = -25 \leq 0$ and $g_2(25, 25) = -25 \leq 0$. So primal feasibility is satisfied.

Complementary Slackness: We know that $g_1(x, y) = g_2(x, y) = -25$. Since neither of these functions is 0, we know that $\lambda_1 = \lambda_2 = 0$. This will force complementary slackness, namely:

$$\lambda_1 g_1(25, 25) = 0$$

$$\lambda_2 g_2(25, 25) = 0$$

Dual Feasibility: We already know that $\lambda_1 = \lambda_2 = 0$. That means we need to find $\mu \in \mathbb{R}$ so that:

$$\nabla A(25, 25) - \mu \nabla h(25, 25) = \mathbf{0}$$

We know that:

$$\nabla A(x, y) = \nabla xy = \begin{bmatrix} y \\ x \end{bmatrix}$$

$$\nabla h(x, y) = \nabla(2x + 2y - 100) = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

Evaluating $\nabla A(25, 25)$ yields:

$$\begin{bmatrix} 25 \\ 25 \end{bmatrix} - \mu \begin{bmatrix} 2 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$$

Thus setting $\mu = 25/2$ will accomplish our goal.

Exercise 63. Find the values of the dual variables for the optimal point in Exercise 53. Show that the KKT conditions hold for the values you found.

7. Relating Back to Game Theory

It's easy to think we've lost our way and wondered into a class on Optimization Theory when really we're in the middle of a class on Game Theory. In reality, the two subjects are intimately related. After all, when you play a game you're trying to maximize your payoff subject to constraints on your moves *and* subject to the actions of the other players. That's what makes games a little more interesting than generic optimization problems, someone else is influencing the decision variables.

Consider a game in normal form $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$. We'll assume that $\mathbf{P} = \{P_1, \dots, P_N$ and $\Sigma_i = \{\sigma_1^i, \dots, \sigma_{n_i}^i\}$. If we assume a fixed mixed strategy $\mathbf{x} \in \Delta$, Player P_i's objective when choosing a response $\mathbf{x}^i \in \Delta_{n_i}$ is to solve the following problem:

$$\text{(7.30)} \quad \text{Player } P_i : \begin{cases} \max & u_i(\mathbf{x}^i, \mathbf{x}^{-i}) \\ s.t. & \mathbf{x}_1^i + \cdots + \mathbf{x}_{n_i}^i = 1 \\ & \mathbf{x}_j^i \geq 0 \quad j = 1, \dots, n_i \end{cases}$$

This is a mathematical programming problem, provided that $u_i(\mathbf{x}^i, \mathbf{x}^{-i})$ is known. However, it assumes that all other players are holding their strategy constant e.g., playing $\mathbf{x}^{-i}$. The interesting part (and the part that makes Game Theory hard) is that each player is solving this problem *simultaneously*. Thus an equilibrium solution is a simultaneous solution to:

$$\text{(7.31)} \quad \forall i : \begin{cases} \max & u_i(\mathbf{x}^i, \mathbf{x}^{-i}) \\ s.t. & \mathbf{x}_1^i + \cdots + \mathbf{x}_{n_i}^i = 1 \\ & \mathbf{x}_j^i \geq 0 \quad j = 1, \dots, n_i \end{cases}$$

This leads to an incredibly rich class of problems in mathematical programming, which we will begin to discuss in the next chapter.

CHAPTER 8

Zero-Sum Matrix Games with Linear Programming

1. Linear Programs

When both the objective and all the constraints in Expression 7.5 are linear functions, then the optimization problem is called a *linear programming problem*. This has the general form:

$$
\left\{
\begin{aligned}
\max\ & z(x_1,\dots,x_n) = c_1x_1 + \cdots + c_nx_n \\
s.t.\ & a_{11}x_1 + \cdots + a_{1n}x_n \leq b_1 \\
& \qquad\vdots \\
& a_{m1}x_1 + \cdots + a_{mn}x_n \leq b_m \\
& h_{11}x_1 + \cdots + h_{n1}x_n = r_1 \\
& \qquad\vdots \\
& h_{l1}x_1 + \cdots + h_{ln}x_n = r_l
\end{aligned}
\right.
\tag{8.1}
$$

Example 8.1. Consider the problem of a toy company that produces toy planes and toy boats. The toy company can sell its planes for \$10 and its boats for \$8 dollars. It costs \$3 in raw materials to make a plane and \$2 in raw materials to make a boat. A plane requires 3 hours to make and 1 hour to finish while a boat requires 1 hour to make and 2 hours to finish. The toy company knows it will not sell anymore than 35 planes per week. Further, given the number of workers, the company cannot spend anymore than 160 hours per week finishing toys and 120 hours per week making toys. The company wishes to maximize the profit it makes by choosing how much of each toy to produce.

We can represent the profit maximization problem of the company as a linear programming problem. Let x_1 be the number of planes the company will produce and let x_2 be the number of boats the company will produce. The profit for each plane is $\$10 - \$3 = \$7$ per plane and the profit for each boat is $\$8 - \$2 = \$6$ per boat. Thus the total profit the company will make is:

$$z(x_1, x_2) = 7x_1 + 6x_2 \tag{8.2}$$

The company can spend no more than 120 hours per week making toys and since a plane takes 3 hours to make and a boat takes 1 hour to make we have:

$$3x_1 + x_2 \leq 120 \tag{8.3}$$

Likewise, the company can spend no more than 160 hours per week finishing toys and since it takes 1 hour to finish a plane and 2 hour to finish a boat we have:

$$x_1 + 2x_2 \leq 160 \tag{8.4}$$

Finally, we know that $x_1 \leq 35$, since the company will make no more than 35 planes per week. Thus the complete linear programming problem is given as:

$$(8.5) \qquad \begin{cases} \max \; z(x_1, x_2) = 7x_1 + 6x_2 \\ s.t. \; 3x_1 + x_2 \leq 120 \\ \quad x_1 + 2x_2 \leq 160 \\ \quad x_1 \leq 35 \\ \quad x_1 \geq 0 \\ \quad x_2 \geq 0 \end{cases}$$

REMARK 8.2. Strictly speaking, the linear programming problem in Example 8.1 is not a true linear programming problem because we don't want to manufacture a fractional number of boats or planes and therefore x_1 and x_2 must really be drawn from the *integers* and not the real numbers (a requirement for a linear programming problem). This type of problem is generally called an integer programming problem. However, we will ignore this fact and assume that we can indeed manufacture a fractional number of boats and planes. If you're interested in this distinction, you might consider taking Math 484, where we discuss this issue in depth.

EXERCISE 64. A chemical manufacturer produces three chemicals: A, B and C. These chemical are produced by two processes: 1 and 2. Running process 1 for 1 hour costs \$4 and yields 3 units of chemical A, 1 unit of chemical B and 1 unit of chemical C. Running process 2 for 1 hour costs \$1 and produces 1 units of chemical A, and 1 unit of chemical B (but none of Chemical C). To meet customer demand, at least 10 units of chemical A, 5 units of chemical B and 3 units of chemical C must be produced daily. Assume that the chemical manufacturer wants to minimize the cost of production. Develop a linear programming problem describing the constraints and objectives of the chemical manufacturer. [Hint: Let x_1 be the amount of time Process 1 is executed and let x_2 be amount of time Process 2 is executed. Use the coefficients above to express the cost of running Process 1 for x_1 time and Process 2 for x_2 time. Do the same to compute the amount of chemicals A, B, and C that are produced.]

2. Intuition on the Solution of Linear Programs

Linear Programs (LP's) with two variables can be solved graphically by plotting the feasible region along with the level curves of the objective function. We will show that we can find a point in the feasible region that maximizes the objective function using the level curves of the objective function. We illustrate the method first using the problem from Example 8.1.

EXAMPLE 8.3 (Continuation of Example 8.1). Let's continue the example of the Toy Maker begin in Example 8.1. To solve the linear programming problem graphically, begin by drawing the feasible region. This is shown in the blue shaded region of Figure 8.1.

After plotting the feasible region, the next step is to plot the level curves of the objective function. In our problem, the level sets will have the form:

$$7x_1 + 6x_2 = c \implies x_2 = \frac{-7}{6}x_1 + \frac{c}{6}$$

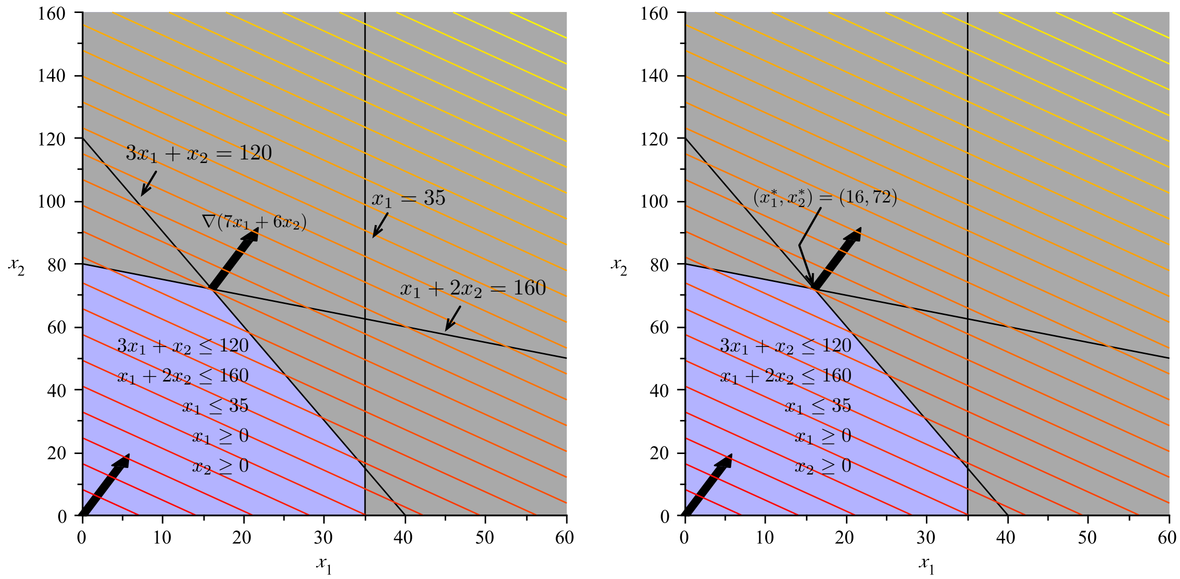


Figure 8.1. Feasible Region and Level Curves of the Objective Function: The shaded region in the plot is the feasible region and represents the intersection of the five inequalities constraining the values of x_1 and x_2. On the right, we see the optimal solution is the "last" point in the feasible region that intersects a level set as we move in the direction of increasing profit.

This is a set of parallel lines with slope $-7/6$ and intercept $c/6$ where c can be varied as needed. The level curves for various values of c are parallel lines. In Figure 8.1 they are shown in colors ranging from red to yellow depending upon the value of c. Larger values of c are more yellow.

To solve the linear programming problem, follow the level sets along the gradient (shown as the black arrow) until the last level set (line) intersects the feasible region. If you are doing this by hand, you can draw a single line of the form $7x_1 + 6x_2 = c$ and then simply draw parallel lines in the direction of the gradient $(7, 6)$. At some point, these lines will fail to intersect the feasible region. The last line to intersect the feasible region will do so at a point that maximizes the profit. In this case, the point that maximizes $z(x_1, x_2) = 7x_1 + 6x_2$, subject to the constraints given, is $(x_1^*, x_2^*) = (16, 72)$.

Note the point of optimality $(x_1^*, x_2^*) = (16, 72)$ is at a corner of the feasible region. This corner is formed by the intersection of the two lines: $3x_1 + x_2 = 120$ and $x_1 + 2x_2 = 160$. In this case, the constraints

$$
\begin{aligned}
3x_1 + x_2 &\leq 120 \\
x_1 + 2x_2 &\leq 160
\end{aligned}
$$

are both *binding*, while the other constraints are non-binding. In general, we will see that when an optimal solution to a linear programming problem exists, it will always be at the intersection of several binding constraints; that is, it will occur at a corner of a higher-dimensional polyhedron.

2.1. KKT Conditions for Linear Programs. As with any mathematical programming problem, we can derive the Karush-Kuhn-Tucker conditions for the a linear programming problem. We'll illustrate this by deriving the KKT conditions for Example 8.1. Note since linear (affine) functions are both convex and concave functions, we know that finding a Lagrange multipliers satisfying the KKT conditions is necessary and sufficient for proving that a point is an optimal point.

EXAMPLE 8.4. Let $z(x_1, x_2) = 7x_1 + 6x_2$, the objective function in Problem 8.5. We have argued that the point of optimality is $(x_1^*, x_2^*) = (16, 72)$. The KKT conditions for Problem 8.5 are:

Primal Feasibility:

$$
(8.6) \qquad \begin{cases} & \underline{\text{Lagrange Multiplier}} \\ g_1(x_1^*, x_2^*) = 3x_1^* + x_2^* - 120 \leq 0 & (\lambda_1) \\ g_2(x_1^*, x_2^*) = x_1^* + 2x_2^* - 160 \leq 0 & (\lambda_2) \\ g_3(x_1^*, x_2^*) = x_1^* - 35 \leq 0 & (\lambda_3) \\ g_4(x_1^*, x_2^*) = -x_1^* \leq 0 & (\lambda_4) \\ g_5(x_1^*, x_2^*) = -x_2^* \leq 0 & (\lambda_5) \end{cases}
$$

Dual Feasibility:

$$
(8.7) \qquad \begin{cases} \nabla z(x_1^*, x_2^*) - \displaystyle\sum_{i=1}^{5} \lambda_i \nabla g_i(x_1^*, x_2^*) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \lambda_i \geq 0 \quad i = 1, \dots, 5 \end{cases}
$$

Complementary Slackness:

$$
(8.8) \qquad \{\lambda_i g_i(x_1^*, x_2^*) = 0 \quad i = 1, \dots, 5
$$

We have $[0 \ \ 0]^T$ in our dual feasible conditions because the gradients of our functions will all be two-dimensional vectors (there are two variables). Specifically, we can compute

(1) $\nabla z(x_1^*, x_2^*) = [7 \ \ 6]^T$
(2) $\nabla g_1(x_1^*, x_2^*) = [3 \ \ 1]^T$
(3) $\nabla g_2(x_1^*, x_2^*) = [1 \ \ 2]^T$
(4) $\nabla g_3(x_1^*, x_2^*) = [1 \ \ 0]^T$
(5) $\nabla g_4(x_1^*, x_2^*) = [-1 \ \ 0]^T$
(6) $\nabla g_5(x_1^*, x_2^*) = [0 \ \ -1]^T$

Notice that $g_3(16, 72) = 16 - 35 = -17 \neq 0$. This means that for complementary slackness to be satisfied we must have $\lambda_2 = 0$. The the same reasoning, $\lambda_4 = 0$ because $g_4(16, 72) = -16 \neq 0$ and $\lambda_5 = 0$ because $g_5(16, 72) = -72 \neq 0$. Thus, dual feasibility can be simplified to:

$$
(8.9) \qquad \begin{cases} \begin{bmatrix} 7 \\ 6 \end{bmatrix} - \lambda_1 \begin{bmatrix} 3 \\ 1 \end{bmatrix} - \lambda_2 \begin{bmatrix} 1 \\ 2 \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ \lambda_i \geq 0 \quad i = 1, \dots, 5 \end{cases}
$$

This is just a set of linear equations (with some non-negativity constraints, which we'll ignore). We have:

$$7 - 3\lambda_1 - \lambda_2 = 0 \implies 3\lambda_1 + \lambda_2 = 7 \tag{8.10}$$

$$6 - \lambda_1 - 2\lambda_2 = 0 \implies \lambda_1 + 2\lambda_2 = 6 \tag{8.11}$$

We can solve these linear equations (and hope that the solution is positive). Doing so yields:

$$\lambda_1 = \frac{8}{5} \tag{8.12}$$

$$\lambda_2 = \frac{11}{5} \tag{8.13}$$

Thus we have found a KKT point:

$$\begin{aligned}
x_1^* &= 16 \\
x_2^* &= 72 \\
\lambda_1 &= \frac{8}{5} \\
\lambda_2 &= \frac{11}{5} \\
\lambda_3 &= 0 \\
\lambda_4 &= 0 \\
\lambda_5 &= 0
\end{aligned} \tag{8.14}$$

This proves (via Theorem 7.31) that the point we found graphically is in fact the optimal solution to the Problem 8.5.

2.2. Problems with an Infinite Number of Solutions. We'll study a specific linear programming problem with an infinite number of solutions by modifying the objective function in Example 8.1.

Example 8.5. Suppose the toy maker in Example 8.1 finds that it can sell planes for a profit of \$18 each instead of \$7 each. The new linear programming problem becomes:

$$\begin{cases}
\max & z(x_1, x_2) = 18x_1 + 6x_2 \\
s.t. & 3x_1 + x_2 \leq 120 \\
& x_1 + 2x_2 \leq 160 \\
& x_1 \leq 35 \\
& x_1 \geq 0 \\
& x_2 \geq 0
\end{cases} \tag{8.15}$$

Applying our graphical method for finding optimal solutions to linear programming problems yields the plot shown in Figure 8.2. The level curves for the function $z(x_1, x_2) = 18x_1 + 6x_2$ are *parallel* to one face of the polygon boundary of the feasible region. Hence, as we move further up and to the right in the direction of the gradient (corresponding to larger and larger values of $z(x_1, x_2)$) we see that there is not *one* point on the boundary of the feasible region that intersects that level set with greatest value, but instead a side of the polygon boundary described by the line $3x_1 + x_2 = 120$ where $x_1 \in [16, 35]$. Let:

$$S = \{(x_1, x_2 | 3x_1 + x_2 \leq 120,\ x_1 + 2x_2 \leq 160,\ x_1 \leq 35,\ x_1, x_2 \geq 0\}$$

that is, S is the feasible region of the problem. Then for any value of $x_1^* \in [16, 35]$ and any value x_2^* so that $3x_1^* + x_2^* = 120$, we will have $z(x_1^*, x_2^*) \geq z(x_1, x_2)$ for all $(x_1, x_2) \in S$. Since there are infinitely many values that x_1 and x_2 may take on, we see this problem has an infinite number of alternative optimal solutions.

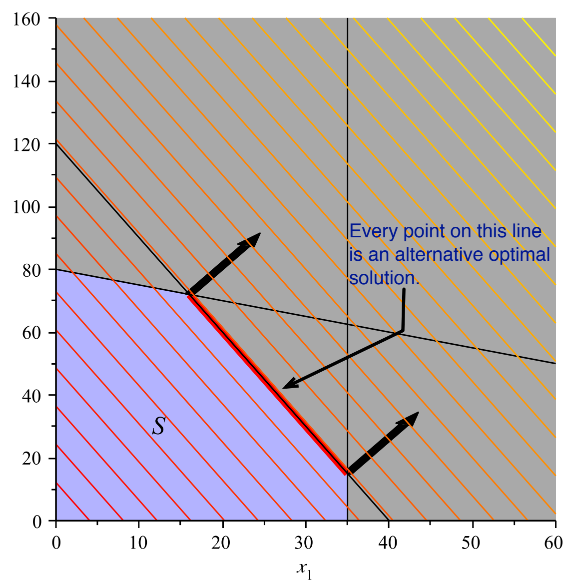


Figure 8.2. An example of infinitely many alternative optimal solutions in a linear programming problem. The level curves for $z(x_1, x_2) = 18x_1 + 6x_2$ are *parallel* to one face of the polygon boundary of the feasible region. Moreover, this side contains the points of greatest value for $z(x_1, x_2)$ inside the feasible region. Any combination of (x_1, x_2) on the line $3x_1 + x_2 = 120$ for $x_1 \in [16, 35]$ will provide the largest possible value $z(x_1, x_2)$ can take in the feasible region S.

Exercise 65. Modify the linear programming problem from Exercise 64 to obtain a linear programming problem with an infinite number of alternative optimal solutions. Solve the new problem and obtain a description for the set of alternative optimal solutions. [Hint: Just as in the example, x_1 will be bound between two value corresponding to a side of the polygon. Find those values and the constraint that is binding. This will provide you with a description of the form for any $x_1^* \in [a, b]$ and x_2^* is chosen so that $cx_1^* + dx_2^* = v$, the point (x_1^*, x_2^*) is an alternative optimal solution to the problem. Now you fill in values for a, b, c, d and v.]

2.3. Other Possibilities. In addition to the two scenarios above in which a linear programming problem has a unique solution or an infinite number of alternative optimal solutions, it is also possible that a linear programming problem can have:

(1) No solution, which occurs when the feasible region is empty,

(2) An unbounded solution, which can occur if the feasible region is an unbounded set.

Fortunately, we will not encounter either of those situations in our study of zero-sum games and so we blissfully ignore these possibilities.

3. A Linear Program for Zero-Sum Game Players

Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A})$ be a zero-sum game with $\mathbf{A} \in \mathbb{R}^{m \times n}$. Recall from Theorem 6.39 that the following are equivalent:

(1) There is a Nash equilibrium $(\mathbf{x}^*, \mathbf{y}^*)$ for $\mathcal{G}$
(2) The following equation holds:

$$v_1 = \max_{\mathbf{x}} \min_{\mathbf{y}} \mathbf{x}^T \mathbf{A} \mathbf{y} = \min_{\mathbf{y}} \max_{\mathbf{x}} \mathbf{x}^T \mathbf{A} \mathbf{y} = v_2 \tag{8.16}$$

(3) There exists a real number v and $\mathbf{x}^* \in \Delta_m$ and $\mathbf{y}^* \in \Delta_n$ so that:
  (a) $\sum_i \mathbf{A}_{ij} \mathbf{x}_i^* \geq v$ for $j = 1, \dots, n$ and
  (b) $\sum_j \mathbf{A}_{ij} \mathbf{y}_j^* \leq v$ for $i = 1, \dots, m$

The fact that $\mathbf{x}^* \in \Delta_m$ implies that:

$$\mathbf{x}_1^* + \cdots + \mathbf{x}^*{}_m = 1 \tag{8.17}$$

and $\mathbf{x}_i^* \geq 0$ for $i = 1, \dots, m$. Similar conditions will hold for $\mathbf{y}^*$.

If we look at Condition (3a) and incorporate the constraints imposed by $\mathbf{x}^* \in \Delta_m$, then we have what looks like the constraints of a linear programming problem. That is:

$$\begin{aligned}
\mathbf{A}_{11}\mathbf{x}_1^* + \cdots + \mathbf{A}_{m1}\mathbf{x}_m^* - v &\geq 0 \\
\mathbf{A}_{12}\mathbf{x}_1^* + \cdots + \mathbf{A}_{m2}\mathbf{x}_m^* - v &\geq 0 \\
&\vdots \\
\mathbf{A}_{1n}\mathbf{x}_1^* + \cdots + \mathbf{A}_{mn}\mathbf{x}_m^* - v &\geq 0 \\
\mathbf{x}_1^* + \cdots + \mathbf{x}_m^* &= 1 \\
\mathbf{x}_i^* &\geq 0 \quad i = 1, \dots, m
\end{aligned} \tag{8.18}$$

In this set of constraints we have $m+1$ variables: $\mathbf{x}_1^*, \dots, \mathbf{x}_m^*$ and v, the value of the game. We know that Player 1 (the row player) is a value *maximizer*, therefore Player 1 is interested in solving the linear programming problem:

$$\begin{aligned}
\max \;& v \\
s.t. \;& \mathbf{A}_{11}x_1 + \cdots + \mathbf{A}_{m1}x_m - v \geq 0 \\
& \mathbf{A}_{12}x_1 + \cdots + \mathbf{A}_{m2}x_m - v \geq 0 \\
& \qquad \vdots \\
& \mathbf{A}_{1n}x_1 + \cdots + \mathbf{A}_{mn}x_m - v \geq 0 \\
& x_1 + \cdots + x_m = 1 \\
& x_i \geq 0 \quad i = 1, \dots, m
\end{aligned} \tag{8.19}$$

By a similar argument, we know that Player 2's equilibrium strategy $\mathbf{y}^*$ is constrained by:

$$
\begin{aligned}
\mathbf{A}_{11}\mathbf{y}_1^* + \cdots + \mathbf{A}_{1n}\mathbf{y}_n^* - v &\leq 0\\
\mathbf{A}_{21}\mathbf{y}_1^* + \cdots + \mathbf{A}_{2n}\mathbf{y}_n^* - v &\leq 0\\
&\vdots\\
\mathbf{A}_{m1}\mathbf{y}_1^* + \cdots + \mathbf{A}_{mn}\mathbf{y}_n^* - v &\leq 0\\
\mathbf{y}_1^* + \cdots + \mathbf{y}_n^* &= 1\\
\mathbf{y}_i^* &\geq 0 \quad i = 1, \dots, n
\end{aligned}
\tag{8.20}
$$

We know that Player 2 (the column player) is a value *minimizer*, therefore Player 2 is interested in solving the linear programming problem:

$$
\begin{aligned}
\min\;\; & v\\
s.t.\;\; & \mathbf{A}_{11}y_1 + \cdots + \mathbf{A}_{1n}y_n - v \leq 0\\
& \mathbf{A}_{21}y_1 + \cdots + \mathbf{A}_{2n}y_n - v \leq 0\\
& \qquad\qquad \vdots\\
& \mathbf{A}_{m1}y_1 + \cdots + \mathbf{A}_{mn}y_n - v \leq 0\\
& y_1 + \cdots + y_n = 1\\
& y_i \geq 0 \quad i = 1, \dots, n
\end{aligned}
\tag{8.21}
$$

EXAMPLE 8.6. Consider the game from Example 6.2. The payoff matrix for Player 1 is given as:

$$
\mathbf{A} = \begin{bmatrix} -15 & -35 & 10\\ -5 & 8 & 0\\ -12 & -36 & 20 \end{bmatrix}
$$

This is a zero sum game, so the payoff matrix for Player 2 is simply the negation of this matrix. The linear programming problem for Player 1 is:

$$
\begin{aligned}
\max\;\; & v\\
s.t.\;\; & -15x_1 - 5x_2 - 12x_3 - v \geq 0\\
& -35x_1 + 8x_2 - 36x_3 - v \geq 0\\
& 10x_1 + 20x_3 - v \geq 0\\
& x_1 + x_2 + x_3 = 1\\
& x_1, x_2, x_3 \geq 0
\end{aligned}
\tag{8.22}
$$

Notice, we simply work our way down each column of the matrix $\mathbf{A}$ in forming the constraints of the linear programming problem. To form the problem for Player 2, we work our way across the rows of $\mathbf{A}$ and obtain:

$$
\begin{aligned}
&\min \;\; v\\
&\quad s.t. \;\; -15y_1 - 35y_2 + 10y_3 - v \leq 0\\
&\qquad\quad -5y_1 + 8y_2 - v \leq 0\\
&\qquad\quad -12y_1 - 36y_2 + 20y_3 - v \leq 0\\
&\qquad\quad y_1 + y_2 + y_3 = 1\\
&\qquad\quad y_1, y_2, y_3 \geq 0
\end{aligned}
\tag{8.23}
$$

Exercise 66. Construct the two linear programming problems for Bradley and von Kluge in the Battle of Avranches.

4. Matrix Notation, Slack and Surplus Variables for Linear Programming

You will recall from your matrices class (Math 220) that matrices can be used as a short hand way to represent linear equations. Consider the following system of equations:

$$
\begin{cases}
a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1\\
a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2\\
\qquad\qquad\vdots\\
a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n = b_m
\end{cases}
\tag{8.24}
$$

Then we can write this in matrix notation as:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \tag{8.25}$$

where $\mathbf{A}_{ij} = a_{ij}$ for $i = 1, \dots, m$, $j = 1, \dots, n$ and $\mathbf{x}$ is a column vector in $\mathbb{R}^n$ with entries x_j $(j = 1, \dots, n)$ and $\mathbf{b}$ is a column vector in $\mathbb{R}^m$ with entries b_i $(i = 1 \dots, m)$. Obviously, if we replace the equalities in Expression 8.24 with inequalities, we can also express systems of inequalities in the form:

$$\mathbf{A}\mathbf{x} \leq \mathbf{b} \tag{8.26}$$

Using this representation, we can write our general linear programming problem using matrix and vector notation. Expression 8.1 can be written as:

$$
\begin{cases}
\max \;\; z(\mathbf{x}) = \mathbf{c}^T\mathbf{x}\\
\qquad s.t. \;\; \mathbf{A}\mathbf{x} \leq \mathbf{b}\\
\qquad\qquad \mathbf{H}\mathbf{x} = \mathbf{r}
\end{cases}
\tag{8.27}
$$

Example 8.7. Consider a zero-sum game with payoff matrix $\mathbf{A} \in \mathbb{R}^{m\times n}$. We can write the problem that arises for Player 1 in matrix notation. The decision variables are $\mathbf{x} \in \mathbb{R}^{m\times 1}$ and $v \in \mathbb{R}$. We can write these decision variables as a single vector $\mathbf{z}$:

$$\mathbf{z} = \left[\begin{array}{c}\mathbf{x}\\ \hline v\end{array}\right]$$

Let:

$$\mathbf{c} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix}$$

Then our objective function is $\mathbf{c}^T\mathbf{z} = v$. Our inequality constraints have the form:

$$\left[\mathbf{A}^T | -\mathbf{e}\right] \mathbf{z} \geq 0$$

Here $\mathbf{e} = [1, 1, \dots, 1]^T$ is a column vector of ones with n elements to make the augmented matrix meaningful. Our equality constraints are $\mathbf{x}_1 + \cdots + \mathbf{x}_m = 1$. This can be written as:

$$\left[\mathbf{e}^T | 0\right] \mathbf{z} = 1$$

Again, $\mathbf{e}$ is an appropriately sized vector of ones (this time with m elements). The resulting linear program is then:

$$\begin{aligned} \max \;\; & \mathbf{c}^T\mathbf{z} \\ s.t. \;\; & \left[\mathbf{A}^T | -\mathbf{e}\right] \mathbf{z} \geq \mathbf{0} \\ & \left[\mathbf{e}^T | 0\right] \mathbf{z} = 1 \\ & \mathbf{e}_i^T\mathbf{z} \geq 0 \quad i = 1, \dots, m \end{aligned}$$

The last constraint simply says that $\mathbf{x}_i \geq 0$ and since v is the $m+1^{\text{st}}$ variable, we do not constraint v to be positive.

Exercise 67. Construct the matrix form of the linear program for Player 2 in a zero-sum game.

4.1. Standard Form, Slack and Surplus Variables.

Definition 8.8 (Standard Form). A linear programming problem is in *standard form* if it is written as:

$$(8.28) \quad \begin{cases} \max \;\; z(\mathbf{x}) = \mathbf{c}^T\mathbf{x} \\ \qquad\quad s.t. \;\; \mathbf{A}\mathbf{x} = \mathbf{b} \\ \qquad\qquad\;\; \mathbf{x} \geq 0 \end{cases}$$

Remark 8.9. It is relatively easy to convert any inequality constraint into an equality constraint. Consider the inequality constraint:

$$(8.29) \quad a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \leq b_i$$

We can add a new *slack variable* s_i to this constraint to obtain:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n + s_i = b_i$$

Obviously this slack variable $s_i \geq 0$. The slack variable then becomes just another variable whose value we must discover as we solve the linear program for which Expression 8.29 is a constraint.

We can deal with constraints of the form:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n \geq b_i \tag{8.30}$$

in a similar way. In this case we subtract a surplus variable s_i to obtain:

$$a_{i1}x_1 + a_{i2}x_2 + \cdots + a_{in}x_n - s_i = b_i$$

Again, we must have $s_i \geq 0$.

Example 8.10. Consider the linear programming problem:

$$\begin{cases} \max & z(x_1, x_2) = 2x_1 - x_2 \\ s.t. & x_1 - x_2 \leq 1 \\ & 2x_1 + x_2 \geq 6 \\ & x_1, x_2 \geq 0 \end{cases}$$

This linear programming problem can be put into standard form by using both a slack and surplus variable.

$$\begin{cases} \max & z(x_1, x_2) = 2x_1 - x_2 \\ s.t. & x_1 - x_2 + s_1 = 1 \\ & 2x_1 + x_2 - s_2 = 6 \\ & x_1, x_2, s_1, s_2 \geq 0 \end{cases}$$

5. Solving Linear Programs by Computer

Solving linear programs can be accomplished by using the Simplex Algorithm or an Interior Point Method [BJS04]. In general, Linear Programming should be a pre-requisite for Game Theory, however we do not have this luxury. Teaching the Simplex Method is relatively straightforward, but it would be better for your to understand the method than to simply memorize a collection of instructions (that's what computers are for). To that end, we will use a computer to find the solution of Linear Programs that arise from our games. There are several computer programs that will solve linear programming problems for you. We'll use Matlab, which is on most computers in Penn State Computer labs. You'll have to make sure that the Matlab Optimization Toolbox is installed.

5.1. Matlab. Matlab (http://www.mathworks.com) is a power tool used by engineers and applied mathematicians for numerical computations. We can solve linear programs in Matlab using the function `linprog`. By default, Matlab assumes it is solving a *minimization* problem. Specifically, Matlab assumes it is solving the following minimization problem:

$$\begin{cases} \min & \mathbf{c}^T\mathbf{x} \\ s.t. & \mathbf{Ax} \leq \mathbf{b} \\ & \mathbf{Hx} = \mathbf{r} \\ & \mathbf{x} \leq \mathbf{u} \\ & \mathbf{x} \geq \mathbf{l} \end{cases} \tag{8.31}$$

Here, $\mathbf{l}$ is a vector of *lower bounds* for the vector $\mathbf{x}$ and $\mathbf{u}$ is a vector of *upper bounds* for the vector $\mathbf{x}$.

In Matlab, almost all input is in the form of matrices. Thus we enter the vector for the objective function $\mathbf{c}$, the matrix $\mathbf{A}$ and vector $\mathbf{b}$ for constraints of the form $\mathbf{Ax} \leq \mathbf{b}$, a matrix $\mathbf{H}$ and vector $\mathbf{r}$ for constraints of the form $\mathbf{Hx} = \mathbf{r}$ and finally the two vectors $\mathbf{l}$ and $\mathbf{u}$ for constraints of the form $\mathbf{x} \geq \mathbf{l}$ and $\mathbf{x} \leq \mathbf{u}$. If a variable in unconstrained, then we can use the value `inf` to indicate an infinite bound. We can solve the Battle of the Networks problem for Players 1 and 2 using Matlab and confirm our saddle point solution from Example 6.2. Recall the game matrix for Battle of the Networks is:

$$\mathbf{G} = \begin{bmatrix} -15 & -35 & 10 \\ -5 & 8 & 0 \\ -12 & -36 & 20 \end{bmatrix}$$

We'll use $\mathbf{G}$ so that we can reserve $\mathbf{A}$ for the inequality matrix for Matlab. Using Equations 8.22 and 8.23, we'll have the linear programming problem for Player 1:

$$\begin{cases} \max \;\; v = 0x_1 + 0x_2 + 0x_3 + v \\ \;\; s.t. \;\; -15x_1 - 5x_2 - 12x_3 - v \geq 0 \\ \qquad\;\; -35x_1 + 8x_2 - 36x_3 - v \geq 0 \\ \qquad\;\; 10x_1 + 0x_2 + 20x_3 - v \geq 0 \\ \qquad\;\; x_1 + x_2 + x_3 + 0v = 1 \\ \qquad\;\; x_1, x_2, x_3 \geq 0 \end{cases}$$

This problem is not in a format Matlab likes, we must convert the greater-than ($\geq$) constraints to less-than ($\leq$) constraints. We must also convert this to a minimization problem. We can do this by multiplying the objective by -1 and each $\geq$ constraint by -1 to obtain:

$$\begin{cases} \min \;\; -v = 0x_1 + 0x_2 + 0x_3 - v \\ \;\; s.t. \;\; 15x_1 + 5x_2 + 12x_3 + v \leq 0 \\ \qquad\;\; 35x_1 - 8x_2 + 36x_3 + v \leq 0 \\ \qquad\;\; -10x_1 + 0x_2 - 20x_3 + v \leq 0 \\ \qquad\;\; x_1 + x_2 + x_3 + 0v = 1 \\ \qquad\;\; x_1, x_2, x_3 \geq 0 \end{cases}$$

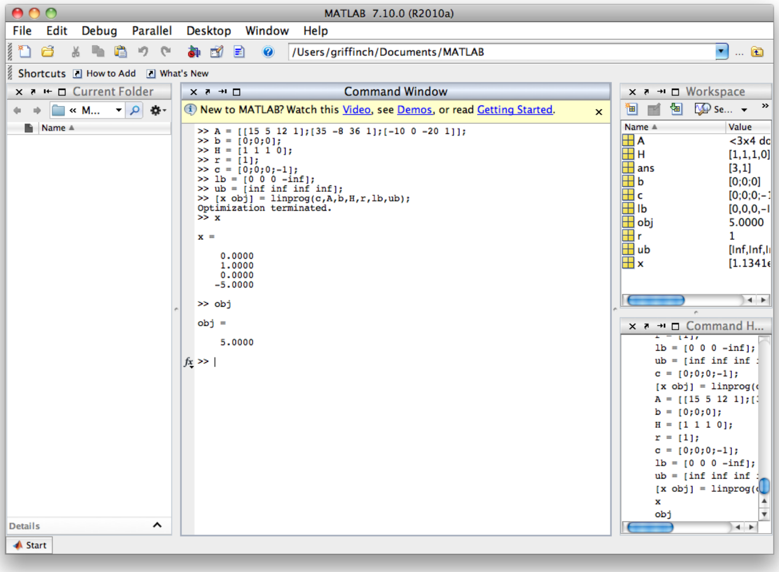
We can read the matrices and vectors for Player 1 as:

$$\mathbf{A} = \begin{bmatrix} 15 & 5 & 12 & 1 \\ 35 & -8 & 36 & 1 \\ -10 & 0 & -20 & 1 \end{bmatrix} \quad \mathbf{b} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

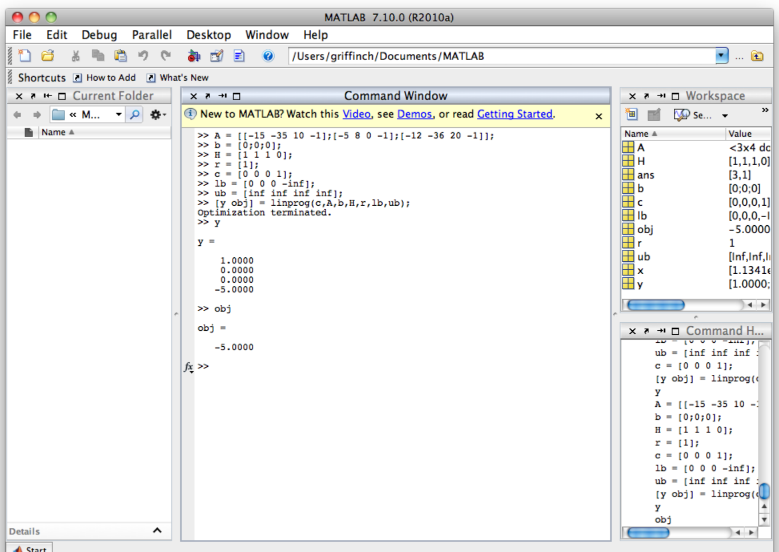
$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 0 \end{bmatrix} \quad \mathbf{r} = \begin{bmatrix} 1 \end{bmatrix}$$

$$\mathbf{c} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -1 \end{bmatrix} \quad \mathbf{l} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ -\infty \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} +\infty \\ +\infty \\ +\infty \\ +\infty \end{bmatrix}$$

Note our lower bound for v is $-\infty$ and our upper bound for all variables is $+\infty$. Though we should note that since $x_1 + x_2 + x_3 = 1$, these values will automatically be less than 1. The Matlab solution is shown in Figure 8.3 (Player 1). We can also construct the Matlab



(a) Player 1



(b) Player 2

Figure 8.3. We solve for the strategy for Player 1 in the Battle of the Networks. Player 1 maximizes v subject to the constraints given in Problem 8.19. The result is Player 1 should play strategy 2 all the time. We also solve for the strategy for Player 2 in the Battle of the Networks. Player 2 minimizes v subject to the constraints given in Problem 8.21. The result is Player 2 should play strategy 1 all of the time. This agrees with our saddle-point solution.

problem for Player 2. Player 2's problem will be

$$\begin{cases} \min \ 0y_1 + 0y_2 + 0y_3 + v \\ s.t. \ -15y_1 - 35y_2 + 10y_3 - v \leq 0 \\ \quad -5y_1 + 8y_2 - 0y_3 - v \leq 0 \\ \quad -12y_1 - 36y_2 + 20y_3 - v \leq 0 \\ \quad y_1 + y_2 + y_3 + 0v = 1 \\ \quad y_1, y_2, y_3 \geq 0 \end{cases}$$

We can construct the matrices and vectors for this problem just as we did before and use Matlab to find the optimal solution. This is shown in Figure 8.3 (Player 2). Notice that it's a lot easier to solve for Player 2's strategy because it's already in a Matlab approved form.

You'll note that according to Matlab, the Nash equilibrium is:

$$\mathbf{x} = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$\mathbf{y} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

That is, Player 1 should always play pure strategy 2, while Player 2 should play pure strategy 1. This agrees exactly with our observation of the minimax value in Figure 6.1 from Example 6.2 in which we concluded that the minimax and maximin values of the game matrix corresponded precisely to when Player 1 played pure strategy 2 and Player 2 played pure strategy 1 (element $(2, 1)$ in the matrix $\mathbf{G}$).

5.2. Closing Remarks. In a perfect world, there would be time to teach you everything you want to know about the Simplex Algorithm (or any other method) for solving linear programs. If you're interested in these types of problems, you should consider taking Math 484 (Linear Programming) or getting a good book on the subject.

6. Duality and Optimality Conditions for Zero-Sum Game Linear Programs

THEOREM 8.11. *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A})$ be a zero-sum two player game with $\mathbf{A} \in \mathbb{R}^{m\times n}$. Then the linear program for Player 1:*

$$\begin{aligned} \max \ & v \\ s.t. \ & \mathbf{A}_{11}x_1 + \cdots + \mathbf{A}_{m1}x_m - v \geq 0 \\ & \mathbf{A}_{12}x_1 + \cdots + \mathbf{A}_{m2}x_m - v \geq 0 \\ & \qquad \vdots \\ & \mathbf{A}_{1n}x_1 + \cdots + \mathbf{A}_{mn}x_m - v \geq 0 \\ & x_1 + \cdots + x_m - 1 = 0 \\ & x_i \geq 0 \quad i = 1, \dots, m \end{aligned}$$

has optimal solution $(x_1, \dots, x_m)$ *if and only if there exists Lagrange multipliers:* $y_1, \dots, y_n$, $\rho_1, \dots, \rho_m$ *and* ν *and surplus variables* $s_1, \dots, s_n$ *such that:*

$$\textit{Primal Feasibility}: \begin{cases} \displaystyle\sum_{i=1}^{m} \mathbf{A}_{ij} x_i - v - s_j = 0 \quad j = 1, \dots, n \\ \displaystyle\sum_{i=1}^{m} x_i = 1 \\ x_i \geq 0 \quad \textit{for } i = 1, \dots, m \\ s_j \geq 0 \quad \textit{for } j = 1, \dots, n \\ v \quad \textit{unrestricted} \end{cases}$$

$$\textit{Dual Feasibility}: \begin{cases} \displaystyle\sum_{j=1}^{n} \mathbf{A}_{ij} y_j - \nu + \rho_i = 0 \quad i = 1, \dots, m \\ \displaystyle\sum_{j=1}^{n} y_j = 1 \\ y_j \geq 0 \quad j = 1, \dots, n \\ \rho_i \geq 0 \quad i = 1, \dots, m \\ \nu \quad \textit{unrestricted} \end{cases}$$

$$\textit{Complementary Slackness}: \begin{cases} y_j s_j = 0 \quad j = 1, \dots, n \\ \rho_i x_i = 0 \quad i = 1, \dots, m \end{cases}$$

Proof. We'll begin by showing the statements that make up Primal Feasibility must hold. Clearly v is unrestricted and $x_i \geq 0$ for $i = 1, \dots, m$. The fact that $x_1 + \cdots + x_m = 1$ is also clear from the problem. We can rewrite each constraint of the form:

$$\mathbf{A}_{1j} x_1 + \cdots + \mathbf{A}_{mj} x_m - v \geq 0 \tag{8.32}$$

where $j = 1, \dots, n$ as:

$$\mathbf{A}_{1j} x_1 + \cdots + \mathbf{A}_{mj} x_m - v + s_j = 0 \tag{8.33}$$

where $s_j \geq 0$. Each variable s_j is a surplus variable. Thus it's clear that if $x_1, \dots, x_m$ is a feasible solution, then at least variables $s_1, \dots, s_n \geq 0$ exist and Primal Feasibility holds.

Let us re-write the constraints of the form in Expression 8.32 as:

$$-\mathbf{A}_{1j} x_1 - \cdots - \mathbf{A}_{mj} x_m + v \leq 0 \quad j = 1, \dots, n \tag{8.34}$$

and each non-negativity constraint as:

$$-x_i \leq 0 \quad i = 1, \dots, m \tag{8.35}$$

We know that each affine function is both concave and convex and therefore, by Theorem 7.31 (the Karush-Kuhn-Tucker theorem), there are Lagrange multipliers $y_1, \dots, y_n$ corresponding to the constraints of the form in Expression 8.34 and Lagrange multipliers $\rho_1, \dots, \rho_m$ corresponding to the constraints of the form in Expression 8.35. Lastly, there is a Lagrange multiplier ν corresponding to the constraint:

$$x_1 + x_2 + \cdots + x_m - 1 = 0 \tag{8.36}$$

We know from Theorem 7.31 that:

$$\begin{aligned}
&y_j \geq 0 \quad j = 1, \dots, n \\
&\rho_i \geq 0 \quad i = 1, \dots, m \\
&\nu \quad \text{unrestricted}
\end{aligned}$$

Before showing that

$$\sum_{j=1}^{n} \mathbf{A}_{ij} y_j - \nu + \rho_i = 0 \quad i = 1, \dots, m \tag{8.37}$$

$$\sum_{j=1}^{n} y_j = 1 \tag{8.38}$$

holds, we show that Complementary Slackness holds. To see this, note that by Theorem 7.31, we know that:

$$\begin{aligned}
&y_j \left(-\mathbf{A}_{1j} x_1 - \cdots - \mathbf{A}_{mj} x_m + v\right) = 0 \quad j = 1, \dots, n \\
&\rho_i(-x_i) = 0 \quad i = 1, \dots, m
\end{aligned}$$

If $\rho_i(-x_i) = 0$, then $-\rho_i x_i = 0$ and therefore $\rho_i x_i = 0$. From Equation 8.33:

$$\mathbf{A}_{1j} x_1 + \cdots + \mathbf{A}_{mj} x_m - v + s_j = 0 \implies s_j = -\mathbf{A}_{1j} x_1 - \cdots - \mathbf{A}_{mj} x_m + v$$

Therefore, we can write:

$$y_j \left(-\mathbf{A}_{1j} x_1 - \cdots - \mathbf{A}_{mj} x_m + v\right) = 0 \implies y_j(s_j) = 0 \quad j = 1, \dots, n$$

Thus we have shown:

$$y_j s_j = 0 \quad j = 1, \dots, n \tag{8.39}$$

$$\rho_i x_i = 0 \quad i = 1, \dots, m \tag{8.40}$$

holds and thus the statements making up Complementary Slackness must be true.

We now complete the proof by showing that Dual Feasibility holds. Let:

$$g_j(x_1, \dots, x_m, v) = -\mathbf{A}_{1j} x_1 - \cdots - \mathbf{A}_{mj} x_m + v \quad (j = 1, \dots, n) \tag{8.41}$$

$$f_i(x_1, \dots, x_m, v) = -x_i \quad (i = 1, \dots, m) \tag{8.42}$$

$$h(x_1, \dots, x_m, v) = x_1 + x_2 + \cdots + x_m - 1 \tag{8.43}$$

$$z(x_1, \dots, x_m, v) = v \tag{8.44}$$

Then we can apply Theorem 7.31 and see that:

$$\nabla z - \sum_{j=1}^{n} y_j \nabla g_j(x_1, \dots, x_n, n) - \sum_{i=1}^{m} \rho_i \nabla f_i(x_1, \dots, x_m, v) - \nu \nabla h(x_1, \dots, x_m, v) = 0 \tag{8.45}$$

Working out the gradients yields:

$$\nabla z(x_1, \ldots, x_m, v) = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} \in \mathbb{R}^{(m+1)\times 1} \tag{8.46}$$

$$\nabla h(x_1, \ldots, x_m, v) = \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 0 \end{bmatrix} \in \mathbb{R}^{(m+1)\times 1} \tag{8.47}$$

$$\nabla f_i(x_1, \ldots, x_m, v) = -\mathbf{e}_i \in \mathbb{R}^{(m+1)\times 1} \tag{8.48}$$

and

$$\nabla g_j(x_1, \ldots, x_m, v) = \begin{bmatrix} -\mathbf{A}_{1j} \\ -\mathbf{A}_{2j} \\ \vdots \\ -\mathbf{A}_{mj} \\ 1 \end{bmatrix} \in \mathbb{R}^{(m+1)\times 1} \tag{8.49}$$

Before proceeding, note that in computing $\nabla f_i(x_1, \ldots, x_m, v)$, $(i = 1, \ldots, m)$, we will have $-\mathbf{e}_1, \ldots, \mathbf{e}_m \in \mathbb{R}^{(m+1)\times 1}$. Thus, we will *never* see the vector:

$$-\mathbf{e}_{m+1} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ -1 \end{bmatrix} \in \mathbb{R}^{(m+1)\times 1}$$

because there is *no* function $f_{m+1}(x_1, \ldots, x_m, v)$. We can now rewrite Expression 8.45 as:

$$\begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} - \left(\sum_{j=1}^{n} y_j \begin{bmatrix} -\mathbf{A}_{1j} \\ -\mathbf{A}_{2j} \\ \vdots \\ -\mathbf{A}_{mj} \\ 1 \end{bmatrix}\right) - \left(\sum_{i=1}^{m} \rho_i(-\mathbf{e}_i)\right) - \nu \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \\ 0 \end{bmatrix} = \mathbf{0} \tag{8.50}$$

Consider element i the first m terms of these vectors. Adding term-by-term we have:

$$0 + \sum_{j=1}^{n} \mathbf{A}_{ij} y_j + \rho_i - \nu = 0 \tag{8.51}$$

This is the i^{th} row of vector that results from adding the terms on the left-hand-side of Expression 8.50. Now consider row $m+1$. We have:

$$1 - \sum_{j=1}^{n} y_j + 0 + 0 = 0 \tag{8.52}$$

From these two equations, we conclude that:

$$\sum_{j=1}^{n} \mathbf{A}_{ij} y_j + \rho_i - \nu = 0 \tag{8.53}$$

$$\sum_{j=1}^{n} y_j = 1 \tag{8.54}$$

Thus, we have shown that Dual Feasibility holds. Necessity and sufficiency of the statement follows at once from Theorem 7.31. This completes the proof. □

THEOREM 8.12. *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A})$ be a zero-sum two player game with $\mathbf{A} \in \mathbb{R}^{m \times n}$. Then the linear program for Player 2:*

$$\begin{aligned}
\min\ & \nu \\
s.t.\ & \mathbf{A}_{11} y_1 + \cdots + \mathbf{A}_{1n} y_n - \nu \leq 0 \\
& \mathbf{A}_{21} y_1 + \cdots + \mathbf{A}_{2n} y_n - \nu \leq 0 \\
& \qquad \vdots \\
& \mathbf{A}_{m1} y_1 + \cdots + \mathbf{A}_{mn} y_n - \nu \leq 0 \\
& y_1 + \cdots + y_n - 1 = 0 \\
& y_i \geq 0 \quad i = 1, \dots, m
\end{aligned}$$

has optimal solution $(y_1, \dots, y_n)$ if and only if there exists Lagrange multipliers: $x_1, \dots, x_m$, $s_1, \dots, s_n$ and v and slack variables $\rho_1, \dots, \rho_m$ such that:

$$\textit{Primal Feasibility}: \begin{cases}
\displaystyle\sum_{j=1}^{n} \mathbf{A}_{ij} y_j - \nu + \rho_i = 0 \quad i = 1, \dots, m \\
\displaystyle\sum_{j=1}^{n} y_j = 1 \\
y_j \geq 0 \quad j = 1, \dots, n \\
\rho_i \geq 0 \quad i = 1, \dots, m \\
\nu \quad \textit{unrestricted}
\end{cases}$$

$$
\textit{Dual Feasibility} : \begin{cases} \displaystyle\sum_{i=1}^{m} \mathbf{A}_{ij} x_i - v - s_j = 0 \quad j = 1, \dots, n \\ \displaystyle\sum_{i=1}^{m} x_i = 1 \\ x_i \geq 0 \quad \textit{for } i = 1, \dots, m \\ s_j \geq 0 \quad \textit{for } j = 1, \dots, n \\ v \quad \textit{unrestricted} \end{cases}
$$

$$
\textit{Complementary Slackness} : \begin{cases} y_j s_j = 0 & j = 1, \dots, n \\ \rho_i x_i = 0 & i = 1, \dots, m \end{cases}
$$

Exercise 68. Prove Theorem 8.12

Remark 8.13. Theorems 8.11 and 8.12 say something very important. They say that the Karush-Kuhn-Tucker conditions for the Linear Programming problems for Player 1 and Player 2 in a zero-sum game are *identical* (only primal and dual feasibility are exchanged).

Definition 8.14. Let P and D be linear programming problems. If the KKT conditions for Problem P are equivalent to the KKT conditions for Problem D with Primal Feasibility and Dual Feasibility exchanged, then Problem P and Problem D are called *dual linear programming problems.*

Proposition 8.15. *The linear programming problem for Player 1 is the dual problem of the linear programming problem for Player 2 in a zero-sum two player game* $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A})$ *with* $\mathbf{A} \in \mathbb{R}^{m \times n}$.

There is a very deep theorem about dual linear programming problems, which is beyond the scope of this course. (We prove it in Math 484.) We state it and make use of it to prove the minimax theorem in a totally new way.

Theorem 8.16 (Strong Duality Theorem). *Let P and D be dual linear programming problems (like the linear programming problems of Players 1 and 2 in a zero-sum game). Then either:*

1. *Both P and D have a solution and at optimality, the objective function value for Problem P is identical to the objective function value for Problem D.*
2. *Problem P has no solution because it is unbounded and Problem D has no solution because it is infeasible.*
3. *Problem D has no solution because it is unbounded and Problem P has no solution because it is infeasible.*
4. *Both Problem P and Problem D are infeasible.*

Theorem 8.17 (Minimax Theorem (redux)). *Let* $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A})$ *be a zero-sum two player game with* $\mathbf{A} \in \mathbb{R}^{m \times n}$*, then there exists a Nash equilibrium* $(\mathbf{x}^*, \mathbf{y}^*) \in \Delta$*. Furthermore, for every Nash equilibrium pair* $(\mathbf{x}^*, \mathbf{y}^*) \in \Delta$ *there is one value* $v^* = \mathbf{x}^{*T} \mathbf{A} \mathbf{y}^*$.

Sketch of Proof. Let Problem P_1 and Problem P_2 be the linear programming problems for Player 1 and 2 respectively that arise from $\mathcal{G}$. That is:

$$P_1 \begin{cases} \max & v \\ s.t. & \mathbf{A}_{11}x_1 + \cdots + \mathbf{A}_{m1}x_m - v \geq 0 \\ & \mathbf{A}_{12}x_1 + \cdots + \mathbf{A}_{m2}x_m - v \geq 0 \\ & \vdots \\ & \mathbf{A}_{1n}x_1 + \cdots + \mathbf{A}_{mn}x_m - v \geq 0 \\ & x_1 + \cdots + x_m - 1 = 0 \\ & x_i \geq 0 \quad i = 1, \dots, m \end{cases}$$

$$P_2 \begin{cases} \min & \nu \\ s.t. & \mathbf{A}_{11}y_1 + \cdots + \mathbf{A}_{1n}y_n - \nu \leq 0 \\ & \mathbf{A}_{21}y_1 + \cdots + \mathbf{A}_{2n}y_n - \nu \leq 0 \\ & \vdots \\ & \mathbf{A}_{m1}y_1 + \cdots + \mathbf{A}_{mn}y_n - \nu \leq 0 \\ & y_1 + \cdots + y_n - 1 = 0 \\ & y_i \geq 0 \quad i = 1, \dots, m \end{cases}$$

These linear programming problems are dual and therefore if Problem P_1 has a solution, then so does problem P_2. More importantly, at these optimal solutions $(\mathbf{x}^*, v^*), (\mathbf{y}^*, \nu^*)$ we know that $v^* = \nu^*$ as the objective function values must be equal by Theorem 8.16.

Consider Problem P_1: we know that $(x_1, \dots, x_m) \in \Delta_m$ and therefore, this space is bounded. The value v clearly cannot exceed $\max_{ij} \mathbf{A}_{ij}$ as a result of the constraints and the fact that $x_i \in [0, 1]$ for $i = 1, \dots, m$. Obviously, v can be made as small as we like, but this won't happen since this is a maximization problem. The fact that v is bounded from above and $(x_1, \dots, x_m) \in \Delta_m$ and P_1 is a maximization problem (on v) implies that there is at least one solution $(\mathbf{x}^*, v^*)$ to Problem P_1. In this case, there is a solution (y^*, ν^*) to Problem P_2 and $v^* = \nu^*$. Since the constraints for Problem P_1 and Problem P_2 were taken from Theorem 6.39, we know that $(\mathbf{x}^*, \mathbf{y}^*)$ is a Nash equilibrium and therefore such an equilibrium must exist.

Furthermore, while we have not proved this explicitly, one can prove that if $(\mathbf{x}^*, \mathbf{y}^*)$ is a Nash equilibrium, then it must be a part of solutions $(\mathbf{x}^*, v^*)$, $(\mathbf{y}^*, \nu^*)$ to Problems P_1 and P_2. Thus, any two equilibrium solutions are simply *alternative optimal solutions* to P_1 and P_2 respectively. Thus, for any Nash equilibrium pair we have:

$$\nu^* = v^* = \mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* \tag{8.55}$$

This completes the proof sketch. □

Remark 8.18 (A remark on Complementary Slackness). Consider the KKT conditions for Players 1 and 2 (Theorems 8.11 and 8.12). Suppose (for the sake of argument) that in an optimal solution of the problem for Player 1, $s_j > 0$. Then, it follows that $y_j = 0$ by complementary slackness. We can understand this from a game theoretic perspective. The expression:

$$\mathbf{A}_{1j}x_1 + \cdots + \mathbf{A}_{mj}x_m$$

is the expected payoff to Player 1 if Player 2 plays column j. If $s_j > 0$, then:

$$\mathbf{A}_{1j}x_1 + \cdots + \mathbf{A}_{mj}x_m > v$$

But that means that if Player 2 *ever* played column j, then Player 1 could do better than the equilibrium value of the game, thus Player 2 has no incentive to ever play this strategy and the result is that $y_j = 0$ (as required by complementary slackness).

EXERCISE 69. Use the logic from the preceding remark to argue that $x_i = 0$ when $\rho_i > 0$ for Player 2.

REMARK 8.19. The connection between zero-sum games and linear programming is substantially deeper than the previous theorem suggests. Luce and Raiffa [**LR89**] show the equivalence between Linear Programming and Zero-Sum games by demonstrating (as we have done) that for each zero-sum game there is a linear programming problem whose solution yields an equilibrium and for each linear programming problem there is a zero-sum game whose equilibrium solution yields an optimal solution.

In the next chapter, we'll continue our discussion of the equivalence of games and optimization problems by investigating general sum two-player games.

CHAPTER 9

Quadratic Programs and General Sum Games

1. Introduction to Quadratic Programming

DEFINITION 9.1 (Quadratic Programming Problem). Let

(1) $\mathbf{Q} \in \mathbb{R}^{n \times n}$,
(2) $\mathbf{A} \in \mathbb{R}^{m \times n}$,
(3) $\mathbf{H} \in \mathbb{R}^{l \times n}$,
(4) $\mathbf{b} \in \mathbb{R}^{m \times 1}$,
(5) $\mathbf{r} \in \mathbb{R}^{l \times 1}$ and
(6) $\mathbf{c} \in \mathbb{R}^{n \times 1}$.

Then a quadratic (maximization) programming problem is:

$$QP \begin{cases} \max & \mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ s.t. & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{H}\mathbf{x} = \mathbf{r} \end{cases} \tag{9.1}$$

EXAMPLE 9.2. Example 7.1 is an instance of a quadratic programming problem. Recall we had:

$$\begin{cases} \max & A(x, y) = xy \\ s.t. & 2x + 2y = 100 \\ & x \geq 0 \\ & y \geq 0 \end{cases}$$

We can write this as:

$$\begin{cases} \max & \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 0 & 1/2 \\ 1/2 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\ s.t. & \begin{bmatrix} 2 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 100 \\ & \begin{bmatrix} x \\ y \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{cases}$$

Obviously, we can put this problem in precisely the format given in Expression 9.1, if so desired.

REMARK 9.3. Quadratic programs are just a special instance of nonlinear (or mathematical) programming problems. There are many applications for quadratic programs that are beyond the scope of these notes. There are also many solution techniques for quadratic programs, which are also beyond the scope of these notes. Interested readers should consult [**BSS06**] for details.

2. Solving QP's by Computer

In this section we show how to solve quadratic programming problems in both Matlab and Maple. Unlike linear programming problems, there is no convenient web-based quadratic programming solver available.

2.1. Matlab. Matlab assumes it is solving the following problem:

$$QP \begin{cases} \min & \dfrac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ s.t. & \mathbf{A}\mathbf{x} \leq \mathbf{b} \\ & \mathbf{H}\mathbf{x} = \mathbf{r} \\ & \mathbf{l} \leq \mathbf{x} \leq \mathbf{u} \end{cases} \tag{9.2}$$

The user will supply the matrices and vectors $\mathbf{Q}$, $\mathbf{c}$, $\mathbf{A}$, $\mathbf{b}$, $\mathbf{H}$, $\mathbf{r}$, $\mathbf{l}$ and $\mathbf{u}$. The function for solving quadratic programs in Matlab is `quadprog`.

If we were to solve the problem from Example 9.2 we would have to multiply the objective function by -1 to transform the problem from a maximization problem to a minimization problem:

$$\begin{cases} \min & \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} 0 & -1/2 \\ -1/2 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} - \begin{bmatrix} 0 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \\ s.t. & \begin{bmatrix} 2 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = 100 \\ & \begin{bmatrix} x \\ y \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \end{cases}$$

Notice we can write:

$$\begin{bmatrix} 0 & -1/2 \\ -1/2 & 0 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \tag{9.3}$$

This leads to the Matlab input matrices:

$$\begin{aligned} &\mathbf{Q} = \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \quad \mathbf{c} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\ &\mathbf{A} = [] \quad \mathbf{b} = [] \\ &\mathbf{H} = \begin{bmatrix} 2 & 2 \end{bmatrix} \quad \mathbf{r} = [100] \\ &\mathbf{l} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad \mathbf{u} = \begin{bmatrix} +\infty \\ +\infty \end{bmatrix} \end{aligned}$$

Note that $\mathbf{Q}$ is defined as it is because Matlab assumes we factor out a $1/2$. Figure 9.1 shows how to call the `quadprog` function in Matlab with the given inputs.

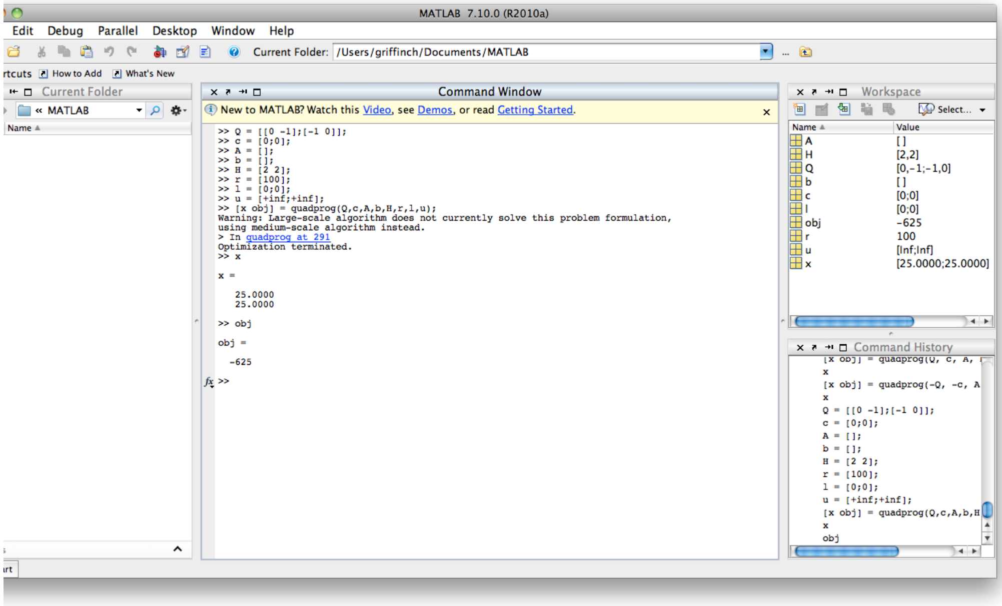


Figure 9.1. Solving quadratic programs is relatively easy with Matlab. We simply provide the necessary matrix inputs *remembering* that we have the objective $(1/2)\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x}$.

3. General Sum Games and Quadratic Programming

A majority of this section is derived from [MS64]. Consider a two-player general sum game $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$. Let $\mathbf{1}_m \in \mathbb{R}^{m\times 1}$ be the vector of all ones with m elements and let $\mathbf{1}_n \in \mathbb{R}^{n\times 1}$ be the vector of all ones with n elements. By Theorem 6.52 there is at least one Nash equilibrium $(\mathbf{x}^*, \mathbf{y}^*)$. If either Player were to play his/her Nash equilibrium, then the optimization problems for the players would be:

$$P_1 \begin{cases} \max & \mathbf{x}^T\mathbf{A}\mathbf{y}^* \\ s.t. & \mathbf{1}_m^T\mathbf{x} = 1 \\ & \mathbf{x} \geq \mathbf{0} \end{cases}$$

$$P_2 \begin{cases} \max & \mathbf{x}^{*T}\mathbf{B}\mathbf{y} \\ s.t. & \mathbf{1}_n^T\mathbf{y} = 1 \\ & \mathbf{y} \geq \mathbf{0} \end{cases}$$

Individually, these are linear programs. The problem is, we don't know the values of $(\mathbf{x}^*, \mathbf{y}^*)$ *a priori*. However, we can draw insight from these problems.

LEMMA 9.4. *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a general sum two-player matrix game with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m\times n}$. A point $(x^*, y^*) \in \Delta$ is a Nash equilibrium if and only if there exists scalar values α*

and β such that:

$$\begin{aligned}
\mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* - \alpha &= 0\\
\mathbf{x}^{*T}\mathbf{B}\mathbf{y}^* - \beta &= 0\\
\mathbf{A}\mathbf{y}^* - \alpha\mathbf{1}_m &\leq \mathbf{0}\\
\mathbf{x}^{*T}\mathbf{B} - \beta\mathbf{1}_n^T &\leq \mathbf{0}\\
\mathbf{1}_m^T\mathbf{x}^* - 1 &= 0\\
\mathbf{1}_n^T\mathbf{y}^* - 1 &= 0\\
\mathbf{x}^* &\geq \mathbf{0}\\
\mathbf{y}^* &\geq \mathbf{0}
\end{aligned}$$

Proof. Assume that $\mathbf{x}^* = [x_1^*, \dots, x_m^*]^T$ and $\mathbf{y}^* = [y_1^*, \dots, y_n^*]^T$. Consider the KKT conditions for the linear programming problem for P_1. The objective function is:

$$z(x_1, \dots, x_n) = \mathbf{x}^T\mathbf{A}\mathbf{y}^* = \mathbf{c}^T\mathbf{x}$$

here $\mathbf{c} \in \mathbb{R}^{n\times 1}$ and

$$c_i = \mathbf{A}_{i\cdot}\mathbf{y}^* = a_{i1}y_1^* + a_{i2}y_2^* + \dots + a_{in}y_n^*$$

The vector $\mathbf{x}^*$ is an optimal solution for this problem if and only if there exists multipliers $\lambda_1, \dots, \lambda_m$ (corresponding to constraints $\mathbf{x} \geq \mathbf{0}$) and α (corresponding to the constraint $\mathbf{1}_m^T\mathbf{x} = 1$ so that:

$$\text{Primal Feasibility}: \begin{cases} x_1^* + \dots + x_m^* = 1 \\ x_i^* \geq 0 \quad i = 1, \dots, m \end{cases}$$

$$\text{Dual Feasibility}: \begin{cases} \nabla z(\mathbf{x}^*) - \displaystyle\sum_{i=1}^m \lambda_i(-\mathbf{e}_i) - \alpha\mathbf{1}_m = \mathbf{0} \\ \lambda_i \geq 0 \quad \text{for } i = 1, \dots, m \\ \alpha \quad \text{unrestricted} \end{cases}$$

$$\text{Complementary Slackness}: \left\{ \lambda_i x_i^* = 0 \quad i = 1, \dots, m \right.$$

We observe first that $\nabla z(\mathbf{x}^*) = \mathbf{A}\mathbf{y}^*$. Therefore, we can write the first equation in the Dual Feasibility condition as:

$$\mathbf{A}\mathbf{y}^* - \alpha\mathbf{1}_m = -\sum_{i=1}^m \lambda_i\mathbf{e}_i \tag{9.4}$$

Since $\lambda_i \geq 0$ and $\mathbf{e}_i$ is just the i^{th} standard basis vector, we know that $\lambda_i\mathbf{e}_i \geq \mathbf{0}$ and thus:

$$\mathbf{A}\mathbf{y}^* - \alpha\mathbf{1}_m \leq \mathbf{0} \tag{9.5}$$

Now, again consider the first equation in Dual Feasibility written as:

$$\mathbf{A}\mathbf{y}^* + \sum_{i=1}^m \lambda_i\mathbf{e}_i - \alpha\mathbf{1}_m = \mathbf{0}$$

If we multiply by $\mathbf{x}^{*T}$ on the left we obtain:

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* + \sum_{i=1}^{m} \lambda_i \mathbf{x}^{*T}\mathbf{e}_i - \alpha \mathbf{x}^{*T}\mathbf{1}_m = \mathbf{x}^{*T}\mathbf{0} = 0 \tag{9.6}$$

But $\lambda_i \mathbf{x}^{*T}\mathbf{e}_i = \lambda_i x_i^* = 0$ by complementary slackness and $\alpha\mathbf{x}^{*T}\mathbf{1}_m = \alpha$ by primal feasibility; i.e., the fact that $\mathbf{x}^{*T}\mathbf{1}_m = \mathbf{1}_m^T\mathbf{x}^* = x_1^* + \cdots + x_m^* = 1$. Thus we conclude from Equation 9.6 that:

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* - \beta = 0 \tag{9.7}$$

If we consider the problem for Player 2, then:

$$z(y_1, \dots, y_n) = z(\mathbf{y}) = \left(\mathbf{x}^{*T}\mathbf{B}\right)\mathbf{y} \tag{9.8}$$

so that the j^{th} component of $\nabla z(\mathbf{y})$ is $\mathbf{x}^{*T}\mathbf{B}_{\cdot j}$. If we consider the KKT conditions for Player 2, we know that $\mathbf{y}^*$ is an optimal solution if and only if there exists Lagrange multipliers $\mu_1, \dots, \mu_n$ (corresponding to the constraints $\mathbf{y} \geq 0$) and β (corresponding to the constraint $y_1 + \cdots + y_n = 1$) so that:

$$\text{Primal Feasibility} : \begin{cases} y_1^* + \cdots + y_n^* = 1 \\ y_j^* \geq 0 \quad j = 1, \dots, n \end{cases}$$

$$\text{Dual Feasibility} : \begin{cases} \nabla z(\mathbf{y}^*) - \displaystyle\sum_{j=1}^{n} \mu_j(-\mathbf{e}_i) - \beta\mathbf{1}_n = \mathbf{0} \\ \mu_j \geq 0 \quad \text{for } j = 1, \dots, n \\ \beta \quad \text{unrestricted} \end{cases}$$

$$\text{Complementary Slackness} : \left\{ \mu_j y_j^* = 0 \quad i = 1, \dots, n \right.$$

As in the case for Player 1, we can show that:

$$\mathbf{x}^{*T}\mathbf{B} - \beta\mathbf{1}_n^T \leq \mathbf{0} \tag{9.9}$$

and

$$\mathbf{x}^{*T}\mathbf{B}\mathbf{y}^* - \beta = 0 \tag{9.10}$$

Thus we have shown (from the necessity and sufficiency of KKT conditions for the two problems) that:

$$\begin{aligned} \mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* - \alpha &= 0 \\ \mathbf{x}^{*T}\mathbf{B}\mathbf{y}^* - \beta &= 0 \\ \mathbf{A}\mathbf{y}^* - \alpha\mathbf{1}_m &\leq \mathbf{0} \\ \mathbf{x}^{*T}\mathbf{B} - \beta\mathbf{1}_n^T &\leq \mathbf{0} \\ \mathbf{1}_m^T\mathbf{x}^* - 1 &= 0 \\ \mathbf{1}_n^T\mathbf{y}^* - 1 &= 0 \\ \mathbf{x}^* &\geq \mathbf{0} \\ \mathbf{y}^* &\geq \mathbf{0} \end{aligned}$$

is a necessary and sufficient condition for $(\mathbf{x}^*, \mathbf{y}^*)$ to be a Nash equilibrium of the game $\mathcal{G}$. $\square$

THEOREM 9.5. *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a general sum two-player matrix game with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$. A point $(x^*, y^*) \in \Delta$ is a Nash equilibrium if and only if there are reals α^* and β^* so that $(\mathbf{x}^*, \mathbf{y}^*, \alpha^*, \beta^*)$, is a global maximizer for the quadratic programming problem:*

$$
\begin{aligned}
\max\ & \mathbf{x}^T(\mathbf{A}+\mathbf{B})\mathbf{y} - \alpha - \beta \\
s.t.\ & \mathbf{A}\mathbf{y} - \alpha \mathbf{1}_m \leq \mathbf{0} \\
& \mathbf{x}^T\mathbf{B} - \beta \mathbf{1}_n^T \leq \mathbf{0} \\
& \mathbf{1}_m^T\mathbf{x} - 1 = 0 \\
& \mathbf{1}_n^T\mathbf{y} - 1 = 0 \\
& \mathbf{x} \geq \mathbf{0} \\
& \mathbf{y} \geq \mathbf{0}
\end{aligned}
\tag{9.11}
$$

PROOF. First observe that:

$$
\mathbf{A}\mathbf{y} - \alpha\mathbf{1}_m \leq \mathbf{0} \implies \mathbf{x}^T\mathbf{A}\mathbf{y} - \alpha\mathbf{x}^T\mathbf{1}_m \leq \mathbf{x}^T\mathbf{0} \implies \mathbf{x}^T\mathbf{A}\mathbf{y} - \alpha \leq 0 \tag{9.12}
$$

Similarly,

$$
\mathbf{x}^T\mathbf{B} - \beta\mathbf{1}_n^T \leq \mathbf{0} \implies \mathbf{x}^T\mathbf{B}\mathbf{y} - \beta\mathbf{1}_n^T\mathbf{y} \leq \mathbf{0}\mathbf{y} \implies \mathbf{x}^T\mathbf{B}\mathbf{y} - \beta \leq 0 \tag{9.13}
$$

Combining these inequalities we see that $z(\mathbf{x}, \mathbf{y}, \alpha, \beta) = \mathbf{x}^T(\mathbf{A}+\mathbf{B})\mathbf{y} - \alpha - \beta \leq 0$. Thus any set of variables $(\mathbf{x}^*, \mathbf{y}^*, \alpha^*, \beta^*)$ so that $z(\mathbf{x}^*, \mathbf{y}^*, \alpha^*, \beta^*) = 0$ is a global maximum.

($\Leftarrow$) We now show that at a global optimal solution, the KKT conditions for the quadratic program are identical to the conditions given in Lemma 9.4. At an optimal point $(\mathbf{x}^*, \mathbf{y}^*, \alpha^*, \beta^*)$, there are multipliers

(1) $\lambda_1, \dots, \lambda_m$ (corresponding to the constraints $\mathbf{A}\mathbf{y} - \alpha\mathbf{1}_m \leq \mathbf{0}$)
(2) $\mu_1, \dots, \mu_n$ (corresponding to the constraints $\mathbf{x}^T\mathbf{B} - \beta\mathbf{1}_n^T \leq \mathbf{0}$),
(3) ν_1 (corresponding to the constraint $\mathbf{1}_m^T\mathbf{x} - 1$),
(4) ν_2 (corresponding to the constraint $\mathbf{1}_n^T\mathbf{y} - 1 = 0$),
(5) $\phi_1, \dots, \phi_m$ (corresponding to the constraints $\mathbf{x} \geq \mathbf{0}$) and
(6) $\theta_1, \dots, \theta_n$ (corresponding to the constraints $\mathbf{y} \geq \mathbf{0}$).

We can compute the gradients of the various constraints and objective as (remembering that we will write $\mathbf{x} \geq \mathbf{0}$ as $-\mathbf{x} \leq \mathbf{0}$ and $\mathbf{y} \geq \mathbf{0}$ as $-\mathbf{y} \leq \mathbf{0}$. Additionally we note that each gradient has $m + n + 2$ components (one for each variable in $\mathbf{x}$, $\mathbf{y}$ and α and β. The vector $\mathbf{0}$ will vary in size to ensure that all vectors have the correct size:

(1)

$$
\nabla z(\mathbf{x}, \mathbf{y}, \alpha, \beta) = \begin{bmatrix} (\mathbf{A}+\mathbf{B})\mathbf{y} \\ (\mathbf{A}+\mathbf{B})^T\mathbf{x} \\ -1 \\ -1 \end{bmatrix}
$$

(2)

$$\nabla\left(\mathbf{A}\mathbf{y}-\alpha\mathbf{1}_m\right)=\begin{bmatrix}\mathbf{0}\\ \mathbf{A}_{i\cdot}^T\\ -1\\ 0\end{bmatrix}$$

(3)

$$\nabla\left(\mathbf{B}\mathbf{x}^T-\beta\mathbf{1}_n\right)=\begin{bmatrix}\mathbf{B}_{\cdot\mathbf{j}}\\ \mathbf{0}\\ 0\\ -1\end{bmatrix}$$

(4)

$$\nabla(\mathbf{1}_m^T\mathbf{x}-1)=\begin{bmatrix}\mathbf{1}_m\\ \mathbf{0}\\ 0\\ 0\end{bmatrix}$$

(5)

$$\nabla(\mathbf{1}_n^T\mathbf{y}-1)=\begin{bmatrix}\mathbf{0}\\ \mathbf{1}_n\\ 0\\ 0\end{bmatrix}$$

(6)

$$\nabla(-x_i)=\begin{bmatrix}-\mathbf{e}_i\\ \mathbf{0}\\ 0\\ 0\end{bmatrix}$$

(7)

$$\nabla(-y_j)=\begin{bmatrix}\mathbf{0}\\ -\mathbf{e}_j\\ 0\\ 0\end{bmatrix}$$

In the final gradients, $\mathbf{e}_i \in \mathbb{R}^{m\times 1}$ and $\mathbf{e}_j \in \mathbb{R}^{n\times 1}$ so that the standard basis vectors agree with the dimensionality of $\mathbf{x}$ and $\mathbf{y}$ respectively. The Dual Feasibility constraints of the KKT conditions for the quadratic program assert that

(1) $\lambda_1, \ldots, \lambda_n \geq 0$
(2) $\mu_1, \ldots, \mu_m \geq 0$
(3) $\phi_1, \ldots, \phi_m \geq 0$,
(4) $\theta_1, \ldots, \theta_n \geq 0$,
(5) $\nu_1 \in \mathbb{R}$, and
(6) $\nu_2 \in \mathbb{R}$

Then final component of dual feasibility asserts that:

$$(9.14)\quad \begin{bmatrix} (\mathbf{A}+\mathbf{B})\mathbf{y} \\ (\mathbf{A}+\mathbf{B})^T\mathbf{x} \\ -1 \\ -1 \end{bmatrix} - \sum_{i=1}^{m}\lambda_i \begin{bmatrix} \mathbf{0} \\ \mathbf{A}_{i\cdot}^T \\ -1 \\ 0 \end{bmatrix} - \sum_{j=1}^{n}\mu_j \begin{bmatrix} \mathbf{B}_{\cdot\mathbf{j}} \\ \mathbf{0} \\ 0 \\ -1 \end{bmatrix} \nu_1 \begin{bmatrix} \mathbf{1}_m \\ \mathbf{0} \\ 0 \\ 0 \end{bmatrix} - \nu_2 \begin{bmatrix} \mathbf{0} \\ \mathbf{1}_n \\ 0 \\ 0 \end{bmatrix} - \sum_{i=1}^{m}\phi_i \begin{bmatrix} \mathbf{e}_i \\ \mathbf{0} \\ 0 \\ 0 \end{bmatrix} - \sum_{j=1}^{n}\theta_j \begin{bmatrix} \mathbf{0} \\ -\mathbf{e}_j \\ 0 \\ 0 \end{bmatrix} = \mathbf{0}$$

We can analyze this expression component by component. Consider the last component (corresponding to variable β), we have:

$$(9.15)\quad -1 - \sum_{j=1}^{n}\mu_j = 0 \implies \sum_{j=1}^{n}\mu_j = 1$$

We can similarly analyze the component corresponding to α and see that dual feasibility implies that:

$$(9.16)\quad -1 - \sum_{i=1}^{m}\lambda_i = 0 \implies \sum_{i=1}^{m}\lambda_i = 1$$

Thus dual feasibility shows that $(\lambda_1, \dots, \lambda_m) \in \Delta_m$ and $(\mu_1, \dots \mu_n) \in \Delta_n$. Let us now analyze the component corresponding to variable y_j. Dual feasibility implies:

$$(9.17)\quad \mathbf{x}^T(\mathbf{A}_{\cdot j} + \mathbf{B}_{\cdot j}) - \sum_{i=1}^{m}\lambda_i\mathbf{A}_{ij} - \nu_2 + \theta_j = 0 \implies \mathbf{x}^T(\mathbf{A}_{\cdot j} + \mathbf{B}_{\cdot j}) - \sum_{i=1}^{m}\lambda_i\mathbf{A}_{ij} - \nu_2 \leq 0$$

We can similarly analyze the component corresponding to variable x_i. Dual feasibility implies that:

$$(9.18)\quad (\mathbf{A}_{i\cdot} + \mathbf{B}_{i\cdot})\mathbf{y} - \sum_{j=1}^{n}\mu_j\mathbf{B}_{ij} - \nu_1 + \phi_i = 0 \implies (\mathbf{A}_{i\cdot} + \mathbf{B}_{i\cdot})\mathbf{y} - \sum_{j=1}^{n}\mu_j\mathbf{B}_{ij} - \nu_1 \leq 0$$

There is now a trick required to complete to proof. Suppose we choose Lagrange multipliers so that $x_i = \lambda_i$ $(i = 1, \dots, m)$ and $y_j = \mu_j$ $(j = 1, \dots, n)$. We are allowed to do so because of the constraints on the λ_i and μ_j. Furthermore, suppose we choose $\nu_1 = \alpha$ and $\nu_2 = \beta$. Then if $\mathbf{x}^*$, $\mathbf{y}^*$, α^*, β^* is an optimal solution, then Equations 9.17 and 9.18 become:

$$\mathbf{x}^{*T}(\mathbf{A}+\mathbf{B}) - \mathbf{x}^{*T}\mathbf{A} - \beta^*\mathbf{1}_n^T \leq \mathbf{0} \implies \mathbf{x}^{*T}\mathbf{B} - \beta^*\mathbf{1}_n^T \leq \mathbf{0}$$
$$(\mathbf{A}+\mathbf{B})\mathbf{y}^* - \mathbf{B}\mathbf{y}^* - \alpha^*\mathbf{1}_m \leq \mathbf{0} \implies \mathbf{A}\mathbf{y}^* - \alpha^*\mathbf{1}_m \leq \mathbf{0}$$

We also know that:

(1) $\mathbf{1}_m^T\mathbf{x}^* = 1$,
(2) $\mathbf{1}_n^T\mathbf{y}^* = 1$,
(3) $\mathbf{x} \geq \mathbf{0}$, and
(4) $\mathbf{y} \geq \mathbf{0}$

Lastly, complementary slackness for the quadratic programming problem implies that:

$$\lambda_i \left(\mathbf{A}_{i\cdot}\mathbf{y} - \alpha\right) = 0 \quad i = 1, \dots, m \tag{9.19}$$

$$\left(\mathbf{x}^T\mathbf{B}_{\cdot j} - \beta\right)\mu_j = 0 \quad j = 1, \dots, n \tag{9.20}$$

Since $x_i^* = \lambda_i$ and $y_j^* = \mu_j$, we have:

$$\sum_{i=1}^m x_i^* \left(\mathbf{A}_{i\cdot}\mathbf{y}^* - \alpha^*\right) = 0 \implies \sum_{i=1}^m x_i^*\mathbf{A}_{i\cdot}\mathbf{y}^* - \sum_{i=1}^m \alpha^* x_i^* = 0 \implies \mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* - \alpha^* = 0 \tag{9.21}$$

$$\sum_{j=1}^n \left(\mathbf{x}^{*T}\mathbf{B}_{\cdot j} - \beta^*\right)\mu_j = 0 \implies \sum_{j=1}^n \mathbf{x}^{*T}\mathbf{B}_{\cdot j}y_j^* - \sum_{j=1}^n \beta^* y_j^* = 0 \implies \mathbf{x}^{*T}\mathbf{B}\mathbf{y}^* - \beta^* = 0 \tag{9.22}$$

From this we conclude that any tuple $(\mathbf{x}^*, \mathbf{y}^*, \alpha^*, \beta^*)$ satisfying these KKT conditions must be a global maximizer because adding these final two equations yields:

$$\mathbf{x}^{*T}(\mathbf{A} + \mathbf{B})\mathbf{y}^* - \alpha^* - \beta^* = 0 \tag{9.23}$$

Moreover, by Lemma 9.4 it must also be a Nash equilibrium.

($\Rightarrow$) The converse of the theorem states that if $(\mathbf{x}^*, \mathbf{y}^*)$ is a Nash equilibrium for $\mathcal{G}$, then setting $\alpha^* = \mathbf{x}^{*T}\mathbf{A}\mathbf{y}^*$ and $\beta^* = \mathbf{x}^{*T}\mathbf{B}\mathbf{y}^*$ gives an optimal solution $(\mathbf{x}^*, \mathbf{y}^*, \alpha^*, \beta^*)$ to the quadratic program. It follows from the Lemma 9.4 that when $(\mathbf{x}^*, \mathbf{y}^*)$ is a Nash equilibrium we know that:

$$\mathbf{x}^{*T}\mathbf{A}\mathbf{y}^* - \alpha^* = 0$$
$$\mathbf{x}^{*T}\mathbf{B}\mathbf{y}^* - \beta^* = 0$$

and thus we know at once that

$$\mathbf{x}^{*T}(\mathbf{A} + \mathbf{B})\mathbf{y}^* - \alpha^* - \beta^* = 0$$

holds and thus $(\mathbf{x}^*, \mathbf{y}^*, \alpha^*, \beta^*)$ must be a global maximizer for the quadratic program because the objective function achieves its upper bound. This completes the proof. $\square$

Example 9.6. We can find a third Nash equilibrium for the Chicken game using this approach. Recall we have:

$$\mathbf{A} = \begin{bmatrix} 0 & -1 \\ 1 & -10 \end{bmatrix}$$
$$\mathbf{B} = \begin{bmatrix} 0 & 1 \\ -1 & -10 \end{bmatrix}$$

Then our quadratic program is:

$$
(9.24) \quad \begin{cases}
\max & \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 0 & 0 \\ 0 & -20 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} - \alpha - \beta \\
s.t. & \begin{bmatrix} 0 & -1 \\ 1 & -10 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} - \begin{bmatrix} \alpha \\ \alpha \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
& \begin{bmatrix} x_1 & x_2 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & -10 \end{bmatrix} - \begin{bmatrix} \beta & \beta \end{bmatrix} \leq \begin{bmatrix} 0 & 0 \end{bmatrix} \\
& \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = 1 \\
& \begin{bmatrix} 1 & 1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = 1 \\
& \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix} \\
& \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \end{bmatrix}
\end{cases}
$$

This simplifies to the quadratic programming problem:

$$
(9.25) \quad \begin{cases}
\max & -20x_2y_2 - \alpha - \beta \\
s.t. & -y_2 - \alpha \leq 0 \\
& y_1 - 10y_2 - \alpha \leq 0 \\
& -x_2 - \beta \leq 0 \\
& x_1 - 10x_2 - \beta \leq 0 \\
& x_1 + x_2 = 1 \\
& y_1 + y_2 = 1 \\
& x_1, x_2, y_1, y_2 \geq 0
\end{cases}
$$

An optimal solution to this problem is $x_1 = 0.9$, $x_2 = 0.1$, $y_1 = 0.9$, $y_2 = 0.1$. This is a third Nash equilibrium in mixed strategies for this instance of Chicken. Identifying this third Nash equilibrium in Matlab is shown in Figure 9.2. In order to correctly input this problem into Matlab, we need to first write the problem as a proper quadratic program. This is done by letting the vector of decision variables be:

$$
\mathbf{z} = \begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ \alpha \\ \beta \end{bmatrix}
$$

Then the quadratic programming problem for Chicken is written as:

$$
(9.26)\quad \begin{cases}
\max\ \dfrac{1}{2}\begin{bmatrix} x_1 & x_2 & y_1 & y_2 & \alpha & \beta \end{bmatrix}\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & -40 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ \alpha \\ \beta \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -1 \\ -1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ \alpha \\ \beta \end{bmatrix} \\
s.t.\ \begin{bmatrix} 0 & 0 & 0 & -1 & -1 & 0 \\ 0 & 0 & 1 & -10 & -1 & 0 \\ 0 & -1 & 0 & 0 & 0 & -1 \\ 1 & -10 & 0 & 0 & 0 & -1 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ \alpha \\ \beta \end{bmatrix} \leq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \\
\begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ \alpha \\ \beta \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \\
\begin{bmatrix} x_1 \\ x_2 \\ y_1 \\ y_2 \\ \alpha \\ \beta \end{bmatrix} \geq \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ -\infty \\ -\infty \end{bmatrix}
\end{cases}
$$

Note, before you enter this into Matlab, you must transform the problem to a minimization problem by multiplying the objective function matrices by -1.

Exercise 70. Use this technique to identify the Nash equilibrium in Prisoner's Dilemma

Exercise 71. Show that when $\mathbf{B} = -\mathbf{A}$ (i.e., we have a zero-sum game) that the quadratic programming problem reduces to the two dual linear programming problems we already identified in the last chapter for solving zero-sum games.

Remark 9.7. It is worth noting that this is still not the most modern method for finding Nash equilibrium of general sum N player games. Newer techniques have been developed (specifically by Lemke and Howson [**LH61**] and their followers) in identifying Nash equilibrium solutions. It is this technique and not the quadratic programming approach that is now used in computational game theory for identifying and studying the computational problems associated with Nash equilibria. Unfortunately, this theory is more complex and outside the scope of these notes.

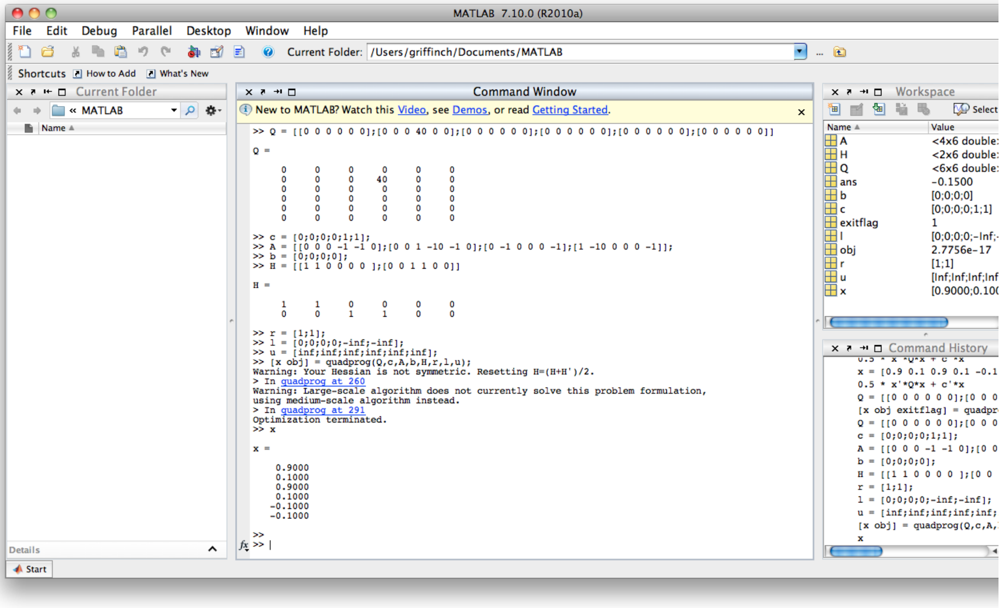


Figure 9.2. We can use the power of Matlab to find a third Nash equilibrium in mixed strategies for the game of Chicken by solving the Problem 9.26. Note, we have to change this problem to a minimization problem by multiplying the objective by -1.

CHAPTER 10

Nash's Bargaining Problem and Cooperative Games

Heretofore we have considered games in which the players were unable to communicate before play began or in which players has no way of trusting each other with certainty (remember Prisoner's dilemma). In this chapter, we remove this restriction and consider those games in which players may put in place a pre-play agreement on their play in an attempt to identify a solution with which both players can live happily.

1. Payoff Regions in Two Player Games

DEFINITION 10.1 (Cooperative Mixed Strategy). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a two-player matrix game with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times m}$. Then a cooperative strategy is a collection of probabilities x_{ij} $(i = 1, \dots, m,\ j = 1, \dots, n)$ so that:

$$\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} = 1$$
$$x_{ij} \geq 0 \quad i = 1, \dots, m,\ j = 1, \dots, n$$

To any cooperative strategy, we can associate a vector $\mathbf{x} \in \Delta_{mn}$.

REMARK 10.2. For any cooperative strategy x_{ij} $(i = 1, \dots, m,\ j = 1, \dots, n)$, x_{ij} gives the probability that Player 1 plays row i while Player 2 players column j. Note, x could be thought of as a matrix, but for the sake of notational consistency, it is easier to think of it as a vector with an strange indexing scheme.

DEFINITION 10.3 (Cooperative Expected Payoff). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a two-player matrix game with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times m}$ and let x_{ij} $(i = 1, \dots, m,\ j = 1, \dots, n)$ be a cooperative strategy for Player 1 and 2. Then:

$$u_1(\mathbf{x}) = \sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{A}_{ij}x_{ij} \tag{10.1}$$

is the expected payoff for Player 1, while

$$u_2(\mathbf{x}) = \sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{B}_{ij}x_{ij} \tag{10.2}$$

DEFINITION 10.4 (Payoff Region (Competitive Game)). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a two-player matrix game with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times m}$. The *payoff region of the competitive game* is

$$Q(\mathbf{A}, \mathbf{B}) = \{(u_1(\mathbf{x}, \mathbf{y}), u_2(\mathbf{x}, \mathbf{y})) : \mathbf{x} \in \Delta_m,\ \mathbf{y} \in \Delta_n\} \tag{10.3}$$

where

$$u_1(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{A} \mathbf{y} \tag{10.4}$$
$$u_2(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T \mathbf{B} \mathbf{y} \tag{10.5}$$

are the standard competitive player payoff functions.

Definition 10.5 (Payoff Region (Cooperative Game)). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a two-player matrix game with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times m}$. The *payoff region of the cooperative game* is

$$P(\mathbf{A}, \mathbf{B}) = \{(u_1(\mathbf{x}), u_2(\mathbf{x})) : \mathbf{x} \in \Delta_{mn}\} \tag{10.6}$$

where u_1 and u_2 are the cooperative payoff functions for Player 1 and 2 respectively.

Lemma 10.6. *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a two-player matrix game with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times m}$. The competitive playoff region $Q(\mathbf{A}, \mathbf{B})$ is contained in the cooperative payoff region $P(\mathbf{A}, \mathbf{B})$.*

Exercise 72. Prove Lemma 10.6. [Hint: Argue that any pair of mixed strategies can be used to generate an cooperative mixed strategy.]

Example 10.7. Consider the following two payoff matrices:

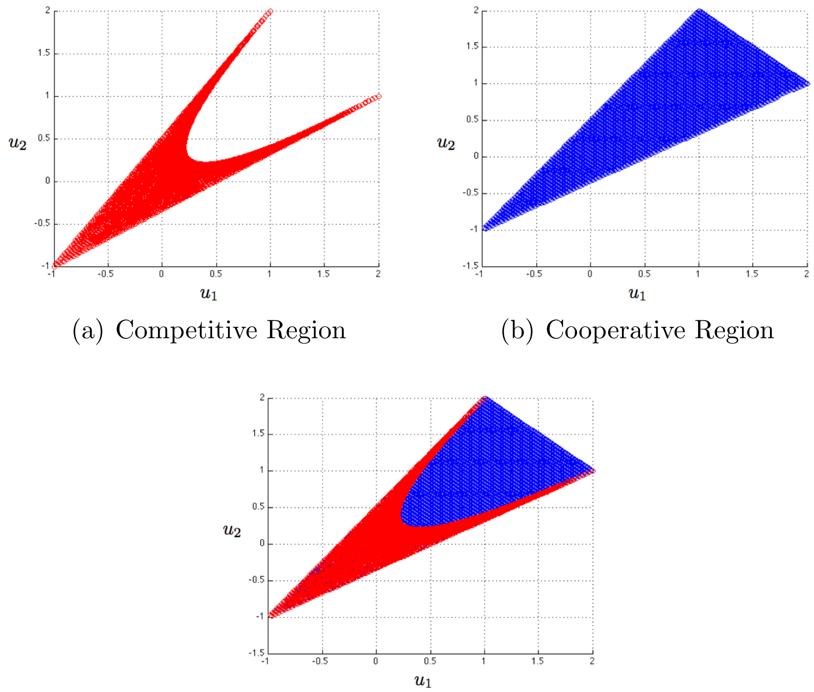
$$\mathbf{A} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$$
$$\mathbf{B} = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$$

The game defined here is sometimes called the *Battle of the Sexes* game and describes the decision making process of a married couple as they attempt to decide what to do on a given evening. The players must decide whether to attend a boxing match or a ballet. One clearly prefers the boxing match (strategy 1 for each player) and the other prefers the ballet (strategy 2 for each player). Neither derives much benefit from going to an event alone, which is indicated by the -1 payoffs in the off-diagonal elements. The competitive payoff region, cooperative payoff region and an overlay of the two regions for the Battle of the Sexes is shown in Figure 10.1. Constructing these figures is done by brute force through a Matlab script.

Exercise 73. Find a Nash equilibrium for the Battle of the Sexes using a Quadratic Programming problem.

Remark 10.8. We will see in the next section that our objective is to choose a cooperative strategy that makes both players as happy as possible.

Theorem 10.9. *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a two-player matrix game with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times m}$. The cooperative payoff region $P(\mathbf{A}, \mathbf{B})$ is a convex set.*



(a) Competitive Region (b) Cooperative Region

(c) Overlap

Figure 10.1. The three plots shown the competitive payoff region, cooperative payoff region and and overlay of the regions for the Battle of the Sexes game. Note that the cooperative payoff region completely contains the competitive payoff region.

Proof. The set $P(\mathbf{A}, \mathbf{B})$ is defined as the set of (u_1, u_2) satisfying the constraints:

$$
\begin{cases}
\displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{A}_{ij}x_{ij} - u_1 = 0 \\
\displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{B}_{ij}x_{ij} - u_2 = 0 \\
\displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} = 1 \\
x_{ij} \geq 0 \quad i = 1, \dots, m, \; j = 1, \dots, n
\end{cases}
\tag{10.7}
$$

This set is defined by equalities associated with linear functions (which are both convex and concave). We can rewrite this as:

$$\begin{cases} \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{A}_{ij}x_{ij} - u_1 \leq 0 \\ \displaystyle -\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{A}_{ij}x_{ij} + u_1 \leq 0 \\ \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{B}_{ij}x_{ij} - u_2 \leq 0 \\ \displaystyle -\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{B}_{ij}x_{ij} + u_2 \leq 0 \\ \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n}x_{ij} \leq 1 \\ \displaystyle -\sum_{i=1}^{m}\sum_{j=1}^{n}x_{ij} = -1 \\ -x_{ij} \leq 0 \quad i = 1, \dots, m, \; j = 1, \dots, n \end{cases}$$

Thus, since linear functions are convex, the set of tuples $(u_1, u_2, \mathbf{x})$ that satisfy these constraints is a convex set by Theorems 7.24 and 7.27. Suppose that $(u_1^1, u_2^1, \mathbf{x}^1)$ and $(u_1^2, u_2^2, \mathbf{x}^2)$ are two tuples satisfying these constraints. Then clearly, $(u_1^1, u_2^1), (u_1^2, u_2^2) \in P(\mathbf{A}, \mathbf{B})$. Since the set of tuples $(u_1, u_2, \mathbf{x})$ that satisfy these constraints form a convex set we know that for all $\lambda \in [0, 1]$ we have:

$$\lambda(u_1^1, u_2^1, \mathbf{x}^1) + (1 - \lambda)(u_1^2, u_2^2, \mathbf{x}^2) = (u_1, u_2, \mathbf{x}) \tag{10.8}$$

and $(u_1, u_2, \mathbf{x})$ satisfies the constraints. But then, $(u_1, u_2) \in P(\mathbf{A}, \mathbf{B})$ and therefore

$$\lambda(u_1^1, u_2^1) + (1 - \lambda)(u_1^2, u_2^2) \in P(\mathbf{A}, \mathbf{B}) \tag{10.9}$$

for all λ. It follows that $P(\mathbf{A}, \mathbf{B})$ is convex. $\square$

Remark 10.10. The next theorem assumes that the reader knows the definition of a closed set in Euclidean space. There are many consistent definitions for a closed set in $\mathbb{R}^n$, however we will take the definition to be that the set is defined by a collection of equalities and *non-strict* (i.e., $\leq$) inequalities.

Theorem 10.11. *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a two-player matrix game with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times m}$. The cooperative payoff region $P(\mathbf{A}, \mathbf{B})$ is a bounded and closed set.*

Proof. Again, consider the defining equalities:

$$\begin{cases} \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{A}_{ij}x_{ij} - u_1 = 0 \\ \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{B}_{ij}x_{ij} - u_2 = 0 \\ \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n}x_{ij} = 1 \\ x_{ij} \geq 0 \quad i = 1, \dots, m, \; j = 1, \dots, n \end{cases}$$

This set must be bounded because $x_{ij} \in [0, 1]$ for $i = 1, \dots, m$ and $j = 1, \dots, n$. As a result of this, the value of u_1 is bounded above and below by the largest and smallest values in $\mathbf{A}$ while the value of u_2 is bounded above and below by the largest and smallest values in $\mathbf{B}$. Closure of the set is ensured by the fact that the set is defined by non-strict inequalities and equalities. □

Remark 10.12. What we've actually proved in these theorems (and more importantly) is that the set of tuples $(u_1, u_2, \mathbf{x})$ defined by the system of equations and inequalities:

$$\begin{cases} \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{A}_{ij}x_{ij} - u_1 = 0 \\ \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{B}_{ij}x_{ij} - u_2 = 0 \\ \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n}x_{ij} = 1 \\ x_{ij} \geq 0 \quad i = 1, \dots, m, \; j = 1, \dots, n \end{cases}$$

is closed, bounded and convex. We will actually use this result, rather than the generic statements on $P(\mathbf{A}, \mathbf{B})$.

2. Collaboration and Multi-criteria Optimization

Up till now, we've looked at optimization problems that had a single objective. Recall our generic optimization problem:

$$\begin{cases} \max \;\; z(x_1, \dots, x_n) \\ s.t. \;\; g_1(x_1, \dots, x_n) \leq 0 \\ \qquad \vdots \\ \qquad g_m(x_1, \dots, x_n) \leq 0 \\ \qquad h_1(x_1, \dots, x_n) = 0 \\ \qquad \vdots \\ \qquad h_l(x_1, \dots, x_n) = 0 \end{cases}$$

Here, $z : \mathbb{R}^n \to \mathbb{R}$, $g_i : \mathbb{R}^n \to \mathbb{R}$ $(i = 1, \dots, m)$ and $h_j : \mathbb{R}^n \to \mathbb{R}$. This problem has one objective function, namely $z(x_1, \dots, x_n)$. A *multi-criteria optimization problem* has several objective functions $z_1, \dots, z_s : \mathbb{R}^n \to \mathbb{R}$. We can write such a problem as:

$$\begin{cases} \max & \begin{bmatrix} z_1(x_1,\dots,x_n) & z_2(x_1,\dots,x_n) & \cdots & z_s(x_1,\dots,x_n) \end{bmatrix} \\ s.t. & g_1(x_1,\dots,x_n) \leq 0 \\ & \vdots \\ & g_m(x_1,\dots,x_n) \leq 0 \\ & h_1(x_1,\dots,x_n) = 0 \\ & \vdots \\ & h_l(x_1,\dots,x_n) = 0 \end{cases}$$

REMARK 10.13. You note that the objective function has now been replaced with a vector of objective functions. Multi-criteria optimization problems can be challenging to solve because (e.g.) making $z_1(x_1, \dots, x_n)$ larger may make $z_2(x_1, \dots, x_n)$ smaller and vice versa.

EXAMPLE 10.14 (The Green Toy Maker). For the sake of argument, consider the Toy Maker problem from Example 8.1. We had the linear programming problem:

$$\begin{cases} \max & z(x_1,x_2) = 7x_1 + 6x_2 \\ s.t. & 3x_1 + x_2 \leq 120 \\ & x_1 + 2x_2 \leq 160 \\ & x_1 \leq 35 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{cases}$$

Suppose a certain amount of pollution is created each time a toy is manufactured. Suppose each plane generates 3 units of pollution, while manufacturing a boat generates only 2 units of pollution. Since x_1 was the number of planes produced and x_2 was the number of boats produced, we could create a multi-criteria optimization problem in which we simultaneously attempt to maximize profit $7x_1 + 6x_2$ and minimize pollution $3x_1 + 2x_2$. Since every minimization problem can be transformed into a maximization problem by negating the objective we would have the problem:

$$\begin{cases} \max & \begin{bmatrix} 7x_1 + 6x_2, & -3x_1 - 2x_2 \end{bmatrix} \\ s.t. & 3x_1 + x_2 \leq 120 \\ & x_1 + 2x_2 \leq 160 \\ & x_1 \leq 35 \\ & x_1 \geq 0 \\ & x_2 \geq 0 \end{cases}$$

REMARK 10.15. For $n > 1$, we can choose many different ways to order elements in $\mathbb{R}^n$. For example, in the plane there are many ways to decide that a point (x_1, y_1) is greater than or less than or equivalent to another point (x_2, y_2). We can think of these as the various ways of assigning a preference relation $\succ$ to points in the plane (or more generally points in $\mathbb{R}^n$). Among other things, we could:

(1) Order them based on their standard euclidean distance to the origin (as points); i.e.,

$$(x_1, y_1) \succ (x_2, y_2) \iff \sqrt{x_1^2 + y_1^2} > \sqrt{x_2^2 + y_2^2}$$

(2) We could *alphabetize* them by comparing the first component and then the second component. (This is called the lexicographic ordering.)
(3) We could specify a parameter $\lambda \in \mathbb{R}$ and declare:

$$(x_1, y_1) \succ (x_2, y_2) \iff x_1 + \lambda y_1 > x_2 + \lambda y_1 2$$

For this reason, a multi-criteria optimization problem may have many equally good solutions. There is a substantial amount of information on solving these types of problems, which arise frequently in the real world. The interested reader might consider [**Coh03**].

Definition 10.16 (Pareto Optimality). Let $g_i : \mathbb{R}^n \to \mathbb{R}$ $(i = 1, \dots, m)$ and $h_j : \mathbb{R}^n \to \mathbb{R}$ and $z_k : \mathbb{R}^n \to \mathbb{R}$ $(k = 1, \dots, s)$. Consider the mult-criteria optimization problem:

$$\begin{cases} \max & \begin{bmatrix} z_1(x_1, \dots, x_n) & z_2(x_1, \dots, x_n) & \cdots & z_s(x_1, \dots, x_n) \end{bmatrix} \\ s.t. & g_1(x_1, \dots, x_n) \leq 0 \\ & \vdots \\ & g_m(x_1, \dots, x_n) \leq 0 \\ & h_1(x_1, \dots, x_n) = 0 \\ & \vdots \\ & h_l(x_1, \dots, x_n) = 0 \end{cases}$$

A payoff vector $\mathbf{z}(\mathbf{x}^*)$ dominates another payoff vector $\mathbf{z}(\mathbf{x})$ (for two feasible points $\mathbf{x}, \mathbf{x}^*$) if:

(1) $\mathbf{z}_k(\mathbf{x}^*) \geq \mathbf{z}_k(\mathbf{x})$ for $k = 1, \dots, s$ and
(2) $\mathbf{z}_k(\mathbf{x}^*) > \mathbf{z}_k(\mathbf{x})$ for at least one $k \in \{1, \dots, s\}$

A solution $\mathbf{x}^*$ is said to be *Pareto optimal* if $\mathbf{z}(\mathbf{x}^*)$ is *not* dominated by any other $\mathbf{z}(\mathbf{x})$ where $\mathbf{x}$ is any other feasible solution.

Remark 10.17. A solution $\mathbf{x}^*$ is Pareto optimal if changing the strategy can only benefit one objective function at the expense of another objective function. Put in terms of Example 10.14, a production pattern (x_1^*, x_2^*) is Pareto optimal if there is no way to change either x_1 or x_2 and both increase profit *and* decrease pollution.

Definition 10.18 (Multi-criteria Optimization Problem for Cooperative Games). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a two-player matrix game with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times m}$. Then the cooperative game multi-criteria optimization problem is:

$$(10.10)\quad \begin{cases} \max & \left[u_1(\mathbf{x}) - u_1^0, \quad u_2(\mathbf{x}) - u_2^0\right] \\ s.t. & \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} = 1 \\ & x_{ij} \geq 0 \quad i = 1, \dots, m \;\; j = 1, \dots, n \end{cases}$$

Where: $\mathbf{x}$ is a cooperative mixed strategy and

$$u_1(\mathbf{x}) = \sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{A}_{ij} x_{ij}$$

$$u_2(\mathbf{x}) = \sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{B}_{ij} x_{ij}$$

are the cooperative expected payoff functions and u_1^0 and u_2^0 are *status quo* payoff values–usually assumed to be a Nash equilibrium payoff value for the two players.

3. Nash's Bargaining Axioms

For a two-player matrix game $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times m}$, Nash studied the problem of finding a cooperative mixed strategy $\mathbf{x} \in \Delta_{mn}$ that would maximally benefit both players–an equilibrium cooperative mixed strategy.

Remark 10.19. The resulting strategy $\mathbf{x}^*$ is referred to as an *arbitration procedure* and is agreed to by the two players before play begins. In solving this problem, Nash quantified 6 axioms (or assumptions) that he wish to ensure.

Assumption 1 (Rationality). If $\mathbf{x}^*$ is an arbitration procedure, we must have $u_1(\mathbf{x}^*) \geq u_1^0$ and $u_2(\mathbf{x}^*) \geq u_2^0$.

Remark 10.20. Assumption 1 simply asserts that we do not wish to do worse when playing cooperatively than we can when we play competitively.

Assumption 2 (Pareto Optimality). Any arbitration procedure $\mathbf{x}^*$ is a Pareto optimal solution to the two player cooperative game multi-criteria optimization problem. That is $(u_1(\mathbf{x}^*), u_2(\mathbf{x}^*)$ is Pareto optimal.

Assumption 3 (Feasibility). Any arbitration procedure $\mathbf{x}^* \in \Delta_{mn}$ and $(u_1(\mathbf{x}^*), u_2(\mathbf{x}^*) \in P(\mathbf{A}, \mathbf{B})$.

Assumption 4 (Independence of Irrelevant Alternatives). If $\mathbf{x}^*$ is an arbitration procedure and $P' \subseteq P(\mathbf{A}, \mathbf{B})$ with $(u_1^0, u_2^0), (u_1(\mathbf{x}^*), u_2(\mathbf{x}^*)) \in P'$, then $\mathbf{x}^*$ is still an arbitration procedure when we restrict out attention to P' (and the corresponding subset of Δ_{mn}).

Remark 10.21. Assumption 4 may seem odd. It was constructed to deal with restrictions of the payoff space, which in turn result in a restriction on the space of feasible solutions to the two player cooperative game multi-criteria optimization problem. It simply says that if our multi-criteria problem doesn't change (because u_1^0 and u_2^0 are still valid status quo values) and our current arbitration procedure is still available (because $(u_1(\mathbf{x}^*, u_2(\mathbf{x}^*)$ is still in the reduced feasible region), then our arbitration procedure will not change, even though we've restricted our feasible region.

Assumption 5 (Invariance Under Linear Transformation). If $u_1(\mathbf{x})$ and $u_2(\mathbf{x})$ are replaced by $u_i'(\mathbf{x}) = \alpha_i u_i(\mathbf{x}) + \beta_i$ $(i = 1, 2)$ and $\alpha_i > 0$ $(i = 1, 2)$ and $u_i^{0\prime} = \alpha_i u_i^0 + \beta_i$ $(i = 1, 2)$ and $\mathbf{x}^*$ is an arbitration procedure for the original problem, then it is also an arbitration procedure for the transformed problem defined in terms of u_i' and $u_i^{0\prime}$.

Remark 10.22. Assumption 5 simply says that arbitration procedures are not affected by linear transformations of an underlying (linear) utility function. (See Theorem 3.25.)

Definition 10.23 (Symmetry of $P(\mathbf{A}, \mathbf{B})$). Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a two-player matrix game with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n \times m}$. The set $P(\mathbf{A}, \mathbf{B})$ is symmetric if whenever $(u_1, u_2) \in P(\mathbf{A}, \mathbf{B})$, then $(u_2, u_1) \in P(\mathbf{A}, \mathbf{B})$.

Assumption 6 (Symmetry). If $P(\mathbf{A}, \mathbf{B})$ is symmetric and $u_1^0 = u_2^0$ then the arbitration procedure $\mathbf{x}^*$ has the property that $u_1(\mathbf{x}^*) = u_2(\mathbf{x}^*)$.

$$u_1(\mathbf{x}') = u_2(\mathbf{x}) \tag{10.11}$$
$$u_2(\mathbf{x}') = u_1(\mathbf{x}) \tag{10.12}$$

Remark 10.24. Assumption 6 simply states that if $(u_1, u_2) \in P(\mathbf{A}, \mathbf{B})$ (for $u_1, u_2 \in \mathbb{R}$), then $(u_2, u_1) \in P(\mathbf{A}, \mathbf{B})$ also. Thus, $P(\mathbf{A}, \mathbf{B})$ is symmetric in $\mathbb{R}^2$ about the line $y = x$. Inspection of Figure 10.1 reveals this is (in fact) true.

Remark 10.25. Our goal is to now show that there is an arbitration procedure $\mathbf{x}^* \in \Delta_{nm}$ that satisfies these assumptions and that the resulting pair $(u_1(\mathbf{x}^*, u_2(\mathbf{x}^*)) \in P(\mathbf{A}, \mathbf{B})$ is unique. This is *Nash's Bargaining Theorem*.

4. Nash's Bargaining Theorem

We begin our proof of Nash's Bargaining Theorem with two lemmas. We will not prove the first as it requires a bit more analysis than is required for the rest of the notes. The interested reader may wish to take Math 312 to see the proof of this lemma.

Lemma 10.26 (Weirstrass' Theorem). *Let S be a non-empty closed and bounded set in $\mathbb{R}^n$ and let z be a continous mapping with $z : S \to \mathbb{R}$. Then the optimization problem:*

$$\begin{cases} \max & z(\mathbf{x}) \\ s.t. & \mathbf{x} \in S \end{cases} \tag{10.13}$$

has at least one solution $\mathbf{x}^* \in S$.

Lemma 10.27. *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a two-player matrix game with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{n\times m}$. Let $(u_1^0, u_2^0) \in P(\mathbf{A}, \mathbf{B})$. The following quadratic programming problem:*

$$
(10.14)\quad \begin{cases}
\max\ (u_1 - u_1^0)(u_2 - u_2^0) \\
s.t.\ \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{A}_{ij}x_{ij} - u_1 = 0 \\
\displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{B}_{ij}x_{ij} - u_2 = 0 \\
\displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} = 1 \\
x_{ij} \geq 0 \qquad\qquad i = 1, \dots, m,\ j = 1, \dots, n \\
u_1 \geq u_1^0 \\
u_2 \geq u_2^0
\end{cases}
$$

has at least one global optimal solution $(u_1^*, u_2^*, \mathbf{x}^*)$. Furthermore if $(u_1', u_2', \mathbf{x}')$ is an alternative optimal solution, then $u_1^* = u_1'$ and $u_2^* = u_2'$.

Proof. By the same argument as in the proof of Theorem 10.11 the feasible region of this problem is a closed bounded and convex set. Moreover, since $(u_1^0, u_2^0) \in P(\mathbf{A}, \mathbf{B})$ we know that there is some $\mathbf{x}^0$ satisfying the constraints given in Expression 10.7 and that the tuple $(u_1^0, u_2^0, \mathbf{x}^0)$ is feasible to this problem. Thus, the feasible region is non-empty. Thus applying Lemma 10.26 we know that there is at least one (global optimal) solution to this problem.

To see the uniqueness of (u_1^*, u_2^*), suppose that $M = (u_1^* - u_1^0)(u_2^* - u_2^0)$ and we have a second solution $(u_1', u_2', \mathbf{x}')$ so that (without loss of generality) $u_1' > u_1^*$ and $u_2' < u_2^*$ but $M = (u_1' - u_1^0)(u_2' - u_2^0)$. We showed that $P(\mathbf{A}, \mathbf{B})$ is convex (see Theorem 10.9). Then there is some feasible $(u_1'', u_2'', \mathbf{x}'')$ so that:

$$
(10.15)\quad u_i'' = \frac{1}{2}u_i^* + \frac{1}{2}u_i'
$$

for $i = 1, 2$. Evaluating the objective function at this point yields:

$$
(10.16)\quad (u_1'' - u_1^0)(u_2'' - u_2^0) = \left(\frac{1}{2}u_1^* + \frac{1}{2}u_1' - u_1^0\right)\left(\frac{1}{2}u_2^* + \frac{1}{2}u_2' - u_2^0\right)
$$

Expanding yields:

$$
(10.17)\quad \left(\frac{1}{2}u_1^* - \frac{1}{2}u_1^0 + \frac{1}{2}u_1' - \frac{1}{2}u_1^0\right)\left(\frac{1}{2}u_2^* - \frac{1}{2}u_2^0 + \frac{1}{2}u_2' - \frac{1}{2}u_2^0\right) = \\
\frac{1}{4}\left((u_1^* - u_1^0) + (u_1' - u_1^0)\right)\left((u_2^* - u_2^0) + (u_2' - u_2^0)\right)
$$

Let $H_i^* = (u_i^* - u_i^0)$ and $H_i' = (u_i' - u_i^0)$ for $i = 1, 2$. Then our expression reduces to:

$$
(10.18)\quad \frac{1}{4}(H_1^* + H_1')(H_2^* + H_2') = \frac{1}{4}\left(H_1^*H_2^* + H_1'H_2^* + H_1^*H_2' + H_1'H_2'\right)
$$

We have the following:

(1) $H_1^* H_2^* = M$ (by definition).
(2) $H_1' H_2' = M$ (by assumption).
(3) $H_1' H_2^* = (u_1' - u_1^0)(u_2^* - u_2^0) = u_1' u_2^* - u_1' u_2^0 - u_2^* u_1^0 + u_1^0 u_2^0$
(4) $H_1^* H_2' = (u_1^* - u_1^0)(u_2' - u_2^0) = u_1^* u_2' - u_1^* u_2^0 - u_2' u_1^0 + u_1^0 u_2^0$

We can write:

$$H_1' H_2^* + H_1^* H_2' = \left(u_1' u_2^* - u_1' u_2^0 - u_2^* u_1^0 + u_1^0 u_2^0\right) + \left(u_1^* u_2' - u_1^* u_2^0 - u_2' u_1^0 + u_1^0 u_2^0\right) \tag{10.19}$$

We can write:

$$\begin{multline} H_1^* H_2^* + H_1' H_2^* + H_1^* H_2' + H_1' H_2' = 2M + H_1^* H_2' + H_1' H_2' = \\ 4M + H_1^* H_2' + H_1' H_2' - 2M = 4M + H_1^* H_2' + H_1' H_2' - H_1^* H_2^* - H_1' H_2' \end{multline} \tag{10.20}$$

Expanding $H_1^* H_2^*$ and $H_1' H_2'$ yields:

(1) $H_1^* H_2^* = (u_1^* - u_1^0)(u_2^* - u_2^0) = u_1^* u_2^* - u_1^* u_2^0 - u_2^* u_1^0 + u_1^0 u_2^0$
(2) $H_1' H_2' = (u_1' - u_1^0)(u_2' - u_2^0) = u_1' u_2' - u_1' u_2^0 - u_2' u_1^0 + u_1^0 u_2^0$

Now, simplifying:

$$\begin{multline} H_1^* H_2' + H_1' H_2' - H_1^* H_2^* - H_1' H_2' = \left(u_1' u_2^* - u_1' u_2^0 - u_2^* u_1^0 + u_1^0 u_2^0\right) + \\ \left(u_1^* u_2' - u_1^* u_2^0 - u_2' u_1^0 + u_1^0 u_2^0\right) - \left(u_1^* u_2^* - u_1^* u_2^0 - u_2^* u_1^0 + u_1^0 u_2^0\right) - \\ \left(u_1' u_2' - u_1' u_2^0 - u_2' u_1^0 + u_1^0 u_2^0\right) \end{multline} \tag{10.21}$$

This simplifies to:

$$\begin{multline} H_1^* H_2' + H_1' H_2' - H_1^* H_2^* - H_1' H_2' = u_1' u_2^* + u_1^* u_2' - u_1^* u_2^* - u_1' u_2' = \\ (u_1^* - u_1')(u_2' - u_2^*) \end{multline} \tag{10.22}$$

Thus:

$$\begin{multline} \frac{1}{4}\left(H_1^* H_2^* + H_1' H_2^* + H_1^* H_2' + H_1' H_2' = 2M + H_1^* H_2' + H_1' H_2'\right) = \\ \frac{1}{4}\left(4M + H_1^* H_2' + H_1' H_2' - H_1^* H_2^* - H_1' H_2'\right) = \\ M + \frac{1}{4}\left(H_1^* H_2' + H_1' H_2' - H_1^* H_2^* - H_1' H_2'\right) = M + \frac{1}{4}(u_1^* - u_1')(u_2' - u_2^*) > M \end{multline} \tag{10.23}$$

because $(u_1^* - u_1')(u_2' - u_2^*) > 0$ by our assumption that $u_1' > u_1^*$ and $u_2' < u_2^*$. But since we assumed that M was the maximum value the objective function attained, we know that we must have $u_1^* = u_1'$ and $u_2^* = u_2'$. This completes the proof. □

Theorem 10.28 (Nash' Bargaining Theorem). *Let $\mathcal{G} = (\mathbf{P}, \Sigma, \mathbf{A}, \mathbf{B})$ be a two-player matrix game with $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{m \times n}$ with $(u_1^0, u_2^0) \in P(\mathbf{A}, \mathbf{B})$ the status quo. Then there is at least one arbitration procedure $\mathbf{x}^* \in \Delta_{mn}$ satisfying the 6 assumptions of Nash and moreover the payoffs $u_1(\mathbf{x}^*)$ and $u_2(\mathbf{x}^*)$ are the unique optimal point in $P(\mathbf{A}, \mathbf{B})$.*

Proof. Consider the quadratic programming problem from Lemma 10.27.

$$
(10.24)\quad \begin{cases} \max & (u_1 - u_1^0)(u_2 - u_2^0) \\ s.t. & \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{A}_{ij}x_{ij} - u_1 = 0 \\ & \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} \mathbf{B}_{ij}x_{ij} - u_2 = 0 \\ & \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} = 1 \\ & x_{ij} \geq 0 \qquad\qquad i = 1, \dots, m,\ j = 1, \dots, n \\ & u_1 \geq u_1^0 \\ & u_2 \geq u_2^0 \end{cases}
$$

It suffices to show that the solution of this quadratic program provides an arbitration procedure $\mathbf{x}$ satisfying Nash's assumptions. Uniqueness follows immediately from Lemma 10.27. Denote the feasible region of this problem by $F(\mathbf{A}, \mathbf{B})$. That is $F(\mathbf{A}, \mathbf{B})$ is the set of all tuples $(u_1, u_2, \mathbf{x})$ satisfying the constraints of Problem 10.24. Clearly $u_1 = u_1(\mathbf{x})$ and $u_2 = u_2(\mathbf{x})$.

Before proceeding, recall that $Q(\mathbf{A}, \mathbf{B})$, the payoff region for the competitive game $\mathcal{G}$ is contained in $P(\mathbf{A}, \mathbf{B})$. Clearly if u_1^0, u_2^0 is chosen as an equilibrium for the competitive game, we know that $(u_1^0, u_2^0) \in P(\mathbf{A}, \mathbf{B})$. Thus there is a $\mathbf{x}^0$ so that $(u_1^0, u_2^0, \mathbf{x}^0) \in F(\mathbf{A}, \mathbf{B})$ and it follows that 0 is a lower bound for the maximal value of the objective function.

Assumption 1: By construction of this problem, we know that $u_1(\mathbf{x}^*) \geq u_1^0$ and $u_2(\mathbf{x}^*) \geq u_2^0$.

Assumption 2: By Lemma 10.27 any solution $(u_1^*, u_2^*, \mathbf{x}^*)$ has unique u_1^* and u_2^*. Thus, any other feasible solution $(u_1, u_2, \mathbf{x})$ must have the property that either $u_1 < u_1^*$ or $u_2 < u_2^*$. Therefore, the (u_1^*, u_2^*) must be Pareto optimal.

Assumption 3: Since the constraints of Problem 10.24 properly contain the constraints in Expression 10.7, the assumption of feasibility is ensured.

Assumption 4: Suppose that $P' \subseteq P(\mathbf{A}, \mathbf{B})$. Then there is a subset $F' \subseteq F(\mathbf{A}, \mathbf{B})$ corresponding to P'. If $(u_1^*, u_2^*) \in P'$ and $(u_1^0, u_2^0) \in P'$, it follows that $(u_1^*, u_2^*, \mathbf{x}^*) \in F'$ and $(u_1^0, u_2^0, \mathbf{x}^0) \in F'$. Then we can define the new optimization problem:

$$
(10.25)\quad \begin{cases} \max & (u_1 - u_1^0)(u_2 - u_2^0) \\ s.t. & (u_1, u_2, \mathbf{x}) \in F \\ & (u_1, u_2, \mathbf{x}) \in F' \end{cases}
$$

These constraints are consistent and since

$$
(10.26)\quad (u_1^* - u_1^0)(u_2^* - u_2^0) \geq (u_1' - u_1^0)(u_2' - u_2^0)
$$

for all $(u_1', u_2', \mathbf{x}') \in F$ it follows that Expression 10.26 must also hold for all $(u_1', u_2', \mathbf{x}') \in F' \subseteq F$. Thus $(u_1^*, u_2^*, \mathbf{x}^*)$ is also an optimal solution for Problem 10.25.

Assumption 5: Consider the problem replacing the objective function with the new objective:

$$(10.27)\quad \left(\alpha_1 u_1 + \beta_1 - (\alpha_1 u_1^0 - \beta_1)\right)\left(\alpha_2 u_2 + \beta_2 - (\alpha_2 u_2^0 - \beta_2)\right) = \alpha_1\alpha_2(u_1 - u_1^0)(u_2 - u_2^0)$$

The constraints of the problem will not be changed since we assume that $\alpha_1, \alpha_2 \geq 0$. To see this note that linear transformation of the payoff values implies the new constraints:

$$(10.28)\quad \sum_{i=1}^{m}\sum_{j=1}^{n}(\alpha_1 \mathbf{A}_{ij} + \beta_1)x_{ij} - (\alpha_1 u_1 + \beta_1) = 0$$
$$\iff \alpha_1 \sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{A}_{ij}x_{ij} + \beta_1 \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} - (\alpha_1 u_1 + \beta_1) = 0 \iff$$
$$\alpha_1 \sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{A}_{ij}x_{ij} + \beta_1 - \alpha_1 u_1 - \beta_1 = 0 \iff \sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{A}_{ij}x_{ij} - u_1 = 0$$

$$(10.29)\quad \sum_{i=1}^{m}\sum_{j=1}^{n}(\alpha_2 \mathbf{B}_{ij} + \beta_2)x_{ij} - (\alpha_2 u_2 + \beta_2) = 0$$
$$\iff \alpha_2 \sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{B}_{ij}x_{ij} + \beta_2 \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} - (\alpha_2 u_2 + \beta_2) = 0 \iff$$
$$\alpha_2 \sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{B}_{ij}x_{ij} + \beta_2 - \alpha_2 u_2 - \beta_2 = 0 \iff \sum_{i=1}^{m}\sum_{j=1}^{n}\mathbf{B}_{ij}x_{ij} - u_2 = 0$$

The final two constraints adjusted are:

$$(10.30)\quad \alpha_1 u_1 + \beta_1 \geq \alpha_1 u_1^0 + \beta_1 \iff u_1 \geq u_1^0$$
$$(10.31)\quad \alpha_2 u_2 + \beta_2 \geq \alpha_2 u_2^0 + \beta_2 \iff u_2 \geq u_2^0$$

Since the constraints are identical, it is clear that the changing the objective function to the function in Expression 10.27 will not affect the solution since we are simply scaling the value by a positive number.

Assumption 6 Suppose that $u^0 = u_1^0 = u_2^0$ and $P(\mathbf{A}, \mathbf{B})$ is symmetric. Assuming that P is symmetric (from Assumption 6), we know that $(u_2^*, u_1^*) \in P(\mathbf{A}, \mathbf{B})$ and that:

$$(10.32)\quad (u_1^* - u_1^0)(u_2^* - u_2^0) = (u_1^* - u^0)(u_2^* - u^0) = (u_2^* - u^0)(u_1^* - u^0)$$

Thus, for some $\mathbf{x}'$ we know that $(u_2^*, u_1^*, \mathbf{x}') \in F(\mathbf{A}, \mathbf{B})$ since $(u_2^*, u_1^*) \in P(\mathbf{A}, \mathbf{B})$. But this feasible solution achieves the same objective value as the optimal solution $(u_1^*, u_2^*, \mathbf{x}^*) \in F(\mathbf{A}, \mathbf{B})$ and thus Lemma 10.27 we know that $u_1^* = u_2^*$.

Again, uniqueness of the *values* $u_1(\mathbf{x}^*)$ and $u_2(\mathbf{x}^*)$ follows from Lemma 10.27. This completes the proof. □

Example 10.29. Consider the Battle of the Sexes game. Recall:

$$\mathbf{A} = \begin{bmatrix} 2 & -1 \\ -1 & 1 \end{bmatrix}$$
$$\mathbf{B} = \begin{bmatrix} 1 & -1 \\ -1 & 2 \end{bmatrix}$$

We can now find the arbitration process that produces the best cooperative strategy for the two players. We'll assume that our status quo is the Nash equilibrium payoff $u_1^0 = u_2^0 = 1/5$ (see Exercise 73). Then the problem we must solve is:

$$(10.33)\qquad \begin{aligned} \max \quad & \left(u_1 - \frac{1}{5}\right)\left(u_2 - \frac{1}{5}\right) \\ s.t. \quad & 2x_{11} - x_{12} - x_{21} + x_{22} - u_1 = 0 \\ & x_{11} - x_{12} - x_{21} + 2x_{22} - u_2 = 0 \\ & x_{11} + x_{12} + x_{21} + x_{22} = 1 \\ & x_{ij} \geq 0 \quad i = 1, 2,\ j = 1, 2 \\ & u_1 \geq \frac{1}{5} \\ & u_2 \geq \frac{1}{5} \end{aligned}$$

The solution, which you can obtain using Matlab (see Figure 10.3), yields $x_{11} = x_{12} = 1/2$, $x_{21} = x_{12} = 0$. At this point, $u_1 = u_2 = 3/2$ (as required by symmetry). This means that Players 1 and 2 should flip a fair coin to decide whether they will both follow Strategy 1 or Strategy 2 (i.e., boxing or ballet). This essentially tell us that in a happy marriage, 50% of the time one partner decides what to do and 50% of the time the other partner decides what to do. This solution is shown on the set $P(\mathbf{A}, \mathbf{B})$ in Figure 10.2.

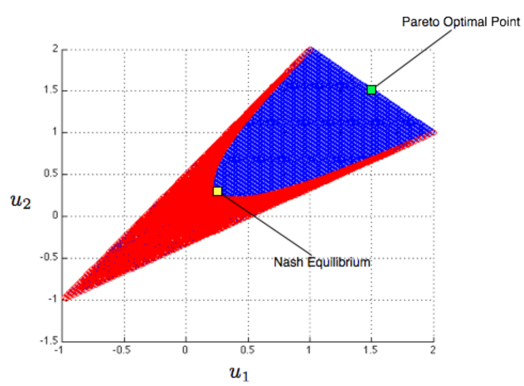


Figure 10.2. The Pareto Optimal, Nash Bargaining Solution, to the Battle of the Sexes is for each player to do what makes them happiest 50% of the time. This seems like the basis for a fairly happy marriage, and it yields a Pareto optimal solution, shown by the green dot.

The following Matlab code will solve the Nash bargaining problem associated with the Battle of the Sexes game. Note that we are solving a *maximization* problem, but Matlab solve *mnimization* problems by default. Thus we change the sign on the objective matrices. As before, calling `quadprog` will solve the maximization problem associated with Battle of the Sexes. We must compute the appropriate matrices and vectors for this problem. In order

```
%%DON'T FORGET MATLAB USES 0.5*x^T*Q*x + c^Tx
Q = [[0 0 0 0 0 0];[0 0 0 0 0 0];[0 0 0 0 0 0];[0 0 0 0 0 0];[0 0 0 0 0 1];[0 0 0 0 1 0]];
c = [0 0 0 0 -1/5 -1/5]';
H = [[2 -1 -1 1 -1 0];[1 -1 -1 2 0 -1];[1 1 1 1 0 0]];
r = [0 0 1]';
lb = [0 0 0 0 1/5 1/5]';
ub = [inf inf inf inf inf inf];
A = [];
b = [];

[x obj] = quadprog(-Q,-c,A,b,H,r,lb,ub);
```

Figure 10.3. Matlab input for solving Nash's bargaining problem with the Battle of the Sexes problem. Note that we are solving a *maximization* problem, but Matlab solve *mnimization* problems by default. Thus we change the sign on the objective matrices.

to see that this is the correct problem, note we can read the $\mathbf{H}$ matrix and $\mathbf{r}$ vector directly from the equality constraints of Problem 10.33. There are no inequality constraints (that are not bounds) thus $\mathbf{A} = \mathbf{b} = []$, the empty matrix. The matrix and vector that make up the objective functions can be found by noting that if we let our vector of decision variables be $[x_{11}, x_{12}, x_{21}, x_{22}, u_1, u_2]^T$, then we have:

$$\left(u_1 - \frac{1}{5}\right)\left(u_2 - \frac{1}{5}\right) = u_1u_2 - \frac{1}{5}u_1 - \frac{1}{5}u_2 + \frac{1}{25} =$$
$$\begin{bmatrix} x_{11} & x_{12} & x_{21} & x_{22} & u_1 & u_2 \end{bmatrix} \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ x_{21} \\ x_{22} \\ u_1 \\ u_2 \end{bmatrix} +$$
$$\begin{bmatrix} 0 & 0 & 0 & 0 & -\frac{1}{5} & -\frac{1}{5} \end{bmatrix} \begin{bmatrix} x_{11} \\ x_{12} \\ x_{21} \\ x_{22} \\ u_1 \\ u_2 \end{bmatrix} + \frac{1}{25} \tag{10.34}$$

Solving a maximization problem with this objective is the same as solving an optimization problem without the added constant 1/25. Thus, the 1/25 is dropped when we solve the problem in Matlab. There is no trick to determining these matrices from the objective function; you just have to have some intuition about matrix multiplication, which requires practice.

Remark 10.30. Nash's Bargaining Theorem is the beginning of the much richer subject of Cooperative Games, which we do not have time to cover in detail in these notes. This area of Game Theory is substantially different from the topics we have covered up till now. The interested reader should consult [**Mye01**] or [**Mor94**] for more details. Regarding Example 10.29, isn't it nice to have a happy ending?

Exercise 74. Use Nash's Bargaining theorem to show that players should trust each other and cooperate rather than defecting in Prisoner's dilemma.

CHAPTER 11

A Short Introduction to N-Player Cooperative Games

In this final chapter, we introduce some elementary results on N-player cooperative games, which extend the work we began in the previous chapter on Bargaining Games. Again, we will assume that the players in this game can communicate with each other. The goals of cooperative game theory are a little different than ordinary game theory. The goal in cooperative games is to study games in which it is in the players' best interest to come together in a *grand coalition* of cooperating players.

1. Motivating Cooperative Games

Definition 11.1 (Coalition of Players). Consider an N-player game $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$. Any set $S \subseteq \mathbf{P}$ is called a *coalition of players*. The set $S^c = \mathbf{P} \setminus S$ is the *dual coalition*. The coalition $\mathbf{P} \subseteq \mathbf{P}$ is called the *grand coalition*.

Remark 11.2. Heretofore, we've always written $P = \{P_1, \ldots, P_N\}$ however for the remainder of the chapter we'll assume that $\mathbf{P} = \{1, \ldots, N\}$. This will substantially simplify our notation.

Let $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ be an N-player game. Suppose within a coalition $S \subseteq \mathbf{P}$ with $S = \{i_1, \ldots, i_{|S|}\}$, the players $i_1, \ldots, i_{|S|}$ agree to play some strategy:

$$\boldsymbol{\sigma}_S = (\sigma_{i_{i_1}}, \ldots, \sigma_{i_{|S|}}) \in \Sigma_1 \times \cdots \times \Sigma_{i_{|S|}}$$

while players in $S^c = \{j_1, \ldots, j_{|S^c|}\}$ agree to play strategy:

$$\boldsymbol{\sigma}_{S^c} = (\sigma_{j_1}, \ldots, \sigma_{j_{|S^c|}})$$

Under these assumptions, we may suppose that the net payoff to coalition S is:

$$K_S = \sum_{i \in S} \pi_i(\boldsymbol{\sigma}_S, \boldsymbol{\sigma}_{S^c}) \tag{11.1}$$

That is, the cumulative payoff to coalition S is just the sum of the payoffs of the members of the coalition from payoff function π in the game $\mathcal{G}$. The payoff to the players in S^c is defined similarly as K_{S^c}. Then we can think of the coalitions as playing a two-player general sum game with payoff functions given by K_S and K_{S^c}.

Definition 11.3 (Two-Coalition Game). Given an N-player game $\mathcal{G} = (\mathbf{P}, \Sigma, \pi)$ and a coalition $S \subseteq \mathbf{P}$, with $S = \{i_1, \ldots, i_{|S|}\}$ and $S^c = \{j_1, \ldots, j_{|S^c|}\}$. The two-coalition game is the two-player game:

$$\mathcal{G}_S = \left(\{S, S^c\}, \left(\Sigma_{i_1} \times \cdots \times \Sigma_{i_{|S|}}\right) \times \left(\Sigma_{j_1} \times \cdots \times \Sigma_{j_{|S^c|}}\right), (K_S \times K_{S^c})\right)$$

Lemma 11.4. *For any Two-Coalition Game $\mathcal{G}_S$, there is a Nash equilibrium strategy for both the coalition S and its dual S^c.*

Exercise 75. Prove the previous lemma. [Hint: Use Nash's theorem.]

Definition 11.5 (Characteristic (Value) Function). Let S be a coalition defined over a N-player game $\mathcal{G}$. Then the value function $v : 2^{\mathbf{P}} \to \mathbb{R}$ is the expected payoff to S in the game $\mathcal{G}_S$ when both coalitions S and S^c play their Nash equilibrium strategy.

Remark 11.6. The characteristic or value function can be thought of as the net worth of the coalition to its members. Clearly

$$v(\emptyset) = 0$$

because the empty coalition can achieve no value. On the other hand,

$$v(\mathbf{P}) = \text{largest sum of all payoff values possible}$$

because a two-player game against the empty coalition will try to maximize the value of Equation 11.1. In general, $v(\mathbf{P})$ answers the question, "If all N players worked together to maximize the sum of their payoffs, which strategy would they all agree to chose and what would that sum be?"

2. Basic Results on Coalition Games

Definition 11.7 (Coalition Game). A coalition game is a pair $(\mathbf{P}, v)$ where $\mathbf{P}$ is the set of players and $v : 2^{\mathbf{P}} \to \mathbb{R}$ is a superadditive characteristic function.

Theorem 11.8. *If $S, T \subseteq \mathbf{P}$ and $S \cap T = \emptyset$, then $v(S) + v(T) \leq v(S \cup T)$.*

Proof. Within S and T, the players may choose a strategy (jointly) and independently to ensure that they receive at least $v(S) + v(T)$, however the value of the game $\mathcal{G}_{S\cup T}$ to Player 1 ($S \cup T$) may be larger than the result yielded when S and T make independent choices, thus $v(S \cup T) \geq v(S) + v(T)$. $\square$

Remark 11.9. The property in the previous theorem is called *superadditivity*. In general, we begin the study of cooperative N player games with the assumption that there is a mapping $v : 2^{\mathbf{P}} \to \mathbb{R}$ so that:

(1) $v(\emptyset) = 0$
(2) $v(S) + v(T) \leq v(S \cup T)$ for all $S, T \subseteq \mathbf{P}$.

The goal of cooperative N-player games is to define scenarios in which the *grand coalition*, $\mathbf{P}$, is stable; that is, it is in everyone's interest to work together in one large coalition $\mathbf{P}$. It is hoped that the value $v(S)$ will be divided (somehow) among the members of the coalition S and that by being in a coalition the players will improve their payoff over competing on their own.

Definition 11.10 (Inessential Game). A game is *inessential* if:

$$v(\mathbf{P}) = \sum_{i=1}^{N} v(\{i\})$$

A game that is not inessential is called *essential*.

Remark 11.11. An inessential game is one in which the total value of the grand coalition does *not* exceed the sum of the values to the players if they each played *against the world*. That is, there is no incentive for any player to join the grand coalition because there is no chance that they will receive more payoff if the total payoff to the grand coalition where divided among its members.

Theorem 11.12. *Let* $S \subseteq \mathbf{P}$. *In an inessential game,*

$$v(S) = \sum_{i \in S} v(\{i\})$$

Proof. We proceed by contradiction. Suppose not, then:

$$v(S) > \sum_{i \in S} v(\{i\})$$

by superadditivity. Now:

$$v(S^c) \geq \sum_{i \in S^c} v(\{i\})$$

and $v(\mathbf{P}) \geq v(S) + v(S^c)$ which implies that:

$$v(\mathbf{P} \geq v(S) + v(S^c) > \sum_{i \in S} v(\{i\}) + \sum_{i \in S^c} v(\{i\}) = \sum_{i \in \mathbf{P}} v(\{i\}).$$

Thus:

$$v(\mathbf{P}) > \sum_{i=1}^{N} v(\{i\})$$

and thus the coalition game is not inessential. □

Corollary 11.13. *A two-player zero sum game produces an inessential coalition game.*

Exercise 76. Prove the previous corollary.

Exercise 77. Consider the following three player cooperative game:
v(123) = 6
v(12) = 2
v(13) = 6
v(23) = 4
v(1) = v(2) = v(3) = 0
Is this an essential game? Why or why not?

3. Division of Payoff to the Coalition

Remark 11.14. Given a coalition game $(\mathbf{P}, v)$ the goal is to find an equitable way to divide $v(S)$ among the members of the coalition in such a way that the individual players prefer to be in the coalition rather than to leave it. This study clearly has implications for public policy and the division of society's combined resources.

The real goal is to determine some set of payoffs to the individual elements of the grand coalition $\mathbf{P}$ so that the grand coalition itself is stable.

Definition 11.15 (Imputation). Given a coalition game $(\mathbf{P}, v)$, a tuple $(x_1, \dots, x_N)$ (of payoffs to the individual players in $\mathbf{P}$) is called a *imputation* if:

(1) $x_i \geq v(\{i\}$ and
(2) $\sum_{i\in\mathbf{P}} x_i = v(\mathbf{P})$

Remark 11.16. The first criterion for a tuple $(x_1, \dots, x_N)$ to be an imputation says that each player must do better in the grand coalition then they would on their own (against the world). The second criterion says that the total allotment of payoff to the players cannot exceed the payoff received by the grand coalition itself. Essentially, this second criterion asserts that the coalition cannot go into debt to maintain its members. It is also worth noting that the condition $\sum_{i\in\mathbf{P}} x_i = v(\mathbf{P})$ is equivalent to a statement on Pareto optimality in so far as players all together can't expect to do any better than the net payoff accorded to the grand coalition.

Definition 11.17 (Dominance). Let $(\mathbf{P}, v)$ be a coalition game. Suppose $\mathbf{x} = (x_1, \dots, x_N)$ and $\mathbf{y} = (y_1, \dots, y_N)$ are two imputations. Then $\mathbf{x}$ *dominates* $\mathbf{y}$ over some coalition $S \subset \mathbf{P}$ if

(1) $x_i > y_i$ for all $i \in S$ and
(2) $\sum_{i\in S} x_i \leq v(S)$

Remark 11.18. The previous definition states that Players in coalition S prefer the payoffs they receive under $\mathbf{x}$ to the payoffs they receive under $\mathbf{y}$. Furthermore, these same players can threaten to *leave the grand coalition* $\mathbf{P}$ because they may actually improve their payoff by playing coalition S.

Definition 11.19 (Stable Set). A stable set $X \subseteq \mathbb{R}^n$ of imputations is a set satisfying:

(1) No payoff vector $\mathbf{x} \in X$ is dominated in any coalition by another coalition $\mathbf{y} \in X$ and
(2) All payoff vectors $\mathbf{y} \notin X$ are dominated by at least one vector $\mathbf{x} \in X$.

Remark 11.20. Stable sets are (in some way) very good sets of imputations in so far as they represent imputations that will make players want remain in the grand coalition.

4. The Core

Definition 11.21 (Core). Given a coalition game $(\mathbf{P}, v)$, the core is:

$$C(v) = \left\{ \mathbf{x} \in \mathbb{R}^n : \sum_{i=1}^{N} x_i = v(\mathbf{P}) \text{ and } \forall S \subseteq \mathbf{P} \left(\sum_{i\in S} x_i \geq v(S) \right) \right\}$$

Remark 11.22. Thus a vector $\mathbf{x}$ is in the core if it is an imputation (since clearly: $\sum_{i\in\mathbf{P}} x_i = v(\mathbf{P})$ and since $\{i\} \subset \mathbf{P}$ we know that $x_i \geq v(\{i\})$. However, it says substantially more than that.

Theorem 11.23. *The core is contained in every stable set.*

Proof. Let X be a stable set. If the core is empty, then it is contained in X. Therefore, suppose $\mathbf{x} \in C(v)$. If $\mathbf{x}$ is dominated by any vector $\mathbf{z}$ then there is a coalition $S \subset \mathbf{P}$ so that

$z_i > x_i$ for all $i \in S$ and $\sum_{i\in S} z_i \leq v(S)$. But then:

$$\sum_{i\in S} z_i > \sum_{i\in S} x_i \geq v(S)$$

by definition of the core. Thus, $\sum_{i\in S} z_i > v(S)$ and $\mathbf{z}$ cannot dominate $\mathbf{x}$, a contradiction. $\square$

Theorem 11.24. *Let $(\mathbf{P}, v)$ be a coalition game. Consider the linear programming problem:*

$$\left\{\begin{array}{l} \min\ x_1 + \cdots + x_N \\ s.t.\ \displaystyle\sum_{i\in S} x_i \geq v(S) \quad \forall S \subseteq \mathbf{P} \end{array}\right. \tag{11.2}$$

If there is no solution $\mathbf{x}^*$ so that $\sum_{i=1}^{N} x_i = v(P)$, then $C(v) = \emptyset$.

Exercise 78. Prove the preceding theorem. [Hint: Note that the constraints enforce the requirement:

$$\forall S \subseteq \mathbf{P} \left(\sum_{i\in S} x_i \geq v(S)\right)$$

while the objective function yields $\sum_{i=1}^{N} x_i$.]

Corollary 11.25. *The core of a coalition game $(\mathbf{P}, v)$ may be empty.*

Exercise 79. Find the core of the previous three player cooperative game.

Theorem 11.26 (Bondarvera-Shapley Theorem). *Let $(\mathbf{P}, v)$ be a coalition game with $|\mathbf{P}| = N$. The core $C(v)$ is non-empty if and only if there exists $y_1, \dots, y_{2^N}$ where each y_i corresponds to a set $S_i \subseteq \mathbf{P}$ so that:*

$$\left\{\begin{array}{l} v(\mathbf{P}) = \displaystyle\sum_{i=1}^{2^N} y_i v(S_i) \\ \displaystyle\sum_{S_i \supseteq \{j\}} y_i = 1 \quad \forall j \in \mathbf{P} \\ y_i \geq 0 \quad \forall S_i \subseteq \mathbf{P} \end{array}\right.$$

Proof. The dual linear programming problem (See Chapter 8.6) for Problem 11.2 is:

$$\left\{\begin{array}{l} \max\ \displaystyle\sum_{i=1}^{2^N} y_i v(S_i) \\ s.t.\ \displaystyle\sum_{S_i \supseteq \{j\}} y_i = 1 \quad \forall j \in \mathbf{P} \\ y_i \geq 0 \quad \forall S_i \subseteq \mathbf{P} \end{array}\right. \tag{11.3}$$

To see this, we note that there are 2^N constraints in Problem 11.2 and N variables and thus there will be N constraints in the dual problem, but 2^N variables and the resulting dual problem is Problem 11.3. By Theorem 8.16 (the Strong Duality Theorem), Problem 11.3 has a solution if and only if Problem 11.2 does and moreover the objective functions at optimality coincide. $\square$

Exercise 80. Prove that Problems 11.2 and 11.3 are in fact dual linear programming problems by showing that they have the same KKT conditions.

Corollary 11.27. *A non-empty core is not necessarily a singleton.*

Exercise 81. Prove the preceding corollary. [Hint: Think about alternative optimal solutions.]

Exercise 82. Show that computing the core is an exponential problem even though solving a linear programming problem is known to be polynomial in the size of the problem.

Remark 11.28. The core can be thought of as the possible "equilibrium" imputations that smart players will agree to and that cause the grand coalition to hold together; i.e., no players or coalition have any motivation to leave the coalition. Unfortunately, the fact that the core may be empty is not helpful.

5. Shapley Values

Definition 11.29 (Shapley Values). Let $(\mathbf{P}, v)$ be a coalition game with N players. Then the Shapley value for Player i is:

$$x_i = \phi_i(v) = \sum_{S \subseteq \mathbf{P} \setminus \{i\}} \frac{|S|!(N - |S| - 1)!}{N!} \left(v\left(S \cup \{i\}\right) - v(S)\right) \tag{11.4}$$

Remark 11.30. The Shapley value is the *average extra value* Player i contributes to each possible coalition that might form. Imagine forming the grand coalition one player at a time. There are $N!$ ways to do this. Hence, in an average, $N!$ is in the denominator of the Shapley value.

Now, if we've formed coalition S (on our way to forming $\mathbf{P}$), then there are $|S|!$ ways we could have done this. Each of these ways yields $v(S)$ in value because the characteristic function does not value how a coalition is formed, only the members of the coalition.

Once we add i to the coalition S, the new value is $v\left(S \cup \{i\}\right)$ and the value player i added was $v\left(S \cup \{i\}\right) - v(S)$. We then add the other $N - |S| - 1$ players to achieve the grand coalition. There are $(N - |S| - 1)!$ ways of doing this.

Thus, the extra value Player i adds in each case is $v\left(S \cup \{i\}\right) - v(S)$ multiplied by $|S|!(N - |S| - 1)!$ for each of the possible ways this exact scenario occurs. Summing over all possible subsets S and dividing by $N!$, as noted, yields the average excess value Player i brings to a coalition.

Remark 11.31. We state, but do not prove, the following theorem. The proof rests on the linear properties of averages. That is, we note that is a linear expression in $v(S)$ and $v\left(S \cup \{i\}\right)$.

Theorem 11.32. *For any coalition game $(\mathbf{P}, v)$ with N players, then:*

1. $\phi_i(v) \geq v(\{i\})$
2. $\sum_{i \in \mathbf{P}} \phi_i(v) = v(\mathbf{P})$
3. *From (1) and (2) we conclude that $(\phi_1(v), \dots, \phi_N(v))$ is an imputation.*
4. *If for all $S \subseteq \mathbf{P}$, $v\left(S \cup \{i\}\right) = v\left(S \cup \{j\}\right)$ with $i, j \notin S$, then $\phi_i(v) = \phi_j(v)$.*
5. *If v and w are two characteristic functions in coalition games $(\mathbf{P}, v)$ and $(\mathbf{P}, w)$, then $\phi_i(v + w) = \phi_i(v) + \phi_i(w)$ for all $i \in \mathbf{P}$.*

(6) *If* $v(S \cup \{i\}) = v(S)$ *for all* $S \subseteq \mathbf{P}$ *with* $i \notin S$ *then* $\phi_i(v) = 0$ *because Player* i *contributes nothing to the grand coalition.*

Exercise 83. Prove the previous theorem.

Exercise 84. Find the Shapley values for each player in the previous three player game.

Remark 11.33. There is substantially more information on coalition games and economists have spent a large quantity of time investigating the various properties of these games. The interested reader should consider [**LR89**] and [**Mye01**] for more detailed information. Additionally, for general game theoretic research the journals, *The International Journal of Game Theory*, *Games and Economic Behavior* and *IEEE Trans. Automatic Control* have a substantial number of articles on game theory, including coalition games.

Bibliography

[BCG01a] E. R. Berlekamp, J. H. Conway, and R. K. Guy, *Winning ways for your mathematical plays*, vol. 1, A. K. Peters, 2001.

[BCG01b] ______, *Winning ways for your mathematical plays*, vol. 2, A. K. Peters, 2001.

[BCG01c] ______, *Winning ways for your mathematical plays*, vol. 3, A. K. Peters, 2001.

[BCG01d] ______, *Winning ways for your mathematical plays*, vol. 4, A. K. Peters, 2001.

[BJS04] Mokhtar S. Bazaraa, John J. Jarvis, and Hanif D. Sherali, *Linear programming and network flows*, Wiley-Interscience, 2004.

[BO82] Tamer Başar and Geert Jan Olsder, *Dynamic noncooperative game theory*, Academic Press, 1982.

[Bra04] S. J. Brams, *Game theory and politics*, Dover Press, 2004.

[BSS06] M. S. Bazaraa, H. D. Sherali, and C. M. Shetty, *Nonlinear programming: Theory and algorithms*, John Wiley and Sons, 2006.

[Coh03] J. L. Cohen, *Multiobjective progamming and planning*, Dover, 2003.

[Con76] J. H. Conway, *On numbers and games*, Academic Press, 1976.

[DJLS00] E. J. Dockner, S. Jørgensen, N. V. Long, and G. Sorger, *Differential games in economics and management science*, Cambridge University Press, 2000.

[Dre81] M. Dresher, *The mathematics of games of strategy*, Dover Press, 1981.

[LH61] C. E. Lemke and J. T. Howson, *Equilibrum points of bimatrix games*, J. Soc. Indust. Appl. Math. **12** (1961), no. 2, 413–423.

[LR89] R. D. Luce and H. Raiffa, *Games and decisions: Introduction and critical survey*, Dover Press, 1989.

[Mor94] P. Morris, *Introduction to Game Theory*, Springer, 1994.

[MS64] O. L. Mangasarian and H. Stone, *Two-Person Nonzero-Sum Games and Quadratic Programming*, J. Math. Analysis and Applications **9** (1964), 348–355.

[MT03] J. E. Marsden and A. Tromba, *Vector calculus*, 5 ed., W. H. Freeman, 2003.

[Mun00] J. Munkres, *Topology*, Prentice Hall, 2000.

[Mye01] R. B. Myerson, *Game theory: Analysis of conflict*, Harvard University Press, 2001.

[PR71] T. Parthasarathy and T. E. S. Raghavan, *Some topics in two-person games*, Elsevier Science LTD, 1971.

[vNM04] J. von Neumann and O. Morgenstern, *The Theory of Games and Economic Behavior*, 60th anniversary edition ed., Princeton University Press, 2004.

[Wei97] J. W. Weibull, *Evolutionary game theory*, MIT Press, 1997.

[WV02] W. L. Winston and M. Venkataramanan, *Introduction to mathematical programming: Applications and algorithms*, vol. 1, Duxbury Press, 2002.