International Series in
Operations Research & Management Science

Alan Washburn

Two-Person Zero-Sum Games

Fourth Edition

informs

Springer

International Series in Operations Research & Management Science

Volume 201

For further volumes:
http://www.springer.com/series/6161

Alan Washburn

Two-Person Zero-Sum Games

Fourth Edition

Springer

Alan Washburn
Operations Research Department
Naval Postgraduate School
Monterey, CA, USA

Supplementary material for this volume is available at www.springer.com.
Search for this title by print ISBN: 978-1-4614-9049-4

ISSN 0884-8289
ISBN 978-1-4614-9049-4 ISBN 978-1-4614-9050-0 (eBook)
DOI 10.1007/978-1-4614-9050-0
Springer New York Heidelberg Dordrecht London

Library of Congress Control Number: 2013950900

Printed on acid-free paper

Springer is part of Springer Science+Business Media (www.springer.com)

Foreword

Practical men, who believe themselves to be quite exempt from any intellectual influences, are usually the slaves of some defunct economist....

J. M. Keynes

This book is unusual among books on game theory in considering only the special case where there are exactly two players whose interests are completely opposed—the two-person zero-sum (TPZS) case. A few words of explanation are in order about why such an apparently small part of John von Neumann and Oskar Morgenstern's (vN&M's) grander vision in their seminal book *Theory of Games and Economic Behavior* should now itself be thought a suitable subject for a textbook. Our explanation will involve a brief review of the history of game theory and some speculation about its future.

The vN&M book appeared in 1944. It played to rave reviews, not only from academics but in the popular press. There are few instances in history of a theory being born and taken so seriously and so suddenly. The May 1949 issue of *Fortune*, for example, contained a 20-page article describing the theory's accomplishments in World War II and projecting further successes in industry. Williams (1954) wrote *The Compleat Strategyst* as a book intended to make the theory more widely accessible to nonmathematicians. There were strong efforts to further develop the theory, particularly at Princeton University and at the RAND Corporation. The American Mathematical Society published a sequence of four volumes between 1950 and 1964 devoted entirely to the subject. Much of this interest has continued to the present; there are now several dedicated journals, two of which are published by the Game Theory Society.

The initial euphoria was doomed to be disappointed. Game theory has had its practical successes, but applications could hardly have kept pace with initial expectations. An explanation of why this has happened requires us to identify two extreme forms of game. One extreme is single player (SP) games, which can be thought of as games in which all decision makers who can affect the outcome have the same goal. The natural mode of social interaction in SP games is cooperation, since there is no point in competition when all parties have the same goal. Use of the word "game" to describe such situations could even be considered a misuse of the term, but we will persist for the moment. The other extreme is where all players who can affect the outcome have opposite goals. Since the idea of "opposite" gets difficult unless there are exactly two players, this is the TPZS case. The natural mode of social interaction in TPZS games is competition, since neither player has anything to gain by cooperation. Games that fall at neither extreme will be referred to as N-person (NP) games. In general, the behavior of players in NP games can be expected to exhibit aspects of both cooperation and competition. Thus, SP and TPZS games are the extreme cases.

vN&M dealt with NP games, as well as the SP and TPZS specializations. The SP theory had actually been widely applied before the book's appearance, although vN&M made an important contribution to decision making under uncertainty that is reviewed in Chap. 1. One major contribution of the book was to put the theory of finite, TPZS games on a solid basis. The famous "minimax" Theorem asserts the existence of rational behavior as long as randomized strategies are permitted. The vN&M definition of "solution" for TPZS games has stood the test of time, with subsequent years seeing the idea generalized, rather than revised. vN&M also considered rational behavior in NP games, and in fact the greater part of their book is devoted to that topic. However, the NP theory can be distinguished from the TPZS theory in being descriptive rather than normative. It would be good to have a normative theory, since one of our hopes in studying games is to discover "optimal" ways of playing them. There were thus two natural tasks for game theorists in the years following 1944:

1. Generalization and application of the minimax Theorem for TPZS games. The vN&M proof was not constructive, so there were computational issues. Also, the original proof applied only to finite games.
2. Further investigation of what "solution" should mean for NP games. The vN&M concept was not fully understood, and there was the problem of finding a normative concept that could serve as a foundation for applications.

Most effort went initially to the first task, shifting gradually toward the second. In the four volumes published by the American Mathematical Society, the percentage of papers devoted to TPZS games in 1950, 1954, 1957, and 1964 was 52, 47, 30, and 25. This decreasing trend has continued. In 1985, only about 20 % of the papers published in the *International Journal of Game Theory* were devoted to TPZS games, and about 10 % currently (2013). There are several reasons for this shifting emphasis:

1. Theoretical progress in TPZS games came rather quickly. The minimax Theorem or something close to it holds in many infinite games. Given the extended minimax Theorem, there is not much more to be said *in principle* about solutions of TPZS games, even though serious computational and modeling issues remain.
2. While many solution concepts were proposed for NP games, none became widely accepted as a standard. The original vN&M concept was shown to be

defective in 1962, when William Lucas exhibited a game with no solution. This unsettled state of affairs in a theory with such obvious potential for application naturally attracts the attention of theoreticians.

3. Although there are industrial and political applications, TPZS games are applicable mainly to military problems and their surrogates—sports and parlor games. NP games are potentially much more broadly applicable.

With the decline in interest in TPZS games among academics has come a similar decline among practitioners, even among the practicing military analysts who have the most to gain from an awareness of the TPZS theory. The inevitable result is neglect of a relevant, computationally tractable body of theory.

This book was written to make TPZS theory and application accessible to anyone whose quantitative background includes calculus, probability, and enough computer savvy to run Microsoft Excel™. Since the theory has been well worked out by this time, a large part of the book is devoted to the practical aspects of problem formulation and computation. It is intended to be used as the basis for instruction, either self study or formal, and includes numerous exercises.

Preface to the Fourth Edition

This edition of *Two-Person Zero-Sum Games*, the first Springer edition, differs significantly from the previous three INFORMS editions.

- The practice of distributing multiple executable files has been discontinued. Instead, this edition comes with a single Microsoft Excel™ workbook *TPZS.xlsb* that the user is expected to download. The Solver addin that comes with Excel will be sufficient for linear programming exercises.
- Chapters 7–9 are new in this edition, and new material has also been added to the first six chapters.

Please check www.springer.com for the latest version of *TPZS.xlsb*, as well as a list of any errata that have been discovered. If you find any errors in either the book or *TPZS.xlsb*, please describe them to me at the email address below.

My thanks to INFORMS for the publishing support I have received over the last three decades.

Monterey, CA, USA

Alan Washburn
awashburn@nps.edu

Prerequisites, Conventions, and Notation

This book assumes knowledge of calculus and probability. We will freely refer to such things as the mean and distribution of a random variable, marginal distributions, and other probabilistic concepts. Some knowledge of mathematical programming, especially linear programming, will also be needed because of the intimate relationship between TPZS games and linear programming. Appendix A gives a brief introduction to mathematical programming, and states the notational conventions in use. Appendix B reviews some aspects of convexity, which is also an important topic in game theory.

The TPZS workbook is not absolutely essential if this book is to be used merely as a reference, but readers who are using the book as a text should definitely download it (see the Preface). Many of the exercises are keyed to it. A good practice would be to immediately save it as *TPZS1.xlsb*, thus avoiding modifications to the original, which is not protected in any way. Macros must be enabled.

Chapters are divided into sections and subsections, with subsection a.b.c being subsection c of section b of chapter a. Equation (a.b-n) is the n$^{\text{th}}$ equation of section a.b. Figure a.b is the figure in section a.b, or the a.b-n notation is used when there are more than one. Theorems are numbered similarly. This section-based numbering system has the advantage of quickly determining the section where a figure can be found, but note that the existence of figure 2.2 does not imply the existence of figure 2.1.

Most sums and integrals have an index set that should be well understood from context, so limits are often not shown explicitly. Thus $\sum_i a_i$ is the sum of the numbers a_i over the index set for i. A similar convention is used with the max and min operators. If the index set for i is $\{1,2,3\}$ and if $a_i = i$, then $\sum_i a_i = 6$, $\max_i a_i = 3$, and $\min_i a_i = 1$. The symbol for set inclusion is $\in$, so it is correct to say that $2 \in \{1,2,3\}$. The number of elements in set S is $|S|$, so it is correct to say that $|\{1,4,2\}| = 3$. In the expression $X = \left\{ \mathbf{x} \middle| \mathbf{x} \geq 0 \text{ and } \sum_{i=1}^{m} x_i \leq b \right\}$, the vertical bar is read "such that," so X consists of nonnegative m-vectors such that the sum of the components does not exceed b.

Scalar real numbers or arbitrary set elements are given an italic symbol like x. Boldface, non-italic symbols denote vectors or matrices, so $\mathbf{x} = (x_i)$ is a vector with components x_i. If you see the definition $z \equiv (1,2,3)$, please send me an email complaining about my misuse of my own notation (z should be in bold type). The symbols for a vector and its components will always be identical, except that the components will usually have subscripts. The symbolic notation will not distinguish between matrices and vectors, but matrix examples will be included in brackets [ ] when written out: thus $\begin{bmatrix} -1 & 0 & 3 \\ 2 & 2 & 7.5 \end{bmatrix}$ is a 2×3 matrix.

The end of a theorem is marked with §§§. The statement $a \equiv b$ is simply a definition of a, so a proof is not necessary. The quantity being defined will always be on the left in equalities that use the $\equiv$ symbol.

Random variables, events and sets will always be given upper case, italic symbols. $P(F)$ is the probability of event F and $E(X)$ is the expected value (the mean) of random variable X.

Most chapters include a set of exercises. A partial list of solutions will be found at the end of the book.

The two players involved in a TPZS game are best thought of as humans, so there will be frequent need for a gender-neutral singular pronoun. I regularly use "he" for that purpose, this being the least of the available evils. I mean no offense.

Contents

Chapter 1
Single Person Background

Take calculated risks.
That is quite different from being rash.

George S. Patton

1.1 Utility

The game players in this book are frequently going to encounter situations where it is necessary to select one alternative out of several when some of the alternatives lead to payoffs that are random. We will invariably advocate selecting the alternative for which the expected value of the payoff is largest. The skeptical reader might protest this apparently cavalier preference for expected values in favor of other statistics such as the median. He might even wonder whether any statistic can satisfactorily summarize a payoff that is in truth random. If the alternatives were \$1,000 for certain or a 50/50 chance at \$3,000, for example, a reasonable person might argue that selecting the certain gain is the right thing to do, even though 0.5(\$0) + 0.5(\$3,000) > \$1,000, and might offer this example as evidence that life is not so simple that one can consistently make decisions by comparing expected values.

Life is that simple, provided payoffs are measured by their utilities. The demonstration of this fact was one of the achievements of von Neumann and Morgenstern (1944). They showed that if a decision maker is rational, there exists a random variable U (the decision maker's utility function) having the property that the best alternative is the one for which the expected value $E(U)$ is largest. In other words, a rational decision maker will make decisions as if he were ranking them by expected utility, even if he never actually makes the computation. It is through utility that personal preferences enter. In the monetary gamble mentioned above, the chance should be taken if the decision maker's utility for \$1,000 is less than the average of his utilities for \$0 and \$3,000. A risk averse decision maker might

A. Washburn, *Two-Person Zero-Sum Games*, International Series in Operations Research & Management Science 201, DOI 10.1007/978-1-4614-9050-0_1,

have $U(\$0) = 0$, $U(\$1{,}000) = 9$, and $U(\$3{,}000) = 10$, in which case he would opt for the \$1,000. Utility functions can be measured, but we will not discuss methods for eliciting the utility functions of actual, human decision makers in this book. Utility functions will always be assumed to be known, sometimes with an argument as to plausibility.

Distinguish between "rational" and "reasonable". A reasonable decision maker (according to this author, at least) will prefer hot coffee to cold coffee, but a rational decision maker is free to prefer the reverse if his tastes run in that direction. All that is required of rationality is that certain postulates be satisfied. One of those postulates involves three alternatives A, B and C, each of which may itself be a gamble. If a rational decision maker prefers A to B and B to C, then he will also prefer A to C—his preferences are transitive, in other words. A small number of additional rationality postulates are sufficient to conclude that the decision maker necessarily makes decisions as if he were comparing expected utilities. When the rationality postulates are criticized for not capturing the behavior of real humans, it is usually the transitivity postulate that gets the most attention.

If $U' \equiv aU + b$, then the linearity of the expectation operator implies that $E(U') = aE(U) + b$. It follows that $E(U')$ and $E(U)$ rank alternatives in the same order as long as $a > 0$, and therefore that U' is operationally the same utility function as U. Thus the origin and scale of utility can be selected for convenience—letting the worst outcome have a utility of 0 and the best a utility of 100 is common. It also follows that no quantitative judgments about utility are required when there are only two possible outcomes; expected utility becomes essentially the probability of obtaining the better outcome, or (as we will often put it) of winning. The reader who is still suspicious of the utility idea may take some comfort in this, since many interesting games have only two possible outcomes. More generally, we will often take the point of view (and we recommend it for the reader) that payoffs represent small amounts of money, since it is reasonable to suppose that utility is a linear function of money when the amounts involved are small. Utility is allowed to be negative, so losses are possible.

Ranking outcomes according to expected utility is admittedly a more comfortable idea when the same decision situation must be faced on multiple occasions, but there is no logical requirement for that to be the case. If the only two outcomes are winning and losing, and if two alternative win probabilities are 1/3 and 2/3, then one is better off with 2/3 than with 1/3, even if the gamble is only to be made once. It is possible, of course, that one will take the 2/3 chance, lose, and then discover that the 1/3 chance would have won. It is possible to regret making the correct decision. Even so, one can hardly advocate taking the gamble that is less likely to succeed.

1.2 The Maxmin Method

As a prelude to studying games, it is useful to consider situations where a lone decision maker is unable to determine the probabilities that are required to calculate the expected utility for each available alternative. Suppose, for example, that

Fig. 1.2-1 An investment problem

	Prosperity	Depression
Stocks	100	60
Bonds	80	70
Blowout	20	0

Joe Fortunato is in the fortunate position of having some money to invest. Three alternatives for investment are represented by rows in Fig. 1.2-1, using the 0–100 scale for utility. The utilities reflect the idea that bonds are less sensitive than stocks to economic fluctuations, as well as Joe's propensity to plan somehow for the future—the utilities associated with spending all of the money in a "blowout" are small in either case. Regardless of what Joe does, his utility depends as much on the state of nature (column) as it does on his decision (row). The two states of nature here are "prosperity" and "depression".

If the probabilities of the states of nature were known, it would be a simple matter to compute $E(U)$ for each of the rows and then select the best one. If Joe thought that both were equally likely, for example, he would choose the first row because 80 is the largest of 80, 75, and 10. However, suppose Joe is convinced that economic fluctuations are fundamentally unpredictable, even as probabilities, and that the decision must therefore somehow be made without having probabilities for the columns. Is there any rational way to proceed?

The world of decision theorists seems to be divided into two camps on this question, the Bayesians and the non-Bayesians. A Bayesian would argue at this point that there is no such thing as an unknown probability, that it is perfectly all right for probabilities to be personal quantities, and that a decision maker who claims he can't make the required estimates is simply exhibiting an unfortunate lack of tough-mindedness. If there is no way to tell whether one state of nature is more likely than another, then it follows that the probabilities are all equal. If there is a way, then they can eventually be quantified. In any case, the Bayesian would argue, the probabilities simply have to be known before a rational decision can be reached. A non-Bayesian, on the other hand, distinguishes between "decisions under risk", where the probabilities are known, and "decisions under uncertainty", where they are not. The two camps agree on what to do about decisions under risk. We can rephrase the question, then, as "Is there any rational way to make decisions under uncertainty?"

We can at least offer a method, reserving comment for the moment on whether it is "rational". It is the method of perfect pessimism: assume that whatever decision is made, the state of nature will turn out to be the worst thing possible. The reader who is convinced he can cause it to rain by washing his car may have some sympathy with the idea. In our investment example, Joe would choose row 2 because the utility will be at least 70 in that case. More generally, let a_{ij} represent the utility (payoff) if row i is chosen when column j is the state of nature. Then $\min_j a_{ij}$ is the worst possible payoff in row i, and $\max_i \min_j a_{ij}$ is the best of the worst. Formally, the maxmin method of making decisions is to always choose the row for which $\min_j a_{ij}$ is largest. The maxmin method is sometimes referred to as "worst-case analysis". There is also of course, the maxmax method where one assumes that the state of nature will be the best thing possible, the method of perfect optimism. The maxmax method would have Joe invest in Stocks, secure in the belief that everything will work out in his favor. The maxmax method has very few proponents.

The maxmin method can be made to look bad by forcing it to design against some extremely unlikely state of nature. To continue our Fortunato example, suppose we add the possibility that a huge meteor destroys Earth sometime between the blowout and the first possible payout from either stocks or bonds, an event that is unlikely but not impossible. The revised decision structure might look like Fig. 1.2-2. The maxmin method will now opt for the blowout. A Bayesian would ridicule this decision, and might offer the example as evidence that any reasonable decision procedure must clearly begin by probabilizing the states of nature. A non-Bayesian might reply that one has simply got to be careful about which states of nature are included when using the maxmin method, and that anyway no one is claiming that the maxmin method ought to be used on every decision problem. Such arguments have been going on for a long time and are likely to go on for a long time yet, since the Bayesians and non-Bayesians are basically reviving the debate over whether science can be fundamentally objective (non-Bayesians) or not (Bayesians).

Fig. 1.2-2 Revised investment problem

	Prosperity	Depression	Meteor
Stocks	100	60	–1000
Bonds	80	70	–1000
Blowout	20	0	–999

The maxmin method receives heavy use in statistics and other branches of mathematics where the states of nature often amount to different hypotheses or values for unknown parameters. Statements that begin "Even in the worst case, . . ." generally have an application of the method lying behind them. However, perhaps the best case for using the maxmin method is in circumstances where the "state of nature" is actually a decision made by a competitor, since a competitor at least has the motivation to make the pessimistic maxmin view of the world an accurate one. In that case a more symmetric nomenclature is desirable, so the original decision maker (the row chooser) will be called player 1 and the competitor (the column chooser) will be called player 2. The maxmin method then amounts to an assumption that player 2 is not only opposed to player 1, but prescient, having the ability to predict player 1's choice and react accordingly.

Prescience is a very strong assumption, so the question arises as to whether player 2's motivation can be preserved without in effect permitting him to examine player 1's strategy before choosing his own. Motivation and prescience are separate issues, and there is nothing illogical about supposing that player 1 is prescient instead of player 2. The comparison of these two extreme cases (player 1 or player 2 being the prescient party) is undertaken in Chap. 2. It does not make logical sense to assume that both players are prescient, but the assumption that neither player can predict the choice of the other is symmetric and often realistic. It leads to the TPZS theory developed and applied in later chapters.

Chapter 2
Maxmin Versus Minmax

I hate definitions.

Disraeli

2.1 Strategies

In this chapter we will consider games where one player or the other is compelled to make the first move, with the other player having the privilege of examining it before making his own choice. A game is usually represented as a rectangular matrix of payoffs (utilities) to player 1. The rows and columns will be referred to as "strategies". In any play of the game, the payoff is at the intersection of the row chosen by player 1 and the column chosen by player 2. Player 1 makes his choice in the hope of making the payoff as large as possible, so he will be referred to as the maximizer. Since the game is zero-sum, player 2 has the opposite motivation, and will therefore be referred to as the minimizer. There is no need to develop an explicit notation for player 2's payoff, but there is a need to remember the convention that "payoff" invariably means "payoff to player 1". The matrix format is the game's "normal" form.

The matrix format is more general than it first appears, since each row and column may actually be a complicated plan for play of an extended game. In general, the term "strategy" in game theory means a rule for action so complete and detailed that a player need not actually be present once his strategy is known, since every eventuality is provided for in the strategy. A Chess playing computer program is a good example of what is meant by the term. This use conflicts somewhat with the common English meaning of "strategy" as being something like "a grand albeit somewhat vague guiding principle". As we use the term, a strategy is not at all vague, but rather dictates exactly what action should be taken in all possible circumstances that can arise during the course of play. The game

A. Washburn, *Two-Person Zero-Sum Games*, International Series in Operations Research & Management Science 201, DOI 10.1007/978-1-4614-9050-0_2,

theoretic adoption of the word “strategy” for such a detailed plan of action was perhaps an early mistake, but by now we are stuck with it. We will not exploit the idea that strategies can be symbols for complicated plans until later chapters, but the reader should nonetheless be aware that the matrix format is actually quite general.

2.2 Maxmin < Minmax

Figure 2.2 shows a typical 3×3 matrix game, together with some marginal computations. For each row, the smallest payoff in the row is written in the right margin. Each of those numbers represents a payoff that player 1 can guarantee even if he has to move first; i.e., even if player 2 is in the favorable position of determining the column in full knowledge of the row. The largest of those numbers is shown in bold type; it represents a “floor” on the payoff that player 1 can guarantee by choosing row 2. This is simply the worst-case analysis of Chap. 1. In general, the floor is $\max_i \min_j a_{ij}$, where a_{ij} is the payoff in row i and column j.

There are also parallel computations that can be made under the assumption that it is player 2 who must move first, rather than player 1. The largest payoff in each column is written in the bottom margin, and the “ceiling” that player 2 can guarantee is the smallest of those numbers, also shown in bold type. Thus player 1 can guarantee a floor of 2 and player 2 can guarantee a ceiling of 4 in the game shown in Fig. 2.2. If player 2 chooses his maxmin column and player 1 chooses his minmax row, the payoff will be neither the floor nor the ceiling, but rather some intermediate value (3, to be precise). Nonetheless, the floor necessarily lies below the ceiling; i.e., it doesn’t pay to move first. This conclusion is formalized in Theorem 2.2, which refers to strategies as x and y because it applies to situations more general than matrix games, where rows and columns are generally called i and j.

Theorem 2.2 Let $A(x,y)$ be a real function defined for $x \in X$ and $y \in Y$, where X and Y are arbitrary sets, and suppose that $v_1 \equiv \max_x \min_y A(x,y)$ and $v_2 \equiv \min_y \max_x A(x,y)$ each exist. Then $v_1 \le v_2$.

Proof Let x^* and y^* be player 1’s maxmin and player 2’s minmax choices, respectively. Then $v_1 \le A(x^*, y)$ for all y in Y, and therefore $v_1 \le A(x^*, y^*)$. Similarly $A(x, y^*) \le v_2$ for all x in X, and therefore $A(x^*, y^*) \le v_2$. Since $A(x^*, y^*)$ lies between v_1 and v_2, necessarily $v_1 \le v_2$. §§§

$$\begin{array}{c}\begin{bmatrix} 4 & 1 & 6 \\ 3 & 2 & 7 \\ 4 & 8 & -1 \end{bmatrix} \begin{array}{c} 1 \\ \mathbf{2} \\ -1 \end{array} \\ \begin{array}{ccc} \mathbf{4} & \;\;8 & \;\;7 \end{array}\end{array}$$

Fig. 2.2 Maxmin and minmax computations for a 3×3 game

There are circumstances where the maxmin and minmax strategies x^* and y^* do not exist, but they always exist when X and Y are finite, as in a matrix game. As long as they exist, v_1 and v_2 are well defined and $v_1 \le v_2$.

2.3 Action and Reaction

Player 1's floor v_1 is $\max_x B(x)$, where $B(x) \equiv A(x, y^*(x))$, where $y^*(x)$ is player 2's minimizing response to x. This is just a different way of stating that $v_1 \equiv \max_x \min_y A(x,y)$, but it emphasizes that player 2's strategy choice (when calculating v_1) must be thought of as a function of x, rather than a constant. This is not true in single person decision making, where the state of nature y is in essence determined by an agent indifferent to player 1's strategy choice. The fact that player 2's choice depends on x in competitive situations results in what might be called an extrapolation error when it is not taken into account. An extrapolation error occurs when player 1 sets x at x_1, observes that player 2's reaction is y_1, changes his strategy to x_2, and expects the payoff to be $A(x_2,y_1)$. This expectation may lead to disappointment when player 2 changes his strategy to be effective against x_2.

Extrapolation errors are easy to describe in theory, but difficult to avoid in practice. Avoidance is difficult because the full set of strategies available to player 2 may not be known to player 1. "As von Moltke remarked to his aides, the enemy seemed to have three alternatives open to him and usually chose the fourth" (van Creveld 1985). A realistic assessment of enemy alternatives is particularly difficult when new technology is introduced. The machine gun was forecast to be effective far beyond what actually turned out to be the case, partially because trench warfare was not anticipated as a reaction. This tendency toward optimism in forecasting the effects of innovation is well known in the military. The dictum "design to the enemy's capabilities, rather than his intentions" would prevent extrapolation errors if followed rigorously, but the dictum is not easily followed, hence the skepticism toward forecasts that characterizes military planners such as von Moltke.

Some of the most important structural changes in warfare have actually been improvements to command and control systems that have had the effect of permitting the improving party to move last. The employment of radar in World War II in the Battle of Britain is a prime example, since it basically permitted the British allocation of fighters to be made after the German allocation of bombers. The party moving last is frequently in an advantageous position in spite of being relatively poor in resources, and can often substantially mitigate the effects of hardware changes that would otherwise be devastating.

2.4 Defense of a Convoy

Suppose that a convoy of ships is defended by a group of destroyers whose responsibility is to defend the convoy from submarine attack. The method of defense is to construct a circular barrier around the convoy in the hope of detecting and attacking any submarine that attempts to penetrate it. The destroyers must determine a radius for that circle. If the circle is too large, the destroyers will be so widely spaced that the barrier can be easily penetrated. If the circle is too small, submarines can attack the convoy without ever penetrating the barrier. We can model the situation as a game where the destroyers' strategy is the radius of the barrier (so the destroyers can choose any nonnegative number r), and where the submarine's strategy is to decide whether to attack from outside (strategy 1) or attempt penetration (strategy 2). Assume that the submarine moves last, since it is concealed as it approaches the convoy and can presumably observe the disposition of the destroyers before attacking. If we let the payoff be the total expected damage (tons sunk, say) in a submarine attack on the convoy, we are dealing with a payoff "matrix" that has two rows and a continuum of columns. More precisely the set X of Theorem 2.2 is $\{1, 2\}$ and the set Y consists of all nonnegative real numbers.

Figure 2.4 shows the relevant technological information. The decreasing curve $D_1(r)$ shows the expected total damage if a live submarine attacks from range r. The increasing curve $D_2(r)$ shows the probability of penetrating a barrier of radius r multiplied by the damage a submarine can cause once it has done so. The increasing curve corresponds to submarine strategy 2 and the decreasing curve to strategy 1. It is now a simple matter to conclude that the minmax payoff v_2 is as illustrated in the figure. This is the smallest expected damage that the destroyers can guarantee. The minmax radius r^* is also illustrated.

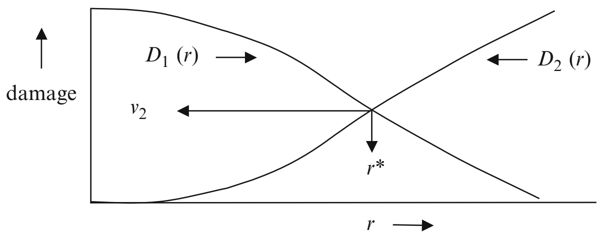


Fig. 2.4 Expected damage for two submarine tactics

The radius r^* can be described as "the radius for which submarines can't decide whether to penetrate or not". This observation can be converted into an observational rule for action that could be useful in a protracted conflict. In most cases, the curves shown in Fig. 2.4 will be either unknown or at best rough approximations to reality. However, the destroyers might adopt an observational rule of the form "increase the radius if most submarines stay outside or decrease it if most of them try to penetrate". This observational rule will have the effect of eventually homing in on the right radius. Game theory solutions often have the equalization property of making the opponent's alternatives equally attractive; the observational rule enforces the property, rather than any particular radius.

It is arguable that the attack of a convoy by a submarine is not a TPZS game at all. A submarine that attacks from outside the barrier stands a good chance of surviving the encounter, but a submarine that fails a barrier penetration attempt is likely to be attacked and sunk. Submarines might show a strong tendency to attack from outside of r^* on account of a concern for their own safety. The destroyers may have the single-minded goal of protecting the convoy, having no concern about sinking submarines *per se*, but the goal of the submarines is not necessarily the opposite of that. In this case, the analysis carried out above is simply pessimistic from the point of view of the destroyers, since the worst-case assumption is that the other player is completely opposed. Choosing r^* guarantees v_2, regardless of submarine tactics or motivation. An expanded point of view here is that the submarine's intrinsic value is due to the fact that it may sink ships in the future, as well as in the present, and that both sides should therefore be concerned about the survival of the submarine as well as any damage done to shipping in the present engagement. An analysis along these lines will be introduced in exercise 13 of Chap. 4. Of course, if the attacker is actually a suicidal terrorist or a single-use missile, then survival of the attacker is beside the point and an analysis like the one above is appropriate.

2.5 Methodology for Large Maxmin Problems

Although we have used "maxmin" in titling this section, the title could as well have been "minmax". A problem of one type can be converted into the other by simply changing the sign of the objective function and the roles of the two players.

Small maxmin problems can be solved by constructing the complete payoff matrix, finding the smallest payoff in each row, and finally finding the largest of all the small payoffs. However, some maxmin problems have so many rows and columns that even enumerating them is unattractive, much less doing any subsequent calculations. For example, player 1 might be able to choose 10 of 50 targets to attack. The number of combinations of 50 things taken 10 at a time is approximately ten billion, and we haven't yet considered strategies for player 2. No matter what the rules of the game are, explicitly writing out the payoff matrix of that game is a task that would be undertaken with great reluctance. Some games may even have an infinite number of strategies, as in Sect. 2.4 above. It does not necessarily follow that the maxmin solution cannot be computed. This section describes some potentially useful computational methods.

2.5.1 The Theory of Maxmin

Consider the problem of finding $v_1 \equiv \max_{x \in X} \min_{y \in Y} A(x,y)$, where X and Y are arbitrary sets. One might proceed by first finding $y^*(x)$, the best response by player 2 to an arbitrary choice x by player 1, then defining the function $B(x) \equiv A(x, y^*(x))$, and finally finding $v_1 = \max_x B(x)$. This procedure always works if X and Y are finite. One might hope to employ it even if X and Y are (say) intervals, equating a derivative to 0 to find $y^*(x)$. Unfortunately, the procedure is complicated by a tendency for $B(x)$ to not be differentiable at player 1's maxmin choice, even if $A(x,y)$ is differentiable everywhere. Similar comments hold, of course, for finding $v_2 \equiv \min_{y \in Y} \max_{x \in X} A(x,y)$. Figure 2.4 shows the difficulty:

letting $x^*(r)$ be the submarine's best reaction to the destroyer's choice of r, the function $C(r) \equiv A(x^*(r), r)$ is the upper envelope of the two curves. Note that the upper envelope is differentiable at every point except at the optimal radius r^*. Therefore r^* could not have been found by equating a derivative to 0. The situation is to some extent saved by the fact that $C(r)$ at least has directional derivatives, even at r^*. Danskin (1967) develops a computational theory that exploits this fact in a generalized context. Consider employing that theory if $A(x,y)$ is differentiable in at least one of its arguments.

2.5.2 Column Generation

Column generation is a useful algorithm in maxmin problems where Y is large or even infinite, but where most strategies in that large set are useless to player 2. The idea is to begin with a small set S of strategies for player 2, and then gradually expand S until it includes all of the strategies that player 2 actually needs. In the flowchart shown in Fig. 2.5, S is initialized to consist of a single arbitrary strategy y_0, and $G(S)$ refers to the maxmin problem where player 2 is restricted to strategies in S.

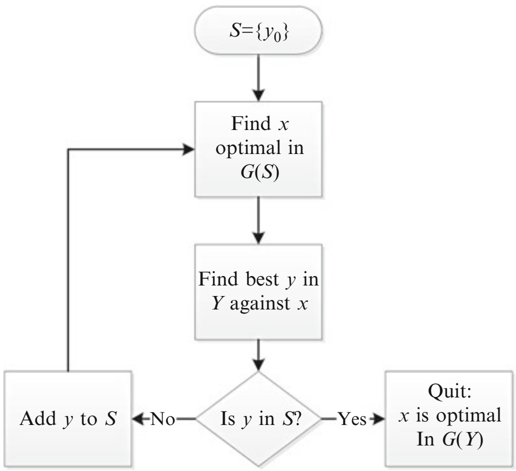


Fig. 2.5 Column generation flowchart

In the step of finding the best x, player 1's whole set X is searched. In the step of finding the best y, if multiple y's are tied for being optimal against x, then termination follows if *any* of them are in S. The basic idea is that, if player 2's best response to x is something that player 1 has already allowed for, then further expansion of S is not necessary. As long as Y is finite, termination is guaranteed because S will eventually grow to Y.

Maxmin problem $G(S)$ has to be solved for the optimal x every time a strategy is added to S, so it is important that $G(S)$ be easy to solve when S is small; after all, the whole procedure could be avoided by just solving $G(Y)$ in one step if Y were not so large. The whole set Y must be searched in finding the best y against x, but finding the best reaction to a specific x is an easier problem that solving a maxmin problem. As long as player 2's optimal strategy is a function mapping X to only a small subset of Y, the column generation procedure will find it efficiently.

For example, consider a "Hostile Grocer" maxmin problem where player 1's (the grocer's) choice x determines $c(x,i)$, the nonnegative price of every object i in the grocery store. The grocer can choose a pricing scheme from some limited set X. There are certain subsets of objects that are satisfactory for player 2. The exact definition of "satisfactory" is unimportant except that a satisfactory subset must remain satisfactory if other objects are added to it. Player 2's goal is to purchase a satisfactory subset as cheaply as possible, and player 1's goal is to maximize the cost of satisfaction. Let $\mathbf{y}$ be the vector describing player 2's spending on the various objects. Now, Y is an infinite set because all nonnegative vectors are permitted, but player 2 knows the object prices because x is known when he chooses $\mathbf{y}$. Once x is known, player 2 has a conventional shopping problem of buying the cheapest satisfactory set of objects. Indeed, the set S can consist of minimal satisfactory subsets, thus assuring convergence because there are only finitely many such subsets. With luck, only a few of those subsets will need to be generated before the algorithm stops. Exercise 2 is an example where the grocer has his choice of four pricing schemes.

The set X does not have to be finite for column generation to be useful. An interesting instance of the Hostile Grocer problem has X being a set of vectors that is limited only by an overall budget level b, so $X = \{\mathbf{x}|\mathbf{x} \geq 0 \text{ and } \sum_{i=1}^{m} x_i \leq b\}$, with the price of object i to player 2 being x_i. Player 2 must match player 1's spending to acquire an object, so the two players are essentially conducting a bidding contest for each object. Player 2 still desires to acquire a satisfactory subset of objects as cheaply as possible. Given a set S of such satisfactory subsets in the column generation algorithm, player 1's solution of $G(S)$ can be accomplished with the following linear program:

$$\begin{aligned}
&\max\nolimits_{v,\mathbf{x}\geq 0} v \\
&\text{subject to} \sum_{i=1}^{m} x_i \leq b \\
&\sum_{i\in s} x_i - v \geq 0; \quad s\in S
\end{aligned}$$

The last group of constraints insists that the cost to player 2 of acquiring each satisfactory subset must exceed v, and player 1's object is to maximize v. This linear program has $|S| + 1$ constraints and $m + 1$ variables. Washburn (2013) applies this to a competitive spending problem involving the United States Electoral College, with a satisfactory subset of the states being one that has a majority of the electoral votes. There are millions of satisfactory subsets, but column generation stops when

|S| is only 19. Regardless of how player 1 spends his budget, player 2's satisfactory subset will always be one of those 19 subsets of states.

Column generation is not always a good idea. Consider a square maxmin problem where the payoff is 1 except on the diagonal, where it is zero. The maxmin value is 0 because player 2, having the privilege of choosing after player 1, can imitate player 1's choice. However, as long as S does not include all columns, player 1 can always find a row where the payoff is 1 for all of player 2's choices in S. The final set S must therefore consist of every column, so we might as well initialize S with all columns and avoid the intermediate steps. The trouble here is that every one of player 2's strategies is useful to him. Column generation will be a useful computational technique only when most of player 2's strategies are useless.

2.5.3 The Dual Trick

Suppose that $\min_{y \in Y} A(x,y)$ can be formulated as a linear program for every x. Every minimizing linear program has a dual linear program that is maximizing, and which has the same value for its optimized objective function. Letting Z be the set of feasible dual variables and $B(x,z)$ be the dual objective function, we therefore have $\max_{x \in X} \min_{y \in Y} A(x,y) = \max_{x \in X, z \in Z} B(x,z)$. Note that the right-hand-side of this equation is written with only a single max operator, and is therefore a mathematical programming problem. This is because maximizations commute—the order in which multiple maximizations are carried out does not matter, so it suffices to indicate a single maximization over an expanded set (in contrast to max and min, where it matters very much which operation is carried out first). We refer to the conversion of a maxmin problem to a max problem by this method as the "dual trick". There are several examples of its employment in Chap. 7. Essentially the same trick can be used to convert a minmax problem into a min problem when the inner maximization problem is a linear program.

The dual trick is worth being aware of. The observation that the inner optimization (be it max or min) can be formulated as a linear program is nearly always significant computationally.

2.6 Exercises

1. Exercise 1 in Chap. 3 lists nine payoff matrices, the first of which has three rows and four columns. What are the maxmin (v_1) and minmax (v_2) values for that matrix?

2. Consider the following Hostile Grocer problem of the type considered in Sect. 2.5.2. In the matrix below, rows correspond to four different pricing schemes for the five objects. One of those rows must be chosen by player 1.

Note that the matrix shown is not a payoff matrix because the columns do not correspond to strategies for player 2—the payoff matrix for this game would have four rows and lots of columns, depending on the definition of satisfaction, but there is no need to construct it. The true values of the objects are $(3, 5, 2, 2, 2)$, and a subset is satisfactory to player 2 if its total true value is at least 6.

$$\begin{bmatrix} 1 & 2 & 3 & 4 & 5 \\ 6 & 7 & 2 & 1 & 1 \\ 4 & 4 & 4 & 4 & 4 \\ 5 & 4 & 3 & 2 & 1 \end{bmatrix}$$

Start S with the single subset $\{1, 2\}$, which is satisfactory because its total value is 8. What subsets are in S when the column generation algorithm terminates?

Chapter 3
Matrix Games

the game is afoot…

Shakespeare

3.1 Holmes Versus Moriarty

In the spirit of consulting the masters, we open this chapter with an example taken from von Neumann and Morgenstern (1944), who in turn got it from Sir Arthur Conan Doyle's story *The Final Solution*:

> Sherlock Holmes desires to proceed from London to Dover and hence to the Continent in order to escape from Professor Moriarty who pursues him. Having boarded the train he observes, as the train pulls out, the appearance of Professor Moriarty on the platform. Sherlock Holmes takes it for granted—and in this he is assumed to be fully justified—that his adversary, who has seen him, might secure a special train and overtake him. Sherlock Holmes is faced with the alternative of going to Dover or of leaving the train at Canterbury, the only intermediate station. His adversary—whose intelligence is assumed to be fully adequate to visualize these possibilities—has the same choice. Both opponents must choose the place of their detrainment in ignorance of the other's corresponding decision. If, as a result of these measures, they should find themselves, *in fine*, on the same platform, Sherlock Holmes may with certainty expect to be killed by Moriarty. If Sherlock Holmes reaches Dover unharmed he can make good his escape.

Holmes' survival probability is 0 if he and Moriarty choose the same railway station. Since Dover is on the coast, it is reasonable to assume that Holmes will surely escape if he chooses Dover while Moriarty chooses Canterbury. However, Holmes is not certain to escape if he chooses Canterbury while Moriarty chooses Dover, the difficulty being that Moriarty still has a chance to track him down before he leaves Britain. Assuming that Holmes' escape probability is 0.5 in this instance, the whole situation can be represented as in Fig. 3.1:

A. Washburn, *Two-Person Zero-Sum Games*, International Series in Operations Research & Management Science 201, DOI 10.1007/978-1-4614-9050-0_3,

Fig. 3.1 Holmes versus Moriarty

		Moriarty: Canterbury	Moriarty: Dover
Holmes:	Canterbury	0	0.5
	Dover	1	0

Letting a_{ij} be the escape probability in row i and column j, it is immediate that $\max_i \min_j a_{ij} = 0$; that is, Holmes' escape probability would be 0 if he had to move first. But plainly Holmes would be daft to make his choice and then telegraph it to Moriarty; his security to a large extent depends on keeping his choice secret. But similarly he cannot expect to know Moriarty's choice before making his own. A new possibility has entered—neither player has to move first. To play the game in the abstract, one would have each player secretly and independently write his choice on a piece of paper, and then compare the two to determine Holmes' fate. In Chap. 2, one of the players was always in the fortunate position of being able to examine the other player's choice before making his own, but that is not the case here.

What should Holmes do? The answer is simple if he happens to have a die in his pocket—roll the die and go to Dover if the result is "1" or "2", else go to Canterbury. In other words, go to Canterbury with probability 2/3. If Holmes does this, he will escape with probability 1/3 even if Moriarty knows about Holmes' use of a die. If Moriarty chooses Canterbury, Holmes will escape with probability (1/3)(1). If Moriarty chooses Dover, Holmes will escape with probability (2/3)(1/2). Either way the result is 1/3, so Moriarty is helpless to prevent Holmes escaping with that probability. Furthermore, although it is perhaps regrettable that Holmes has less than an even chance of escaping, there is no point in his seeking some other method for choosing where to detrain. To see this, suppose that Moriarty were to choose Canterbury with probability 1/3 by rolling a die. Even if Holmes knew about Moriarty's die rolling, he could not escape with probability more than 1/3 unless he could actually predict how the die would roll. This, then, is the basis for our recommendation: Holmes can achieve 1/3 by rolling a die, and should not hope for more than 1/3 because Moriarty can also roll a die. We say rolling a die as advised is "optimal" and that 1/3 is "the value of the game".

The Holmes versus Moriarty example has been deliberately chosen to make the author's job of convincing a skeptical reader as easy as possible. There are only two outcomes, so questions of utility don't enter except that Holmes and Moriarty are opposed. The payoff is a probability even if the players don't randomize, so the reader can hardly object that randomization should not be considered on the grounds that the game is only to be played once. If one is going to gamble (and Holmes has no choice but to do so), surely the gamble with the greatest probability of success is to be preferred.

Possibly the reader still has some reservations about advising Holmes to abandon control of his destiny to a die. Sir Arthur did not have Holmes roll a die to decide what to do, and would probably not have done so even if the above analysis had been pointed out to him. There is something unromantic about the idea. It is much more interesting to conceive of the problem as being one of utilizing subtle clues to anticipate the opponent's behavior, and the observation that all such attempts are doomed to failure if the opponent's behavior is actually governed by a die is disappointing, even if true. The reader who wishes some sympathy for this position may enjoy reading Hofstadter's (1982) account of the disappointment he felt when a clue-seeking computer program that he wrote could not beat a much shorter program employing a random number generator that simply made choices at random.

We need to come to an agreement about this, you the reader and I the author, since the rest of this book is about calculating optimal strategies and game values. We can hardly continue by me saying such things as "the optimal strategy for Eisenhower was to land in the Pas de Calais with probability 0.73 and in Normandy with probability 0.27", while you say to yourself that real decisions just don't get made that way. If you are reluctant about this business of making decisions by rolling dice, I suggest the following compromise. I will make no further attempts to convince you that making decisions formally at random is actually a reasonable thing to do. "Let him believe in the subtle clue theory of decision making if he likes", I will tell myself, "since in practice relying on subtle clues could very well result in random behavior anyway". In return, you must agree to take at least the game value seriously as a way of summarizing a game by a number. To continue the example, suppose that Holmes has in his pocket a pill that will temporarily change his appearance to such an extent that Moriarty could not recognize him, but which will, with probability 0.5, have death as a side effect. In other words, Holmes can either escape with probability 0.5 by taking the pill or else play the game. Since $1/2 > 1/3$, Holmes should take the pill. If you resist even this conclusion (perhaps saying to yourself that Holmes should go to Dover to enjoy the possibility of escaping for certain), then the rest of this book is going to be rather meaningless.

If you are not put off by the idea of Holmes rolling a die, then so much the better. There are some precedents. Military officers can usually quote a small number of instances where randomization is formally built into tactics, as well as a larger number of instances where disasters could have been prevented by doing so. Some organizations randomize paydays to make payday thievery of employees unattractive. Pita et al. (2008) describe how random numbers are used to schedule police patrols at LAX airport. Randomization is simply the formalization of the intuitive idea that tactics should be unpredictable. It is the key to resolving the "…if he thinks that I think that he thinks that…" impasse—you can't outwit somebody who refuses to think.

3.2 von Neumann's Theorem

In general, let a_{ij} be the payoff to player 1 (his utility) if he chooses row i (strategy i) while player 2, who is opposed, chooses column j. Let x_i be the probability that player 1 chooses row i, and $\mathbf{x}$ the vector of such probabilities. $\mathbf{x}$ will be called player 1's "mixed strategy", or sometimes, if there is no danger of confusion with the rows themselves, simply his "strategy". The average payoff when player 1 uses $\mathbf{x}$ against column j is $\sum_i x_i a_{ij}$. Recall that player 1's preferences are based entirely on this average utility. Similarly let **y** be a vector of probabilities for player 2, in which case $\sum_j a_{ij} y_j$ is the average payoff if player 2 uses **y** against player 1's row i.

Definition Given a matrix (a_{ij}), the corresponding TPZS game is solvable if there exist probability distributions **x*** over the rows, **y*** over the columns, and a number v such that

1. $\sum_{i=1}^{\infty} a_{ij} x_i^* \geq v$ for $j = 1, \ldots,$ and
2. $\sum_{j=1}^{\infty} a_{ij} y_j^* \leq v$ for $i = 1, \ldots$

In that case, **x*** and **y*** are said to be optimal (mixed) strategies, and v is said to be the value of the game.

The first condition in the definition states that **x*** guarantees at least v against any of player 2's (pure) strategies, so it would be hard to defend any number smaller than v as a value for the game. The second condition states that **y*** guarantees at most v against any of player 1's pure strategies, so it would also be hard to defend any number larger than v as a value of the game. Either player can guarantee v from his own perspective, regardless of what the other player does, even if the other player knows the mixed strategy being used. These are powerful reasons for thinking of a play of the game between two sentient, opposed players as being equivalent to a single outcome with utility v to player 1.

The definition does not require that **x*** be player 1's best reaction to an arbitrary strategy of player 2. If player 1 somehow knows that player 2's strategy choice will not be optimal, then player 1 may be well advised to make a non-optimal choice himself to exploit player 2's mistake. If Holmes knew that Moriarty planned to detrain at Dover, then his best choice would be Canterbury, rather than the randomization implied by **x***. Player 1 cannot lose (relative to v) by playing **x***, but he may not exploit mistakes on player 2's part to the fullest extent possible. The argument for using an "optimal" strategy thus rests on a fundamentally pessimistic viewpoint about how the opponent's choice will be made. As in Chap. 2, the opponent is assumed to be opposed and sentient. The main difference here is that the opponent is no longer assumed to be prescient.

In the Holmes versus Moriarty game, **x*** = (1/3, 2/3), **y*** = (2/3, 1/3), and $v = 1/3$. But do solutions exist for games that are larger than 2×2? John von Neumann (1928) proved the following remarkable theorem:

Theorem 3.2-1 All finite matrix games have a solution. §§§

It will later be shown that the one method of computing **x***, **y***, and v is by linear programming (LP), and that a proof of Theorem 3.2-1 can be based on the LP Duality Theorem. That theorem was not available in 1928, and historically the

Fig. 3.2-1 John von Neumann in the 1940s. He started it all

proof of the Duality Theorem was based on Theorem 3.2-1, rather than vice versa. Nonetheless, the reader who has studied LP has the tools available for an easy proof of Theorem 3.2-1. We will not reproduce von Neumann's proof, which was non-constructive and lengthy.

Theorem 3.2-1 establishes the existence of a function *val*() such that *val*(**a**) is the value of an arbitrary finite payoff matrix **a**. It is not true in general that *val*(**a** + **b**) = *val*(**a**) + *val*(**b**), but at least the *val*() function behaves predictably under changes of scale and location. The next theorem establishes this, as well as the fact that *val*() is a continuous function.

Theorem 3.2-2 If **a** and **b** are two matrices with the same dimensions, and if c and d are two scalars with $c \geq 0$, then, if $c\mathbf{a} + d$ is the matrix $(ca_{ij} + d)$, and if $\Delta = max_{ij}|a_{ij} - b_{ij}|$, then

1. $val(c\mathbf{a} + d) = c\ val(\mathbf{a}) + d$, and
2. $|val(\mathbf{a}) - val(\mathbf{b})| \leq \Delta$.

Proof The proof of the first conclusion is left to the reader in exercise 3. To prove the second, let **x*** be optimal for player 1 in game **a** and let **y*** be optimal for player 2 in game **b**. Since $a_{ij} \leq b_{ij} + \Delta$ and since **y*** is optimal in **b**, we must have for all i

$$\sum_j a_{ij} y_j^* \leq \sum_j (b_{ij} + \Delta) y_j^* = \sum_j b_{ij} y_j^* + \Delta \leq val(\mathbf{b}) + \Delta, \tag{3.2-1}$$

where the last inequality is because **y*** is optimal in **b**. Therefore, multiplying (3.2-1) through by x_i* and summing on i,

$$\sum_i x_i^* \left(\sum_i a_{ij} y_j^* \right) \leq val(\mathbf{b}) + \Delta \tag{3.2-2}$$

On the other hand, $\sum_i x_i^* a_{ij} \geq val(\mathbf{a})$ for all j because **x*** is optimal for **a**, so

$$val(\mathbf{a}) \leq \sum_j \left(\sum_i x_i^* a_{ij} \right) y_j^* \tag{3.2-3}$$

But the sums in (3.2-2) and (3.2-3) are equal, so $val(\mathbf{a}) \le val(\mathbf{b}) + \Delta$. By symmetry, it is also true that $val(\mathbf{b}) \le val(\mathbf{a}) + \Delta$, which completes the proof. §§§

Neither Theorem 3.2-1 nor Theorem 3.2-2 can be generalized to infinite games without qualification. A simple counterexample is the game where each player guesses a number and the winner is the one who has guessed the larger. The payoff is perfectly well defined in that game, but there is no $\mathbf{x}^*$, $\mathbf{y}^*$, v that satisfies the inequalities demanded of a solution.

While calculating $\mathbf{x}^*$, $\mathbf{y}^*$, and v is not always easy, it is usually easy to check whether any proposed solution is actually optimal, finding bounds on the game value in the process. Figure 3.2-2 shows how this can be done for a particular 3×4 game. A mixed strategy $\mathbf{x}$ is shown as the left column, and the associated quantities $\sum_i x_i a_{ij}$ are shown as the bottom row (for example $\sum_i x_i a_{i2} = 0.14$). The smallest of the latter quantities is shown in bold type; it represents a floor that is guaranteed by $\mathbf{x}$ regardless of player 2's strategy. Similarly $\mathbf{y}$ is shown in the top row, $\sum_j a_{ij} y_j$ in the right column, and the largest of these latter quantities is also shown in bold type, establishing a ceiling that is guaranteed by $\mathbf{y}$. In general the floor will lie below the ceiling, but if the bold numbers are all equal the game has in fact been solved. Evidently the value of the game in Fig. 3.2-2 is 0.12. If the bold numbers were not equal the diagram would instead establish them as bounds on the value.

Fig. 3.2-2 Testing optimality

$$
\begin{array}{c c c}
 & \begin{matrix} .4 & 0 & 0 & .6 \end{matrix} & \\
\begin{matrix} .4 \\ 0 \\ .6 \end{matrix} & \begin{bmatrix} .3 & .2 & .1 & 0 \\ .1 & .5 & .5 & .1 \\ 0 & .1 & .2 & .2 \end{bmatrix} & \begin{matrix} \mathbf{.12} \\ .10 \\ \mathbf{.12} \end{matrix} \\
 & \begin{matrix} \mathbf{.12} & .14 & .16 & \mathbf{.12} \end{matrix} &
\end{array}
$$

The matrix in Fig. 3.2-2 could have the following interpretation. A single weapon is available for use against a submarine. The submarine's depth is unknown but under the control of the submarine, which realizes that it is about to be attacked. The submarine has four feasible depths, and the weapon has "shallow", "medium", and "deep" settings. The probability of damaging the submarine is known for all 12 combinations, and all this information is available to both sides. The submarine will be damaged in 0.12 of the engagements if both sides play optimally, since either side can guarantee this number against all possible strategies of the other. Note that the "medium" weapon setting should never be used, in spite of the fact that row 2 has some large numbers in it. This example shows that the meaning of words such as "unknown depth" can be very important. If "unknown" is interpreted to mean that the submarine is equally likely to be at any of the four feasible depths, then the "medium" setting should be used exclusively because it has the largest row sum. If "unknown" is interpreted in the sense of game theory, the "medium" setting should never be used because it is ineffective at the depths where the submarine is likely to be.

In studying games we always assume that both sides know the rules, or equivalently that both sides know the payoff matrix. This is not always true in the real world. It occasionally happens that a Checkers game is played by someone who does not understand that pieces advanced to the last rank become kings, and radioelectronic combat in World War II was characterized by innovations the existence of which were temporarily unknown to the other side. Our definitions and calculations will not be useful in such situations, since we always assume complete knowledge of the rules. Harsani (1967) refers to games where everybody knows the rules as "games of complete information", to be distinguished from the other kind. Since all of the games studied in this book are games of complete information in Harsani's sense, subsequent use of the term will not be necessary. We can still study the effects of uncertainty even when both sides know the rules, as subsequent examples will make clear.

3.3 Games with Saddle Points

Using the terminology of Chap. 2, let i^* be player 1's maxmin row and v_1 the floor that i^* guarantees, and similarly let j^* be the minmax column and v_2 the associated ceiling. Figure 2.2 shows how these quantities can be easily calculated for any finite game, and Theorem 2.2 states that $v_1 \leq v_2$. If $v_1 = v_2$, the game is said to have a saddle point. In that case, it is not hard to prove that $\mathbf{x}^*$ can be the only probability vector for which $x_{i^*} = 1$, and that $\mathbf{y}^*$can be the only probability vector for which $y_{j^*} = 1$. In other words, randomized strategies are not needed to play games that have saddle points.

Games with saddle points are sometimes called "strictly determined". Strictly determined games are distinguished in that only ordinal utilities are required to describe them. The strategy pair (i^*, j^*) is a saddle point of $\mathbf{a}$ if and only if $a_{i^*j} \leq a_{i^*j^*} \leq a_{ij^*}$ for all (i, j), where the middle payoff is v. If $b_{ij} = g(a_{ij})$, where $g()$ is any strictly increasing function, then it is not hard to establish that (i^*, j^*) must also be a saddle point of $\mathbf{b}$, and that the value of the transformed game is $g(v)$. The fact that the saddle point is independent of such increasing transformations means that only the ordering of the payoffs is important, rather than the cardinal values.

Haywood (1954) describes a World War II game where General Kenney had to decide whether to commit the bulk of his aircraft to either the North or South side of New Britain in the Bismarck Sea, hoping to intercept and bomb a crucial Japanese convoy whose commander had to make a similar choice. Haywood's assessment of the payoff in terms of "days of bombing" is shown in Fig. 3.3-1. The asymmetry of the figure is due to the asymmetry of the weather on the two sides of New Britain. There is a saddle point where both sides choose "North"(N). Now, it could be argued that Haywood's choice of measure of effectiveness should have been something more militarily relevant. Figure 3.3-2 shows a hypothetical matrix where "days of bombing" has been converted to "expected tons of shipping sunk", an increasing transformation because the latter increases with the former. The large entry in the lower right-hand corner is due to the supposed exhaustion of the convoy's defenses after two days of bombing. However, all this subtlety is lost

Fig. 3.3-1 Days of bombing

$$\begin{array}{cc} & \text{Japan} \\ & \begin{array}{cc} \text{N} & \text{S} \end{array} \\ \text{Kenney} \begin{array}{c} \text{N} \\ \text{S} \end{array} & \begin{bmatrix} 2 & 2 \\ 1 & 3 \end{bmatrix} \end{array}$$

Fig. 3.3-2 Tons of shipping sunk

$$\begin{array}{cc} & \text{Japan} \\ & \begin{array}{cc} \text{N} & \text{S} \end{array} \\ \text{Kenney} \begin{array}{c} \text{N} \\ \text{S} \end{array} & \begin{bmatrix} 20000 & 20000 \\ 1000 & 80000 \end{bmatrix} \end{array}$$

in the present case because Fig. 3.3-2 has a saddle point in exactly the same place as Fig. 3.3-1. The transformation is strategically neutral, and the value of Fig. 3.3-2 (20,000 tons) could just as well have been obtained by directly transforming the value of Fig. 3.3-1. This would be an important observation if the increasing transformation were expensive to evaluate—a detailed computer simulation, perhaps. Either the original problem or the transformed version can be solved; whichever is more convenient.

In general, only increasing <u>linear</u> transformations are strategically neutral, as proved in Theorem 3.2-2. Even without knowing that Fig. 3.3-1 has a saddle point, it is essentially immaterial whether all of the entries are multiplied by 24, thus converting them to "hours of bombing". The reason why increasing linear transformations are always permitted is that utility functions are only unique to within such transformations in the first place. But it is only in strictly determined games where <u>arbitrary</u> increasing transformations can be employed for convenience.

If a game has a saddle point, that fact can sometimes be demonstrated without knowing all of the payoffs. Figure 3.3-3 shows a matrix game where there is a saddle point at $(i^*, j^*) = (3,1)$, regardless of how the other entries are filled in. This could be important if evaluation of entries is expensive. Suppose, for example, that evaluating all entries in a $1,000 \times 1,000$ game is not feasible, but that on intuitive grounds it is suspected that there is a saddle point at (1,1). The truth of this hypothesis can be tested by evaluating only 1999 entries (the first row and column), rather than all 1,000,000. Even if the hypothesis is false, the calculations may lead to conclusions about which entries to evaluate next.

Fig. 3.3-3 A game with a saddle point

$$\begin{bmatrix} 0 & & & \\ .5 & & & \\ 3 & 8 & 9 & 3 \end{bmatrix}$$

Why would one ever suspect that a game has a saddle point? The intuitive feeling that usually leads to this suspicion is that there is no need for secrecy. If Haywood had it right, there was no need for Kenney to keep secret his plans for concentrating his forces on the Northern route. There have been certain football teams (Notre Dame of 1928 comes to mind) that felt they were playing a game with a saddle point when it came to the run/pass decision. They were so good at running (or so poor at passing) that they would run with the ball on every opportunity, even though the opposing team knew and relied on the fact. Tic-Tac-Toe is another example.

Either player can avoid losing Tic Tac Toe, regardless of what the other player does, so there is no need for secrecy when playing it. Tic Tac Toe is an example of a class of games that includes Chess, Backgammon, and most commercial "war games", all of which have saddle points.

3.4 Solving 2-by-2 Games

Recall that the solution of Holmes versus Moriarty had the property that, when Holmes used his optimal strategy **x***, the payoff was independent of player 2's choice; that is, **x*** is "equalizing". This idea can be exploited to solve 2 × 2 games in general. Figure 3.4 shows a general 2 × 2 matrix, together with the three equations that one obtains by requiring that **x** be a probability vector for which the payoff is the same number u in both columns:

Fig. 3.4 The general 2 × 2 game

$$\begin{array}{cc} x_1 & \\ x_2 & \\ & \end{array}\begin{array}{c} \begin{bmatrix} a & b \\ c & d \end{bmatrix} \\ \begin{matrix} u & \quad u \end{matrix} \end{array} \qquad \begin{aligned} ax_1 + cx_2 &= u \\ bx_1 + dx_2 &= u \\ x_1 + x_2 &= 1 \end{aligned}$$

The three equations can be solved for the three unknowns x_1, x_2, and u. With equal logic, one could look at the problem from player 2's standpoint and solve the three equations $ay_1 + by_2 = w$, $cy_1 + dy_2 = w$, and $y_1 + y_2 = 1$ for y_1, y_2, and w. It turns out that u and w are equal, and it is tempting to conclude that the common value must be v, the value of the game that is guaranteed to exist by Theorem 3.2-1. That conclusion is correct unless the game has a saddle point. If the game has a saddle point, then one or both of **x** and **y** will have a negative component and therefore cannot serve as a probability vector. The solution obtained by assuming that both sides equalize is:

$$\begin{aligned} \text{Let} \quad D &= a + d - b - c \\ \text{Then} \quad \mathbf{x}^* &= (d - c, a - b)/D, \\ \mathbf{y}^* &= (d - b, a - c)/D \\ v &= (ad - bc)/D. \end{aligned} \tag{3.4-1}$$

It can be easily verified that the Holmes versus Moriarty results can be obtained by applying (3.4-1). For a different example consider the game with payoff matrix $\begin{bmatrix} 2 & 3 \\ 1 & 0 \end{bmatrix}$. Applying (3.4-1) results in $\mathbf{x}^* = (0.5,\ 0.5)$, $\mathbf{y}^* = (1.5,\ -0.5)$, and $v = 1.5$, but the correct solution is $\mathbf{x}^* = (1,0)$, $\mathbf{y}^* = (1,0)$, and $v = 1$. The game has a saddle point, so (3.4-1) does not apply.

The same "exhaustion of cases" technique that works with 2 × 2 games also works with larger games, but the proofs become progressively more complicated because there are more cases. Borel actually solved 3 × 3 and 4 × 4 games by this technique before von Neumann proved Theorem 3.2-1, but the proofs were so complicated that Borel speculated that Theorem 3.2-1 was false for larger games. A good account of the history of Theorem 3.2-1 and its relation to linear programming can be found in Fréchet (1953).

3.5 Sensor Selection

Suppose that one of two sensors must be acquired for use in detecting submarines. Either sensor has "shallow"(S) and "deep"(D) settings, but sensor 2 sacrifices long-range detections for robustness—its detection range is not 0 even if the submarine's depth (which is also either S or D) is guessed incorrectly. The detection ranges are shown in Fig. 3.5-1. The question is which kind of sensor is more effective.

One way of dealing with the question is to regard each matrix in Fig. 3.5-1 as the payoff matrix in a game, with the value of the game being an equivalent detection range. Letting v_A and v_B be the values of the sensor A and sensor B games, (3.4-1) can be used to calculate $v_A = 1.33$ and $v_B = 1.50$. Sensor B is better according to this criterion.

Another way of dealing with the question is to reason that the sensors will be used within a large area of size C to detect hidden submarines. If n sensors are used, the fraction of submarines detected will be the area ratio $(\pi R_1{}^2 + \ldots + \pi R_n{}^2)/C$, where R_i represents the detection range of the ith sensor, a random variable because it depends on the S/D choices of both sides. On the average, this ratio is the probability of detection $n\pi E(R^2)/C$, where $E(R^2)$ is the value of one of the games shown in Fig. 3.5-2. These are the same games as in Fig. 3.5-1, except that each entry has been squared. The values of these two games are 3.2 and 2.5, so now sensor 1 is better! Which of these two methods of answering the question is correct?

Fig. 3.5-1 Detection ranges for two sensors

$$\begin{array}{cc} & \begin{array}{cc} \text{S} & \text{D} \end{array} \\ \begin{array}{c} \text{S} \\ \text{D} \end{array} & \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix} \end{array} \qquad \begin{array}{cc} & \begin{array}{cc} \text{S} & \text{D} \end{array} \\ \begin{array}{c} \text{S} \\ \text{D} \end{array} & \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix} \end{array}$$

Sensor A Sensor B

Fig. 3.5-2 Squared detection ranges for two sensors

$$\begin{array}{cc} & \begin{array}{cc} \text{S} & \text{D} \end{array} \\ \begin{array}{c} \text{S} \\ \text{D} \end{array} & \begin{bmatrix} 4 & 0 \\ 0 & 16 \end{bmatrix} \end{array} \qquad \begin{array}{cc} & \begin{array}{cc} \text{S} & \text{D} \end{array} \\ \begin{array}{c} \text{S} \\ \text{D} \end{array} & \begin{bmatrix} 4 & 1 \\ 1 & 4 \end{bmatrix} \end{array}$$

Sensor A Sensor B

The second method is correct if the sensors are indeed to be used in a large area, whereas the first would be correct if the sensors will be used to construct a linear barrier. Intuitively, the long-range detections of the first sensor are wasted on a line, but have great effect in two dimensions. Figures 3.5-1 and 3.5-2 lead to different conclusions, in spite of the fact that the latter was obtained from the former by simply squaring every element—an increasing transformation. The games under consideration are not strictly determined, so arbitrary increasing transformations are not strategically neutral. Note that 3.2 is not the square of 1.33, and that 2.5 is not the square of 1.5. Another way of putting this is that, if player 1's problem is to cover a large area, then his utility should be the square of the detection range, rather than the detection range.

Now suppose instead that the sensors are to be used to detect tanks, rather than submarines, and that the option for the tanks is to cross either a swamp (S) or a desert (D). Each sensor has a Swamp/Desert switch on it, and Fig. 3.5-1 still shows the detection ranges. However, there is an essential difference from either of the previous situations. Sensors deployed in swamps are of course going to use the S setting, and sensors in deserts the D setting. The physical separation of swamps and deserts essentially means that the sensor side logically has the last move, regardless of the chronological order of events. This being the case, the "value" associated with each matrix in Fig. 3.5-1 should be its minmax value. The minmax value is 2 in each case, or 4 in Fig. 3.5-2, so it makes no difference which type of sensor is employed.

3.6 Graphical Solution of 2-by-n and m-by-2 Games

Consider the 2×4 game in Fig. 3.6-1. In solving this game, we will take advantage of the fact that an optimal mixed strategy does not have to be kept secret to be effective in guaranteeing v. Even though the strategy choices themselves must in general be kept secret, the distribution from which they are drawn can be public knowledge. Figure 3.6-1 graphically shows the consequences of using the strategy $\mathbf{x} = (1 - x_2, x_2)$ against each of player 2's four strategies. For example, when player 2 uses the first column against $\mathbf{x}$, the payoff is $1(1 - x_2) + 9x_2$, a linear function of x_2. This line can be drawn by simply connecting 1 on the left-hand vertical axis to 9 on the right-hand vertical axis, as in the figure. Similar lines are shown for the other three columns. The four lines can be used to predict player 2's optimal response to any $\mathbf{x}$. If $x_2 = 0.1$, for example, the best response is column 4 because the line for column 4 lies below all of the others at that point. Evidently any point on the lower envelope of the four lines (the two heavy line segments) is achievable by player 1. The best value of x_2 is x_2*, the highest point of the lower envelope, and the value of the game is the height of that point. Thus player 1's optimal strategy and the game value can be determined graphically.

Player 2's first two strategies give payoffs that are larger than v at x_2*, so neither of them should be used in player 2's optimal mixed strategy. We can therefore find an exact solution by solving a 2×2 game where player 2 uses only columns 3 and 4, the two columns whose lines determine the highest point of the lower envelope in Fig. 3.6-1. This is the game $\begin{bmatrix} 6 & 0 \\ 0 & 4 \end{bmatrix}$, and the solution is $v = 2.4$, $\mathbf{x}^* = (0.4, 0.6)$, and $\mathbf{y}^* = (0.4, 0.6)$. The marginal calculations in Fig. 3.6-1 verify

$$
\begin{array}{rc l}
 & \begin{matrix} 0 & 0 & 0.4 & 0.6 \end{matrix} & \\
0.4 = 1 - x_2 & \multirow{2}{*}{$\begin{bmatrix} 1 & 3 & 6 & 0 \\ 9 & 3 & 0 & 4 \end{bmatrix}$} & 2.4 \\
0.6 = x_2 & & 2.4 \\
 & \begin{matrix} 5.8 & 3.0 & \mathbf{2.4} & \mathbf{2.4} \end{matrix} &
\end{array}
$$

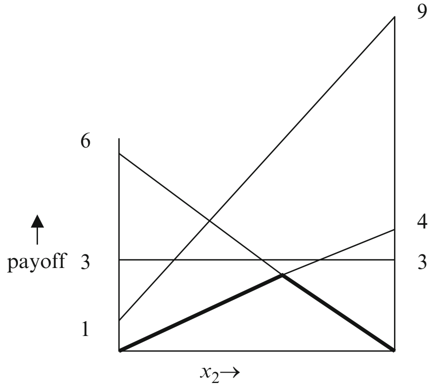


Fig. 3.6-1 Graphical solution of a 2 × 4 game

that the 2 × 4 game has in fact been solved, the optimal solution being the same as for the 2 × 2 game except that **y*** must be padded with zeros to obtain **y*** = (0, 0, 0.4, 0.6). The main function of the graph is to determine which 2 × 2 game to solve. Note that as long as player 1 uses **x***, the payoff will be v if player 2 uses (0, 0, 1 − y, y) for any y in the unit interval. Nonetheless, the unique optimal strategy for player 2 is **y***. It is also true that the payoff will be 2.4 as long as player 2 uses **y***, regardless of how player 1 chooses his strategy. It does not follow, however, that all strategies are optimal for player 1. Player 1's <u>only</u> optimal strategy is **x*** because **x*** is the only strategy that will guarantee 2.4 regardless of player 2's choice. In other instances **x*** may not be unique because the upper envelope can have a flat segment at its peak.

The procedure for solving m × 2 games is similar, except that (y_2*, v) is the lowest point on the upper envelope of the m lines corresponding to player 1's strategies. Unless the game has a saddle point, there will always be some lowest point that is at the intersection of a non-decreasing line and a non-increasing line. Player 1 can safely use only these two lines, and the value of the game is the same as the value of the resulting 2 × 2 game. Figure 3.6-2 shows a worked example.

Since all three lines in Fig. 3.6-2 go through the minimum point of the upper envelope, either row 1 or row 2 (but not row 3, the only decreasing line) can be crossed out to obtain a 2 × 2 game. If row 1 is crossed out, the solution is **x*** = (0, 2/3, 1/3), **y*** = (0.5, 0.5), and v = 2. If row 2 is crossed out, the solution is **x*** = (0.5, 0, 0.5), **y*** = (0.5, 0.5), and v = 2. Player 1's optimal strategy is not unique, but v and **y*** are. In degenerate cases (a game where the payoff is always 0, for example) both sides may have multiple optimal strategies, but even such games have a unique value. The value is unique in every finite game, even though the optimal strategies may not be.

$$\begin{array}{c} \\ 0 \\ 2/3 \\ 1/3 \\ \\ \end{array} \begin{array}{c} \begin{array}{cc} 1-y_2 & y_2 \end{array} \\ \left[\begin{array}{cc} 0 & 4 \\ 1 & 3 \\ 4 & 0 \end{array}\right] \\ \begin{array}{cc} \mathbf{2} & \mathbf{2} \end{array} \end{array} \begin{array}{c} \\ \mathbf{2} \\ \mathbf{2} \\ \mathbf{2} \\ \\ \end{array}$$

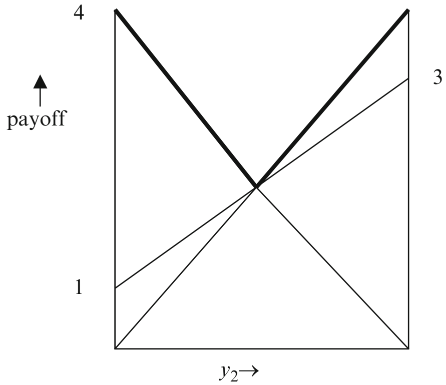


Fig. 3.6-2 Graphical solution of a 3 × 2 game

The graphical procedure may also be employed in three dimensions (Luce and Raiffa 1957), but it becomes clumsy because of the need to imagine and illustrate intersecting planes. If a game cannot be reduced to two dimensions by employing the dominance techniques discussed in the next section, it is generally best to solve it using some non-graphical method.

3.7 Dominance

Compare columns 1 and 4 in Fig. 3.6-1. Each element of column 4 is smaller than the corresponding element of column 1, so player 2 should prefer column 4 to column 1 no matter which row is chosen by player 1. Column 4 *dominates* column 1, so column 1 can effectively be crossed out—player 2 can use it with probability 0 in his optimal strategy. More generally, column j dominates column k if $a_{ij} \leq a_{ik}$ for all i. Similarly, player 1 need not use row k if it is uniformly smaller than row i; row i dominates row k if $a_{ij} \geq a_{kj}$ for all j. Roughly speaking, large columns and small rows can be safely removed from the payoff matrix.

There is no instance of row dominance in Fig. 3.6-1, but consider Fig. 3.7. In diagram a, column 4 dominates column 2, and row 2 dominates row 3. If column 2 and row 3 are crossed out, diagram b results. Notice that column 3 dominates columns 1 and 4 in diagram b, even though it did not do so in diagram a. Crossing out these two columns results in diagram c. There is no further row or column dominance in diagram c, but the game can now be solved as a 3 × 2 game. The final solution is shown in diagram d, with marginal calculations that verify optimality.

a

$$\begin{bmatrix} 1 & 4 & 0 & 2 & 9 \\ 3 & 6 & 2 & 2 & 8 \\ 1 & 1 & 2 & 0 & 3 \\ 7 & 6 & 5 & 6 & 1 \end{bmatrix}$$

b

$$\begin{bmatrix} 1 & . & 0 & 2 & 9 \\ 3 & . & 2 & 2 & 8 \\ . & . & . & . & . \\ 7 & . & 5 & 6 & 1 \end{bmatrix}$$

c

$$\begin{bmatrix} . & . & 0 & . & 9 \\ . & . & 2 & . & 8 \\ . & . & . & . & . \\ . & . & 5 & . & 1 \end{bmatrix}$$

d

$$\begin{array}{ccc}
 & \begin{matrix} 0 & 0 & .7 & 0 & .3 \end{matrix} & \\
\begin{matrix} 0 \\ .4 \\ 0 \\ .6 \end{matrix} & \begin{bmatrix} 1 & 4 & 0 & 2 & 9 \\ 3 & 6 & 2 & 2 & 8 \\ 1 & 1 & 2 & 0 & 3 \\ 7 & 6 & 5 & 6 & 1 \end{bmatrix} & \begin{matrix} 2.7 \\ \mathbf{3.8} \\ 2.3 \\ \mathbf{3.8} \end{matrix} \\
 & \begin{matrix} 5.4 & 6.0 & \mathbf{3.8} & 4.4 & \mathbf{3.8} \end{matrix} &
\end{array}$$

Fig. 3.7 Dominance in a 4×5 game

The principal value of the dominance idea is to reduce the effective size of the game. Actually, if any linear combination of rows dominates row k, where the weights in the linear combination form a probability distribution, then row k can be crossed out. Columns can be treated similarly. For example, column 2 in Fig. 3.6-1 is not dominated by any other column, but it is dominated by the average of the third column and the fourth.

3.8 Poker, Extensive Forms, and the PEG

The game of Poker, particularly its element of bluffing, has always fascinated game theorists. Poker's arcane system of ranking hands adds more to the color than to the essence of the game, so most abstract Poker models begin by simply assigning a random number to each player, with the winner ultimately being whichever surviving player has the biggest number. The chance of a tie in Poker is so small that it is reasonable to base a model on the assumption that ties are impossible, in which case the random number can be assumed to lie uniformly in the unit interval. A model of this type will be considered in Chap. 5. Analyses of the case where only a few values for a hand are possible have also been conducted (Kuhn 1950), but in this case one has to make provision for the possibility of a tie. The Basic Poker model is of this type, but it avoids the possibility of a tie by incorporating the additional assumption that only one of the two players gets a hand. The asymmetry is at first unappealing, but surprisingly many situations in (real) Poker involve two players, one of whom has a middle-ranking hand that is more or less obvious on account of cards that are exposed, while the other player has a mainly concealed hand that is either very powerful or worthless. In such a situation the player with the concealed hand might consider "bluffing" even if his hand is actually worthless, which is what makes the game interesting. Here are the rules of Basic Poker:

- Both players put 1 unit, the ante, into the pot.
- Player 1 is dealt either a high card (probability p) or a low card (probability $q = 1 - p$). The card is unseen by player 2, but both players know p.
- After observing his hand, player 1 may either put an additional b units into the pot ("bet"), or not ("fold"). If he folds, player 2 gets the pot.
- If player 1 bets, player 2 may either put an additional b units into the pot himself ("call") or not ("fold"). If he folds, player 1 gets the pot.
- If player 1 bets and player 2 calls, the pot goes to player 1 if he has a high card, or otherwise to player 2.

Since the game proceeds in stages, it is tempting to diagram the rules as the "extensive form" shown in Fig. 3.8-1.

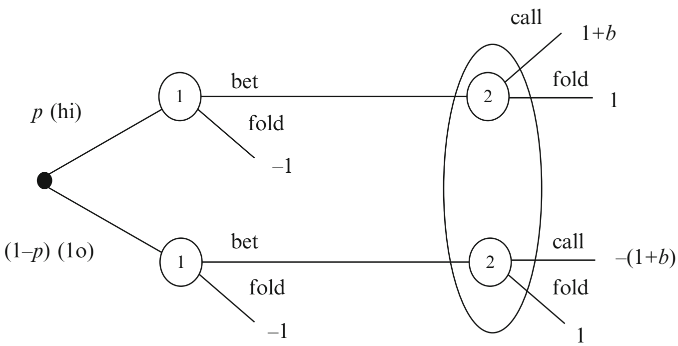


Fig. 3.8-1 Extensive form of Basic Poker

The initial (left) node in Fig. 3.8-1 is unlabeled, which means that the exit branch is randomly chosen with specified probabilities. In this case player 1 gets a high hand with probability p or a low hand with probability $1 - p$. In either case player 1 must next decide whether to bet or fold. Player decisions are shown as circled player numbers—there are four such, two for each player. The elliptical enclosure is an "information set"; it is needed because player 2 is only supposed to know player 1's action, and not whether it was based on a high or low card, in choosing an action himself. There are 2 nodes in the information set, the idea being that player 2 does not know which one represents the state of the game. Player 1's two information sets contain only a single node each, so enclosures are not needed. Information sets are an unwelcome (since what started out to be a tree quickly gets complicated by contorted ellipses crossing over branches) but necessary feature in diagramming games in this manner. The payoffs shown at the ends of terminal branches are "net gain to player 1". In spite of the need for an information set enclosure, the extensive form in Fig. 3.8-1 is a clearer description of the game than the list of rules that precede it.

If the extensive form of a game takes the form of a tree with no information set enclosures, then it is a "Tree game". Checkers and Backgammon are examples of

Tree games. The term "game of perfect information" is also used and equivalent to "Tree game", but we prefer the latter because it is more evocative of the nature of such games, and also because the term "perfect information" is easy to confuse with "complete information", which has a separate meaning. Finite Tree games can be solved by an efficient method that will be detailed in Sect. 4.3.

Figure 3.8-1 has an information set enclosure, so Basic Poker is not a Tree game. We therefore have the problem of reducing the extensive form to the "normal form" (a matrix) in order to solve it. The key to accomplishing this is to define a strategy to be a rule for playing the game that anticipates all possible eventualities. There are four distinct strategies for player 1, the first being "bet whether the card is high or low", the second "bet on a high card, fold on a low card", etc. These are the four rows of Fig. 3.8-2. In general, the number of strategies for a player is a product of

Fig. 3.8-2 Normal form of Basic Poker

$$\begin{array}{ll} \text{hi} & \text{low} \\ \text{bet} & \text{bet} \\ \text{bet} & \text{fold} \\ \text{fold} & \text{bet} \\ \text{fold} & \text{fold} \end{array}
\begin{array}{c} \begin{array}{ll} \quad\text{call} \quad\quad\quad & \text{fold} \end{array} \\ \begin{bmatrix} (p-q)(b+1) & 1 \\ p(b+1)-q & p-q \\ -1-qb & q-p \\ -1 & -1 \end{bmatrix} \end{array}$$

factors, one for each information set, with each factor being the number of choices in that information set. Player 1 has four strategies because $4 = 2 \times 2$. Player 2 has only one information set, and that information set generates only two strategies. Player 2's two strategies are shown as the columns of Fig. 3.8-2. Each entry in Fig. 3.8-2 is the average payoff to player 1 given the pair of strategy choices; in taking averages we are implicitly assuming that utility is linear in money for both players. The payoff in row 1 and column 1, for example, is $p(b + 1) + q(-b - 1)$—since both players always bet and call, the pot is always $b + 1$, and player 1's card determines who gets it. The other seven payoffs can be similarly derived as averages. Note that the payoffs depend on parameter p. Even though player 2 does not know whether player 1's card is high or low, he must know p in order to know the payoff matrix—as always, we assume that both players know the rules.

The last two rows of Fig. 3.8-2 are dominated (no surprise, since there is not a lot of intuitive appeal to folding with a high card). After eliminating them, all that remains is to solve a 2×2 game. If $(p - q)(b + 1) \geq 1$, the game has value 1 at a saddle point where player 1 always bets and player 2 always folds. Otherwise, using the 2×2 formula (3.4-1),

$$\begin{aligned} x_1^* &= pb/(q(b+2)), \\ y_1^* &= 2/(b+2), \text{and} \\ v &= \big(b+2(p-q)(b+1)\big)/(b+2). \end{aligned}$$

Note that the value of the game is positive when $p = q = 1/2$, but that the game can be made fair by adjusting p and b so that $v = 0$. With p fixed, x_1* and y_2* are both increasing functions of b. As the stakes increase, there should be more betting and less calling. The "official" version of Basic Poker is a fair game where $p = 1/3$ and $b = 2$, in which case $x_1{*} = 0.25$, $y_1{*} = 0.5$ and $v = 0$. When played optimally, this version results in a bluff being called one time in twelve (exercise 6).

Figure 3.8-2 is simple enough, but it should be clear that extensive forms that could easily fit on one page can have very large normal forms on account of the multiplication rule for counting strategies. Most of these strategies may ultimately be dominated, but that is scant comfort to the poor analyst who has to develop the matrix. This strategy explosion problem is similar to Bellman's "curse of dimensionality", but we prefer to call it the Problem of Exponential Growth (PEG) of strategies. The acronym is irresistible because there is some truth to the statement that "As a practical matter, TPZS game theory often gets hung up on the PEG". While there is tutorial value in constructing the normal form for small games such as Basic Poker, solution of practical games by painstakingly reducing the extensive form (or some other version of the rules of the game) to the normal form is rare on account of the PEG.

Although every game in extensive form can in principle be reduced to a unique game in normal form by listing all of the strategies for the two sides, the converse is not true. Multiple games in extensive form can have the same normal form.

3.9 Bare Hands and the Principle of Equality

Let $\mathbf{x}*$, $\mathbf{y}*$, and v be the solution of a game with payoff matrix $\mathbf{a}$, and assume that $\sum_j a_{ij} y_j{*} < v$. It follows that $x_i{*} = 0$, since row i achieves less than v against $\mathbf{y}*$ (exercise 4). Equivalently, $x_i{*} > 0$ implies $\sum_j a_{ij} y_j{*} = v$. Let A_1 be the set of rows for which $x_i{*} > 0$—the set of "active" strategies for player 1. Then $\sum_j a_{ij} y_j{*} = v$ for i in A_1. This is the Principle of Equality: player 2's optimal strategy is equalizing in the sense that all active alternatives for player 1 have the same expected payoff. The principle can be used to solve for $\mathbf{y}*$ and v in games for which A_1 is for some reason obvious. Alternatively, the equations $\sum_i x_i{*}\, a_{ij} = v$ for j in A_2 can sometimes be used to solve for $\mathbf{x}*$ and v when player 2's active set A_2 is obvious.

Let m_1 be the number of active rows and m_2 be the number of active columns. There are m_1 equations of the form $\sum_j a_{ij} y_j{*} = v$, one for each i in A_1, and m_2 unknown quantities $y_j{*}$, one for each j in A_2. The game value v is also unknown, but there is also one more equation to the effect that each player's probabilities must sum to 1. Barring the possibility of redundant equations (exercise 12), we expect $m_1 = m_2$ because there should be one equation for each unknown. This observation leads to the trial and error or "bare hands" method for solving games:

1. Guess the sets of active strategies A_1 and A_2, with each set having the same number of strategies.
2. Solve the appropriate linear equations to obtain $\mathbf{x}$, $\mathbf{y}$, and v.
3. Test whether $\mathbf{x}$, $\mathbf{y}$, and v solve the game. If not, go back to step 1.

Solving sets of linear equations is sufficiently tedious that one would like to guess A_1 and A_2 correctly the first time. In square games it is sometimes plausible to argue that all strategies must be active, in which case the game is said to be "completely mixed". Consider Fig. 3.9-1, which represents a search situation where there are three cells in which to hide. The detection probability is zero unless

Fig. 3.9-1 A three-cell search game

$$\begin{bmatrix} 1/2 & 0 & 0 \\ 0 & 2/3 & 0 \\ 0 & 0 & 3/4 \end{bmatrix} \qquad \begin{aligned} y_1\ (1/2) &= v \\ y_2\ (2/3) &= v \\ y_3\ (3/4) &= v \\ y_1 + y_2 + y_3 &= 1 \end{aligned}$$

the searcher (player 1) looks in the hider's cell. Intuitively, the game is completely mixed because a strategy that never uses any specific row or column is exploitable by the other side. If player 2 were to never hide in cell 3 for example, then player 1 should never look there, but if player 1 never looks in cell 3, then player 2 should hide there. The initial assumption leads to a contradiction, as would the assumption that player 2 never hides in any specific cell. All three columns must be active, and the same is true of the three rows.

The assumption that all three rows are active leads to the solution of the four simple equations that are shown in the figure. The solution is $\mathbf{y}^* = (12, 9, 8)/29$ and $v = 6/29$. It is easy to see that player 1 can guarantee 6/29 using exactly the same strategy, so 6/29 is indeed the value of the game. There is no dominance in Fig. 3.9-1, so the game could not have been solved by methods introduced earlier. It should be clear that all games that have zero payoffs except for positive numbers on the main diagonal can be solved by this insight, regardless of how large they are. Note that the searcher is most likely to do what he is worst at! His rationale for being attracted to cell 1, in spite of his low detection probability there, would presumably be that cell 1 is more likely to contain the hider.

The bare hands procedure is fail-safe as long as one makes sure that both sets of linear equations produce probability distributions. To illustrate the necessity for solving both sets, suppose that player 1 feels that 6/29 is not a high enough detection probability, and that he therefore installs in cell 1 a sensor that will detect hiders in cell 1 with probability p whether player 1 searches there or not. Player 2 knows about this revision to the rules of the game. The revised game matrix is shown in Fig. 3.9-2, together with the four equations (u is the tentative game value) that result from assuming that all rows are active ($A_1 = \{1,2,3\}$).

Fig. 3.9-2 Modified search game

$$\begin{bmatrix} .5+5p & 0 & 0 \\ p & 2/3 & 0 \\ p & 0 & 3/4 \end{bmatrix} \qquad \begin{aligned} (1+p)\,y_1/2 &= u \\ py_1 + 2y_2/3 &= u \\ py_1 + 3y_3/4 &= u \\ y_1 + y_2 + y_3 &= 1 \end{aligned}$$

Letting $p_0 = 6/17$ to avoid the necessity for repeatedly writing the fraction, the solution of the equations in Fig. 3.9-2 is

$$\begin{aligned} y_1 &= 2p_0/(2p_0 + 1 - p) \\ y_2 &= (3/2)p_0\big(1 - p\big)/\big(2p_0 + 1 - p\big) \\ y_3 &= (4/3)p_0\big(1 - p\big)/\big(2p_0 + 1 - p\big) \\ u &= p_0(1 + p)/\big(2p_0 + 1 - p\big) \end{aligned}$$

All three components of **y** are positive, so the value of the game is at most u. However, the possibility that the value of the game is smaller than u remains because the assumption that all rows are active may be wrong. To complete the solution, assume that all columns are active and solve the resulting four equations to obtain:

$$\begin{aligned} x_1 &= 2(p_0 - p)/\big(2p_0 + 1 - p\big) \\ x_2 &= (3/2)p_0\big(1 + p\big)/\big(2p_0 + 1 - p\big) \\ x_3 &= (4/3)p_0\big(1 + p\big)/\big(2p_0 + 1 - p\big) \\ u &= p_0(1 + p)/\big(2p_0 + 1 - p\big) \end{aligned}$$

The expression given for x_1 is negative for $p > p_0$, so the assumption that the game is completely mixed turns out to be correct only for $p \le p_0$. It is only in that case where the above expressions for **x**, **y**, and u are **x***, **y***, and v, the solution of the game. Analysis of the case $p > p_0$ is continued in exercise 7.

Many of the game solutions reported in the literature have been obtained by the bare hands technique, which always involves solving some equations based on an insight into which strategies should be active. These equations determine a pair of mixed strategies and a possible game value. Holding his breath lest some necessary condition be violated, the analyst then attempts to verify that he has actually solved the game.

3.10 Linear Programming

Linear programming (LP) software can solve the problem of optimizing one linear function of several variables subject to constraints on other linear functions. This large class of optimization problems turns out to include the solution of TPZS matrix games. For more information on LP and the notation in use here, see appendix A.

Theorem 3.2-1 makes two statements. The first involves only player 1's strategy **x**, and requires that **x** must guarantee v, on the average, against every column. In other words it is required that $\sum_{i=1}^{m} x_i a_{ij} \ge v;\ \ j = 1, \ldots, n$. As long as **x** and v satisfy all of these inequalities, **x** guarantees a floor of v for player 1. The problem of finding the highest possible floor therefore emerges naturally for player 1 as linear program LP1, written out below for the case where **a** is an $m \times n$ matrix:

$$\begin{aligned}&\max_{v,\mathbf{x}\geq 0} v\\&\text{subject to} \sum_{i=1}^{m} x_i a_{ij} - v \geq 0;\;\; j = 1, \ldots, n\\&\sum_{i=1}^{m} x_i = 1\end{aligned} \tag{3.10-1}$$

As is customary in stating mathematical programs, constraints in LP1 do not involve variables on the right-hand side. The last constraint of LP1 requires that the nonnegative components of **x** must sum to 1, which makes **x** a probability distribution. The objective function and constraints are all linear functions of the variables, so LP1 is a linear program with $m + 1$ variables (counting v) and (counting the m constraints implied by requiring that **x** be a nonnegative vector) $m + n + 1$ constraints.

Had we chosen to look at the problem from player 2's viewpoint, we would have concentrated on statement 2 of Theorem 3.2-1 and formulated player 2's linear program LP2:

$$\begin{aligned}&\min_{v,\mathbf{y}\geq 0} v\\&\text{subject to} \sum_{j=1}^{n} y_j a_{ij} - v \leq 0\\&\sum_{j=1}^{n} y_j = 1\end{aligned} \tag{3.10-2}$$

LP2 has $n + 1$ variables and $m + n + 1$ constraints.

There is nothing deep about any of the above. LP1 and LP2 could have been written down before 1928 as perfectly definite mathematical problems, LP1 to find the highest floor and LP2 to find the lowest ceiling. It was not until 1928, however, that it became known that LP1 and LP2 must have the same optimized value—essentially the Duality Theorem of linear programming. LP1 and LP2 are dual linear programs, so either one can be solved to find the value of the game. In fact, if the LP code being used produces dual variables, as most do, then the dual variables of the value constraints are the optimal strategy for the other player, so solution of a single program provides $\mathbf{x}^*$, $\mathbf{y}^*$, and v.

The game of Morra provides a good example of the use of LP to solve a game. Morra is an old game dating back to the Roman empire, and perhaps even earlier. In Morra (Fig. 3.10), each of two players exposes from one to three fingers and simultaneously predicts the number exposed by the opponent. There is no payoff unless exactly one player predicts correctly, in which case he collects from the other player the sum of both exposures. Thus if the row player exposes 2 and predicts 1, while the column player exposes 2 and predicts 2, then the column player collects 2 + 2 from the row player; that is, the payoff is − 4.

The value of Morra is clearly 0 by symmetry, but there is no dominance and the sets of active strategies are not obvious. The complete 9 × 9 payoff matrix is shown

Fig. 3.10 A Morra game in progress (Bartolomeo Pinelli 1809)

in exercise 16 and also on page "Morra" of the TPZS workbook, where the reader can either try to guess the solution or use linear programming to solve LP2. The optimal mixed strategy turns out to use only 3 of the 9 available strategies, and the probabilities of those 3 are not equal. Among the solution techniques discussed so far, linear programming is the only one capable of discovering this solution.

The solution of Morra by LP requires only ten variables and constraints, but commercial codes exist that can solve LP problems with thousands of variables and constraints in reasonable amounts of time. Still, the PEG described in Sect. 3.8 can frustrate even such an impressive capability. The number of rows and columns required to describe the normal (matrix) form of an interesting game can easily swamp any LP code. Without trying to be overly precise, it seems that TPZS games are of two kinds. In one kind it is natural to describe the game in normal form. Holmes versus Moriarty falls in this class, as does Morra, and more generally so do all games where one player is trying to guess some property of the other. For such games, the capabilities of modern LP codes are more than sufficient. In the other kind of game the normal form has to be forcibly imposed by exhaustively enumerating all the strategies for each side, in the process often losing whatever structure is present in a more natural representation. Figures 3.8-1 and 3.8-2 each precisely define the game of Basic Poker, but the nature of the game is far clearer in the extensive form than in the normal form. This second kind of game often involves multiple decisions or multiple circumstances for making the same decision. Even if one is willing to destroy the structure of such a game by dealing with it in normal form, the LP procedure is problematic because of the PEG. It is easy for the normal form of such a game to involve millions or even billions of strategies, rather than thousands. Therefore, while LP applied to the normal form is a powerful and useful tool for solving TPZS games, it is not the end of the story.

3.11 Methods for Large Games

The bare hands approach is one method for solving large games, but it relies on insight and luck. This subsection describes three additional methods that are more systematic.

3.11.1 The Brown-Robinson Method

The idea of solving a game by simply playing it repeatedly is appealing. Consider the idea of taking turns, with each player selecting a strategy that is optimal against his opponent's previous choice. The usual result is that the players end up going around a cycle repeatedly with no convergence to a game solution, a disappointing result. However, a closely related idea has been shown to work. The essential change is that both players need to remember all of the other player's previous choices, not just the most recent one, each player selecting a strategy that is optimal against the observed empirical distribution of the strategy choices of the other player. This is the Brown Robinson (BR) method.

To make the BR method definite, let vectors $\mathbf{X}$ and $\mathbf{Y}$ be counts of the number of times each player has selected each of his pure strategies in a repeatedly played $m \times n$ game, and let the number of plays so far be $K \equiv \sum_{i=1}^{m} X_i = \sum_{j=1}^{n} Y_j$. To avoid division by 0, initialize $\mathbf{X}$ and $\mathbf{Y}$ to have a single count for some arbitrary strategy, so $K = 1$ after this first play. The empirical mixed strategies observed after the first K plays are $\mathbf{x} = \mathbf{X}/K$ and $\mathbf{y} = \mathbf{Y}/K$. In the next play, player 1 chooses a strategy that is optimal against $\mathbf{y}$, and player 2 chooses a strategy that is optimal against $\mathbf{x}$. The appropriate components of $\mathbf{X}$ and $\mathbf{Y}$ are then incremented by unity, new empirical distributions are calculated, and play proceeds. Note that the players are automatons here—actual humans are not needed because each player's behavior is entirely dictated by our assumptions.

Finding the best response to a definite mixed strategy is a much easier task than solving a linear program, so the BR method may be feasible even when the normal form is not directly available or too large for LP. Sheet "MorraBR" of the TPZS workbook shows the operation of the BR method in approximately solving Morra. The command button on that sheet accomplishes a single play as described above.

Every distribution $\mathbf{x}$ has an associated lower bound on the game value and every distribution $\mathbf{y}$ has an associated upper bound. Robinson (1951) proved that the BR technique works in the sense that the game bounds associated with the best empirical distributions will approach each other in the limit as the number of plays approaches infinity. However, the convergence is not fast and the bounds may never be exactly equal, so BR is a method for approximation, rather than solution.

3.11.2 Strategy Generation

The BR method incorporates memory of past play in the count vectors **X** and **Y**. The strategy generation (SG) technique also incorporates memory, but only the sets of strategies that have been employed in the past need to be remembered, rather than the associated empirical frequencies. As play proceeds these sets gradually expand until they are sufficiently large, similar to the use of column generation in Sect. 2.5.2.

Consider a large $m \times n$ game **a**. One employs SG in the hope that, even though m and n may be large, the optimal mixed strategies will employ only a small fraction of the available pure strategies. Let $G(S_1,S_2)$ be the game where the two players are restricted to using strategies in S_1 (player 1) or S_2 (player 2). Sets S_1 and S_2 are initialized to be small but nonempty sets of strategies. The SG algorithm for solving the game is then:

1. Solve the game $G(S_1, S_2)$, and let a solution be $\mathbf{x}$, $\mathbf{y}$, v.
2. Test whether there is some j not in S_2 such that $\sum_{i \in S_1} a_{ij} x_i < v$. If so, add j to S_2.
3. Test whether there is some i not in S_1 such that $\sum_{j \in S_2} a_{ij} y_j > v$. If so, add i to S_1.
4. If nothing is added to either set in steps 2 or 3, stop with $\mathbf{x}$, $\mathbf{y}$, v being the solution of the game. Otherwise go to step 1.

As long as the original game is finite, the SG algorithm will eventually stop because S_1 and S_2 will expand to include all rows and columns, and all of the inequalities required of a solution to game **a** will be satisfied.

One employs SG when steps 2 and 3 are easy to carry out in spite of the large size of m and n. SG is superior to BR in the sense that it will eventually find an exact optimal solution, and inferior in the sense that each employment of step 1 involves the solution of a game. Of course one hopes that termination of SG will happen quickly, since strategy generation is undertaken only because one cannot accomplish step 1 when S_1 and S_2 are complete. The perfect application would be to a game that has a saddle point that one is clever enough to guess, initializing S_1 and S_2 accordingly. Convergence will happen in a single step. However, this desirable property doesn't extend to games without saddle points. Even if S_1 and S_2 are initialized to be the sets of strategies that are active in the complete game, convergence may still take a long time. Morra is an example of this. Letting "x-y" be the strategy of exposing x fingers and guessing y for the opponent, the optimal active set for both players (the game is symmetric) is {1-3, 2-2, 3-1}. If S_1 and S_2 are both initialized to be this set, $G(S_1,S_2)$ will be a 3×3 game where all payoffs are 0, so any strategies whatever are optimal. Unless tremendous luck leads to selection of the strategies that are optimal in G, the sets S_1 and S_2 will be augmented. In fact, the SG procedure is likely to end up with both sets of strategies being complete, a disappointing outcome. One might say that SG is something to try on large games that "almost have saddle points", and that Morra is an example of a game that isn't like that. On the other hand Jain et al. (2011) report success using SG on a large network interdiction game.

3.12 Experiments with TPZS Games

Experiments have been performed to test whether humans play games in the manner that logic would seem to dictate. In the simplest experiment, two players repeatedly play a strictly determined game. The experimenter observes whether the choices are "correct" (the saddle point choices), and whether the tendency to be correct improves with experience. The experimental evidence is mixed, but favors the conclusion that humans usually do play the saddle point once they have appreciated its significance and convinced themselves that the opponent is rational. Lieberman (1960) asked undergraduate students to play a 3×3 saddle point game 200 times. About half of the students played the saddle point throughout. Saddle point choices were more common in the later stages, and non-saddle point choices were, according to player accounts, sometimes motivated by a desire to alleviate the boredom of doing the same thing 200 times in a row. Morin (1960) reports that non-saddle point rows for which the row sum is large are competitive with the saddle point row—apparently subjects are reluctant to abandon the hypothesis that the opponent is equally likely to choose any column.

There have also been some experiments with games that do not have saddle points. In this case it is not possible to declare that any given choice is in agreement with theory or not, so the experimenter usually compares $\mathbf{x}^*$ and $\mathbf{y}^*$ to the frequencies with which particular rows or columns actually get chosen in repeated play. These frequencies are sometimes estimated in blocks of trials to test whether any convergence takes place. Broadly speaking, humans do not appear to play even 2×2 games according to optimal mixed strategies. Kaufman and Lamb (1967) report flatly that "under the conditions of the present experiment, players do not learn to play a game theory optimal strategy". They note that spinners left lying about in case players felt the need to randomize were seldom used. Other investigators have reported a tendency toward optimality, but only a slight one and only for some players.

Messick (1967) had subjects repetitively play a game with value $7/45 = 0.23$ against a computer that was programmed to play either the optimal mixed strategy or a Brown-Robinson (BR) strategy that always chooses the next column to achieve the minimal expected gain against the currently observed row frequencies. The BR strategy does well against subjects who choose rows independently on each trial from a non-optimal distribution, but is itself exploitable because its choices are theoretically predictable. Subjects were told only that the computer was programmed to maximize its own (minimize the subject's) winnings. The average payoff to the subjects was 0.10 against the optimal strategy and 0.66 against BR, the difference between 0.66 and the game value being statistically significant. The human players exploited BR, rather than vice versa! This is consistent with the idea that humans approach game playing by trying to outwit the opponent, rather than by choosing strategies independently at random on each trial. Against a BR strategy that only remembers the last 5 row choices, humans did even better (0.95). Against the optimal randomized strategy, humans were of course unable to discern a pattern in the computer's choices, and suffered from trying to do so. Messick concluded that "the study reported here unambiguously indicates that human subjects do not behave in a manner consistent with minimax theory". Other investigators have put it less strongly, and some have been unwilling to reject the hypothesis that humans do behave in accordance with theory, but nonetheless Messick's conclusion isn't a bad summary.

Exercise 18 is to solve a game called Undercut that Hofstadter (1982) and a companion invented to relieve boredom on a long bus trip. His experience in playing Undercut led to his subsequent construction of a computer program designed to play the game repeatedly, searching for patterns in the opponent's choices that permit a forecast of his next move. Hofstadter's favorite program eventually met a competing program that simply made its choices using a random number generator. The competitor could not be beat because it used the optimal frequencies, and of course a random number generator cannot be forecast. Hofstadter's emotional reaction to this was disappointment that such a mindless technique should be so effective. Playing a game optimally is sort of a joyless exercise compared to the fascinating business of trying to predict the opponent's next move. There is not much to be said for optimal strategies other than that they are optimal.

The fact that humans do not gravitate toward optimal play should be worrisome. It is tempting to approximate the value of a difficult game through operational gaming—playing it repeatedly with human players and averaging the results. In fact, it is tempting to argue that the analyst's goal should be to measure the average payoff when strategies are chosen by humans (call it v_H), since humans will ultimately be the decision makers anyway. But suppose $v_H > v$. Surely player 2 should change his behavior, since there is a way to guarantee that the payoff will not exceed v, no matter what player 1 does. The game theoretic definition of "optimal" is sufficiently impressive that deviations from it should motivate corrections. But this kind of correction is not possible if v is unknown, and humans are unlikely to discover v by simply playing the game. This problem is particularly acute when the game is a modification of one with which the humans are familiar, since the human tactics may not intuit the potential of the modification. Logic dictates the necessity of knowing the game value, but human habits and abilities make its discovery difficult.

3.13 Exercises

1. Solve all of the games shown below that are solvable using the techniques of Sects. 3.3–3.7.

(a) $\begin{bmatrix} 3 & 6 & 1 & 4 \\ 5 & 2 & 4 & 2 \\ 5 & 5 & 3 & 5 \end{bmatrix}$	(f) $\begin{bmatrix} 2 & 1 & 1 \\ 1 & 8 & 8 \\ -2 & 0 & 6 \\ 0 & 0 & 0 \\ 9 & 0 & 0 \end{bmatrix}$

(continued)

(continued)

(b) $\begin{bmatrix} 4 & 1 \\ 2 & 3 \end{bmatrix}$	(g) $\begin{bmatrix} 8 & 3 & 0 & 7 \\ 0 & 3 & 4 & 2 \end{bmatrix}$
(c) $\begin{bmatrix} 8 & 0 & 4 \\ -2 & 2 & 4 \end{bmatrix}$	(h) $\begin{bmatrix} 2 & 1 & 0 & 6 \\ 5 & 3 & 4 & 9 \\ 1 & 2 & 0 & 3 \\ 5 & 1 & 2 & 8 \\ 9 & 3 & 1 & 1 \\ 4 & 4 & 6 & 2 \end{bmatrix}$
(d) $\begin{bmatrix} 10 & 2 & 0 \\ 8 & 2 & 3 \\ 0 & 1 & 7 \end{bmatrix}$	(i) $\begin{bmatrix} 2 & 4 & 9 & 0 & 5 \\ 8 & 3 & 2 & 5 & 0 \\ 1 & 3 & 8 & 0 & 4 \end{bmatrix}$
(e) $\begin{bmatrix} 0 & 1 & 4 \\ 0 & -3 & 0 \\ 1 & 2 & 1 \\ -2 & 0 & 6 \end{bmatrix}$	

2. Messick (1967) used the game shown below in the experiment mentioned in Sect. 3.12. Verify that the solution is $\mathbf{x}^* = (18/45, 5/45, 22/45)$, $\mathbf{y}^* = (25/45, 9/45, 11/45)$, and $v = 7/45$.

$$\begin{bmatrix} 0 & 2 & -1 \\ -3 & 3 & 5 \\ 1 & -2 & 0 \end{bmatrix}$$

3. Prove the first conclusion of Theorem 3.2-2. Why is the assumption that $c \geq 0$ required?

4. If $\mathbf{x}^*$, $\mathbf{y}^*$, v solve a game with $m \times n$ payoff matrix $\mathbf{a}$, prove that $x_i^* = 0$ if $R_i < v$, where $R_i = \sum_{j=1}^{n} a_{ij} y_j^*$.

5. The text claims that rows 3 and 4 of Fig. 3.8-2 are dominated. Which row or rows dominate row 3? Which row or rows dominate row 4?

6. Poker players often argue over the ideal ratio of the maximum bet to the ante. In Basic Poker, at least, the question is answerable. Define the ideal game to be the fair game ($v = 0$) for which the probability of a called bluff is greatest in optimal play. Show that the ideal Basic Poker game has $p = \sqrt{2}/4$ and $b = \sqrt{2}$. In what fraction of the hands dealt will player 1 bet on a low card (bluff) and be called by player 2?

7. Continue the analysis of the search game considered in Sect. 3.9 for the case $p > p_0$. Sketch v as a function of p for $0 \leq p \leq 1$.

8. Change the rules of Basic Poker so that the single card dealt is exposed for all to see. The modified game is clearly silly, since bluffing is impossible. Nonetheless, diagram the game in extensive form and count the number of strategies that each side would have if the game were reduced to normal form. The

extensive form should be in the form of a tree with no information sets that enclose multiple decisions.

9. Referring to Sect. 3.9, verify that the assumption $A_2 = \{1, 2, 3\}$ for the game below results in $\mathbf{x}^* = (29, 7, 16)/52$ and $v = 101/52$ —solving linear equations is not required to do this.
   (a) Is 101/52 a bound on the value of the game, and if so is it an upper or lower bound?
   (b) Verify that the correct set of active strategies for player 1 is actually $A_1 = \{1, 2\}$.

$$\begin{bmatrix} 3 & 1 & 0 \\ 2 & 8 & 3 \\ 0 & 1 & 5 \end{bmatrix}$$

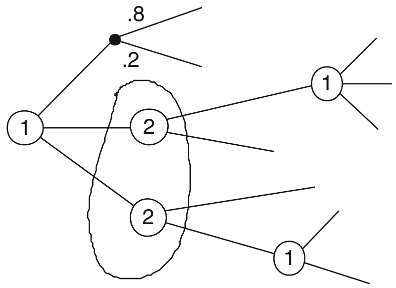
10. There are two contested areas, and two players must secretly divide their units between the two areas. Player 1 captures an area if and only if he assigns strictly more units than player 2, and the payoff is the number of areas captured by player 1. If player 1 has a total of 3 units and player 2 has 4 units, and if units can only be assigned in integer quantities, construct the payoff matrix and solve the game. "Blotto" games such as this are the subject of Chap. 6.

11. Modify and solve the games of exercises 1c and 1d above to reflect the idea that player 2 has a spy in player 1's camp. More precisely, with probability 0.5 the spy will discover player 1's strategy choice and report it to player 2 before player 2 makes his own choice, or otherwise the spy will make no report at all. Both sides are aware of the fact that player 1's strategy will be "leaked" to player 2 with probability 0.5, but player 1 will not know whether the leak actually occurs. The modification should not change the number of strategies for either side, but player 2's second strategy in the modified game, for example, is "choose column 2 if there is no leak, or else choose the best response to player 1's choice".

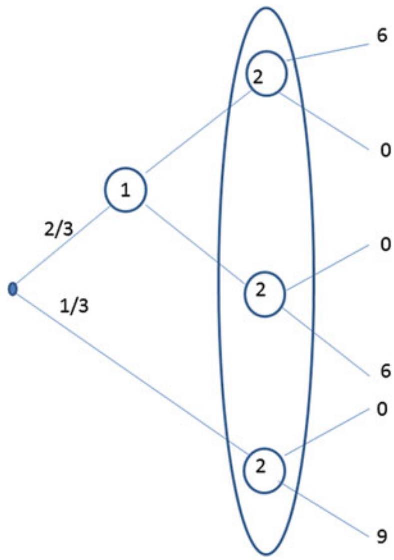
12. (a) Show that $\mathbf{x}^* = (0.5, 0.5)$, $\mathbf{y}^* = (0, 1, 0)$, and $v = 1$ are a solution to the game below, thereby demonstrating that the two players need not always have the same number of active strategies.
   (b) Give a different solution where both players do have the same number of active strategies.

$$\begin{bmatrix} 2 & 1 & 0 \\ 0 & 1 & 2 \end{bmatrix}$$

13. If the game in extensive form shown below were converted to normal form, how many rows and columns would there be? Payoffs at the end of each branch have been omitted because actual construction of the normal form is not necessary.



14. The paper by Aumann and Machsler (1972) is a rare example of game theorists objecting to the use of mixed strategies in a TPZS game. Their objection is based on the idea that the normal form of a game can miss some important features that are present in the extensive form. They consider the game in extensive form shown below. Player 2 has only one information set, so his decision must be the same at all three of the decisions that he controls. He is essentially required to say "up" or "down" before learning anything about either the random move or the possible move of player 1. Find the normal form of the game and the optimal mixed strategies for the two sides. If you were player 1, would you use his "optimal" strategy?



15. Consider a game where each player can choose a positive integer and where the payoff is

$$a_{ij} = \begin{cases} j^2 & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}$$

Does this infinite game have a solution? If so, what is it?

Hint: $\sum_{j=1}^{\infty} j^{-2} = \frac{\pi^2}{6}$

Linear Programming Exercises

16. Use LP to solve the 2×3 game $\begin{bmatrix} 2 & 1 & 6 \\ 3 & 5 & 0 \end{bmatrix}$.

17. In Morra, each player extends from one to three fingers and simultaneously calls out a guess at the number extended by the opponent. There is no payoff unless exactly one player guesses correctly, in which case he gets an amount from the other player proportional to the total number of fingers extended by both players. Using the code that "x-y" means "extend x fingers and call out y", the payoff matrix is shown below, as well as on page "Morra" of the TPZS workbook.

    (a) Suppose each side were to use a mixed strategy where every strategy is used with the same probability. Find the corresponding bounds on the game value.
    (b) Use LP to find a solution, perhaps using sheet "Morra" of the TPZS workbook.

$$\begin{array}{c} \\ \text{1-1} \\ \text{1-2} \\ \text{1-3} \\ \text{2-1} \\ \text{2-2} \\ \text{2-3} \\ \text{3-1} \\ \text{3-2} \\ \text{3-3} \end{array}
\begin{array}{c}
\begin{array}{ccccccccc} \text{1-1} & \text{1-2} & \text{1-3} & \text{2-1} & \text{2-2} & \text{2-3} & \text{3-1} & \text{3-2} & \text{3-3} \end{array} \\
\begin{bmatrix}
0 & 2 & 2 & -3 & 0 & 0 & -4 & 0 & 0 \\
-2 & 0 & 0 & 0 & 3 & 3 & -4 & 0 & 0 \\
-2 & 0 & 0 & -3 & 0 & 0 & 0 & 4 & 4 \\
3 & 0 & 3 & 0 & -4 & 0 & 0 & -5 & 0 \\
0 & -3 & 0 & 4 & 0 & 4 & 0 & -5 & 0 \\
0 & -3 & 0 & 0 & -4 & 0 & 5 & 0 & 5 \\
4 & 4 & 0 & 0 & 0 & -5 & 0 & 0 & -6 \\
0 & 0 & -4 & 5 & 5 & 0 & 0 & 0 & -6 \\
0 & 0 & -4 & 0 & 0 & -5 & 6 & 6 & 0
\end{bmatrix}
\end{array}$$

18. Hofstadter (1982) describes his experiences playing "Undercut" with a friend on a trip to Prague, and his subsequent attempts to program a computer to play the game optimally. The rules are that each player names an integer from 1 to 5, being paid that amount by the other player with one exception. The exception is that if the two numbers differ by exactly 1, the player naming the lower number is paid the sum of the two numbers by the other player. For example the net payoff is -3 if player 1 names 1 and player 2 names 4, or the net payoff is +3 if player 1 names 1 while player 2 names 2.

    (a) Show Undercut as a 5×5 matrix game, and calculate the optimal strategies. The game is symmetric, so the value should be 0.
    (b) As Hofstadter did, modify the rules so that player 2 picks a number in the range 2 to 6, rather than 1 to 5. Which player is favored by the change?

19. For matrix games with positive values, a trick can be employed to reduce the number of variables and constraints by 1. Consider LP2 of Sect. 3.10, and let $u_j = y_j/v$ for $j = 1, \ldots, n$.

    (a) Show that LP2 is equivalent to

$$\begin{aligned}&\text{Maximize}\sum_{j=1}^{n} u_j\\&\text{subject to : }\sum_{j=1}^{n} a_{ij}u_j \le 1; i = 1, \ldots, m.\\&u_j \ge 0; j = 1, \ldots, n\end{aligned}$$

    This is a linear program with n variables and $m + n$ constraints. Hint: v is minimized if $1/v$ is maximized.

    (b) Describe how **y** and v can be recovered from the solution of part (a)

    (c) Use LP to solve exercise 16 this way.

20. The payoff matrix partially shown below is the normal form of a game where player 2 guesses cells until he correctly chooses the one of three consecutively numbered cells that player 1 has chosen to hide in, being informed whether each guess is too high or too low. The payoff is the number of guesses required, counting the last guess even when there is only one cell remaining. Player 1 has three strategies because there are only three cells, but player 2 has more than three. The third strategy for player 2, for example, is "guess cell 2 first, and then either cell 1 or cell 3 on the second guess, depending on whether the first is too high or too low".

$$\begin{bmatrix} 1 & 1 & 2 & . & . \\ 2 & 3 & 1 & . & . \\ 3 & 2 & 2 & . & . \end{bmatrix}$$

    (a) Player 2 has two additional strategies. Complete the 3×5 payoff matrix.

    (b) What is the value of the game?

    (c) Is the strategy (1/3, 1/3, 1/3) optimal for player 1?

    Johnson (1964) gives the solution of all such games with 11 or fewer cells.

21. Three points A, B, C are located in the plane. They form a triangle with sides a, b, c, with a being the length of the side opposite point A, etc. Player 1 hides at one of the three points and stays there. Player 2 starts at one of the three points and then moves from point to point until he encounters player 1. The payoff is the total distance covered by player 2. Thus player 2 has six strategies, one of which we might call BCA. If player 2 uses BCA when player 1 hides at C then the payoff is the distance from B to C, which is a. There are 17 more payoffs, six of which are zero (those where player 2 starts at player 1's position). Assume $(a, b, c) = (3, 4, 5)$.

(a) Create the complete 3×6 payoff matrix. Is there any dominance among the rows and columns?
(b) What is the solution of the game?

22. Consider a search problem where the searcher tries to guess the cell that the evader is hiding in. The searcher has two chances, and wins unless he fails on both of them. At time 1 the evader can hide in either cell 1 or cell 2. At time 2 (but not time 1) he can also hide in cell 3.

(a) Model this situation as a game where each side has 6 strategies, and prove (use LP if you need to) that the value is 2/3
(b) Add the restriction that the evader cannot hide in cell 3 at time 2 if he hides in cell 1 at time 1—this reflects a restriction on the evader's speed. Solve the revised game.

Meinardi (1964) generalizes the revised game to include more than two time periods. Such games can be thought of as discrete, one-dimensional versions of the "Flaming Datum" problem, in which the evader tries to escape from some known spot without being caught (the Flaming Datum problem will return in Sect. 8.2.3).

23. Consider a game where $A(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{\infty} x_i y_i$. Player 1 can use any mixed strategy $\mathbf{x}$, but player 2 must observe the constraint $\sum_{i=1}^{\infty} j y_j \leq Y$, where Y is given, in addition to the constraint $\sum_{i=1}^{\infty} y_i = 1$. One interpretation is that x_i is the fraction of mines in a minefield that are programmed to detonate on the actuation that follows the i^{th} actuation, and y_i is the number of times that the minefield is swept before a single important transitor attempts passage. In that case $x_i y_i$ is the probability that a mine is "ripe" for the transitor, $A(\mathbf{x},\mathbf{y})$ is the average number of ripe mines, and the constraint on the average number of sweeps is present because there is only so much time available for minesweeping. Solve the game for $Y = 3$. You may be able to use the bare hands approach of Sect. 3.9. If you use linear programming, approximate ∞ by 7 or any larger number.

Chapter 4
Markov (Multistage) Games

Great fleas have little fleas upon their backs to bite 'em,
and little fleas have lesser fleas, and so ad infinitum.

Augustus de Morgan

4.1 Introduction, Inspection Game

In games that involve a sequence of moves, it can be useful to regard certain payoffs as themselves being games, and those payoffs might in turn have other games as payoffs, or possibly even the original game as a payoff. The idea of representing a game in this manner is often parsimonious and natural. The Inspection Game is a prototype.

Suppose that an Agent wishes to carry out some action without being noticed by an Inspector. In any time period the Agent can either "act" or "wait", and the Inspector can either "inspect" or "wait". The Inspector can only inspect once, so if he ever inspects during a period when the Agent has chosen to wait, the Agent can observe this and safely act in the next period. The only catch for the Agent is that there may not be a next period; with probability $1 - \beta$ at the end of each period, the possibility of a successful action disappears forever. This is an abstract version of several real life situations. For example, "act" could be "ship the supplies", "inspect" could be "interdict the supply route", and $1 - \beta$ could represent the probability that the need for the supplies will be overtaken by events. If we call the game G, then G can be represented as in Fig. 4.1, where the second row and column each corresponds to "wait" and the payoff is the probability that the Agent (player 1) acts successfully.

The unusual feature is in the lower right-hand corner of the 2×2 payoff matrix—G is used in describing itself. If both sides choose "wait", then with probability β the two players find themselves playing exactly the same game again. Determining the value is no longer a simple matter of solving a 2×2

A. Washburn, *Two-Person Zero-Sum Games*, International Series in Operations Research & Management Science 201, DOI 10.1007/978-1-4614-9050-0_4,

Fig. 4.1 The inspection game G

$$G = \begin{bmatrix} 0 & 1 \\ \beta & \beta G \end{bmatrix}$$

game, since one of the entries in the matrix involves the object of the computation. However, if the value v exists, then surely it should solve the equation

$$v = val\left(\begin{bmatrix} 0 & 1 \\ \beta & \beta v \end{bmatrix}\right) \tag{4.1-1}$$

where $val()$ is the function that finds the game value of a matrix. Using the 2×2 formula (3.4-1), (4.1-1) is $v = (-\beta)(\beta v - 1 - \beta)$, a quadratic equation with solution

$$v = \left(1 + \beta - \sqrt{(1+\beta)^2 - 4\beta^2}\right) \Big/ (2\beta) \tag{4.1-2}$$

Once v is known, $\mathbf{x}^*$ and $\mathbf{y}^*$ are easily obtainable by solving an ordinary 2×2 game. For example, when $\beta = 0.5$ we have $v = \left(3 - \sqrt{5}\right)/2 = 0.382$, $\mathbf{x}^* = (0.236, 0.764)$, and $\mathbf{y}^* = (0.618, 0.382)$. The sense in which v, $\mathbf{x}^*$, and $\mathbf{y}^*$ are a solution of the Inspection Game will be stated clearly in the next section; the important point is that the required calculations, at least in this simple game, are not particularly difficult.

The ease of obtaining a solution should not obscure the fact that that the Inspection Game G is a new kind of object. Given only the verbal description, G might be represented as a game where the Inspector chooses any positive integer as the inspection time, and where the Agent chooses any positive integer as the time at which to act if no inspection has yet occurred. This would avoid using G in describing itself, but only at the cost of making the payoff matrix ∞-dimensional. Theorem 3.2-1 applies only to finite payoff matrices, so there is no guarantee that the game can be solved. Indeed, we will see later that the Inspection Game has no solution in the usual sense when $\beta = 1$, even though (4.1-2) is perfectly well behaved at that point.

In the context of arms control inspections, Dresher (1962) analyzes a different inspection game G_{nk} where the Agent must act at one of n opportunities, k of which may be inspected by the Inspector (player 1), with the payoff being the probability that the Agent is inspected when he acts. The payoff is 1 when $k = n$, or 0 when $k = 0$. For $0 < k < n$, with row 1 being "inspect" and column 1 being "act", the payoff matrix is

$$G_{nk} = \begin{bmatrix} 1 & G_{n-1,k-1} \\ 0 & G_{n-1,k} \end{bmatrix}.$$

The payoffs in column 2 are subscripted by the remaining number of opportunities and inspections, so we are implicitly assuming that both sides can observe or calculate those quantities. Dresher's game and G are qualitatively different in the sense that G can go on indefinitely and appears in its own payoff matrix, whereas G_{nk} cannot and does not. Nonetheless, both are examples of Markov games.

There are many other formulations of inspection games in the literature, e.g. Ferguson and Melolidakis (1998).

4.2 Definition of Markov Games, Sufficiency Theorems

A Markov game G has finitely many finite game elements $G_1 \ldots G_K$, one of which is specified as the initial game element. Dresher's game is an example, since multiple subscripts can in principle be reduced to one by simply counting them. The play of each game element is called a "stage" (Markov games are often called "multi-stage" games for that reason), with the game element encountered at the next stage depending on the strategies chosen. The subscript(s) of the current game element will be called the "state" of G. The outcome when row i and column j are chosen in state k is written

$$a_{ij}(k) + \sum_{l=1}^{K} p_{ij}^{kl} G_l, \tag{4.2-1}$$

the interpretation being that player 1 receives a reward $a_{ij}(k)$ and in addition G_l is played with probability p_{ij}^{kl} (the superscripts k and l do not represent exponents, but only a dependence on k and l as well as i and j). The symbol "+" does not have the usual interpretation, since G_l is not a number, but the meaning of (4.2-1) should nonetheless be clear from the English description. Let $q_{ij}(k) \equiv \sum_l p_{ij}^{kl}$, so that $1 - q_{ij}(k)$ is the probability of termination when row i and column j are chosen in state k. If $q_{ij}(k) = 0$, the sum in (4.2-1) can be omitted because all terms are 0, and the game element is called "ordinary". Letting $A(t)$ be the reward at the t^{th} stage and M the number of stages before termination, the payoff in G is the total reward $X = A(1) + \ldots + A(M)$.

Definition If the initial state is s, we say that the value of G is v_s if there is a (mixed) strategy $\mathbf{x}^*$ for player 1 such that $E(X) \geq v_s$ regardless of the strategy $\mathbf{y}$ employed by player 2, and also a strategy $\mathbf{y}^*$ for player 2 that guarantees $E(X) \leq v_s$ regardless of the strategy $\mathbf{x}$ employed by player 1.

In principle, strategies are probability distributions over the rows and columns of the current game element that depend on the entire history of G (the current game element, all previous game elements, and all previous observable row or column choices). If the probability distribution depends only on the current game element, the corresponding strategy is called "stationary".

If a game value v_s exists for each initial state s, then the vector $\mathbf{v} = (v_1, \ldots, v_K)$ must satisfy a generalization of (4.1-1). For an arbitrary K-vector of real numbers $\mathbf{w}$, let the vector $\mathbf{T}(\mathbf{w})$ have components

$$T_k(\mathbf{w}) = val\Big(a_{ij}(k) + \sum\nolimits_l p_{ij}^{kl} w_l\Big); k = 1, \ldots, K. \tag{4.2-2}$$

The generalization referred to is the vector value equation

$$\mathbf{v} = \mathbf{T}(\mathbf{v}). \tag{4.2-3}$$

Equation (4.2-3) plays an important role in the analysis of Markov games, but the fact that it possesses a solution $\mathbf{v}$, even if the solution is unique, does not necessarily mean that $\mathbf{v}$ is the game value (more precisely the vector of game values) in the sense of the above definition. The difficulty is always that X might be ill-defined because M might be infinite. A condition that prevents this is required if solutions of (4.2-3) are to have the desired interpretation. It is sufficient to assume that the average length of play $E(M)$ is finite, as in the following theorem.

Theorem 4.2-1 For all states k let $\mathbf{x}(k)$ be a probability distribution over the rows of G_k, and let $\mathbf{x} \equiv (\mathbf{x}(1), \ldots, \mathbf{x}(K))$. Suppose there is a vector v such that

$$\sum\nolimits_i x_i(k) \Big(a_{ij}(k) + \sum\nolimits_l p_{ij}^{kl} v_l - v_k\Big) \geq 0; j \text{ in } G_k, \quad k = 1, \ldots, K. \tag{4.2-4}$$

Then, if a Markov game starts in state s, and if for all k player 1 uses $\mathbf{x}(k)$ to select a row on every occurrence of G_k, and if $E(M)$ is uniformly bounded above under these conditions regardless of the strategy employed by player 2, then $E(X) \geq v_s$.

Proof Let the random variables $J(t)$ and $S(t)$ be the column choice and state at stage t, respectively. Also let $V(t) = v_{S(t)}$, so that $V(1) = v_s$, and let $A(t)$ be the reward at stage t. It is convenient to imagine that termination is merely a transition to an absorbing state 0 with 0 reward, and to define $v_0 = 0$, so that (4.2-4) holds even for $k = 0$. With these definitions, (4.2-4) can be restated as

$$E\big(A(t) + V(t+1) - V(t)\big|S(t) = k, J(t) = j\big) \geq 0; \; j \text{ in } G_k, k = 0, \ldots, K. \tag{4.2-5}$$

By averaging over $S(t)$ and $J(t)$, it follows from (4.2-5) and the Conditional Expectation Theorem that

$$E(A(t) + V(t+1) - V(t)) \geq 0 \quad \text{for} \quad t \geq 1. \tag{4.2-6}$$

Since $E(M)$ is finite, (4.2-6) can be summed from $t = 1, \ldots, M$. The result is

$$E(A(1) + \ldots + A(M) + V(M+1) - V(1)) \geq 0. \tag{4.2-7}$$

But $A(1) + \ldots + A(M) = X$, $V(M+1) = v_0 = 0$, and $V(1) = v_s$. The conclusion follows. §§§

The following theorem gives sufficient conditions for the finite play assumption of Theorem 4.2-1 to be true.

Theorem 4.2-2 For all states k let $\mathbf{x}(k)$ be a probability distribution over the rows of G_k, and let $\mathbf{x} \equiv (\mathbf{x}(1), \ldots, \mathbf{x}(K))$ be the stationary strategy of player 1. Suppose that there is a vector $\mathbf{u}$ such that

$$1 + \sum_i x_i(k) \left(\sum_l p_{ij}^{kl} u_l - u_k \right) \leq 0; j \text{ in } G_k, k = 1, \ldots, K. \quad (4.2\text{-}8)$$

Then, regardless of the strategy employed by player 2, $E(M) \leq u_s$ when the game starts in state s.

Proof Let termination lead to an absorbing state 0, and define $u_0 \equiv 0$. Let $I(t)$ be 0 if $S(t) = 0$, or otherwise 1. Random variable $I(t)$ indicates whether the game continues at stage t. Also let $M(n) \equiv I(1) + \ldots + I(n)$, and let $U(t) \equiv u_{S(t)}$. Equation (4.2-8) implies that

$$I(t) + E\big(U(t+1) - U(t) \big| S(t) = k, J(t) = j\big) \leq 0; \;\; j \text{ in } G_k, k = 1, \ldots, K, t > 0.$$

This also holds if $k = 0$, since in that case $I(t) = S(t) = U(t) = U(t+1) = 0$. By averaging over $S(t)$ and $J(t)$, we obtain, regardless of the strategy employed by player 2,

$$E(I(t) + U(t+1) - U(t)) \leq 0 \text{ for } t \geq 1. \quad (4.2\text{-}9)$$

By summing (4.2-9) from $t = 1$ to n and canceling terms, we have

$$E(M(n) + U(n+1) - U(1)) \leq 0. \quad (4.2\text{-}10)$$

There are only finitely many states, so $|U(n+1) - U(1)|$ is bounded. Therefore $M(n)$ is a nondecreasing sequence of random variables with bounded mean, and the Dominated Convergence Theorem implies that $M \equiv \lim_{n \to \infty} M(n)$ exists and is finite with probability one. The result of summing (4.2-9) from $t = 1$ to M is then

$$E(M + U(M+1) - U(1)) \leq 0. \quad (4.2\text{-}11)$$

But $U(M+1) = 0$ and $U(1) = u_s$. The conclusion follows. §§§

For completeness, and also because these sufficiency theorems underlie the results in the rest of this chapter, we also state without proof the parallel results for player 2:

Theorem 4.2-3 Let $\mathbf{y}(k)$ be a probability distribution over the columns of G_k, and let $\mathbf{y} \equiv (\mathbf{y}(1), \ldots, \mathbf{y}(K))$. If there are vectors $\mathbf{w}$ and $\mathbf{z}$ such that

$$\sum_j \left(a_{ij}(k) + \sum_l p_{ij}^{kl} w_l - w_k \right) y_j(k) \leq 0; \quad i \text{ in } G_k, \quad k = 1, \ldots, K, \text{and} \quad (4.2\text{-}12)$$

$$1 + \sum_j \left(\sum_l p_{ij}^{kl} z_l - z_k \right) y_j(k) \le 0; \quad i \text{ in } G_k, \quad k = 1, \ldots, K, \tag{4.2-13}$$

then, by using the stationary strategy $\mathbf{y}$, player 2 can guarantee $E(M) \le z_s$ and $E(X) \le w_s$ as long as the game starts in state s. §§§

Any solution $\mathbf{v}$ of (4.2-3), together with the associated stationary strategies $\mathbf{x}$ and $\mathbf{y}$, must solve (4.2-4) and, with $\mathbf{w} = \mathbf{v}$, (4.2-12). Provided vectors $\mathbf{u}$ and $\mathbf{z}$ can be found that satisfy (4.2-8) and (4.2-13), the three theorems above imply that $\mathbf{v}$ is the vector value of the game and that $\mathbf{x}$ and $\mathbf{y}$ can be called $\mathbf{x}^*$ and $\mathbf{y}^*$, the optimal stationary mixed strategies for the two players. Equivalently $\mathbf{x}$, in addition to guaranteeing $\mathbf{v}$, must also guarantee that the average number of stages is finite regardless of the strategy employed by player 2, and similarly for $\mathbf{y}$. If it is for some reason obvious that finite play is inevitable, then it is not necessary to define $\mathbf{u}$ or $\mathbf{z}$ or employ (4.2-8) or (4.2-13).

In the Inspection Game of Sect. 4.1, we already know that v, $\mathbf{x}$, $\mathbf{y}$ solves (4.2-4) and (4.2-12). It remains only to find numbers (rather than vectors, since there is only one state) u and z satisfying (4.2-8) and (4.2-13). Formula (4.2-8) requires that $1 - u \le 0$ and $1 + x_2 \beta u - u \le 0$, both of which are satisfied by $u = 1/(1 - \beta)$. Similarly $z = 1/(1 - \beta)$ solves both inequalities of (4.2-13). As long as $\beta < 1$, the quantities calculated in Sect. 4.1 are therefore the solution of the Inspection Game. When $\beta = 1$, (4.2-8) has no finite solution because $x_2 = 1$ when $\beta = 1$. For $\beta = 1$, the Inspection Game has no solution in the sense of our definition. Dresher's game of Sect. 4.1 is simpler—it always has a solution because $M \le n$ no matter what the players do.

Example Consider the Bankruptcy Game where at each stage a game element is played to determine how much money player 2 pays player 1, with the ultimate winner being whoever can bankrupt his opponent. The sum of the two players' wealth is constant, so the wealth of player 1 can be the state of the game. Let G_i be the game where player 1's wealth is i, with $val(G_i)$ being the probability that player 1 ultimately wins. The typical game element shown in Fig. 4.2 is a guessing game where player 1's guess controls the stakes, which are 1 in the first row or 2 in the second. Player 1's wealth goes up if his strategy matches player 2's, or otherwise goes down. There is no upper bound on the length of play.

Fig. 4.2 The Bankruptcy Game where player 1 has n units

$$G_n = \begin{bmatrix} G_{n+1} & G_{n-1} \\ G_{n-2} & G_{n+2} \end{bmatrix}$$

If we assume that the total wealth involved is 4 units, then G_0 and G_{-1} can be replaced by 0, and G_4 and G_5 can be replaced by 1. The game thus consists of three game elements where player 1's wealth is 1, 2, or 3. Assuming no saddle points, (4.2-3) consists of the three equations $v_1 = v_2 v_3/(v_2 + v_3)$, $v_2 = v_3/(1 + v_3 - v_1)$, and $v_3 = (1 - v_1 v_2)/(2 - v_1 - v_2)$. A solution of these three equations is $\mathbf{v} = \left(1 - \sqrt{2}/2, 1/2, \sqrt{2}/2\right) = (0.293, 0.500, 0.707)$, which together with the associated optimal mixed strategies solves (4.2-4) and, with $\mathbf{w} = \mathbf{v}$, (4.2-12). The optimal mixed strategy of player 2 is $\mathbf{y}^*(1) = (0.586, 0.414)$, $\mathbf{y}^*(2) = (0.5, 0.5)$, and $\mathbf{y}^*(3) = (0.414, 0.586)$. Intuitively, the game must end when player 2 uses this mixed strategy because the only instance where player 1 can avoid the risk of the game ending is in G_2, where he could avoid the risk by using the first row, but in that case the next element will be G_1 or G_3, where there is a risk of terminating no matter what he does. Formally, the six inequalities of (4.2-13) are

$$\begin{array}{ll} 1 + .586z_2 - z_1 \leq 0 & 1 + .414z_3 - z_1 \leq 0 \\ 1 + .5z_3 + .5z_1 - z_2 \leq 0 & 1 - z_2 \leq 0 \\ 1 + .586z_2 - z_3 \leq 0 & 1 + .414z_1 - z_3 \leq 0 \end{array}$$

All six are satisfied by (say) $z = (10, 12, 10)$, so player 2's strategy guarantees finite play. It can also be easily checked that player 1's strategy guarantees finite play, so $\mathbf{v}$ is the value of this Bankruptcy Game.

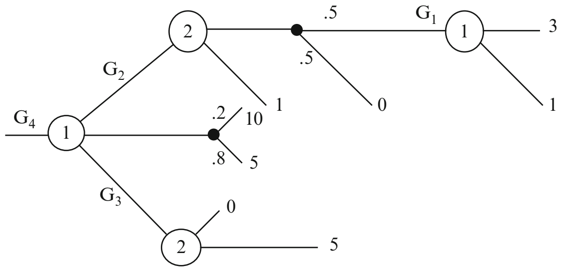
In addition to justifying the use of (4.2-3), Theorems 4.2-1, 4.2-2, and 4.2-3 lead naturally to a mathematical programming method for solving Markov games. First, note that any solution of (4.2-4) and (4.2-8) will work for all initial states s, since s plays no distinguished role in the equations. It is therefore natural to consider the mathematical program "maximize $v_1 + \ldots + v_K$ subject to (4.2-4)" for player 1, and to the corresponding "minimize $w_1 + \ldots + w_K$ subject to (4.2-12)" for player 2. If the optimal value vectors of the two programs are equal, and if in addition vectors $\mathbf{u}$ and $\mathbf{z}$ can be found that satisfy (4.2-8) and (4.2-13), then the game is solved. There is no guarantee that this will happen because neither mathematical program is a linear program ((4.2-4) includes products of $\mathbf{x}$- and $\mathbf{v}$-variables, for example), so there is no duality theorem that guarantees that the two objective functions will be equal. Nonetheless, the procedure fails safely, and often succeeds in solving the game.

The next several sections discuss special classes of Markov games. Tree games (Sect. 4.3) and their generalization exhaustive games (Sect. 4.4) force finite play through their structure. Since finite play is guaranteed for all strategies, there is no need to check (4.2-8) or (4.2-13) once a solution of (4.2-3) is obtained. The length of play has no upper bound in Stochastic games (Sect. 4.5), but an upper bound on the average length of play (which is all that is required) can be established. Once this bound is established, it is again not necessary to check (4.2-8) and (4.2-13). In general, the procedure outlined above is only necessary for games such as the Bankruptcy Game where infinite play is possible for some strategy pairs, but not when player 1 uses $\mathbf{x}^*$ or player 2 uses $\mathbf{y}^*$.

The restriction to finitely many game elements in Markov games is essential. Gale and Stewart (1953) give an example of a Tree game with infinitely many game elements for which there is no solution.

4.3 Tree Games

Figure 4.3-1 shows the extensive form of a Tree game and an equivalent description as a Markov game. In general, the extensive form of a Tree game is simply a tree with a root, three kinds of nodes, and leaves that are marked with player 1's payoff.



$$G_1 = \begin{bmatrix} 3 \\ 1 \end{bmatrix} \qquad G_2 = [1 \quad .5G_1] \qquad G_3 = [\ 0 \quad 5\ ] \qquad G_4 = \begin{bmatrix} G_2 \\ 6 \\ G_3 \end{bmatrix}$$

Fig. 4.3-1 Tree game example

Play begins at the root, and continues until one of the leaves is encountered. Nodes where player 1 or 2 chooses the exit branch are so labeled, and nodes where the exit branch is randomly chosen are labeled with a solid dot, with a probability distribution written on the branches. At each node at most one of the two players has a choice to make. Markov games may generally include game elements where both players simultaneously make decisions, but neither simultaneous moves nor information sets are permitted in tree games. Tree games are sometimes called games of "perfect information", since each player is perfectly aware of where he is in the tree when it comes time to make a choice.

The branch labeled G_1 in Fig. 4.3-1 leads to a choice by player 1 of whether he would prefer a payoff of 1 or 3. He would prefer 3, of course, so G_1, considered as a game by itself, has value $v_1 = 3$. The branch labeled G_2 leads to a choice by player 2 of whether he would prefer 1 or a 50/50 gamble at either 0 or G_1. The gamble is equivalent to $0.5(0) + 0.5(v_1) = 1.5$, so player 2 should not gamble and consequently $v_2 = 1$. Continuing in this manner, working by "back substitution" from the end to the beginning, repeatedly employing the three reductions shown in Fig. 4.3-2, the value of the game can eventually be determined to be $v_4 = 6$. The game has a saddle point that is determined in the process of solving it.

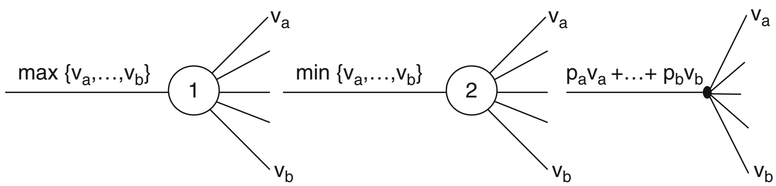


Fig. 4.3-2 Reductions for Tree games

Any finite Tree game has a saddle point that can be found by back substitution, using the reductions shown in Fig. 4.3-2. A proof of this statement could be made by induction, since any reduction of a terminal node results in a shorter game. Note that the normal form of the tree shown in Fig. 4.3-1 would be a 6×4 matrix, so the back substitution technique somehow avoids consideration of a lot of bad strategies in finding the saddle point. Skipping all these strategies is the source of the technique's efficiency. Analysis of Tree games has a history that antedates that of games in general. Well before Theorem 3.2-1 was known to be true, Zermelo (1912) showed that finite Tree games with only three possible payoffs ("win", "lose", and "tie") are solvable.

Figure 4.3-3 shows the beginning of a representation of Tic Tac Toe as a Tree game, using symmetry to avoid unnecessary branching. The X player can choose one of three essentially different places for his first mark, so there are only three branches from the empty playing board. The "state" is just the positions of all marks made so far, with the player to move being obvious from the marks. Note that there are two branches leading into the only state shown with three marks, so strictly speaking the completed diagram will not be a tree. The tree representation could be preserved by replicating that state at the end of each of the two branches, but clearly there is no point in expanding the size of the network merely to preserve the form of a tree.

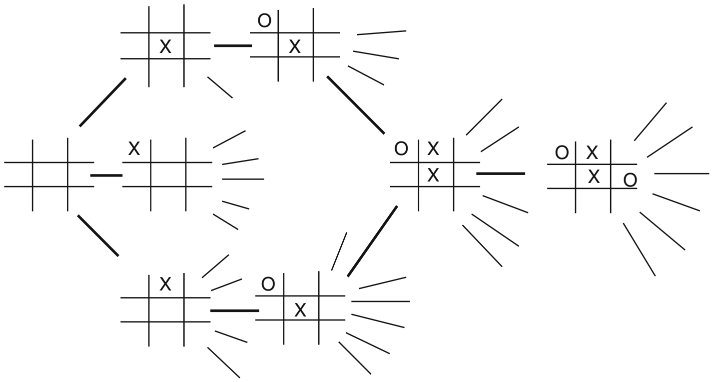


Fig. 4.3-3 A partial graph of Tic Tac Toe. The state moves from left to right as the game is played. Back substitution moves from right to left

Back substitution is still possible in Tic Tac Toe because play always proceeds to the right. In principle, one could begin by solving all states with nine marks, then (since states with 8 marks always lead to states with 9 marks) all states with 8 marks, etc. In practice, the value of some states is obvious without writing out all succeeding states. The value of the rightmost state shown is "win" for example, because player 1 (the X player) can complete an X column on the next move.

With a large sheet of paper and considerable patience, Tic Tac Toe can be solved by back substitution. Figure 4.3-4 partially shows an optimal strategy for the X player, again taking advantage of symmetry to represent the strategy compactly. The number of marks is always odd in Fig. 4.3-4; branches correspond to moves for player 2, and every route that player 2 chooses through the network results in either "win" or "tie"(routes that lead to "win" are not shown). A similar diagram could show a strategy for player 2 where all routes chosen by player 1 lead to either "lose" or "tie", so the value of the game is "tie".

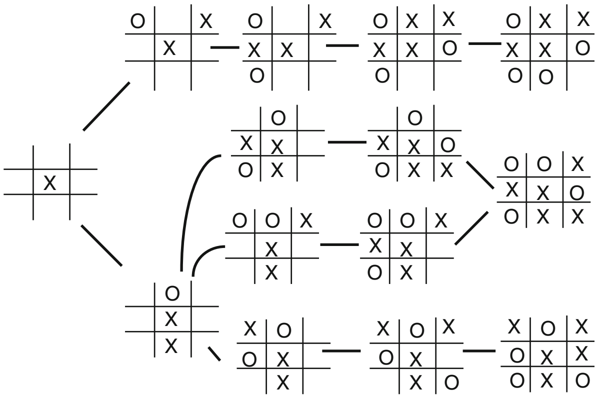


Fig. 4.3-4 An optimal strategy for the *X* player in Tic Tac Toe

Tic Tac Toe could also be represented in normal form as a very large matrix game, with the above pair of strategies being a saddle point. The strategy represented by Fig. 4.3-4 would be a single row in that payoff matrix. Constructing the normal form of a Tree game is pointless because back substitution is a more efficient solution technique, but it could be done—the concept of "solution" is the same for Tree games as for matrix games.

The important thing about Fig. 4.3-3 is that it is a finite, acyclic, directed graph where at each node either player 1 alone or player 2 alone or chance determines the exit branch, and where each exit branch leads either to another node or to a payoff. By replicating nodes, all such games can in principle be represented as Tree games, and are therefore solvable by back substitution. The acyclic restriction is important, since it guarantees that no state can ever repeat, and therefore that the number of stages can never exceed the number of states.

Tree games have been described above as games where all payoffs are terminal, but that restriction is not essential. The back substitution method is easily adapted to games where there are additive payoffs associated with the branches. Indeed, since there is a unique path from the root to every leaf, any intermediate payoffs could always be summed and collected on the leaves.

4.3.1 Combinatorial Games, Nim

The game of Nim involves several heaps of stones. The players alternate in removing stones from heaps, and the winner is the player who removes the last stone. The rules for stone removal are that at least one stone must be removed in each turn, and that stones can only be removed from a single heap. It is permissible to remove an entire heap. If n_k is the positive number of stones in heap k, then $\mathbf{n} = (n_1, \ldots, n_K)$, together with a flag for whose move it is, is the state of the game. Nim is an example of a Tree game that can be solved without employing back substitution.

There is a particular function (see exercise 17) NimSum($\mathbf{n}$) that maps the state of Nim onto the nonnegative integers. This function is the basis of an optimal strategy for playing Nim. The important facts about NimSum($\mathbf{n}$) are that

1. NimSum($\mathbf{0}$) $= 0$,
2. If NimSum($\mathbf{n}$) > 0, there is always a move to a state $\mathbf{n}'$ where NimSum($\mathbf{n}'$) $= 0$,
3. If NimSum($\mathbf{n}$) $= 0$, there is no move to a state $\mathbf{n}'$ where NimSum($\mathbf{n}'$) $= 0$.

If you think about it, you can win whenever NimSum($\mathbf{n}$) is positive and it is your move—just make any move that reduces NimSum($\mathbf{n}$) to 0. If the resulting state is 0, you win immediately. If not, your opponent will have to make NimSum($\mathbf{n}$) nonzero again, so you will win later. If you happen to start from a state where NimSum($\mathbf{n}$) is 0, then all you can do is hope that your opponent makes a mistake, since he of course has the same options. Once the NimSum($\mathbf{n}$) function is discovered, the optimal strategy for playing Nim becomes immediately apparent.

Nim is an example of a Combinatorial game, the distinguishing features being that there are no random moves, the first player who is unable to move is the loser, and play is finite. Our solution of Nim has been by the bare hands technique of guessing the optimal solution. Combinatorial games are not all as easily solved as Nim, but this subclass of Tree games does have some exploitable properties. There is a book (Guy, 1991) devoted to the topic.

4.4 Exhaustive Games

Exhaustive games are Markov games in which no state can occur more than once in play. Tree games are a subset with no simultaneous moves. It turns out that back substitution also works in this larger class.

All exhaustive games must have at least one ordinary game element G that doesn't involve any of the others. That game element can be solved first and its value substituted wherever it occurs in the others, treating the "+" of Equation (4.2-1) as ordinary addition when substituting $val(G)$ for G. A proof of this can be constructed by contradiction. If the initial game element (call it G_1) is not ordinary, then transition to a different game element G_2 must be possible. If G_2 is not ordinary, transition to a game element G_3 must be possible, and furthermore G_3 can be neither G_2 nor G_1 because no game element can repeat. Continuing in this manner we must eventually come either to an ordinary game element or the last game element, and in the latter case the last game element must necessarily be ordinary because it can involve neither itself nor any of the others. Furthermore, once the value of this ordinary game element is obtained and substituted in the others, the "back substitution" process can be repeated until all game elements have been solved. We formalize our conclusions in

Theorem 4.4 All exhaustive games are solvable by back substitution. §§§

Figure 4.4-1 shows a small example of an exhaustive game. The values of the four elements must be computed in the order $v_3 = -0.5, v_4 = 2, v_2 = 0,$ and $v_1 = 1.75,$ with all game elements except G_3 having saddle points.

$$G_1 = \begin{bmatrix} 2+.5G_3 & 6 \\ 1 & G_2 \end{bmatrix} \qquad G_2 = \begin{bmatrix} G_3 & 0 \\ 0 & G_4 \end{bmatrix} \qquad G_3 = \begin{bmatrix} -1 & 0 \\ 0 & -1 \end{bmatrix} \qquad G_4 = \begin{bmatrix} 2 \\ G_3 \end{bmatrix}$$

Fig. 4.4-1 Exhaustive game example

4.4.1 Women and Cats Versus Mice and Men

This is a whimsical example of an exhaustive game due to Blackwell (1954). Player 1 is initially given two women and two cats, while player 2 is given two mice and two men. Each player secretly selects one of his animals to fight a battle, one being eliminated according to the circular rule "woman eliminates man eliminates cat eliminates mouse eliminates woman". The survivors then fight another battle until one side (the loser) is wiped out. The state of the game is $\mathbf{s} = (w, c, r, m)$, with w being the number of women remaining, etc. (r is for "rats", an animal whose name doesn't begin with the same letter as "men" and which the male author has found to be more effective in this role anyway). Strategy choices are made simultaneously in secret, so this exhaustive game is not a Tree game. Figure 4.4-2 shows the normal form of the typical game element, except that some of the payoffs should be 1 if all mice or all men are eliminated, or 0 if all women or all cats are eliminated.

$$G(w,c,r,m) = \begin{array}{r c} & \begin{array}{cc} \text{mouse} & \text{man} \end{array} \\ \begin{array}{r} \text{woman} \\ \text{cat} \end{array} & \begin{bmatrix} G(w-1,c,r,m) & G(w,c,r,m-1) \\ G(w,c,r-1,m) & G(w,c-1,r,m) \end{bmatrix} \end{array}$$

Fig. 4.4-2 Women and Cats Versus Mice and Men

Let $v(\mathbf{s}) = val(G(\mathbf{s}))$, and suppose our goal is to find $v(2, 1, 1, 1)$ and its accompanying optimal strategies. All game elements where one component is 0 are trivial, and clearly $v(1, 1, 1, 1) = 0.5$ by symmetry. We therefore have

$$\begin{aligned} v(2,1,1,1) &= val\left(\begin{bmatrix} v(1,1,1,1) & v(2,1,1,0) \\ v(2,1,0,1) & v(2,0,1,1) \end{bmatrix}\right) \\ &= val\left(\begin{bmatrix} 0.5 & 1 \\ 1 & 0 \end{bmatrix}\right) = 2/3 \end{aligned} \tag{4.4-1}$$

with player 1 entering a woman with probability 2/3 and player 2 entering a mouse with probability 2/3. All game elements with 5 or fewer animals remaining can be solved similarly, after which we can consider game elements with 6 animals, etc. In the process of solving any of these game elements, we will have also found the optimal way to play all possible successors.

4.4.2 Attrition Battles

Again we consider a situation where two sides fight with each other until one side is completely eliminated, but the mechanics here are different and the game is not whimsical. We will refer to player 1 as Blue and player 2 as Red, since that is customary in modeling battles. Let **x** be an m-vector of nonnegative integers, where x_i is the number of Blue units of type i. If Blue had tanks, squads, and helicopters, then m would be 3. Also let **y** be an n-vector with similar meaning for Red. The battle proceeds in continuous time until a unit is eliminated on one side or the other, which event terminates the game element. The next game element has one less unit on the side that suffered the loss, and the battle continues until one side or the other is completely eliminated. As long as resupply of units is impossible, the game is exhaustive.

Let $P(\mathbf{x},\mathbf{y})$ be the probability that Red is eventually eliminated in the game that begins with vectors **x** and **y**, given optimal play on both sides. $P(\mathbf{0}, \mathbf{0})$ is not defined, $P(\mathbf{x}, \mathbf{0})$ is 1 as long as **x** is not **0** and $P(\mathbf{0}, \mathbf{y})$ is 0 as long as **y** is not **0**. Those statements cover all of the ordinary game elements. The rest of the game elements depend on the attrition process. To describe it, let λ_j(or μ_i) be the rate per unit time at which a Red unit of type j (or a Blue unit of type i)is eliminated, which parameters we take to be given for the moment, and let

$$\tau \equiv \left(\sum_{i=1}^{m} \mu_i + \sum_{j=1}^{n} \lambda_j \right)^{-1} \tag{4.4-2}$$

The interpretation of λ_j is that, in a small interval of time of length δ, the probability that a Red unit of type j is eliminated is $\delta\lambda_j$, and similarly for μ_i. With those interpretations, τ is the average length of time until some unit is lost and $\tau\lambda_j$ (or $\tau\mu_i$) is the probability that the next game element will have one less type j unit (or type i unit). As a result, the following formula connects the value of game element $(\mathbf{x}, \mathbf{y})$ with the values of its successors:

$$P(\mathbf{x},\mathbf{y}) = \tau \left(\sum_{j=1}^{n} \lambda_j P(\mathbf{x},\mathbf{y}\backslash j) + \sum_{i=1}^{m} \mu_i P(\mathbf{x}\backslash i,\mathbf{y}) \right) \tag{4.4-3}$$

where the $\backslash\, k$ suffix means “minus one unit of type k”.

So far we have merely described a continuous time Markov process where neither player has any control of events, rather than a Markov game. Our derivation of (4.4-3) has been cursory because it has little to do with game theory—a more careful derivation can be found in Isbell and Marlow (1956). We now turn to a Markov game that starts with (4.4-3) but allows the players to control the attrition rates by deciding who will shoot at whom. The given attrition parameters for Blue are λ_{ij}, the rate at which one type i Blue unit kills type j Red units given that the type i unit chooses to shoot at type j units. Each Blue unit is free to shoot at any Red unit, so let u_{ij} be 1 if type i units shoot at type j units, or otherwise 0. Blue's strategy is represented by $\mathbf{u}$. We then have $\lambda_j = \sum_{i=1}^{m} x_i u_{ij} \lambda_{ij}$ for all j, plus the constraint that $\sum_{j=1}^{n} u_{ij} = 1$ for all i. Similarly let μ_{ij} be the given attrition rates of Red type j against Blue type i, and let v_{ij} be Red's assignments. Then $\mu_i = \sum_{j=1}^{n} y_j \mu_{ij} v_{ij}$, and we must have $\sum_{i=1}^{m} v_{ij} = 1$ for all j. Red's strategy is represented by $\mathbf{v}$, and both $\mathbf{u}$ and $\mathbf{v}$ can depend on the game element. We can now enumerate all of the strategies available to the two sides and solve each game element in normal form. This is a game of exhaustion because one unit is lost at every stage, so it can be solved using back substitution.

Example Suppose that Blue consists of two units labeled a and b, while Red consists of two units labeled c and d. Take the attrition rates to be $\boldsymbol{\lambda} = \begin{bmatrix} 9 & 1 \\ 2 & 1 \end{bmatrix}$ and $\boldsymbol{\mu} = \begin{bmatrix} 2 & 3 \\ 1 & 2 \end{bmatrix}$, with a and c labeling the first row and the first column, respectively. Note that unit a dominates unit b in terms of lethality, particularly against unit c, and that unit d dominates unit c (compare the columns of $\boldsymbol{\mu}$). Our goal is to solve the 2-on-2 game, but we must first begin with smaller games. The 2-on-2 game will lead to either a 2-on-1 game or a 1-on-2 game, and each of those will either lead to termination or a 1-on-1 game. We must begin by solving all four of the 1-on-1 games. The four values are $\begin{bmatrix} 0.818 & 0.250 \\ 0.667 & 0.333 \end{bmatrix}$, with rows and columns labeled as in $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$. The most favorable game for Blue is when a faces c; when a faces d, Blue wins only one time in four. Having solved these four game elements, we can next solve the 2-on-1 and 1-on-2 elements using 4.4-3 and back substitution. Each of those elements involves a decision on the part of the lone unit about the order in which he should engage his two opponents. After solving all four of those, we can finally consider the 2-on-2 element, which takes the form of the 4×4 matrix in Fig. 4.4-3:

Fig. 4.4-3 "$x > y, u > v$" means that x shoots at y and u shoots at v

$$\begin{array}{c} \\ a>c,\ b>c \\ a>c,\ b>d \\ a>d,\ b>c \\ a>d,\ b>d \end{array} \begin{array}{c} \begin{array}{cccc} c>a,\ d>a & c>a,\ d>b & c>b,\ d>a & c>b,\ d>b \end{array} \\ \begin{bmatrix} 0.464583 & 0.483651 & 0.483254 & 0.505867 \\ 0.478803 & 0.500249 & 0.499823 & 0.525451 \\ 0.416346 & 0.450314 & 0.449464 & 0.496596 \\ 0.39011 & 0.425366 & 0.424374 & 0.475916 \end{bmatrix} \end{array}$$

There is a saddle point in the second row (where a shoots at c and b shoots at d) and the first column (where both Red units shoot at a), so the value of the game element shown in Fig. 4.4-3 is 0.48. Red has a slight advantage because unit a's effectiveness against c does not compensate for the fact that unit d is reasonably effective and very hard to kill. Further detail, including the values of the 2-on-1 and 1-on-2 elements, can be found on page "Battle42" of the TPZS workbook. There you can explore the effects of different attrition rates $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$ on tactics and values.

The example given above is a small one. There is some good news for those who would like to solve larger games:

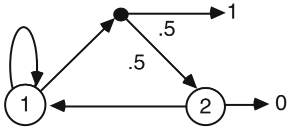
- Every game element has a saddle point, so it is never necessary to consider mixed strategies. The reason for this is that each game element takes the form of a Race time game (see Sect. 9.1).
- If there are multiple units of the same type, then all of them should shoot at the same enemy type, a fact that greatly reduces the number of strategies that have to be considered. This is shown in Kikuta (1986). We have implicitly assumed this in writing the formulas for λ_j and μ_i above.

Unfortunately there is also some bad news, that being the PEG. If all units are distinct, then Blue has n^m strategies and Red has m^n strategies, numbers that grow very quickly as the size of the game increases, and remember that all smaller games need to be solved before an m-on-n game can even be considered. Kikuta (1986) gives some special cases where not all of these strategies need to be enumerated, but there seems to be no simple, general procedure for solving these games. Even the case where $m = 1$ is nontrivial, since it is not obvious whether the lone Blue unit should first shoot at a Red unit that is easy to kill, or at a Red unit that is dangerous (Friedman, 1977; Kikuta, 1983).

4.5 Graphical and Recursive Games

Recursive games are Markov games where only terminal payoffs are permitted. In the notation of Sect. 4.2, either $q_{ij}(k) = 0$ or $a_{ij}(k) = 0$ for all i, j, k. Except for the game in Fig. 4.4-1, which is not recursive because of G_1, all examples considered so far in this chapter are Recursive. Graphical games are a special case where in each state only one player has a choice, a feature that makes a graphical representation possible. Figure 4.5-1 shows an example in both graphical and normal form, using the convention that G_1 (G_2) is the game where player 1 (player 2) has the choice. Graphical games can also be regarded as a generalization of Tree games where the acyclic restriction is removed. The freedom permitted by the generalization has a price; as the loop connecting G_1 to itself in Fig. 4.5-1 makes clear, non-terminating play is possible. The convention in all Recursive games is that the payoff for non-terminating play is zero. This convention is not a significant restriction to application because addition of a constant to all the payoffs of any game is strategically neutral, but the convention should nonetheless be borne in mind when determining other payoffs. For the game in Fig. 4.5-1 we have $val(G_1) = 0.5$ and $val(G_2) = 0$. Player 1 is motivated to force termination, so the theoretical possibility of non-termination does not arise in optimal play. Change the "1" payoff to a "−1" payoff, however, and player 1 would never leave state 1.

Fig. 4.5-1 Example of a Graphical game



$$G_1 = \begin{bmatrix} G_1 \\ .5+.5G_2 \end{bmatrix}$$

$$G_2 = \begin{bmatrix} 0 & G_1 \end{bmatrix}$$

Most parlor games that involve a playing board (Chess, Checkers, Backgammon) can be represented as Graphical games. They are not usually exhaustive games because of the possibility that a state (configuration of pieces on the board) might repeat. In practice, the possibility that play of the game might not terminate is taken care of by modifying the rules so that only a fixed number of moves (or state repetitions) is permissible before declaring the outcome to be a "tie". The modification effectively prevents a player who is involved in an infinitely repeating situation from converting "tie" to victory by continually refusing his opponent's "let's call it a tie" offers, a concession to human patience. However, limiting the number of moves converts the Graphical game into a much larger Tree game by forcing "number of moves remaining" to be part of the state description. The tree feature at first seems attractive because back substitution becomes a feasible solution technique, but in practice the size of the tree is often awkwardly large. Modifying the game to simply prohibit state repetitions introduces theoretical difficulties of a different sort (Pultr and Morris (1984)). For theoretical purposes, it is usually best to accept the possibility of infinite play.

Graphical games with no chance moves (Checkers and Chess, but not Backgammon) are called Deterministic Graphical games, and have a solution procedure that is not much more complicated than back substitution (Washburn (1990)). It is not known whether (unmodified) Chess terminates when played optimally. If it does terminate, however, then at least one of the players has an optimal strategy for which state repetition is impossible. Chess as a Deterministic Graphical game is hopelessly large (recall Fig. 4.3-3 for Tic Tac Toe!), but there are interesting smaller games of the same type whose solutions are known (exercise 10). Being Markov games, Deterministic Graphical games can have only a finite number of states. Baston and Bostok (1993) consider the implications of allowing the number of states to be countably infinite.

Deterministic Graphical games always have saddle points, but Recursive games in general may not. In some cases Recursive games are not solvable at all in the usual sense. An example of this is the Inspection Game of Fig. 4.1 when $\beta = 1$. Equation (4.1-1) has a unique solution when $\beta = 1$, namely $v = 1$. However, the second row of Fig. 4.1 is dominant when $v = 1$, so apparently the Agent can guarantee to commit his act unnoticed while choosing the "wait" option at every opportunity. The trouble is that there is a great deal of difference between taking a very long time to act, which is wise because it makes it difficult for player 2 to guess exactly when the action will occur, and taking "forever" to act, which makes the payoff 0 because the game never terminates. There is a "discontinuity at infinity". In spite of our intuitive feeling that the value of the game ought to be 1, in a formal sense there is no solution.

On account of such difficulties, it has become customary to say that the value of a Recursive game element is v as long as each player can guarantee values arbitrarily close to v, even if v itself is unattainable. If the Agent in the Inspection Game with $\beta = 1$ uses the mixed strategy $(\varepsilon, 1 - \varepsilon)$, he can guarantee $1 - \varepsilon$ against "inspect" and, given that the game ends, 1 against "wait". Since the game will certainly end as long as $\varepsilon > 0$, the Agent can guarantee any payoff that is strictly smaller than 1, and we therefore say that the value of the game is 1. Everett (1957) shows that, in this modified sense, every Recursive game has a solution.

Except for Graphical games, there is no simple way to separate Recursive games where values can actually be guaranteed from Recursive games where values can almost be guaranteed. It is common, however, for the latter type to arise when "wait" is an option for one or both of the players, as in the Inspection Game.

Example Assume that Colonel Blotto outnumbers the enemy by 2 units to 1, and is therefore capable of capturing the enemy's camp because he is in the majority. The difficulty is that the enemy might use his one unit to attack Blotto's camp, thereby capturing it if Blotto is indeed using both of his units to attack the enemy. Blotto wants to know if there is any way to capture the enemy's camp without risking the loss of his own. There is no time constraint, so the natural model is a Recursive game G where the payoff is 1 if Blotto's (player 1's) goal is eventually achieved, else 0.

Fig. 4.5-2 Colonel Blotto's dilemma

$$G = \begin{array}{c c} & \begin{array}{cc} 0 & 1 \end{array} \\ \begin{array}{c} 0 \\ 1 \\ 2 \end{array} & \begin{bmatrix} G & G \\ G & 1 \\ 1 & 0 \end{bmatrix} \end{array}$$

Strategies in Fig. 4.5-2 are named for the number of units chosen for attack, and the payoff matrix reflects the assumption that attackers simply retire unless they outnumber the defenders. For example the payoff in row 0, column 1 is G because Blotto does not attack, and his two units are strong enough to cause his opponent to retire, thus repeating the game. Blotto's strategies 0 and 1 are both essentially "wait", since he can't lose in either case, but strategy 1 is intuitively preferable because it provides at least the possibility of a win. A human player might choose strategy 1 several days in a row, and then slip in a choice of strategy 2 if player 2 consistently chooses his strategy 0. Blotto's nearly optimal mixed strategy imitates this, choosing strategy 1 with probability $1 - \varepsilon$, and strategy 2 with probability ε. The game will terminate with probability 1 as long as $\varepsilon > 0$, and in that case Blotto's mixed strategy guarantees at least $1 - \varepsilon$ against either column. The value of G is therefore 1—Blotto can accomplish his objective if he is patient enough. Note that $v = 1$ solves (4.2-3), but so does any number larger than 1.

The Recursive games considered in this section are all examples where the value equations of Theorem 4.2-1 are satisfied by some solution of (4.2-3), but where the finite play inequalities (4.2-8) and (4.2-13) cannot be satisfied simultaneously. It is this possibility that forces the explicit consideration of infinite play in Recursive games. There are also Recursive games where the number of stages is always finite under optimal play. The Bankruptcy Game of Sect. 4.2 is an example.

4.6 Stochastic Games

Recall from Sect. 4.1 that the payoff when row i and column j are chosen in state k of a Markov game is $a_{ij}(k) + \sum_l p_{ij}^{kl} G_l$. In a Stochastic game it is required that the continuation probability $q_{ij}(k) \equiv \sum_l p_{ij}^{kl}$ be smaller than 1 for all i, j, and k. Because there are only finitely many possibilities for i, j, and k, the maximum continuation probability $\beta \equiv \max_{ijk} q_{ij}(k)$ is also smaller than 1. The probability of ending is at least $1 - \beta$ in every game element, so a Stochastic game will surely end and there is no need to define a payoff for infinite play as in Recursive games. Stochastic games are nonetheless more difficult to solve than exhaustive games because there is no "last" game element, so the back substitution method has no place to start. In fact, the Inspection Game of Sect. 4.1 with $\beta = 0.5$ demonstrates that nothing resembling back substitution can possibly suffice, since the game's value is irrational in spite of the fact that all of its inputs are rational. No finite procedure using only the four basic operations of arithmetic (+, -, *, /) is capable of calculating such a value. The first question, then, is how can Stochastic games be solved or at least approximated?

Since a Stochastic game will always end, there would seem to be little harm in amending the rules so that the game terminates after some large number N of stages, awarding player 1 a final reward that depends on the final game element. If N is large enough, the value of the modified game should be near the value of the original game regardless of the final reward. This idea turns out to be correct, and leads to the "value iteration" method of solution. The modified game is an exhaustive game with state (n,k), where n is the maximum number of stages remaining to be played and k is the current game element. It can therefore be solved by back substitution. Let $\mathbf{v}(n) = (v_1(n), \ldots, v_K(n))$ be the vector value of a K-element game when there are n stages remaining. With $\mathbf{v}(0)$ being the final reward vector, the back substitution formula is, for $n \geq 0$ and all k,

$$v_k(n+1) = T_k(\mathbf{v}(n)), \tag{4.6-1}$$

where $T_k()$ is the value transformation defined in (4.2-2). By iterating (4.6-1) N times, the final approximation $v_k(N)$ is obtained, one value for each game element k. In the Inspection Game, (4.6-1) is (suppressing k because there is only one state),

$$v(n+1) = val\left(\begin{bmatrix} 0 & 1 \\ \beta & \beta v(n) \end{bmatrix}\right) = \beta/(1+\beta-\beta v(n)). \tag{4.6-2}$$

If $\beta = 0.5$ and $v(0) = 0$, the sequence $v(n)$ is 0, 1/3, 3/8, 8/21, 21/55, ... This sequence will never include the value of the Inspection Game because every element is rational. Nonetheless, 21/55 = 0.382 is correct to three decimal places. In this example, at least, the convergence of $v(n)$ to v appears to be rapid.

In order to make a general study of the behavior of $\mathbf{v}(n)$ for large n, it is useful to first establish some facts about the value transformation $\mathbf{T}()$ defined in (4.2-3). Let the norm $\|\mathbf{w}\|$ of an arbitrary vector be the largest absolute value of any of its components, so that $\|\mathbf{w}\|$ measures the closeness of $\mathbf{w}$ to the 0 vector. The following theorem establishes that for any two value vectors $\mathbf{u}$ and $\mathbf{w}$, $\mathbf{T}(\mathbf{w})$ is closer to $\mathbf{T}(\mathbf{u})$ than $\mathbf{w}$ is to $\mathbf{u}$.

Theorem 4.6 For all value vectors $\boldsymbol{u}$ and $\boldsymbol{w}$, $\|\mathbf{T}(\mathbf{w}) - \mathbf{T}(\mathbf{u})\| \leq \beta\|\mathbf{w} - \mathbf{u}\|$, where β is as defined earlier in this section.

Proof Define payoff matrix $\mathbf{W}_k \equiv (a_{ij}(k) + \sum_l p_{ij}^{kl} w_l)$, so that $T_k(\mathbf{w}) = val(\mathbf{W}_k)$. Similarly let $\mathbf{U}_k \equiv (a_{ij}(k) + \sum_l p_{ij}^{kl} u_l)$, and let $\Delta_k \equiv \max_{ij}|\sum_l p_{ij}^{kl}(w_l - u_l)|$. From Conclusion 2 of Theorem 3.2-2, $|T_k(\mathbf{w}) - T_k(\mathbf{u})| \leq \Delta_k$, and $\Delta_k \leq \max_{ij} \sum_l p_{ij}^{kl}\|\mathbf{w} - \mathbf{u}\| = \max_{ij} q_{ij}(k)\|\mathbf{w} - \mathbf{u}\| \leq \beta\|\mathbf{w} - \mathbf{u}\|$ for all k. Therefore $\max_k |T_k(\mathbf{w}) - T_k(\mathbf{u})| \leq \beta\|\mathbf{w} - \mathbf{u}\|$, which completes the proof. §§§

Theorem 4.6 states that $\mathbf{T}()$ is a contraction mapping. Since the normed value vectors are a complete metric space, $\mathbf{T}()$ must have a unique fixed point $\mathbf{v}$ that is the limit of the sequence $\mathbf{v}(1)$, $\mathbf{v}(2)$,..., and furthermore $\mathbf{v}$ must be close to $\mathbf{v}(n)$ in the sense that (see Shilov (1965))

$$\begin{aligned} \|\mathbf{v} - \mathbf{v}(n)\| &\leq (\beta/(1-\beta))\|\mathbf{v}(n) - \mathbf{v}(n-1)\| \\ &\leq (\beta^n/(1-\beta))\|\mathbf{v}(1) - \mathbf{v}(0)\|. \end{aligned} \tag{4.6-3}$$

If value iteration stops after computing $\mathbf{v}(n)$, (4.6-3) establishes a bound on the error in approximating $\mathbf{v}$. Evidently value iteration is especially useful when β is small, since in that case the right-hand side of (4.6-3) decreases rapidly. The fixed point $\mathbf{v}$, together with the associated strategies $\mathbf{x}$* and $\mathbf{y}$*, is the solution of the game in the sense offered in Sect. 4.2. Variables u_k and z_k in Theorems 4.2-2 and 4.2-3, respectively, can each be taken to be $1/(1-\beta)$.

Value iteration is often a useful technique even when $\beta = 1$. Consider the Markov game shown in Fig. 4.6-1. The continuation probabilities $q_{22}(1)$ and $q_{12}(2)$ are both 1, so $\beta = 1$. Even so, it should be clear that the game will eventually end, even if the players were to cooperate in trying to extend its length. Therefore it makes sense to apply value iteration. Arbitrarily letting $\mathbf{v}(0) = (3,2,1)$ to make the point that any initial guess is theoretically acceptable, the successive value vectors are (0.5, 1, 1.5), (0.56, 1, 0.25), (0.14, 1, 0.28), (0.15, 1, 0.07), (0.04, 1, 0.08),... Note that $v_2(n) = 1$ for $n \geq 1$ because G_2 has a saddle point regardless of the value of G_1. The continued shrinkage (somewhat jerky, but shrinkage nonetheless) of $v_1(n)$ and $v_3(n)$ might make one suspect that $v_1 = v_3 = 0$. If this were true, G_1 would have a saddle point at (1,2). It is not difficult to verify that each player can indeed force a 0 payoff in G_1 by playing his part of that saddle point. The vector (0,1,0) thus solves the value equation $\mathbf{v} = \mathbf{T}(\mathbf{v})$, and, since the game ends in finite time, Theorems 4.2-1, 4.2-2, and 4.2-3 can be used to show that (0,1,0) is the value of G. An odd fact is that both players use mixed strategies in the approximated first game element for all n, in spite of the fact that G_1 has a saddle point.

$$G_1 = \begin{bmatrix} 1+0.5G_2 & 0 \\ -1 & G_3 \end{bmatrix} \qquad G_2 = \begin{bmatrix} 0 & G_1 \\ 1 & 8 \end{bmatrix} \qquad G_3 = [\,0.5G_1\,]$$

Fig. 4.6-1 A Markov game that is not Stochastic because $\beta = 1$

Value iteration is one of several methods that have been proposed for solving Stochastic games. Another method is to solve the value Equation (4.2-3) directly for $\mathbf{v}$, as in the Inspection Game. However, direct solution of the value equation is often difficult because *val*() is not an analytically "nice" function. In 2×2 games, for example, *val*() has one form for games with saddle points and a completely different form otherwise. The difficulty is that it is usually not possible to predict *a priori* which strategies will be active (used with positive probability) because of dependence on the as yet unknown vector $\mathbf{v}$. The following hybrid procedure therefore suggests itself:

1. Guess $\mathbf{v}(0)$.
2. Iterate (4.6-1) until it is "obvious" which strategies are active.
3. Solve (4.2-3) with all non-active rows and columns crossed out.
4. Check that the solution actually solves (4.2-3). If not, go to Step 2.

The hybrid procedure was used to solve the game in Fig. 4.6-1. For another example, consider the 1×2 game $G = [1 + 0.5G,\ 1.9]$. If $v(0) = 0$, value iteration produces the sequence 1, 1.5, 1.75, 1.875 $= v(4)$, with player 2 using only column 1 in each instance. Under the assumption that player 2 uses only column 1, (4.2-3) is $v = 1+.5v$, which has the solution $v = 2$. However, step 4 fails because $2 \neq val([2\ 1.9])$, revealing that the "obvious" conclusion is actually wrong; player 2 should always choose column 2, as would have been revealed by continuing the sequence one more step, and $v = 1.9$. This example has been deliberately contrived to show that Step 4 is necessary. In practice, the obvious conclusion is usually correct.

Even when it is possible to accurately identify active rows and columns, solution of (4.2-3) may be sufficiently difficult that value iteration is preferable. In that case, the values converge more quickly if $\mathbf{v}(0)$ is close to $\mathbf{v}$, so a little preliminary thought may pay off. Thomas et al. (1983) discuss computational experience with a variety of modified value iteration schemes for Markov Decision Processes, essentially Markov games where player 2 cannot influence the outcome. The same conclusions may apply to Markov games in general.

The Inspection Game shows that there is no hope of finding a finite algorithm using only (+, −, *, /) for finding the value of a Markov game. On the other hand ordinary matrix games are solvable using the simplex method of linear programming, which will always find a solution using finitely many of those operations. The question arises, "What Markov games are solvable using only finitely many of the four fundamental operations of arithmetic?" The answer is not exactly known, but several such subclasses have been identified (Raghavan and Filar (1991)). One such subclass is "single controller" games where the next game element, if any, is dependent entirely on the strategy choice of one of the players. There is an example of such a game in Sect. 8.1.3.

4.7 Football as a Markov Game

American football can be modeled as a Markov game, but the problem is to construct a model that resembles the real game without introducing so many states that the model is unsolvable. Suppose that the model were to include

60	possible amounts of time remaining
4	possibilities for the current down
50	possibilities for the relative score
25	possibilities for the number of yards to go for first down
100	possibilities for the yard line, and
16	possibilities for the number of time outs remaining to the two sides

Since these six factors can occur in any combination, the number of states required would be the product of all six numbers, which is 480 million. This is too large a number to be practical, even though the model omits less quantifiable aspects of the state of the game that an actual coach must take into account. The less important aspects of state will evidently have to be reduced or omitted in order to achieve a tractable model of the game. It makes sense to construct a hierarchy of models, from very simple ones to models sufficiently complex to be of practical use. Our object here is to look at the simple end of the spectrum.

Assume that there are only three offensive and two defensive plays, with the results being shown in Fig. 4.7-1. In that figure L means "lose the ball", "*x* or *y*" means that there is a 50/50 chance of either alternative happening, and numbers represent yards gained.

$$\begin{array}{l} \\ \textbf{run} \\ \textbf{pass} \\ \textbf{punt} \end{array} \begin{array}{c} \begin{array}{cc} \textbf{run defense} & \textbf{pass defense} \end{array} \\ \begin{bmatrix} 0 \text{ or } 6 & 4 \text{ or } 8 \\ 10 \text{ or } 20 & 10 \text{ or L} \\ 20\text{+L} & 20\text{+L} \end{bmatrix} \end{array}$$

Fig. 4.7-1 Play results in football

Even the simplest model needs to count the number of downs remaining. Our initial model G has only four states, with the payoff being the average number of yards gained in at most four downs—we are temporarily ignoring the possibility of getting four more downs by gaining 10 yards. The implied vision of football is that it is played on an infinitely long field, with the winner being the one with the most yards gained after a long interval of play. G is shown in Fig. 4.7-2, where it should be understood that rows and columns are named as in Fig. 4.7-1, that the game element subscript is the index of the current down, and that $G_5 = [0]$. The G model is an exhaustive game, and can be solved by calculating the values of the game elements in reverse order. The "punt" strategy is dominant in G_4, so $v_4 = 20$. We then calculate $v_3 = 24.57$, $v_2 = 28.99$, and $v_1 = 33.31$, all of which involve mixed strategies that never punt. The optimal probability of passing, which risks losing the ball, is only 0.11 on first down, rising to 0.13 on third down. It is encouraging that some familiar features of football (punt only on fourth down, increasing pass probability over the first three downs) emerge even in a model as simple as this one.

$$G_i = \begin{bmatrix} 3 + G_{i+1} & 6 + G_{i+1} \\ 15 + G_{i+1} & 5 + .5G_{i+1} \\ 20 & 20 \end{bmatrix}$$

Fig. 4.7-2 The G model for $\mathbf{i} = \mathbf{1}, \ldots, \mathbf{4}$

It is an essential feature of football that sufficient progress earns a new series of downs, so our next model (model F) incorporates this feature. To simplify the computations assume only two downs are available in which to gain the required 10 yards. The "state" is now (d,y), with d being the current down and y being the number of yards yet to go. The initial state is (1, 10) when a team receives the ball. The number of yards gained is always even and never 2, so $F(2, 4) = F(2, 2)$. Thus only four distinct states can actually arise: (1, 10), (2, 2), (2, 6), and (2, 10). Game elements (1, 10) and (2, 2) are shown in Fig. 4.7-3.

$$\begin{bmatrix} 3+.5F(2,10)+.5F(2,4) & 6+.5F(2,6)+.5F(2,2) \\ 15+F(1,10) & 5+.5F(1,10) \\ 20 & 20 \end{bmatrix} \qquad \begin{bmatrix} 3+.5F(1,10) & 6+F(1,10) \\ 15+F(1,10) & 5+.5F(1,10) \\ 20 & 20 \end{bmatrix}$$

Fig. 4.7-3 F(1,10) on the left and F(2,2) on the right

Model F is not exhaustive, but player 1 clearly cannot arrange to accumulate infinitely many yards by keeping the ball forever because player 2 has the option of always flipping a coin to decide what defense to play next. It therefore makes sense to use value iteration (Sect. 4.6). If the initial guess is that all elements have value 20 except for $F(1, 10)$, which has value 23, the first 10 iterations are shown in Fig. 4.7-4, together with the true value of the game (obtained in this case by continuing value iteration 100 times). The expected gain per possession is 29.45 yards. The punt strategy is never used except in $F(2, 10)$, where it is always used.

Associated with the game values are the optimal run and pass probabilities, which would be the objects of tactical interest.

Fig. 4.7-4 Results of value iteration on model F

iteration	(1,10)	(2,2)	(2,6)	(2,10)
0	23.00	20.00	20.00	20.00
1	24.84	23.97	20.00	20.00
2	26.74	25.37	20.00	20.00
3	27.51	26.82	20.00	20.00
4	28.21	27.41	20.00	20.00
5	28.53	27.94	20.00	20.00
6	28.79	28.18	20.16	20.00
7	28.96	28.38	20.29	20.00
8	29.10	28.51	20.37	20.00
9	29.19	28.62	20.44	20.00
10	29.26	28.69	20.49	20.00
100	29.45	28.88	20.62	20.00

A natural improvement to model *F* would be model *H* where there are four downs instead of two, in which case 10 game elements are required. Sheet "FootballH" of the TPZS workbook implements value iteration for model *H* (exercise 18). The optimism of the second row of Fig. 4.7-1 is evident in model *H* because it is better to gamble on a pass in state (4, 10), rather than punt.

In addition to enriching the selection of plays, any model of football that hopes to be of practical use has got to keep track of field position, in addition to the current down and the number of yards to go. One can hardly decide whether to kick a field goal, for instance, without knowing field position. These three variables, however, seem sufficient to model football to a level where the optimal calculated play frequencies are interesting to coaches. Winston and Cabot (1984) describe how a model at that level of detail was developed to aid Indiana University in play selection. The payoff in that game is the expected net difference of points (rather than yards). Making the objective "probability of winning the game" would require the addition of the current score difference and the amount of time left to the list of state variables.

Additional abstractions of football, baseball and other games are studied in Ladany and Machol (1977) and Winston (2009).

4.8 Tactical Air War

Military aircraft are capable of fulfilling multiple roles, but not simultaneously. Questions about what kind of missions should be undertaken and when they should be undertaken naturally arise. The decisions made can have a dramatic influence on battle outcome, witness the great aircraft carrier battles of World War II, particularly the Midway battle of 1942. Such situations lend themselves naturally to a treatment by Markov games. Our objective in this section is first to analyze an elementary model with only two aircraft missions, and then to describe some more detailed models.

Consider a tactical air war of known but protracted length where each aircraft can be assigned to either counter air (CA) or ground support (GS) at the beginning of each day. We will assume no replenishment of aircraft, so that only the survivors of one day's battle will be available to fight the next. CA aircraft are never lost, but GS aircraft are vulnerable to the other side's CA aircraft. The ultimate purpose of an air force is assumed to be ground support, so we take the payoff to be the net accumulated difference of GS sorties over the remaining length of the war. Let $V(x, y, n)$ be the value of the game where player 1 has x aircraft, player 2 has y aircraft, and n days remain. If u and w are integers representing the two players' secretly selected GS allocations, and if random variables U and W are the corresponding numbers of surviving sorties that actually conduct ground support, then the net payoff is U minus W plus the net difference of ground support missions over the rest of the war. Formally, we have

$$V(x,y,n) = val(E(U - W + V(x - u + U, y - w + W, n - 1))) \tag{4.8-1}$$

Since $V(x, y, 0)$ is known to be 0, (4.8-1) can be used to calculate $V(x,y,n)$ iteratively by back substitution. $V(x, y, 1)$ is just $x - y$, since all aircraft should be allocated to GS on the last day. To compute $V(x, y, 2)$, we need to be more specific about U and W.

Assume that the battle between CA and GS aircraft takes place as follows: First, the CA aircraft distribute themselves as evenly as possible over the GS aircraft. Each CA then independently destroys its assigned GS with probability 0.5. Surviving GS then conduct ground support and return to base for reassignment on the succeeding day. Thus, if 2 GS are met by 3 CA, the number of GS survivors will be 0, 1, or 2 with probability 3/8, 4/8, 1/8. Or if 2 GS are met by 1 CA, the number of GS survivors is equally likely to be 1 or 2. In the specific case $(x,y) = (2,1)$, $V(2,1,n)$ is the value of the 3×2 game below where the rows and columns are labeled with the choice of u and w, respectively. Other game elements can be similarly made explicit by first determining the possibilities and probabilities of the number of surviving GS aircraft.

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \end{array}\begin{array}{c} \begin{array}{cc} \qquad\qquad 0 \qquad\qquad & \qquad\qquad 1 \qquad\qquad \end{array} \\ \begin{bmatrix} V(2,1,n-1) & -.25 + .25V(2,1,n-1) + .75V(2,0,n-1) \\ .5 + .5V(2,1,n-1) + .5V(1,1,n-1) & .5 + .5V(2,1,n-1) + .5V(2,0,n-1) \\ 1.5 + .5V(2,1,n-1) + .5V(1,1,n-1) & 1 + V(2,1,n-1) \end{bmatrix} \end{array}$$

For $n = 2$ and $x, y \leq 3$, it turns out that the optimal strategies are again to allocate all aircraft to GS, so $V(x, y, 2) = 2(x - y)$. However, this is not true for $n = 3$. $V(3, 1, 3)$, for example, is the value of the 4×2 game

$$\begin{array}{c} \\ 0 \\ 1 \\ 2 \\ 3 \end{array}\begin{array}{c} \begin{array}{cc} 0 & \quad 1 \end{array} \\ \begin{bmatrix} 4.0 & 5.625 \\ 3.5 & 6.25 \\ 4.5 & 6.5 \\ 5.5 & 6.0 \end{bmatrix} \end{array}$$

The optimal strategy for player 2 is to assign his only aircraft to CA, rather than GS, and $V(3, 1, 3) = 5.5$. In fact, it turns out that for $x \leq 3$ and $y \leq 3$ the weaker player uses pure CA for all $n \geq 3$. The stronger player uses pure GS except in the cases $(x,y) = (3,2)$ or $(2,3)$ and $n \geq 3$, in which case the stronger player also uses pure CA. If one of those states is ever reached, it will persist until there are only two days left in the war, at which point both players will switch from pure CA to pure GS. In other words, as long as the war is not nearly over, the weaker player finds it more profitable to use his aircraft to prevent future enemy ground support than to conduct current ground support himself. Even though the objective function is "net ground support", in a long war the weaker player should not use his aircraft in that role until the war is almost over or his CA efforts have substantially reduced the opponent's air force. As a result, $\lim_{n \to \infty} V(x,y,n)$ is always finite as long as neither x nor y is 0. It is a curious fact that in no case is a mixed strategy ever needed; in fact, in all cases each player assigns either all or none of his air force to GS.

More realistic variations of the above model have been considered. Berkovitz and Dresher (1959) describe a deterministic model where forces are treated as continuous variables, and where all opposing aircraft (not just GS) are subject to CA attack. Curiously, they find again that every game element has a saddle point. They also consider a game where aircraft can be assigned to a third "air defense" (AD) mission that puts the opponent's CA aircraft at risk, as when fighters escort bombers. The nature of the optimal solution changes considerably when this is done. Mixed strategies appear, and strong and weak players use qualitatively different tactics. A strong player should split his force between all three tasks, but a weak player should ignore the GS mission and randomly choose whether to allocate all his aircraft to CA or AD. Oddly, the GS mission reappears if a player is very weak, almost as if the player wanted to get in one good bash before being wiped out. The precise nature of the strong player's split and the weak player's probabilities defy simple description. This complexity is one of the authors' reasons for pessimism about the possibility of studying complicated games by having humans play them. They opine that the randomness of the weak player's strategy is unlikely to be discovered, particularly if the number of plays is limited.

Some very large versions of models such as this have been developed under the sponsorship of the U.S. Air Force and the Institute for Defense Analyses. The technique illustrated earlier will be lengthy in games that are long or include many aircraft, so there are serious computational issues involved in making such games more realistic. Schwartz (1979) describes a technique that takes advantage of the fact that a game value can sometimes be computed without knowing the whole payoff matrix, particularly if the game has a saddle point or nearly so. The U.S. Air Force (1971), in connection with work on the TAC CONTENDER model, has investigated a technique where games are solved forward (beginning with the first day), rather than backward as in the technique illustrated above. The technique requires estimates of residual values for aircraft surviving each day's combat. Results achieved depend on the quality of those estimates, but the computational burden is substantially reduced.

None of the models described above come very close to actual combat, even statistically. The desire to actually solve a game dictates dealing with stark, simple models that try to capture the essence of a situation in spite of missing most of the detail. The advent of increasingly powerful digital computers is unlikely to change this situation, since analysts have consistently shown themselves capable of quickly consuming large increases in computing power to make relatively minor progress towards realism. Models that are not designed to be explicitly solved as games can incorporate much more realism and richness of detail, but only at the cost of relying on operational gaming for "solution", the same technique that Berkovitz and Dresher (1959) disparage (see also Thomas and Deemer (1957) for a discussion of this issue). We are thus left with a situation where the analyst who attempts to model a real military situation seemingly has two choices, neither of which is satisfactory in itself:

1. Risk losing the essence of the situation, along with surely losing the detail, by abstracting to a TPZS game
2. Construct a more realistic model for which no objective solution technique is available. Operational gaming, with all its pitfalls, will then be the only available method for studying strategy and value.

The dilemma should come as no surprise, since Man's most difficult problems always seem to have humans in them. The game theory opponent is sentient, so it is natural to expect that the analysis will be taxing.

4.9 The Hamstrung Squad Car and Differential Games

Games where a pursuer chases an evader can sometimes be formulated as Markov games. The state of the game generally includes the positions of the two players, sometimes together with other state components. The state changes as time goes by in response to the joint decisions of the two players. A Markov game results if time is discrete. In this section we give an example, and then allude to the theory that applies (differential games) when time is continuous.

In the Hamstrung Squad Car (HSC) game the police are chasing a criminal in a city whose street structure is an unbounded rectangular grid. The police move first and then the players take turns. When his turn comes, the criminal can move one step to any of the four nearest neighbor road intersections. The faster police can move two steps, but are "hamstrung" by traffic laws that prohibit U turns and left turns; the police have only the two options of either going two steps straight ahead or of first turning right and then doing the same thing. The criminal, being a criminal, simply ignores these laws. The two players take turns and are assumed to be able to see each other at all times, so HSC is a Deterministic graphical game. The question is "Can the fast but awkward police catch the slow but nimble criminal?"

The most natural state representation for HSC would include six components: four for the two dimensional positions of the two players, one to keep track of the direction in which the police were last going, and one to keep track of whose move it is. Since only relative position is important, we can adopt a less natural but more compact reference system where the police are always at the origin traveling "up", as in Fig. 4.9. The advantage of this reference system is that it minimizes the dimensionality of the state vector. The disadvantage is that a right turn on the part of the police corresponds to a counterclockwise rotation of the criminal's relative position.

The positions marked with a "1" in Fig. 4.9 all correspond by definition to "capture"; more positions than just the origin are included in this termination region to prevent the possibility that the police might overrun the criminal without capturing him. These nine positions correspond to the nine ordinary game elements. All of the other game elements involve subsequent moves.

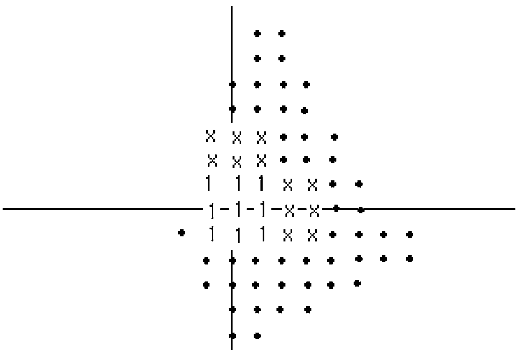


Fig. 4.9 The Hamstrung Squad Car is at the origin going "up"

If (x,y) is the criminal's position relative to the police, let $P(x,y)$ be the game element where the next move belongs to the police, and let $C(x,y)$ be the game element where the next move belongs to the criminal. The P/C notation is more convenient than introducing a third state component to keep track of whose move it is. We can now represent HSC as a Markov game:

$$C(x,y) = [P(x,y+1)\ P(x,y-1)\ P(x+1,y)\ P(x-1,y)]$$
$$P(x,y) = \begin{bmatrix} C(x,y-2) \\ C(-y,x-2) \end{bmatrix} \tag{4.9-1}$$

except that $C(x,y) = P(x,y) = 1$ at each of the nine terminal positions. A solution in pure strategies necessarily exists, so the value will always be either 0 (the criminal cannot be captured) or 1 (capture is certain). Note that the convention of assigning 0 payoff to infinite play is the correct one, given the context.

The solution of HSC can be obtained directly on Fig. 4.9, which we now assume to represent a situation where it is the police's turn to move. If either of the police strategies moves (x,y) to either a terminal position or to a position where all four neighbors are already marked "1", then capture is possible. The 12 positions marked X in Fig. 4.9 are therefore capturable. After the 12 X's are themselves marked "1", repetition of the argument will discover other capturable positions (there are four of them), etc. After several iterations of this technique one discovers that all of the positions shown dotted are capturable. Neither police strategy will move any position outside this "capture region" to a point where it is surrounded by points inside the capture region, so no outside point is capturable. Thus, the value of the game is zero at all positions outside the capture region. If the criminal is initially discovered at an outside position, he has a strategy for motion (taking account of his own position, the police position, and the police direction of travel) that will avoid capture indefinitely. The strategy is "if he ever moves you inside the capture region, then move back out". If the criminal is initially discovered inside the capture region, the police can capture him no matter what he does by eventually maneuvering (x,y) into the termination region. The HSC is now completely solved.

The constraints on the motion of the two players in the HSC are somewhat artificial, as is the assumption that the players take turns in moving. In many pursuit games it would be more natural to measure time continuously, have the players make decisions continuously, and introduce constraints on speed or acceleration. The theory of differential games then applies. The reader who is interested in pursuit games and has no aversion to the use of differential equations is referred to any of several books on the topic. Owen (1995) includes a brief introduction.

The HSC was introduced by Isaacs (1965), who also introduces several intriguingly named differential pursuit games such as the Homicidal Chauffer and the Isotropic Rocket. In the latter the pursuing rocket has a constraint on its acceleration, while the evader instead has a constraint on its speed. The rocket can go as fast as it likes, but cannot make sharp turns like the evader. The evader has no hope of simply outrunning the pursuer, but may nonetheless avoid being caught through nimbleness in close quarters. The HSC phenomenon that capture is possible from certain initial positions, but not from others, reappears The initial positions from which the rocket can capture the evader within a certain time limit might be called a "launch envelope" and displayed for use by the rocket launcher. Once the evader is within the launch envelope, he is doomed if the rocket is launched and optimally controlled. Ho et al. (1965) describe some pursuit-evasion differential games in this vein.

Another approach to keeping the evader from simply leaving the area would be to confine both parties to some fixed region, so that even a slow pursuer might have some chance of cornering a fast evader. An example of this is the Lion and Man differential game (Isler et al. 2005) where a lion attempts to capture a faster man who is confined to a circular arena.

Differential games are games of perfect information because both players are aware of the state at all decision points. The Princess and Monster (P&M) game has the same rules as the Lion and Man game, except that neither the Princess nor the Monster can see each other until the Monster comes within a given capture distance. The lack of observability means that the P&M game is not a differential game. It will be reviewed in Sect. 8.2.3.

4.10 Non-Terminating Markov Games

If there is no expectation that a Markov game will stop, the definition of solvability offered in Sect. 4.2 must be abandoned. In the general case where rewards can accumulate, it is hard to compare one strategy with another because the rewards for both may be infinite. To get around this problem one can either discount the rewards (in which case Sect. 4.6 applies with all transition probabilities multiplied by the discount factor), or else make "average reward per stage" the criterion for comparing one strategy with another. The latter course will be explored in this section.

Let X_n be the total reward at the end of n stages. We now expect X_n to grow without bound as n increases, but if $E(X_n - nv_s)$ is bounded for large n, where v_s is some number that possibly depends on the starting state s, it is reasonable to describe v_s as the "average reward per stage".

Definition In a Markov game G with initial state s, let **x** and **y** be (possibly nonstationary) strategies for players 1 and 2, respectively. We say that **x**LAv_s if $E(X_n - nv_s)$ is bounded below for all **y** and for all n whenever player 1 uses **x**. Similarly we say v_sLA**y** if $E(X_n - nv_s)$ is bounded above for all **x** and for all n whenever player 2 uses **y**. If **x**LAv_s and also v_sLA**y**, we say that **x** and **y** are LA optimal and that v_s is the LA value of G.

If (**x**,**y**,v_s) is the LA solution of a Markov game and if both players use their optimal strategies, then $E(X_n - nv_s)$ is bounded above and below, and therefore $\lim_{n \to \infty} E(X_n/n) = v_s$. Hence the term "limiting average". As in Sect. 4.2, the LA definition refers to strategies that are in principle nonstationary, but our first object will be to establish conditions where stationary strategies suffice.

Theorem 4.10 Let Markov game G consist of game elements $G_1, \ldots, G_K$. For all states k let $\mathbf{x}(k)$ be a probability distribution over the rows of G_k, let $\mathbf{x} = (\mathbf{x}(1), \ldots, \mathbf{x}(K))$ be a stationary strategy for player 1, and let v_k and u_k be real numbers. Suppose that

$$\sum\nolimits_i x_i(k)\left(\sum\nolimits_l p_{ij}^{kl} v_l - v_k\right) \geq 0;\ j \text{ in } G_k \text{ and } k = 1, \ldots, K, \text{and also that} \tag{4.10-1}$$

$$\sum\nolimits_i x_i(k)\left(a_{ij}(k) + \sum\nolimits_l p_{ij}^{kl} u_l - u_k - v_k\right) \geq 0;\ j \text{ in } G_k \text{ and } k = 1, \ldots, K. \tag{4.10-2}$$

Then **x**LAv$_s$. Similarly, let $\mathbf{y}(k)$ be a probability distribution over the columns of G_k, with $\mathbf{y} = (\mathbf{y}(1), \ldots, \mathbf{y}(K))$, and let w_k and z_k be real numbers. Suppose that

$$\sum\nolimits_j \left(\sum\nolimits_l p_{ij}^{kl} w_l - w_k\right) y_j(k) \leq 0; i \text{ in } G_k \text{ and } k = 1, \ldots, K, \text{and also that} \tag{4.10-3}$$

$$\sum_j \Big(a_{ij}(k) + \sum_l p_{ij}^{kl} z_l - z_k - w_k \Big) y_j(k) \leq 0; i \text{ in } G_k \text{ and } k = 1, \ldots, K. \quad (4.10\text{-}4)$$

Then $w_s LA\mathbf{y}$.

Proof We will prove only that **x**LAv_s. Let the random variables $A(t)$, $J(t)$, and $S(t)$ be the reward, column choice, and state at stage t, respectively. Also let $V(t) \equiv v_{S(t)}$, $U(t) \equiv u_{S(t)}$ and $\overline{u} \equiv \max_k u_k$. For $t \geq 1$, (4.10-2) can be rewritten

$$E\big(A(t) + U(t+1) - U(t) - V(t) \big| S(t) = k, J(t) = j\big) \geq 0; \quad j \text{ in } G_k \text{ and } \quad k = 1, \ldots, K. \quad (4.10\text{-}5)$$

By the Conditional Expectation Theorem, (4.10-5) implies

$$E(A(t) + U(t+1) - U(t) - V(t)) \geq 0 \text{ for } t \geq 1. \quad (4.10\text{-}6)$$

By summing (4.10-6) from $t = 1$ to n, we obtain

$$E(X_n + U(n+1) - U(1) - V(1) - \ldots - V(n)) \geq 0. \quad (4.10\text{-}7)$$

But (4.10-1) implies $E(V(t+1) - V(t)) \geq 0$. Therefore $E(V(1)) \leq E(V(2)) \leq \ldots \leq E(V(n))$, and it follows from (4.10-7) that

$$E(X_n + U(n+1) - U(1) - nV(1)) \geq 0. \quad (4.10\text{-}8)$$

Since $V(1) = v_s$, $U(1) = u_s$, and $U(n+1) \leq \overline{u}$, it follows from (4.10-8) that $E(X_n - nv_s) \geq u_s - \overline{u}$, as was to be shown. The proof that $E(X_n - nv_s) \leq z_s - \overline{z}$ as long as player 2 uses **y**, where $\overline{z} \equiv \min_k z_k$, is similar. §§§

For many Markov games with LA solutions we expect that v_s will be the same number g for all game elements, with g being the average gain per stage and $\mathbf{v} = (g, \ldots, g)$. One would expect this in any situation where the long run payoff should be independent of the starting point. In that case, suppose that **u**, **v** is a solution of $\mathbf{v} + \mathbf{u} = T(\mathbf{u})$, where $T(\ )$ is the transformation (4.2-2). Then the solution, together with the associated mixed strategies, and taking $\mathbf{w} = \mathbf{v}$, $\mathbf{z} = \mathbf{u}$, will satisfy all four equations required in Theorem 4.10. Stationary strategies suffice for such games, and finding the LA solution is equivalent to solving the value equation $\mathbf{v} + \mathbf{u} = \mathrm{T}(\mathbf{u})$ (Raghavan and Filar 1991).

Example Consider the Markov game shown in Fig. 4.10-1. Note that there is one possibility in G_2 where the game ends after a reward of -1, a possibility that player 2 would relish because all other rewards in both game elements are nonnegative. However, player 1 can prevent termination by choosing row 2 in G_2, so we should expect the game to go on forever. We consequently look for an LA solution.

Fig. 4.10-1 A Markov game with an LA solution

$$G_1 = \begin{bmatrix} 2+G_2 & G_1 \\ G_2 & 1+G_1 \end{bmatrix} \qquad G_2 = \begin{bmatrix} G_2 & -1 \\ 1+G_1 & 2+G_1 \end{bmatrix}$$

Letting $\mathbf{v} = (g,g)$, the two components of $\mathbf{v} + \mathbf{u} = \mathrm{T}(\mathbf{u})$ are shown in Fig. 4.10-2. Assuming for the moment that player 1 does indeed always choose row 2 in G_2, the second game element requires $g + u_2 = 1 + u_1$ at the supposed saddle point in the lower left-hand corner. Since only the difference $u_2 - u_1$ is important, set $u_1 = 0$ for convenience. Assuming that the first game element has no saddle point, solve the first component (with $g = 1 - u_2$) for u_2. The solution is $u_2 = 1/4$, with associated optimal strategies $\mathbf{x}(1) = (1/4, 3/4)$ and $\mathbf{y}(1) = (1/3, 2/3)$. Substituting $\mathbf{u} = (0, 1/4)$ in the second element, we verify that it has a saddle point at $\mathbf{x}(2) = (0,1)$ and $\mathbf{y}(2) = (1,0)$. The LA value of the game is therefore $g = 3/4$, regardless of the starting state. The stated mixed strategies each guarantee an LA payoff of 3/4 per stage in the long run. The components of $\mathbf{u}$ can be interpreted as relative values of being in the referenced state; player 1 prefers G_2 to G_1 because $u_2 - u_1 = 1/4$. The statements "the LA value is 3/4 regardless of the starting state" and "player 1 prefers state 2" are not inconsistent—the starting state does not affect the long run gain per stage, but it does affect player 1's immediate expectations.

Fig. 4.10-2 Finding u_1, u_2 in a two-element game

$$g + u_1 = val\left(\begin{bmatrix} 2+u_2 & u_1 \\ u_2 & 1+u_1 \end{bmatrix}\right) \qquad g + u_2 = val\left(\begin{bmatrix} u_2 & -1 \\ 1+u_1 & 2+u_1 \end{bmatrix}\right)$$

Theorem 4.10 suggests a pair of mathematical programs for finding the limiting average reward or at least bounds on it. Player 1's program is to maximize v_s subject to (4.10-1) and (4.10-2), and player 2's program is to minimize w_s subject to (4.10-3) and (4.10-4). Necessarily $v_s \leq w_s$, but if the optimized values v_s^* and w_s^* are equal, an LA solution has been found. Filar and Schultz (1987) show that $v_s^* = w_s^*$ provided optimal stationary strategies exist in the LA sense; that is, a stationary strategy solution can be found by mathematical programming if one exists.

There are examples of Markov games where there is no LA solution in stationary strategies. Figure 4.10-3 shows the Big Match of Blackwell and Ferguson (1968). Note that G_2 and G_3 repeat indefinitely once entered, with the payoff at each stage being 1 in G_2 and 0 in G_3. Player 1 seems to be trapped in G_1. If he uses any stationary strategy of the form $(x, 1 - x)$ with $x > 0$, then player 2 can force an average reward of 0 by consistently choosing column 2 (sooner or later G_3 will be entered). But if player 1 makes $x = 0$, then player 2 can accomplish the same thing by using column 1. If player 1 solves the mathematical program suggested in the previous paragraph, he will find that the maximum possible value for v_1 is 0 (exercise 11). Nonetheless, Blackwell and Ferguson (1968) show that player 1 can guarantee an average payoff arbitrarily close to 0.5 by using a non-stationary strategy where row 2 is more likely to be chosen when the observed frequency of column 2 in previous stages is large, and that the value of the Big Match is therefore

0.5, not 0. Player 2 can guarantee 0.5 by flipping a coin in G_1 at every opportunity, a stationary strategy.

Fig. 4.10-3 The Big Match

$$G_1 = \begin{bmatrix} 1+G_2 & G_3 \\ G_1 & 1+G_1 \end{bmatrix} \qquad G_2 = [1+G_2] \qquad G_3 = [G_3]$$

Mertens and Neyman (1981) have shown that all Markov games have LA solutions if non-stationary strategies are permitted. However, there is as yet no simple method by which to determine whether a given Markov game requires such strategies.

It is usually obvious from context whether a solution in the sense of Sect. 4.2 (a TR solution—TR for "total reward") or an LA solution should be sought, but conceivably the appropriate type is not obvious from the problem statement. It is not difficult to establish the following facts:

1. If a Markov game has a TR solution, then its LA value is zero.
2. If a Markov game has a non-zero LA solution, it has no TR solution.

It is not definitely known whether Markov games with an LA solution of 0 must necessarily have a TR solution. However, the trivial Recursive game that repeats indefinitely with a zero reward per stage provides an example of a Markov game with a zero LA value where play continues forever, and therefore of a game that cannot possibly satisfy the finite play requirements of Sect. 4.2.

4.11 Exercises

1. The negative square root was taken in solving (4.1-2). Explain why.
2. The text describes how the Inspection Game could be represented as an ∞-dimensional matrix. For β=.5, what is the upper left-hand 4 × 4 corner of that matrix? Using linear programming, solve that 4 × 4 game, and compare it with the solution of the Inspection Game. Hint: $a_{41} = 0.5$.
3. Calculate the value and optimal strategies in the Women and Cats versus Mice and Men game element $G(2,1,2,1)$.
4. It was shown in Sect. 4.4 that every exhaustive game has an ordinary game element. Give an example showing that the converse statement is false; that is, give an example of a Markov game with an ordinary game element that is not exhaustive.
5. Sheet "FootballH" of the TPZS workbook implements the final version of football as introduced in Sect. 4.7.
   (a) If punts only go 20 yards, the punt strategy is never used. How long do punts have to be to make them attractive?
   (b) The state "second and two" (2&2) gets a higher score than state 1&10. Is this because more additional yards can be gained from 2&2, or because 8 yards must already have been gained to get from 1&10 to 2&2?

6. In the bankruptcy game of Sect. 4.2, show that player 1's strategy guarantees finite play by finding **u** that solves (4.2-8).

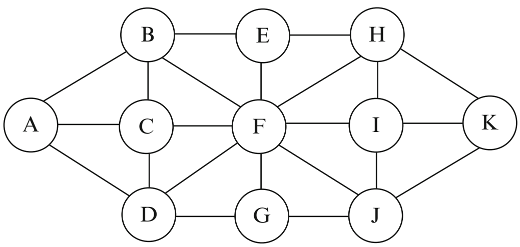
7. Starting from $\mathbf{v}(0) = (0,0,0)$, apply value iteration to the bankruptcy game of Sect. 4.2. What is $\mathbf{v}(3)$?

8. Change $a_{22}(2)$ from 2 to 0 in the game shown in Fig. 4.10-1 and find the LA solution.

9. Consider the GS/CA game of Sect. 4.8.

   (a) For $(x,y,n) = (3,2,2)$, verify that the game has a saddle point where both sides assign all aircraft to GS.
   (b) Assuming that $V(x,y,2) = 2(x - y)$, find V(3,2,3).
   (c) What is $\lim_{n\to\infty} V(2,1,n)$? Hint: Find an equation that relates $V(2,1,n)$ to $V(2,1,n-1)$.

10. The French Military Game is played on the board shown below, with player 1's three pennies and player 2's nickel each occupying one of the 11 circles. The initial configuration has pennies on circles A, B, and D, with the nickel on circle F. The pennies move first and the players take turns. Player 2 must move the nickel to any unoccupied feasible (connected by a line) neighboring circle, and player 1 must move exactly one of his pennies to any unoccupied feasible circle that does not involve movement to the left (a penny on F could move to G, but not D). The object of the game is for player 1 to trap the nickel in a circle from which there is no feasible move not occupied by pennies—the nickel must necessarily be on either E, G, or K for this to happen. Player 1 wins in this case, and loses in the case of infinite play. In practice, player 1 concedes when it becomes clear that he cannot trap the nickel. The game was at one time popular among officers in the French army, hence the name. It is an example of a graphical game with no random moves, the same category that includes Tic-Tac-Toe, Chess and Checkers. The assignment is to simply play the game until you are prepared to guess whether player 1 has a strategy that will always trap the nickel. The only other possibility is that player 2 has a strategy that makes him impossible to trap. Either copy the figure below and play the game with a partner, or find an interactive version on the internet. After playing the game for a while and looking at the answer, what do you think of operational gaming as a solution technique?



11. Change the payoff of the Hamstrung Squad Car game to be the number of moves required by the squad car to effect capture. The game $P(x,y)$ then becomes $[1 + C(x, y - 2)\ 1 + C(-y, x - 2)]$, a row vector because the police are now minimizing, with $P()$ and $C()$ both being 0 at each of the nine terminal positions. Solve this version within the capture region, either by hand or by writing a computer program. The capture times should range from 0 (in the nine terminal positions) to 11. This is the version originally considered by Isaacs (1965).

12. For the Graphical game shown in Fig. 4.5-1, give (u_1,u_2) satisfying (4.2-8) and show that there is no (z_1,z_2) satisfying (4.2-13). This shows that the premises of Theorem 4.2-3 are sufficient but not necessary for a Markov game to have a solution.

13. In Sect. 2.4 it was pointed out that the submarine's survival, as well as damage to the convoy, was at stake. The submarine's survival is important because a surviving submarine can attack other convoys in the future. The analysis of Sect. 2.4 essentially assumed that the future was unimportant, but suppose now that a submarine attacks convoys until it is itself sunk. Assume further that a submarine that attacks from outside r will sink a ship with probability $P_1(r)$, and always survive, whereas a submarine that chooses to penetrate will survive the attempt with probability $P_2(r)$, and will always sink a ship if the penetration attempt is successful. Between attacks, a submarine will survive with known probability $q < 1$. Model this situation as a one element Markov game where the submarine moves last at every stage, knowing the barrier radius r. Let v be the total number of ships that a submarine that is about to make an attack will sink over its remaining lifetime. What equations must v and the optimal barrier radius r^* satisfy? Since there are two unknowns, you need two equations.

14. Section 4.10 mentions a pair of mathematical programs for finding bounds on the limiting average value. Show that these programs produce bounds of $v_1 = 0$ and $w_1 = 0.5$ for the Big Match.

15. Show that the value of Dresher's game G_{nk} introduced in Sect. 4.1 is k/n, and find the optimal mixed strategies for the two sides.

16. The example in Sect. 4.4.2 requires the solution of some 2-on-1 and 1-on-2 battles. Solve all four of them, and use the results to verify that the value in the first row (all Blue units shoot at c) and first column (all Red units shoot at a) of the 4×4 matrix is actually 0.46. There is a solution on page "Battle42" of the TPZS workbook.

17. To evaluate the NimSum function introduced in Sect. 4.3.1, begin by expressing each of the numbers n_k in binary format. For example 4 is 100, 5 is 101 and 7 is 111. Cross out pairs of equal powers, add up the remainders, and you have the binary expansion of NimSum. Thus, after crossing out four 1 s, NimSum(4,5,7) = 110 in binary format, or decimal 6. If it is

your move, remove 6 stones from the pile of 7, leaving your opponent with state (4,5,1). That state has NimSum 0 because, considering all three piles, there are an even number of 1s in all three positions of the binary expansions.

(a) Now that you know the secret, amaze your friends at how good you are at playing Nim.
(b) Revise the rules so that the player who chooses the last stone loses, rather than wins. Can you still base an optimal strategy on the NimSum function?

18. There is a game called "99" where player 1 moves first and then the players alternate in adding something to a sum S, the initial value of S being 0. The amount added by each player must be an integer between 1 and 10, including the endpoints, and S must always remain strictly smaller than 100. The winner is the player who makes $S = 99$, since this leaves the other player with no feasible move. Who wins, and what is an optimal strategy for that player? This is a combinatorial Tree game, and you are being asked to solve it using the bare hands approach. Do not attempt to write out the whole tree!

Chapter 5
Games with a Continuum of Strategies

We didn't lose the game;
we just ran out of time.

Vince Lombardi

It is not unusual to encounter games where the number of available pure strategies is infinite. Any game where the two players each select a time for action is an example, or a submarine can dive to any depth up to some maximum limit. Intervals of real numbers can of course be artificially subdivided to make the number of strategies finite, but that is merely an approximation technique. Sometimes it may even be enlightening to approximate a subdivided interval by a continuous one. The radio frequency spectrum, for example, contains only finitely many frequencies as far as modern digital receivers are concerned, but there are so many frequencies that for some purposes one might as well think of the spectrum as being continuous. In this chapter we consider games where the choice of strategy is not limited to a finite set.

5.1 Games on the Unit Square, Continuous Poker

In the Basic Poker model of Sect. 3.8, player 1 receives a hand while player 2 does not. In Continuous Poker both players receive hands. Each player puts an ante of a units in the pot and is then given his hand, a random number uniformly distributed in the unit interval. After looking at his hand, player 1 may put an additional b units in the pot ("bet"); if he does not, player 2 gets the $2a$ units in the pot. If player 1 bets, player 2 (after inspecting his own hand) may put b units in the pot ("call"); if he does not, player 1 gets the pot. If a bet is called, the player with the better hand (the larger number) wins. The game is not symmetric because player 1 acts before player 2.

A. Washburn, *Two-Person Zero-Sum Games*, International Series in Operations Research & Management Science 201, DOI 10.1007/978-1-4614-9050-0_5,

Let random variables X and Y be the hands of players 1 and 2. For simplicity we will assume that player 1 never bluffs, which is equivalent to assuming that there is some level x such that he bets only if his hand is larger than x. Similarly player 2 calls only if $Y > y$, where y is his calling level. If $X \le x$, then, regardless of y, the payoff is $-a$ because player 1 loses his ante. Accounting for all possibilities of both X and Y, the payoff is

$-a$	if $X \le x$, or
a	if $X > x$ and $Y \le y$, or
$a + b$	if $X > x$, $Y > y$, and $X > Y$, or
$-a - b$	if $X > x$, $Y > y$, and $X \le Y$.

By averaging over X and Y and simplifying, we find that the average payoff is

$$A(x,y) \equiv \begin{cases} x(-a) + (1-x)(y)(a) + (1-y)(a+b)(x-y) \text{ if } x \le y \\ x(-a) + \big(1-x\big)\big(y\big)\big(a\big) + \big(1-x\big)\big(a+b\big)\big(x-y\big) \;\text{ if } x > y \end{cases}. \quad (5.1\text{-}1)$$

Note that $A(x,y) \neq -A(y,x)$. As noted above, the game is not symmetric.

A game where each player's strategy space is the unit interval is called a game on the unit square. Continuous Poker is an example. The important thing about the unit square is that it fits the assumptions of Theorem 5.1.

Theorem 5.1 Let the pure strategy sets B and C be closed, bounded, non-empty convex subsets of finite dimensional Euclidean space, and suppose that the payoff $A(x,y)$ is continuous for x in B and y in C. Then there exist cumulative distribution functions (CDFs) $F^*()$ and $G^*()$ and a number v (the value of the game) such that

$$\int_B A(x,y)dF^*(x) \le v \;\text{ for all } y \text{ in } C, \text{ and}$$

$$\int_C A(x,y)dG^*(y) \le v \;\text{ for all } x \text{ in } B.$$

The Stieltjes notation for integrals is used because the CDFs may have both discrete and continuous parts. A proof can be based on the idea that all finite games have a solution, and a continuous game can be closely approximated by a finite game where the strategy sets are fine partitions of B and C (e.g. Burger 1963, his Theorem 5).

In particular, since the unit interval is a closed, bounded convex subset of the real line, and since (5.1-1) is continuous on the unit square, Continuous Poker has a solution.

Some kind of continuity restriction is essential in Theorem 5.1. To see why, consider Ville's game on the unit square where the winner is the player who chooses the larger number in the interval [0,1], except that a player loses if he chooses "1" while his opponent chooses anything smaller. If both players choose the same number, the outcome is "tie". The payoff function is discontinuous, and the game has no solution.

The value of Continuous Poker can be approximated as closely as desired by a discrete game with a large number of strategies (exercise 1). This will work for any continuous game, but bear in mind that the discrete approximation of a continuous game may disguise important features of the solution. Continuous Poker actually has a saddle point (exercise 2), but discrete approximations may not reveal that fact.

Newman (1959) studies a game called "real" poker that is the same as Continuous Poker except that player 1 can bet any non-negative amount. Player 1's optimal strategy has him betting 0 (thus abandoning the ante to player 2) if his hand lies in the interval [1/7,4/7], but betting a positive amount otherwise. Note that he bets with very weak hands ("bluffs"), but not with medium hands! The ability to control the size of the bet is apparently an advantage, since the value of real poker is positive ($a/7$, to be exact), rather than negative as in Continuous Poker (exercise 2). Karlin (1959) analyzes Poker models where the number of legal bets is larger than 1 but still finite. All of these models of Poker have only two players. If there were more than two, as there usually are in reality, we would not have a TPZS game.

It is not unusual for TPZS games on the unit square to have solutions employing only finitely many strategies, even though infinitely many are available. Continuous Poker is an example, as are certain search games. Consider a search game where y is a place for player 2 to hide, x is player 1's guess at y, and $A(x - y)$ is the probability that player 1 detects or captures player 2. We might have $A(x - y) \equiv \exp(-k(x - y)^2)$, where k is positive. This function has its maximum (1.0) when $x = y$, the perfect guess on player 1's part. This game is "bell-shaped", a class of games on the unit square where it is known (Karlin 1959) that the two players will always employ only finitely many strategies. Exercise 18 is another example.

Another special class of games on the unit square has a payoff in the form of a finite sum:

$$A(x,y) \equiv \sum_i \sum_j a_{ij} r_i(x) s_j(y). \tag{5.1-2}$$

These games are sometimes called "polynomial" or "polynomial-like" or "separable"—we will use the latter term. The functions and coefficients in (5.1-2) are not always unique. For example consider the payoff function $A(x,y) \equiv x\ \sin(y) + x\ \exp(y) + 3x^2$. To put that function into the required format, we might take

$$r_1(x) = x,\ \ r_2(x) = x^2,\ \ s_1(y) = \sin(y),\ \ s_2(y) = \exp(y),\ \ s_3(y) = 1, \text{and}$$
$$\mathbf{a} = \begin{bmatrix} 1 & 1 & 0 \\ 0 & 0 & 3 \end{bmatrix}. \tag{5.1-3}$$

However, we could also take

$$r_1(x) = x,\ \ r_2(x) = x^2,\ \ s_1(y) = \sin(y) + \exp(y),\ \ s_2(y) = 3, \text{and}$$
$$\mathbf{a} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}. \tag{5.1-4}$$

Either of these decompositions establishes that $A(x,y)$ is separable, although the second is probably preferable because it involves four functions instead of five.

The payoff function $A(x,y) \equiv \sin(4xy)$ is not separable, since there is no way to express it as a finite sum of products of functions of one variable. Neither is (5.1-1)—either of the two expressions in (5.1-1) by itself would be separable, but (5.1-1) *in toto* cannot be put into the form (5.1-2).

Separable games have special methods of solution, but we defer the subject to Sect. 5.5 because of their close relationship to bilinear games.

5.2 Concave-Convex Games

Continuous games sometimes have solutions in pure strategies, Continuous Poker being an example. The following theorem gives sufficient conditions. See Appendix B for background material on convex sets and functions.

Theorem 5.2 If B, C, and $A(x,y)$ are as in Theorem 5.1, and if in addition $A(x,y)$ is a concave function of x for each y, then player 1 has an optimal pure strategy and $v = \max_x \min_y A(x,y)$. Similarly, if $A(x,y)$ is a convex function of y for each x, then player 2 has an optimal pure strategy and $v = \min_y \max_x A(x,y)$.

Proof Let $F^*()$ be an optimal mixed strategy for player 1 and let $\mu \equiv \int_B x dF^*(x)$ be the average value chosen by player 1 using $F^*()$. Since B is a convex set, μ is in B. Since $A(x,y)$ is a concave function in x for each y, there is some gradient $k(y)$ such that

$$A(x,y) \le A(\mu,y) + k(y)(x-\mu) \quad \text{for all } x. \tag{5.2-1}$$

By averaging (5.2-1) with respect to $F^*()$, we obtain

$$v \le \int_B A(x,y)dF^*(x) \le A(\mu,y) + k(y)(\mu-\mu) = A(\mu,y) \tag{5.2-2}$$

The first inequality is true because $F^*()$ is optimal, and the second because of (5.2-1). Since (5.2-2) is true for all y in C,

$$v \le \min_y A(\mu,y) \le \max_x \min_y A(x,y) \le v. \tag{5.2-3}$$

The last inequality holds because the maxmin value can never be larger than the game value. Since all three quantities in (5.2-3) must be equal, the statements about player 1 are proved. The proof of the second part is similar. §§§

If $A(x,y)$ is both concave in x for each y and convex in y for each x, then both parts of Theorem 5.2 hold and the game has a saddle point. Finding the saddle point of such a concave-convex game is essentially a matter of finding a pair of strategies (x,y) such that y is player 2's optimal response to x and also x is player 1's optimal response to y. Formally, let $U(x)$ be player 2's optimal response to x (so $A(x,U(x)) = \min_y A(x,y)$), and similarly let $V(y)$ be such that $A(V(y), y) = \max_x A(x,y)$. Then we want to find a pair (x, y) such that $x = V(y)$ and simultaneously $y = U(x)$. If those two equations possess a solution (x^*, y^*), then (x^*, y^*) is a saddle point.

Example Consider the separable game on the unit square

$$A(x, y) = -x^2 + y^2 + 3xy - x - 2y \tag{5.2-4}$$

This function is concave in x for each y and also convex in y for each x. By solving the equation $(d/dx)A(x,y) = 0$ for x, we obtain $x = (3y - 1)/2$. This is the optimal response to y if it is in the unit interval. Allowing for the possibility that the optimal response is on a boundary of the unit interval,

$$V(y) = \begin{cases} (3y-1)/2 & ; 1/3 \le y \le 1 \\ 0 & ; 0 \le y \le 1/3 \end{cases} \tag{5.2-5}$$

Similarly, by equating $(d/dy)A(x,y) = 0$ and solving for y, we find that

$$U(x) = \begin{cases} (2-3x)/2 & ; 0 \le x \le 2/3 \\ 0 & ; 2/3 \le x \le 1 \end{cases} \tag{5.2-6}$$

To find the saddle point, it is simplest to sketch both functions on the unit square; $U(x)$ in the conventional manner and $V(y)$ with y measured on the y-axis. The result is shown in Fig. 5.2. The saddle point is revealed as the place where the two curves intersect.

In this case, the intersection is in the interior of the square, so we solve the equations $x = (3y - 1)/2$ and $y = (2 - 3x)/2$ to obtain the saddle point $(x^*, y^*) = (4/13, 7/13)$, as well as the value of the game $v = A(x^*, y^*) = -9/13$.

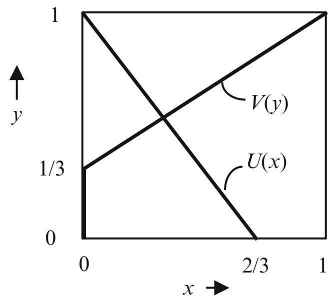


Fig. 5.2 Finding the saddle point of a concave-convex game

The purpose of drawing figures such as Fig. 5.2 is mainly to determine which line segments intersect at the saddle point. There will always be an intersection if $U(x)$ and $V(y)$ are each continuous on the unit interval, but continuity is not a necessary condition. As long as the two curves intersect, the intersection is a saddle point. The optimal response functions can be defined for analytical convenience at points where the optimal response is not unique.

Even if the payoff function satisfies only one part of Theorem 5.2, the value of the game and one player's optimal strategy can still be determined without ever considering mixed strategies. This can be important in minimizing the amount of computational effort required, as will be illustrated in the next section.

5.3 Logistics Games and Extensions

Logistics games arise when player 2 must divide up some kind of "force" totaling c over n regions without knowing the region in which player 1 will test him. If player 1 chooses region i and if player 2's allocation to that region is y_i, then the payoff is $f_i(y_i)$. Each of these functions is assumed to be convex and decreasing on the interval $[0,c]$. One application would be to a situation where player 1 must choose one of n supply routes subject to interdiction by player 2's forces, so we will refer to such games as "Logistics games".

Let x_i be the probability that player 1 chooses region i, so that the average payoff when player 2 uses $\mathbf{y}$ is $A(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} x_i f_i(y_i)$. Since $f_i(y_i)$ is convex over $[0,c]$ for all i, Theorem 5.2 applies and $v = \min_{\mathbf{y}} \max_{\mathbf{x}} A(\mathbf{x},\mathbf{y})$, with the minimizing vector being optimal for player 2. The inner maximization is trivial because $\max_{\mathbf{x}} A(\mathbf{x},\mathbf{y}) = \max_i f_i(y_i)$. However, determining v when c is given is still an n-dimensional minimization problem. It is usually easier to determine the smallest c for which the game value is v, since the minimization problem then decouples into n one-dimensional minimization problems. If we let $y_i(v)$ be the smallest non-negative number u such that $f_i(u) \leq v$,[1] then the smallest possible total resource is $c = Y(v) \equiv \sum_{i=1}^{n} y_i(v)$. If S is the set of subscripts i for which $f_i(0) > v$, then $f_i(y_i) = v$ for $i \in S$, an equation that can be solved for y_i, or otherwise $y_i = 0$.

Example Suppose $n = 2, f_1(y_1) \equiv 3/(1 + y_1)$ and $f_2(y_2) = 1/(2 + y_2)$. If $v = 2$, we make $y_1 = 0.5$ and $y_2 = 0$ to make sure that neither function exceeds 2. Thus $S = \{1\}$ and $Y(2) = 0.5 + 0 = 0.5$. If $v = 0.1$, we make $y_1 = 29$ and $y_2 = 8$ for a total of 37. If c is neither 0.5 nor 37, then search for the achievable value of v must continue.

Example (IED Warfare) The extended conflicts in Iraq and Afghanistan introduced a kind of warfare based on Improvised Explosive Devices (IEDs). IEDs are essentially homemade mines that player 1 places on roads in the hope that they will be activated by player 2's logistic traffic. One of the counters to IEDs is to employ sweepers that continuously patrol the roads searching for IEDs, safely disposing of those that are found. Player 2 is limited in the total number of sweepers available.

[1] $y_i(v)$ and $f_i(u)$ are inverse functions.

A game results where player 1 chooses a road segment for each IED and player 2 allocates sweepers to road segments, with the payoff being the probability that player 1's IED will damage one of player 2's logistic vehicles.

To construct a model of this situation, we begin with the idea that every IED that is placed on a road segment will eventually be removed for one of three reasons: it might be removed by one of player 2's logistic vehicles (this is what player 1 hopes for), it might be removed by a sweeper, or it might be removed for some other reason that neither side has any control over. Each of these phenomena has a rate at which it happens, and the total rate of removal is the sum of the three. The data that we need to quantify this are

$d_k \equiv$ rate at which an IED on segment k is removed by player 1's logistic traffic,
$s_k \equiv$ rate at which an IED on segment k is removed by sweeping, per sweeper, and
$s_k \equiv$ rate at which an IED on segment k is removed for other reasons.

If there are y_k sweepers assigned to segment k, then the total rate at which a given IED on segment k is removed is $c_k + d_k + s_k y_k$. The probability that the removal of an IED is due to logistic traffic is therefore the rate ratio $d_k/(c_k + d_k + s_k y_k)$. Let p_k be the fraction of IEDs removed by logistic traffic that actually damage the logistic vehicle. The probability that a given IED damages some logistic vehicle is then

$$A(\mathbf{x}, \mathbf{y}) = \sum_k x_k p_k d_k / (c_k + d_k + s_k y_k)$$

where x_k is the probability that the IED is placed on segment k. This expression is of the required form, so we have modeled IED warfare as a Logistics game. See sheet "IED" of the TPZS workbook for a quantitative example, or Washburn and Ewing (2011) for further detail, generalization, and some ideas on data estimation. In other IED models d_k might also be under player 1's control because he can route logistic vehicles as he likes, but here the logistic traffic is taken to be fixed.

In the IED example, the question arises as to whether $\mathbf{y}$ must consist of integers. If so then analysis as a Logistics game is not possible, so there are analytic reasons for not restricting $\mathbf{y}$ in that manner. It may also be more realistic to omit the restriction. The IED game is played over a long time period, so the assignment of 4.8 sweepers to a segment, for example, could be implemented by occasionally assigning 4 and more frequently assigning 5, or perhaps by recognizing that missions for individual sweepers may involve multiple arcs.

Player 1's optimal strategy is generally mixed in Logistics games. Given the rest of the game's solution, that strategy can also be recovered. If S is empty, then any strategy is optimal for player 1. Otherwise, let y_i* be player 2's optimal allocation to region i, and observe that, since $f_i(y_i)$ is convex and decreasing, there exist positive gradients s_i such that $f_i(y_i) \geq f_i(y_i*) - s_i(y_i - y_i*)$ for $i \in S$. By making x_i inversely proportional to s_i for $i \in S$, or otherwise 0, player 1 can force the payoff to be at least v no matter what player 2 does (exercise 4). That mixed strategy is therefore optimal for player 1.

The same method of solution (find c as a function of v, rather than vice versa) works even if additional constraints are placed on $\mathbf{y}$. If additional constraints to the effect that $y_i \le 1$ are imposed, for example, then $y_i(v)$ simply becomes "the smallest number u in the unit interval for which $f_i(u) \le v$". If there is no such number, then v is not the game value for any c. Imposing such constraints would be natural if y_i were itself a probability (exercise 15).

If player 1 also has a divisible resource $b > 0$ to allocate to the n regions, and if the payoff in any region is proportional to player 1's allocation, then a game results where $A(\mathbf{x},\mathbf{y})$ is as before, but where $\mathbf{x}$ is constrained to satisfy $\sum_i x_i = b$. In IED warfare, b might be the total rate at which player 1 can place IEDs on roads. This new game is equivalent to a Logistics game where x_i/b is interpreted as the probability that player 1 will choose region i, and can therefore be solved in the same manner. The value of the new game is just b times the value of the corresponding Logistics game.

Finally, consider a modified Logistics game where $A(\mathbf{x},\mathbf{y}) \equiv \sum_i f_i(x_i - y_i)$, where $f_i()$ is an increasing, convex function on the real line for all i and player 1 has b units in total. The payoff is still convex in $\mathbf{y}$, although it is not concave (it is convex, in fact) in $\mathbf{x}$. Such games have been proposed as models for the attack ($\mathbf{x}$) and defense ($\mathbf{y}$) of a group of targets, with damage to the i^{th} target depending on the difference $x_i - y_i$. The convexity assumption is a bit unnatural in this regard, but it does at least permit a relatively simple solution through the employment of Theorem 5.2. Player 2 has an optimal pure strategy because of the convexity assumption. On account of the same assumption, player 1 should devote the entirety of his b units to a single target, so the value of the game is

$$v = \min_{\mathbf{y}} \max_{\mathbf{x}} A(\mathbf{x},\mathbf{y}) = \min_{\mathbf{y}} \max_i f_i(b - y_i). \tag{5.3-1}$$

Formula (5.3-1) makes possible a solution by the same method used in Logistics games, since it is a simple matter to determine the smallest possible value for y_i when v is given. In special cases explicit solutions are possible (see exercise 11 or Blackett 1954).

Logistics games are special cases of Blotto games. The general case will be studied in Chap. 6.

5.4 Single-Shot Duels

The trend of modern warfare seems to be towards smaller numbers of more lethal (and expensive) weapons. The situation could eventually arise where two opposed parties have a single weapon each, with the associated problem of deciding when to launch it. If the weapon is launched too early, it is likely to miss its target. On the other hand, waiting too long has its own perils. The situation is in effect a duel where lethality increases as time passes.

We first consider the situation where both sides have the same lethality function. The probability of killing the target begins at 0 and gradually increases to 1 as the players approach each other. We assume that only the survival of player 2 is important, and use the probability that he is killed as the payoff. This situation is not symmetric—player 1 is indifferent to his own fate except that he wants to survive long enough to kill player 2. The reader may feel that a more realistic formulation would be a symmetric situation where each player thinks of himself as being worth b units and his opponent as being worth c units, with c smaller than b for both players. Unfortunately this viewpoint makes the game non-zero sum. Intuitively, the two players ought to consider calling the whole thing off if $b > c$, or to "cooperation" in a mutual destruction pact if $b < c$. To have a zero sum situation where such cooperation is pointless, one must either consider the asymmetric case, as we will do here, or else assume $b = c$, as in exercise 10. The asymmetric assumption is natural when player 1 is a single-mission weapon system that will be useless after the encounter in any case, or perhaps a suicide bomber.

Duels can be either "noisy" or "silent", the issue being whether an unsuccessful launch is detectable by the other party. In a noisy duel, the party making an unsuccessful launch is doomed because his opponent can wait until retribution is certain. That kind of waiting is impossible in a silent duel because neither player gets any information about the other player's launch time. We first consider the silent case. In a silent duel, a strategy for each player consists of choosing a launch time, or equivalently choosing a hit probability. Letting x and y be the hit probabilities chosen by player 1 and player 2, respectively, the payoff in the silent, asymmetric case is

$$A(x,y) = \begin{cases} x & \text{if } \; x \le y \\ x(1-y) & \text{if } \; x > y \end{cases} \tag{5.4-1}$$

Selecting $x = 1$ is feasible, but not optimal, for player 1. A good counter to $x = 1$ would be $y = 0.9$, since $A(1,0.9) = 0.1$. Setting $y = 0.999$ would be even better, but setting $y = 1$ would be a disaster for player 2 because the payoff would be 1. We have a game on the unit square, but the function $A(x,y)$ is not continuous across the line $y = x$.

Theorem 5.1 does not apply, but the silent duel turns out to be solvable in spite of the discontinuity in $A(x,y)$. Actually the game is solvable <u>because</u> of the discontinuity, since the technique that will be used would not work if $A(x,y)$ were continuous.

We will use the bare hands technique to solve the game. We begin by postulating a reasonable form for a mixed strategy for player 1, together with some properties that we expect if it is optimal. The general form of player 1's cumulative distribution function (CDF) is shown in Fig. 5.4. Player 1 never fires when $x < a$, and there is a positive probability p of waiting until $x = 1$.

The best values for a and p, as well as the form of $F(x)$ for $a \le x < 1$, need to be established. Let $E(F,y)$ be the payoff if player 2 chooses y while player 1 uses the CDF $F()$. Player 2 should never choose $y < a$ against $F()$ because it would be better

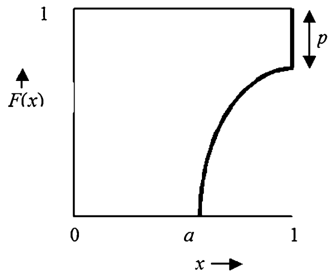


Fig. 5.4 Player 1's optimal cumulative distribution function

to wait a bit longer. We will attempt to choose a, p, and $F()$ to make $E(F,y)$ a constant for $a \le y \le 1$; that is, player 1's strategy is expected to be equalizing over that interval. We have

$$E(F,y) = \int_a^y xF'(x)dx + \int_y^1 x(1-y)F'(x)dx + p(1-y); \quad a \le y \le 1. \tag{5.4-2}$$

Equation (5.4-2) is an integral equation involving the unknown derivative $F'()$. In an attempt to simplify it, we take another derivative with respect to y, getting 0 on the left-hand side because $E(F,y)$ is supposed to be constant:

$$\begin{aligned} 0 &= yF'(y) - y(1-y)F'(y) - \int_y^1 xF'(x)dx - p \\ &= y^2F'(y) - \int_y^1 xF'(x)dx - p; \quad a \le y \le 1. \end{aligned} \tag{5.4-3}$$

The integral persists, so one more derivative is necessary:

$$0 = 2yF'(y) + y^2F''(y) + yF'(y) = 3yF'(y) + y^2F''(y); \quad a \le y \le 1. \tag{5.4-4}$$

Equation (5.4-4) is an ordinary differential equation involving the first and second derivatives of $F()$. The general solution is $F'(y) = ky^{-3}$, where k is an arbitrary constant. As long as $F'(y)$ is of this form, the first derivative of $E(F,y)$ with respect to y will be a constant. To ensure that the constant is 0, it suffices to require that (5.4-3) hold at any point in the stated interval. Evaluating (5.4-3) at $y = 1$, we see that the first derivative is 0 if we set $k = p$. This ensures that $E(F,y)$ is constant for all y in the interval. Evaluating (5.4-2) at $y = 1$, we see that the constant is $v = \int_a^1 y\left(ky^{-3}\right)dy = k\left(a^{-1} - 1\right)$. Because $F()$ must be a probability distribution, we must also have

$$\int_a^1 F'(y)dy = 0.5k\left(a^{-2} - 1\right) = 1 - p \tag{5.4-5}$$

where the last equality is because of the jump of magnitude p at $y = 1$. We now identify k with p, solve (5.4-5) for p in terms of a, and substitute the result into v. The result is $v = 2(a^{-1} - 1)/(a^{-2} - 1)$. The parameter a should be selected to maximize v. Equating the derivative to 0 reveals that $a = \sqrt{2} - 1 = v = 0.414$ and $p = k = 1/\left(2 + \sqrt{2}\right) = 0.293$. We have found a mixed strategy that guarantees a payoff of at least 0.414 for player 1.

To finish solving the game we must show that player 2 can also guarantee the same v. Let $G(y)$ be player 2's mixed strategy over some interval $[b,1]$. Then the payoff when player 1 uses x against player 2's mixed strategy is

$$E(x,G) = \int_0^1 A(x,y)dG(y) = \int_b^x x(1-y)G'(y)dy + \int_x^1 xG'(y)dy + xq; \quad b \le x \le 1 \tag{5.4-6}$$

where q is the probability that player 2 waits until the very end. Player 2 wants $E(x,G)$ to be a small constant over the interval $b \le x \le 1$. We leave it as an exercise to show that the smallest possible constant is $\sqrt{2} - 1$, achieved by making $b = \sqrt{2} - 1$, $q = 0$, and $G'(y) = \left(\sqrt{2} - 1\right)y^{-3}$ for $b \le y \le 1$. Since player 2 can guarantee the same payoff as player 1, the game is solved.

This silent duel is our first example of a game where a continuum of strategies is employed. Note that the value of the game is smaller than 0.5, even though both sides are identically armed. Player 2 has a better than even chance of survival in spite of the fact that both players have the same lethality function—there exists no strategy for player 1 that will kill player 2 with a probability that exceeds 0.414.

Now consider the noisy version of the same duel. In the noisy version, a strategy x has the interpretation "fire at x unless player 2 fires first and misses, in which case wait until the hit probability is 1", and similarly for player 2. The revised payoff function is

$$A(x,y) = \begin{cases} x & \text{if } x \le y \\ 1 - y & \text{if } x > y \end{cases}. \tag{5.4-7}$$

We leave it as an exercise to show that the noisy game has a saddle point at $x = 0.5$, $y = 0.5$.

Duels were the object of considerable attention at the RAND Corporation just after World War II. Results are known where there are multiple shots and where the accuracy functions are not equal. Dresher (1961) provides a comprehensive summary.

5.5 Bilinear Games

Consider the payoff function $A(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}x_i y_j$. In Chap. 3 we interpreted **x** and **y** as mixed strategies over finite sets of alternatives, but there is no mathematical reason why **x** and **y** have to be interpreted in that way. We could also say that **x** and **y** are pure strategy vectors constrained to lie in sets X and Y, where each of those sets is a simplex—a set of nonnegative vectors whose components sum to 1—and that a solution in pure strategies always exists. Theorem 3.2-1 can thus be interpreted in two ways: it is a proof that finite games always have saddle points in mixed strategies, but it is also a proof that a particular kind of infinite game has a saddle point in pure strategies. A bilinear game retains the same payoff function, but generalizes the sets X and Y to be arbitrary nonempty convex sets that are closed and bounded. A simplex is an example of such a set, but there are others.

In his generalized minmax Theorem, von Neumann (1937) shows that bilinear games have saddle points as long as m and n are finite, thus generalizing Theorem 3.2-1. We will not reproduce his proof, but it is instructive to prove the theorem for the special case where X and Y are polygons having K and L extreme points, respectively. Since X is convex, any point $x \in X$ can be represented as a linear combination of the polygonal extreme points $\mathbf{x}_1, \ldots, \mathbf{x}_K$:

$$\mathbf{x} = \sum_{k=1}^{K} \mathbf{x}_k u_k, \tag{5.5-1}$$

where **u** is a nonnegative weighting vector whose components sum to 1. Note that a weighting vector has exactly the properties of a probability distribution. Similarly **y** can be represented as a linear combination of the L extreme points of Y. Letting **v** be the required weighting vector for **y** and substituting into the expression for $A(\mathbf{x},\mathbf{y})$, we have

$$\begin{aligned} A(\mathbf{x},\mathbf{y}) &= \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}x_i y_j \\ &= \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}\left(\sum_{k=1}^{K} x_{ki}u_k\right)\left(\sum_{l=1}^{L} y_{lj}v_j\right) = \sum_{k=1}^{K}\sum_{l=1}^{L} u_k v_l d_{kl}, \end{aligned} \tag{5.5-2}$$

where $d_{kl} \equiv \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij}x_{ki}y_{lj}$. In these expressions x_i is the i^{th} component of **x**, x_{ki} is the i^{th} component of the k^{th} extreme point of X, and similarly for y_j and y_{lj}. The rearrangement of terms and factors in the last equality of (5.5-2) makes it clear that the payoff in terms of the weighting vectors **u** and **v** is simply the expected payoff in a $K \times L$ game with payoff matrix **d**. Theorem 3.2-1 guarantees that there are optimal weighting vectors for that $K \times L$ game. Given the optimal weighting vectors $\mathbf{u}^*$ and $\mathbf{v}^*$, we can reconstruct the optimal pure strategies $\mathbf{x}^*$ and $\mathbf{y}^*$ by using (5.5-1) and its counterpart for player 2. We have thus constructively proved that a saddle point exists.

A polygon can also be defined as the set of points that satisfy a system of linear constraints. Suppose that player 1's polygon consists of nonnegative m-vectors $\mathbf{x}$ that satisfy the constraints $\sum_{i=1}^{m} x_i b_{ik} \le B_k$ for $k = 1, \ldots, K$, and that player 2's polygon consists of nonnegative n-vectors $\mathbf{y}$ that satisfy the constraints $\sum_{j=1}^{n} y_j c_{jl} \ge C_l$ for $l = 1, \ldots, L$. Then the solution of the bilinear game is available from the following linear program LP

$$\begin{aligned}
&\max_{\mathbf{x}\ge 0,\mathbf{u}\ge 0} \sum_{l=1}^{L} u_l C_l \\
&\text{subject to} \sum_{i=1}^{m} x_i b_{ik} \le B_k; \quad k = 1, \ldots, K \\
&\sum_{l=1}^{L} u_l c_{jl} - \sum_{i=1}^{m} x_i a_{ij} \le 0; \quad j = 1, \ldots, n
\end{aligned} \tag{5.5-3}$$

The components of $\mathbf{x}$ have the usual interpretation. To interpret $\mathbf{u}$, first note that player 1 would like to have the constraint bounds $\mathbf{C}$ be as large as possible, since the components appear in the objective function of LP. Variable u_l is the value to player 1 of a unit increase in C_l. The objective function of LP is a lower bound on $A(\mathbf{x},\mathbf{y})$ that player 1 can guarantee because

$$\sum_{l=1}^{L} u_l C_l \le \sum_{l=1}^{L} u_l \sum_{j=1}^{n} y_j c_{jl} \le \sum_{j=1}^{n} y_j \sum_{i=1}^{m} x_i a_{ij}.$$

The first inequality is true as long as $\mathbf{y}$ is feasible, the second because of the constraints of LP, and the right-hand side is $A(\mathbf{x},\mathbf{y})$.

Alternatively player 2 could solve LP′, the dual of LP, which is

$$\begin{aligned}
&\min_{\mathbf{y}\ge 0,\mathbf{v}\ge 0} \sum_{k=1}^{K} v_k B_k \\
&\text{subject to} \sum_{j=1}^{n} y_l c_{jl} \ge C_l; \quad l = 1, \ldots, L \\
&\sum_{k=1}^{K} v_k b_{ik} - \sum_{j=1}^{n} y_j a_{ij} \ge 0; \quad i = 1, \ldots, m
\end{aligned} \tag{5.5-4}$$

Here v_k is the value (to player 1) of a unit increase in B_k. One could also say that a unit decrease in B_k is worth v_k to player 2. The objective function of LP′ is an upper bound on $A(\mathbf{x},\mathbf{y})$ that player 2 can guarantee as long as $\mathbf{x}$ is feasible (exercise 22).

Since LP and LP′ are duals, the optimized objective function of either LP or LP′ is the value of the game, player 1's optimal strategy is the optimized $\mathbf{x}$ from (5.5-3), and player 2's optimal strategy is the optimized $\mathbf{y}$ from (5.5-4). Since neither $\mathbf{x}$ nor $\mathbf{y}$ is necessarily required to sum to 1 in LP or LP′, Theorem 3.2-1 is a special case where those constraints are included (exercise 21). The "constrained" games of Charnes (1953) are also a special case. See Dresher et al. (1950), McKinsey (1952), or Karlin (1959) for other methods of solving bilinear games.

Example 1 (air marshals) Suppose $A(\mathbf{x},\mathbf{y}) \equiv \sum_{k=1}^{n} a_k x_k y_k$ in a problem where player 2 assigns air marshals to flights in the hope of intercepting hijackers. If player 1 puts his single hijacker on flight k, and if there is no air marshal assigned to that flight, then the payoff to player 1 is a_k, the number of passengers on the flight. If an air marshal is present, then he foils the hijack and the payoff to player 1 is 0. The off-diagonal terms can all be taken to be 0, so $A(\mathbf{x},\mathbf{y})$ is expressed as a single sum. Variable x_k is the probability that player 1 chooses flight k, variable y_k is the probability that player 2 does not assign an air marshal to flight k, and the average payoff is the expected number of passengers inconvenienced, robbed or killed by the hijacker. The simplest case would be where player 2 has a fixed number m of air marshals, in which case the single constraint on $\mathbf{y}$ would be $\sum_{k=1}^{m} y_k \geq n - m$. In that case it is not hard to show after an examination of LP that $v_k x_k$ should be independent of k and that that value of the game is $(n-m)\Big/\sum_{k=1}^{n} v_k^{-1}$. There may be other scheduling constraints, but, as long as they are linear, we have a bilinear game that can be solved with either LP or LP′.

Bilinear games are closely related to "separable" games, which are games on the unit square where the payoff is

$$A(x,y) \equiv \sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij} r_i(x) s_j(y) \tag{5.5-5}$$

As x moves from 0 to 1, the vector $\mathbf{r}(x)$ traces out a path in m-dimensional Euclidean space, any point of which is a feasible strategy for player 1. Let R be the convex hull of that path, and similarly let S be the convex hull in n-space of player 2's traced path. Also let random variable U represent player 1's choice of x, and let V represent player 2's choice of y. Then the expected payoff is

$$E(A(U,V)) = E\left(\sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij} r_i(U) s_j(V)\right) = \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij} E(r_i(U)) E\big(s_j(V)\big)$$
$$\equiv \sum_{i=1}^{m}\sum_{j=1}^{n} a_{ij} r_i s_j \qquad (5.5\text{-}6)$$

where r_i and s_j are by definition the expected values of the associated random variables. The central equality in (5.5-6) is because the expected value of a sum is the sum of the expected values, and also because the expected value of a product of independent random variables is the product of the expected values—$r_i(U)$ and $s_j(V)$ are independent because U is chosen by player 1 and V is chosen by player 2, neither knowing anything about the choice of the other. The right-hand-side of (5.5-6) is the payoff in a bilinear game where the constraining convex sets are R and S. Thus separable games are essentially bilinear games.

If $m = n = 1$ in a separable game, then the constraining sets are simply intervals. Player 1 can choose r anywhere in the interval $R = [r_{\min}, r_{\max}]$ and player 2 can choose s anywhere in the interval $S = [s_{\min}, s_{\max}]$. Any mean in either interval can be obtained by mixing the two extreme points, so the game is equivalent to a 2×2 game where the strategies are the extreme points. Furthermore, because of the multiplicative nature of the objective function, that 2×2 game will always have a saddle point.

Example 2 Suppose $m = n = 1$, $R = [-1, 1]$, $S = [2,3]$, and $a_{11} = -3$. The 2×2 payoff matrix $\mathbf{d}$ of (5.5-2) is $\begin{bmatrix} 6 & 9 \\ -6 & -9 \end{bmatrix}$, which has a saddle point with value 6. Player 1 always chooses $r_{\min}$ and player 2 always chooses $s_{\min}$.

Example 3 Suppose that $m = n = 2$ in a separable game. Assume $s_1(y) = y$ and $s_2(y) = 1 - y$, so S is the line connecting the two points (0,1) and (1,0). Also assume that $r_1(x) = x$, and that $r_2(x)$ is the "function" shown in Fig. 5.5. That discontinuous function increases from 0.5 to 1 as x goes from 0 to 0.5, then jumps to 0 and stays there until x gets to 1. We take the "function" to have both values when $x = 0.5$. The meaning of that statement is simply that player 1 has his choice. In the present context it is not important that the "function" be uniquely defined, but only that the set of alternatives for player 1 be made clear at every point. To emphasize that point, the figure displays another line segment slanting downwards, every point of which is an alternative for player 1. That segment has no operational significance because it lies entirely within the convex hull R that is shown as the dashed-solid border in Fig. 5.5. Since R is a polygon with four extreme points (corners), player 1 effectively has $K = 4$ four pure strategies: (0.5,0) (0,0.5), (0.5,1), and (1,0). Player 2 has only $L = 2$ pure strategies, namely (0,1) and (1,0). To complete the description of the game, take $\mathbf{a} = \begin{bmatrix} 1 & 2 \\ 3 & 0 \end{bmatrix}$ in (5.5-6). Listing each player's strategies in the order given above, we can then compute the 4×2 payoff matrix

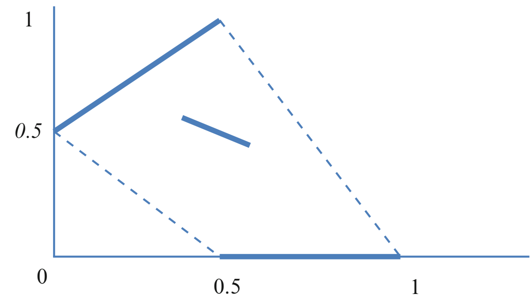


Fig. 5.5 The convex *dashed-solid* border shows the feasible set R for player 1

$\mathbf{d} = \begin{bmatrix} 1 & 0.5 \\ 0 & 1.5 \\ 1 & 3.5 \\ 2 & 1 \end{bmatrix}$ as in (5.5-2). For example $d_{32} = (0.5)(1) + (1)(3) = 3.5$. The first two rows of **d** are dominated by row 3, and the value of the remaining 2×2 game is 12/7. Player 1's optimal mixed strategy in **d** is (0,0,2,5)/7, so player 1's optimal mixed strategy in the separable game is to select corner (0.5,1) with probability 2/7, or corner (1,0) with probability 5/7. Player 2's optimal strategy in **d** is (5,2)/7, which means that he chooses (0,1) with probability 5/7 or (1,0) with probability 2/7. The solution of this game is now complete. Sheet "Separable" of the TPZS workbook permits the reader to continue this example by calculating **d** for different assumptions about **a**.

5.6 Submarines and Machine Guns

What may be the first serious application of game theory to military tactics is recounted by Morse and Kimball (1950) in their book on World War II analytical methods and problems. The application concerns a submarine that would like to remain submerged all the time as it transits a channel, since it is undetectable when submerged, but which is forced to surface part of the time because of battery limitations. The submarine must endure the possibility that it will be on the surface when an enemy aircraft is present, an aircraft that the submarine cannot detect in sufficient time to submerge. The aircraft's endurance limits it to searching only one point in the channel. What are the best tactics for the two sides?

Let M be the length of the channel, and assume that the submarine can only remain submerged for a smaller length a. The period of submergence need not be continuous, so the operative constraint on the submarine's tactics is that $\int_0^M y(t)dt = a$, where $y(t)$ is the probability of being submerged at point t. The aircraft chooses a point in the channel at which to conduct a patrol, but is not guaranteed success even if the submarine is surfaced at the chosen point. Let $P(t)$ be the conditional probability of detecting the submarine by an aircraft patrol at point t, given that the submarine is surfaced at that point. This probability varies with t because of variations in channel width or background noise level. Let $x(t)$ be the probability density of the aircraft's choice, so that $\int_0^M x(t)dt = 1$. The unconditional probability of detecting the submarine is then

$$d \equiv \int_0^M x(t)P(t)(1 - y(t))dt, \tag{5.6-1}$$

to be maximized by the aircraft and minimized by the submarine.

This game is conceptually simple enough that we can try the bare hands approach to finding a solution, the general idea being that each player should act in an equalizing manner. A natural tactic for the aircraft is to make $x(t)$ positive only at points where $P(t)$ is larger than some constant K, call the set of points $S(K)$. An equalizing strategy will make $x(t)P(t)$ a constant $C(K)$ within that set (see Fig. 5.6-1).

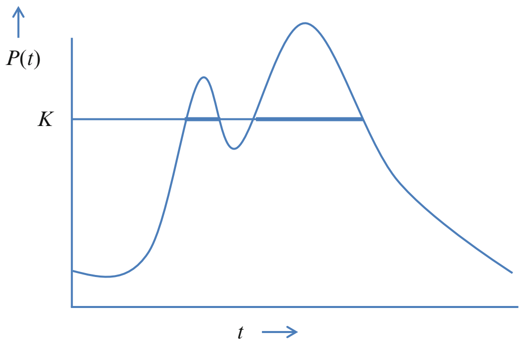


Fig. 5.6-1 The set $S(K)$ is shown as two heavy horizontal segments. The length-measure $L(K)$ is the sum of the two segment lengths

The constant $C(K)$ must satisfy $C(K)\int_{S(K)} P(t)^{-1}dt = 1$. If $L(K)$ is the length-measure of $S(K)$, we then have

$$d = \int_{S(K)} C(K)(1 - y(t))dt = C(K)\left(L(K) - \int_{S(K)} y(t)dt\right) \geq C(K)(L(K) - a)$$

The aircraft can guarantee that the detection probability will be at least $C(K)(L(K) - a)$ as long as $y(t)$ meets the integral constraint imposed on it. The aircraft should choose K to make this lower bound on the detection probability as large as possible, but let us defer that maximization for a moment.

Just as the aircraft patrols in such a manner that it doesn't matter where the submarine submerges, the submarine can submerge in such a manner that it doesn't matter where the aircraft patrols. If $P(t)$ is smaller than some constant H; that is, at points that are not in $S(H)$, the submarine never submerges ($y(t) = 0$). At points t in $S(H)$, the submarine chooses $y(t)$ so that the product $P(t)(1 - y(t))$ is H. If the submarine does this, then d cannot exceed H. The submarine would like to make H as small as possible, but really has no choice because H must meet the constraint

$$\int_{S(H)} \left(1 - HP(t)^{-1}\right) dt = L(H) - H/C(H) = a,$$

which can be expressed as $H = C(H)(L(H) - a)$.

The submarine's upper bound is H, but the aircraft can always choose K to be H, which will make the aircraft's lower bound equal to the submarines upper bound. Since lower bounds never exceed upper bounds, the best value of K must be H. We have evidently guessed the solution of this game. Either player can guarantee a detection probability of H, no matter what the other does.

Example Suppose the submarine leaves a narrowly constricted port located at $t = 0$ through a channel that is $M = 10$ miles long. Take $P(t) = 1 - t/M$ throughout the channel, so the detection probability decreases gradually as the submarine leaves port and the channel gets increasingly wide, falling to 0 at the end of the channel. The submarine can only stay submerged for $a = 4.781$ miles. To solve this game we first establish that $L(H) = M(1 - H)$ and (after evaluating the necessary integral) $C(H)^{-1} = -M\ln(H)$. The equation that must be solved for H is therefore $M(1 - H + H\ln(H)) = a$. The solution is $H = 0.2$.[2] Figure 5.6-2 shows the optimal strategies for the submarine and the aircraft, or see sheet "Submarine" of the TPZS workbook for further detail. It may seem odd that $x(t)$ is an increasing function of t, since the conditional detection probability decreases with t. The intuitive explanation is that the submarine is less likely to be submerged when t is large. The value of the game is 0.2.

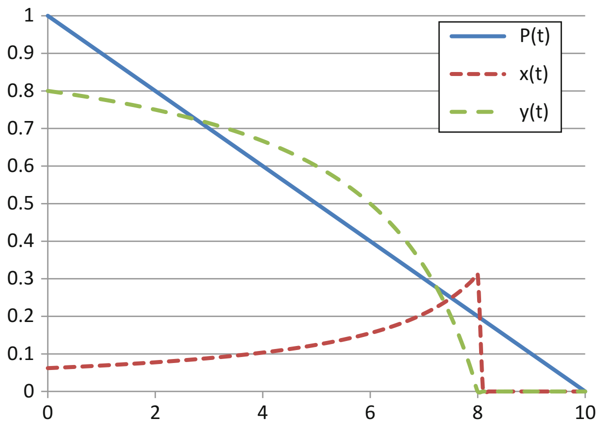


Fig. 5.6-2 Aircraft (x(t)) and submarine (y(t)) strategies when a = 4.781

Thomas (1964), in reviewing applications of game theory up to 1964, refers to the aircraft versus submarine game described above, but gives more significance to

[2] The equation is transcendental, so, as the reader may have already guessed, the value for a was actually obtained by plugging in 0.2 for H. This is the advantage of writing textbooks instead of working real problems—you can make up the data to be convenient.

what we might call the fighter versus bomber game. When a fighter attacks a bomber, there is an interval of time [0,T] over which the guns of both parties become increasingly effective as the range between the two decreases. There is an element of dueling about this, since each party is tempted to begin firing early in order to kill the other before the other can shoot back effectively, but is also tempted to wait until his own guns become more effective. However, this duel is not of the single shot kind covered in the previous section. Instead, each side is armed with machine guns that could well fire hundreds of bullets over the length of the engagement. Interest in the game just after World War II came mainly from aircraft designers, who had to decide what kind of machine guns to use for armament. It is not obvious, for example, whether to include a highly lethal gun that will quickly run out of ammunition, or a less lethal gun that can operate for a longer period of time. The maximum rate of fire is also involved. Tactical questions must be settled before armaments can be evaluated, so even a very abstract version of such engagements can be informative.

Here is an abstraction where each side's strategy is interpreted as a rate of fire. The constraints for the fighter are that $\int_0^T x(t)dt = A_1$, where A_1 is a magazine capacity, and in addition $x(t)$ can never exceed the maximum rate of fire R_1. There are similar constraints for the bomber's $y(t)$, except that the amount of ammunition is A_2 and the maximum rate of fire is R_2. If two lethality functions are given, the payoff can be written as an integral involving both $x(t)$ and $y(t)$, a function of functions as in the aircraft versus submarine game. Questions of optimal tactics can then be investigated, and the dependence of the game value on various parameters can be explored. We will not give the details here, but instead refer the reader to Thomas (1964), who is also worth reading to get his perspective on the uses of game theory in general up to that time. One more point is worthy of mention. Both sides are at risk here, so there are two kill probabilities involved. A TPZS game can only have one payoff, so, as in the duels considered in Sect. 5.4, there is an issue of how these two numbers are to be combined. Given that the bomber at the time was likely to be carrying nuclear bombs, the payoff was usually taken to be the probability of killing the bomber.

The generalized fighter versus bomber problem was an object of intense analysis by theorists in the late 1940s. Karlin (1959) gives a summary formulation that is sufficiently general to include both of the games described above, referring to them as "infinite classical games not played over the unit square".

5.7 Exercises

1. Sheet "Poker" of the TPZS workbook shows a 9×9 approximation of Continuous Poker as described in Sect. 5.1, using (5.1-1) for the payoff. The continuous version of this game has a saddle point that is stated in exercise 2 below.

a) Solve the discrete game using Solver when $a = b = 1$. You should find that the optimal strategy for the column player is mixed, and can conclude that the discrete game does not have a saddle point. How close is the value of the discrete game to the value of the continuous game?
b) Solve the game using Solver when $a = 1$, $b = 2$. This discrete game has a saddle point. Where is it?

2. Show that the payoff function given in (5.1-1) is concave-convex. The difficult part will be to show that the derivatives are continuous across the line $x = y$. Then show that the saddle point occurs in the region $x < y$, with $y^* = b/(b + 2a)$, $x^* = (y^*)^2$, and $v = -ax^*$. This optimal strategy for player 1 does not involve bluffing. Karlin (1959) discovers that there are also optimal strategies that do involve bluffing, although of course all of these alternate optima lead to the same game value.

3. Consider the Logistics game where $n = 3$, $f_1(y_1) = 3/(1+y_1)$, $f_2(y_2) = 2/(1+y_2)$, and $f_3(y_3) = 1/(1 + 5y_3)$. Sketch v as a function of c for $0.2 \le v \le 2$.

4. For the Logistics game in exercise 3, find the optimal strategy for player 1 when $c = 3$. Hint: $c = 3$ corresponds to $v = 1$, and $\boldsymbol{y}^* = (2,1,0)$ when $v = 1$ from exercise 3.

5. Logistics games can be generalized so that the payoff is $f_i\left(\sum_{j=1}^{J} a_{ij}y_j\right)$, where a_{ij} is the effectiveness coefficient for forces of type j in area i. The motivation for doing this might be that that some of player 2's activities have effects in multiple areas, as in mine warfare where a given minesweeper configuration might be effective in sweeping multiple types of mine. The constraint on the y vector is now $\sum_{j=1}^{J} y_j \le c$, where J is the number of different kinds of force available to player 2. The functions $f_i()$ are assumed convex and decreasing, as in ordinary Logistics games. Construct a linear program that will find the minimal value of c for which the value of the game is v. Hint: First write a non-linear program, and then take advantage of the fact that decreasing functions have inverses.

6. Suppose that the functions $f_i()$ of exercise 5 are all the same function $f()$, and that $c = 1$ (any other value of c could be accommodated by redefining a_{ij}). Let u^*, $\mathbf{x}^*$, and $\mathbf{y}^*$ be the solution of the ordinary matrix game (a_{ij}) where player 1 is minimizing and player 2 is maximizing (note the role reversal). Show that the solution of the generalized Logistics game is $f(u^*)$, $\mathbf{x}^*$, $\mathbf{y}^*$. Hint: Let $u_i = \sum_j a_{ij}y_j$, so that the payoff is $\sum_i x_i f(u_i)$. Since $f()$ is convex decreasing, there is some $k \ge 0$, possibly depending on u^*, such that $f(u_i) \ge f(u^*) - k(u_i - u^*)$.

7. Consider the "Logistics" game where $c = 2$, $n = 2$, $f_1(y_1) = 1 - y_1/3$, and $f_2(y_2) = 1$ for $y_2 < 1$, or 0 for $y_2 \ge 1$. The quotes are necessary because $f_2(y_2)$

is not convex for $0 \leq y_2 \leq 2$. Show that $\min_y \max_i f_i(y_i) = 2/3$, but that player 2 can hold the payoff to 1/2 by flipping a coin to use either 0 or 1 unit in region 2.

8. Verify that player 2 can guarantee $E(x, G) \leq \sqrt{2} - 1$ for the silent asymmetric duel considered in Sect. 5.4.

9. Verify that the noisy asymmetric duel considered in Sect. 5.4 has a saddle point.

10. Solve the silent symmetric duel discussed in Sect. 5.4. For this duel,

$$A(x, y) = \begin{cases} x(1) + (1 - x)(y)(-1) = x - y + xy & \text{if } x < y \\ 0 & \text{if } x = y \\ y(-1) + (1 - y)(x)(1) = x - y - xy & \text{if } x > y \end{cases}$$

Hint: Note that $A(x,y) = -A(y,x)$. Since the game is symmetric, the value is 0 and it suffices to find a strategy for player 1 that guarantees $\mathrm{E}(F,y) \geq 0$ for all y.

11. Consider the continuous attack-defense game $A(\mathbf{x}, \mathbf{y}) \equiv \sum_{i=1}^{3} k_i \max(0, x_i - y_i)$, where $\mathbf{k} \equiv (1,2,3)$. Player 1's strategy $\mathbf{x}$ is constrained so that $\mathbf{x}$ is nonnegative and $x_1 + x_2 + x_3 \leq 4$, and player 2's strategy is constrained so that $\mathbf{y}$ is nonnegative and $y_1 + y_2 + y_3 \leq 3$. Show that the value of the game is 6, and give the optimal pure strategy $\mathbf{y}^*$. As a point of interest, the optimal mixed strategy for player 1 is $\mathbf{p} = (0, 0.6, 0.4)$, where p_i is the probability that player 1 attacks target i with all 4 units. Player 1 never attacks except in full strength, and target 2 is more likely to be attacked than the more valuable target 3. Blackett (1954) gives a general solution for games of this type.

12. Solve the game on the unit square where $A(x,y) = -2x^2 + 5y^2 + 3xy - x - 2y$.

13. Explain why the technique used in Sect. 5.2 for solving games on the unit square will not work for all continuous games. It may help to attempt to apply it to the game $A(x,y) = (x - y)^2$.

14. Solve the game on the unit square $A(x,y) = (x - y)^2$. Determine at least the value and the optimal strategy of player 2.

15. Consider a game where $A(x,y) = \sum_i p_i x_i (1 - y_i)$, where p_i is the probability that an aircraft search of cell i will detect a surfaced submarine in that cell, x_i is the probability that the aircraft searches cell i, and y_i is the probability that the submarine is submerged (hence undetectable) in cell i. The aircraft can search only one of the n cells, so $\sum_i x_i = 1$. The submarine can submerge in at most c cells on account of battery limitations, so $\sum_i y_i \leq c$ and of course $0 \leq y_i \leq 1$ because y_i is a probability. Parameter c is nonnegative, but is not required to be an integer. This is a discrete counterpart of the aircraft versus submarine

game considered in Sect. 5.6. It can be solved using the methods of Sect. 5.3. If there are four cells and $\mathbf{p} = (0, 0.1, 0.1, 0.5)$, what value of c is required to make $v = 0.05$?

16. Consider the Logistics game where $f_i(y_i) = \exp(-\alpha_i y_i)$. If the positive constants α_i are interpreted as search rates, the payoff can be interpreted as the probability that player 2 fails to detect player 1 when player 1 hides in cell i.
    a) Solve the game when $n = 2, \alpha_1 = -\ln(0.4) = 0.916, \alpha_2 = -\ln(0.6) = 0.511$, and $c = 2$.
    b) For the general case prove that $y_i{}^*\alpha_i$ is independent of i.
    c) For the general case prove that $x_i{}^*\alpha_i$ is independent of i.
    d) Find a formula for the value of the game in terms of $\alpha_1,\ldots,\alpha_n$.
17. Consider exercise 16a under the additional restriction that y_i must be an integer for all i. The revised game is not a Logistics game, and player 2's strategy may very well be mixed. In general, such games are hard to solve because of the PEG, but 16a with the integer restriction is small enough to solve as a matrix game. Do so, and compare the solution with the solution of exercise 16a.
18. Consider the game on the unit square where player 1 wins if and only if $|x - y| \leq 0.2$:

$$A(x, y) = \begin{cases} 1 & \text{if} |x - y| \leq 0.2 \\ 0 & \text{otherwise} \end{cases}.$$

    a) Is $A(x,y)$ concave in x for all y? Is it convex in y for all x?
    b) What value can player 1 guarantee by choosing x to be uniform in [0,1]?
    c) What value can player 2 guarantee by choosing y to be uniform in [0,1]?
    d) Can you guess the solution? Hint: The optimal mixed strategies are discrete.
19. Consider a game where $(x_1,\ldots,x_m)$ is a probability vector, $(y_1,\ldots,y_n)$ is in some convex set Y, and $A(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} x_i f\left(\sum_{j=1}^{n} a_{ij} y_j\right)$, where $f()$ is a convex, decreasing function. This is a generalization of a Logistics game in that the payoff in area i depends on $\mathbf{y}$, rather than just y_i, and a restriction in that $f()$ is not subscripted. Show that the value of this game is $f(u^*)$, where u^* is the optimized value of the following mathematical program:

$$\begin{aligned} &\max u \\ &\text{subject to} \sum_{j=1}^{n} a_{ij} y_j - u \geq 0; \quad i = 1, \ldots, m \\ &y \text{ in } Y. \end{aligned}$$

    If Y itself can be expressed with linear inequalities, then the program is linear. The main point here is that the precise nature of the function $f()$ turns out to be irrelevant, since optimal tactics can be determined without reference to it. One application of this is in mine warfare where x_i is the fraction of mines of type i, y_j is the amount of minesweeping of type j, $f\left(\sum_{j=1}^{n} a_{ij}y_j\right)$ is the fraction of mines of type i that survive sweeping, and $A(\mathbf{x},\mathbf{y})$ is the average number of surviving mines. The set Y might consist of nonnegative vectors $\mathbf{y}$ such that $\sum_{j=1}^{n} t_j y_j \le T$, where t_j is the amount of time required for a sweep of type j and T is the total amount of time allowed for minesweeping. The data (a_{ij}) depend on sweep widths, sweeper speeds, etc.

20. Suppose $A(x,y) = \cos(y - x)$, with $|y| \le 9$ and $|x| \le 0.2$. The value of the game is $-\cos(0.2)$. Use the bare hands technique to find the optimal strategies for the two sides.

21. By specializing (5.5-3) of Sect. 5.5, show that Theorem 3.2-1 is a special case of solving a bilinear game. You will need two inequality constraints to force the equality constraint in Theorem 3.2-1 that requires probabilities to sum to 1, so take $K = 2$, and take advantage of the fact that an unconstrained variable can be represented as the difference of two nonnegative variables.

22. Prove that the objective function of LP′ in Sect. 5.5 is indeed an upper bound on $A(\mathbf{x},\mathbf{y})$ as long as $\mathbf{x}$ is feasible. The proof can parallel the proof that the objective function of LP is a lower bound.

23. Repeat example 3 of Sect. 5.5, except assume $\mathbf{a} = \begin{bmatrix} 2 & -1 \\ -1 & 0 \end{bmatrix}$. What is the solution of the separable game? Make use of sheet "Separable" of the TPZS workbook.

24. Consider again the case where $m = n = 1$ in a separable game. This is the case where $R = [r_{\min}, r_{\max}]$ and $S = [s_{\min}, s_{\max}]$. Of course $r_{\min} \le r_{\max}$, so there are three possibilities for R:

    1. $0 \le r_{\min}$
    2. $r_{\min} < 0 < r_{\max}$
    3. $r_{\max} < 0$

    There are also three similar possibilities for S.

    a) If 0 is included in both R and S, prove that the value of the game is 0 by giving the optimal mixed strategies for the two players in a 2×2 game.
    b) There are eight more possibilities, one of which, for example, is that all four limits are nonnegative. Find the value of the game in all eight of them. All eight of the 2×2 games should have saddle points.

25. Example 2 of Sect. 5.3 concerns IED warfare. Suppose there are five road segments, and that the data for the problem are

$$\begin{bmatrix} 1 & 2 & 3 & 2 & 20 \\ 2 & 4 & 4 & 3 & 6 \\ 1 & 2 & 1 & 1 & 1 \\ 0.1 & 0.2 & 0.3 & 0.4 & 0.5 \end{bmatrix},$$

where the four rows are for data vectors **c**, **d**, **s**, and **p**. Page “IED” of the TPZS workbook shows this problem as a Logistics game.

a) Suppose you were player 2 with 17 sweepers available. Segment 5 has the most logistic traffic, and IEDs on that segment are particularly lethal, but on the other hand IEDs on that segment are quickly removed even if you don’t put any sweepers there. Would you put most of your sweepers on segment 5?
b) With 17 sweepers the best you can do is make it so that only one IED in 10 will cause any damage. How many sweepers would you need to make that 1 in 20?

Chapter 6
Blotto Games

Another such victory over the Romans, and we are undone.

Pyrrhus

6.1 The Symmetric Case

Colonel Blotto made his first appearance in *Caliban's Weekend Book* of 1924, where he faced the problem of trying to capture as many forts as possible with a limited number of divisions. Blotto's problem was made difficult because an opposing commander was at the same time distributing defensive forces among the forts in an effort to frustrate him. The class of games where each player must divide an aggregate force among several areas of contention has since been named after the Colonel.

> **Definition** In a Blotto game with $n \geq 1$ areas, the payoff is $\sum_{k=1}^{n} A_k(x_k, y_k)$.
> Player 1 chooses a vector of allocations $\mathbf{x}$ subject to $\sum_k x_k \leq f$, and player 2 chooses a vector of allocations $\mathbf{y}$ subject to $\sum_k y_k \leq g$. All allocations are required to be non-negative, and f and g are given, non-negative quantities. If the functions $A_k()$ do not depend on k, the game is called "symmetric."

The total payoff in a Blotto game is a sum of n payoffs, each of which depends only on the allocations to the subscripted area. It will be useful to think of the allocations as random variables, since there are some powerful theorems about sums of random variables. Allocations x_k and y_k will be capitalized to emphasize this when that is the intended meaning. In the symmetric case the subscript on $A()$

A. Washburn, *Two-Person Zero-Sum Games*, International Series in Operations Research & Management Science 201, DOI 10.1007/978-1-4614-9050-0_6,

will simply be omitted. In this section as well as in Sects. 6.2 and 6.3 we assume that all allocations are integer valued; the continuous case will be considered in Sect. 6.4. The principal application of Blotto games has been to problems where a group of targets is attacked and defended by missiles (Eckler & Burr, 1972), in which case $A_k(X_k,Y_k)$ is the probability that target k is destroyed and the meaning of the total payoff is "average number of targets destroyed." That work will be reviewed in Sect. 6.5.

Exercise 10 at the end of Chap. 3 is an example of a symmetric Blotto game where $f = 3$, $g = 4$, $n = 2$, and where $A(x,y) = 1$ if $x > y$, else 0. There are 4 pure strategies for player 1 and 5 for player 2, counting only those that totally consume the resources. There will be a finite number regardless of (f,g,n), so there is no question about the existence of solutions. However, the number of pure strategies increases very fast as the problem becomes larger. The number of pure strategies for player 1, even if we count only those for which f is totally consumed, is $\binom{n+f-1}{f}$, the number of combinations of $n + f - 1$ things taken f at a time. This grows so fast with n and f that some method of solution that does not begin by listing all possible strategies must be found if Blotto games are to be solvable as a practical matter.

The expected payoff in a Blotto game is

$$E\left(\sum_{k=1}^{n} A_k(X_k, Y_k)\right) = \sum_{k=1}^{n} E(A_k(X_k, Y_k)) \tag{6.1-1}$$

There may be dependence among the components of $\mathbf{X}$ and $\mathbf{Y}$, but the equality in (6.1-1) is nonetheless valid because expected values and sums commute. The equality is important because evaluation of the left-hand side requires the joint distributions of $\mathbf{X}$ and $\mathbf{Y}$, whereas the right-hand side requires only the marginal distributions of X_k and Y_k; $k = 1,\ldots,n$. This observation leads to a method for obtaining a bound on the value of the Blotto game that is enforceable by player 1. The bound is obtained by relaxing the constraint $\sum_k X_k \le f$ to require only $\sum_k E(X_k) \le f$ and also $X_k \le f$ for all k; that is, player 1 is allowed to occasionally exceed f in total provided he does not do so on the average, and provided also that he never exceeds f in any single area. Relaxing the constraints in this manner for both players results in what we will call the "relaxed" game.

In the relaxed game, player 1's problem is merely to determine n marginal distributions of $X_1,\ldots, X_n$, rather than a joint distribution for $\mathbf{X}$. In the symmetric case we expect that player 1 will use the same marginal distribution in all areas, in which case that distribution cannot have a mean exceeding f/n. Letting x_i be the probability that i units are used in the typical area (so $\mathbf{x} \equiv (x_0, \ldots, x_f)$ is the marginal distribution), we therefore require $\sum_{i=0}^{f} ix_i \le f/n$. Note the notation shift here—x_4 is now the probability of using 4 units in any area, rather than player 1's allocation to area 4. If player 1 uses $\mathbf{x}$ in every area, the average payoff in any area when player 2's allocation is j is $\sum_{i=0}^{f} x_i A(i,j)$. If that payoff is bounded below by some linear function $a - bj$ for all feasible values of j, with $b \geq 0$, then

$$\sum\nolimits_k E(A(X_k, Y_k)) \geq \sum\nolimits_k E(a - bY_k) = na - bE\left(\sum\nolimits_k Y_k\right) \geq na - bg, \qquad (6.1\text{-}2)$$

where the last inequality is because player 2 cannot use more than g units in total. The quantity on the right-hand side of (6.1-2) does not depend on how player 2 allocates his force, so player 1 can regard it as the objective function in linear program LP1:

$$\begin{aligned} &\max\nolimits_{a,b\geq 0,\mathbf{x}\geq 0} na - bg \\ \text{subject to :}\quad &\sum_{i=0}^{f} x_i A(i,j) \geq a - bj; \quad j = 0, \ldots, g, \\ &\sum_{i=0}^{f} i x_i \leq f/n, \\ &\sum_{i=0}^{f} x_i = 1. \end{aligned}$$

LP1 has f +3 variables, including a and b, and g +3 constraints. Let v be the maximized objective function and $\mathbf{x}^*$ the optimal marginal distribution. We can then say that

If the rules for player 1 are relaxed as above in a symmetric game, he can guarantee v (per area) by using $\mathbf{x}^*$ *in every area.*

If there exists some joint distribution for $\mathbf{X}$ such that $\mathbf{x}^*$ is the marginal distribution of X_k for all k and also $\sum_k X_k = f$ always, rather than just on the average, then we say that $\mathbf{x}^*$ is *playable* in the Blotto game. Playability is discussed further in the next section, but at this point we can at least say that

If $\mathbf{x}^*$ *is playable in the (unrelaxed) Blotto game, then player 1 can guarantee v by playing* $\mathbf{x}^*$ in every area.

We can also consider relaxing the rules for player 2 in a similar manner, in which case we are led to linear program LP2 for player 2:

$$\begin{aligned} &\min\nolimits_{c,d\geq 0,\mathbf{y}\geq 0} nc + df \\ \text{subject to :}\quad &\sum_{j=0}^{g} A(i,j) y_j \leq c + di; \quad i = 0, \ldots, f, \end{aligned}$$

$$\sum_{j=0}^{g} jy_j \leq g/n,$$

$$\sum_{j=0}^{g} y_j = 1.$$

The meaning of **y** in LP2 is the probability distribution that player 2 uses in every area. LP2 has g +3 variables and f +3 constraints. It can be verified that LP2 and LP1 are dual linear programs, so the objective functions are necessarily equal. In addition to making one-sided statements about player 2 that parallel those made above about player 1, we can therefore say that

If **x*** *and* **y*** *are both playable, then the value of the symmetric Blotto game is v times the number of areas.*

Thus, subject to some reservations about playability, symmetric Blotto games can be solved by linear programming. Note that the number of variables in LP1 or LP2 is nowhere near the number of possible pure strategies. When $n = f = 10$ there are 92,378 pure strategies for player 1, but only 13 variables in LP1.

6.2 Playability

Does there exist a joint distribution for $\mathbf{X} = (X_1,\ldots,X_n)$ such that the components have given marginal distributions, and also such that $\sum_k X_k$ is constant? If so, we say that the given marginals are *playable*. In this section we permit the possibility that the marginals are not identical, even though that generality is not necessary in the symmetric case. We will first give some examples, then discuss techniques for playing specified marginals, and finally make some observations about large-scale problems.

Consider again exercise 10 at the end of Chap. 3, a symmetric Blotto game with $n = 2, f = 3$, and $g = 4$. On solving LP1 or LP2, we find that the solution of the relaxed game is $\mathbf{x}^* = (1/4, 1/4, 1/4, 1/4)$, $\mathbf{y}^* = (1/6, 1/6, 1/6, 1/2, 0)$, and $v = 0.5$. Player 1's strategy is playable—if he lets X_1 be uniform over the set {0, 1, 2, 3} and $X_2 = 3 - X_1$, then both X_1 and X_2 have the uniform distribution **x*** while always $X_1 + X_2 = 3$. Player 2's strategy is not playable; we must have $P(Y_2 = 1) = P(Y_1 = 3)$ in the Blotto game because Y_1 and Y_2 must always sum to 4, but $y_1^* \neq y_3^*$. The marginal distribution **y*** uses the correct number of units on the average (namely 2), but is nonetheless not playable when $n = 2$. It follows from the playability of **x*** that the value of the game is at least 0.5 per area, but the only way to prove that the value of the Blotto game is exactly 0.5 is by solving it as in Chap. 3.

Now "scale up" the same exercise so that $(n, f, g) = (4, 6, 8)$. Player 1 and player 2 have the same number of units per area, but there are 4 areas instead of 2. We now find that $\mathbf{x}^* = (0.4, 0.15, 0.15, 0.15, 0.15, 0, 0)$, $\mathbf{y}^* = (0.2, 0.2, 0.2, 0.2, 0.2, 0, 0, 0, 0)$, and $v = 0.6$. Notice that player 1 has a positive probability of using 4 attackers, a possibility that was forced to be 0 when $(n, f, g) = (2, 3, 4)$ because player 1 had only 3 attackers in total. Both players' strategies are now playable (playability isn't obvious for $\mathbf{x}^*$, but see below). Therefore the value of the 4-area Blotto game is 2.4. That conclusion is achieved despite that fact that the normal form of the 4-area game has 84 rows and 165 columns. For an even more impressive example, scale up to $(n, f, g) = (20, 30, 40)$. The optimal strategies are unchanged, except for being extended with 0's for higher numbered probabilities, so the value of the 20-area game is 6. Since $n\mathbf{x}^*$ and $n\mathbf{y}^*$ are both vectors of integers, playability can be accomplished by simply putting 20 appropriately labeled balls in a hat and drawing without replacement. Player 1 includes 8 balls with "0" on them, 3 balls with "1" on them, etc. Furthermore, the same strategies are playable if n is any integer multiple of 20, or for that matter any integer multiple of 4. Apparently playability is easiest to establish in large problems, rather than in small ones. This is encouraging because it is the large games that are most difficult to solve directly.

The fact that $\mathbf{x}^*$ above is playable when $n = 4$ can be demonstrated using a technique that resembles the use of antithetic variables in variance reduction. Consider the general situation where each of n random variables X_k has a given distribution, not necessarily the same for all k. Let $F_k()$ be the cumulative distribution function (CDF) for X_k, and let U be a uniform [0,1] random number. Then $X_k = {F_k}^{-1}(U)$ has CDF $F_k()$, where the superscript "-1" denotes an inverse function—this is the basis of the "inverse transform" method of generating random variable X_k from a standard uniform random variable U. Let m be an area index different from k. Since $1 - U$ is also uniform [0,1], $X_m = F_m^{-1}(1 - U)$ has CDF $F_m()$. X_k and X_m both depend on the same random number U, so they will not in general be independent, but there is no requirement for them to be independent. In fact, since small values of X_k are associated with large values of X_m, there is reason to hope that the variance of $X_k + X_m$ will be small, and that repetition of the idea might result in the variance of $\sum_{k=1}^{n} X_k$ being 0, which is the goal. Specifically, here is the antithetic method for generating $\mathbf{X}$, an n-vector of random variables with specified marginal CDFs. It begins by placing all n CDFs on a list.

1. (Selection): If only one CDF remains on the list, go to Step 3. Otherwise, suppose $F_a()$ and $F_b()$ have the largest variances among those on the list.
2. (Reduction): Let U_{ab} be uniform in [0,1], and let $F_{ab}()$ be the CDF of the random variable ${F_a}^{-1}(U_{ab}) + {F_b}^{-1}(1 - U_{ab})$. Cross out $F_a()$ and $F_b()$ from the list and insert $F_{ab}()$, thus reducing the number of CDFs on the list by 1. Go to Step 1.
3. (Sampling): If $F_c()$ is the last CDF, then X_c is the sum of two random variables X_a and X_b. Sample X_c from $F_c()$, and consider the level-set of values u such that $X_c = {F_a}^{-1}(u) + {F_b}^{-1}(1 - u)$. Select a uniform random value U_{ab} from within

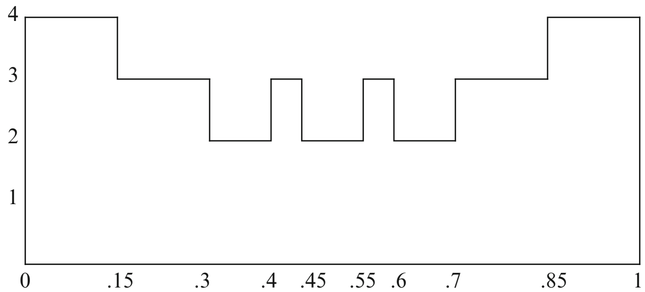


Fig. 6.2 Developing the CDF $F_{12}()$

that set, and let $X_a = F_a^{-1}(U_{ab})$ and $X_b = F_b^{-1}(1 - U_{ab})$. If either a or b represent sums, repeat this procedure until all n components of $\mathbf{X}$ are determined.

It can be shown by induction that the distribution of X_k will be $F_k()$ for all k. The sum of all the random variables will be constant if and only if the last CDF is actually the CDF of a constant, in which case the method has succeeded in playing the given list of CDFs.

Example Suppose $F_k()$ is the CDF corresponding to the aforementioned probability mass function $\mathbf{x}^* = (0.4, 0.15, 0.15, 0.15, 0.15, 0, 0)$, for $k = 1,\ldots,4$. All four CDFs are the same, so select the first and second in step 1. Figure 6.2 shows $F_1^{-1}(u) + F_2^{-1}(1 - u)$ as a function of u, so X_{12} is either 2, 3, or 4. After measuring the length of each of the three level-sets, the probability mass function for X_{12} is (0, 0, 0.3, 0.4, 0.3) on the integers (0, 1, 2, 3, 4). This completes step 2, so return to step 1. Considering the three CDF's $F_{12}()$, $F_3()$, and $F_4()$, choose $F_3()$ and $F_4()$ for the next pair, discovering that $F_{34}()$ is the same as $F_{12}()$. Since $F_{12}^{-1}(u) + F_{34}^{-1}(1 - u) = 6$ for all u, the last CDF $F_{1234}()$ is simply the CDF of the constant 6, so the antithetic method is successful. Actual samples of $\mathbf{X}$ can be generated using a random number generator. If $U_{1234} = 0.78$, for example, we would have $X_{12} = F_{12}^{-1}(0.78) = 4$ and $X_{34} = F_{34}^{-1}(0.22) = 2$. We would next select a point at random in the 4 level-set of Fig. 6.2, this set being the union of the two intervals [0, 0.15] and [0.85, 1]. The selected point U_{12} determines X_1 and X_2. Next we similarly select a point U_{34} at random in the 2 level-set to obtain X_3 and X_4. The sum $X_1 + X_2 + X_3 + X_4$ is always 6 regardless of which random numbers are chosen, and all of the X_k have mass function $\mathbf{x}^*$. Note that $4\mathbf{x}^*$ is not a vector of integers; that criterion is sufficient for playability, but not necessary.

The antithetic method described above is based on the idea of considering the areas in pairs. It is also possible to consider them in larger groups, since it is possible to generate any number $n > 1$ of uniform [0,1] random variables U_i such that $U_1 + \ldots + U_n = n/2$. One method for doing this is based on the trigonometric identity

$$\sum_{i=1}^{n} cos\,(T + 2\pi i/n) = 0 \tag{6.2-1}$$

If T is uniform $[0,2\pi]$, all n of the quantities summed in (6.2-1) are identically distributed and sum to 0. Let R be any random variable independent of T, and let $V_i = R\cos(T + 2\pi i/n)$. The distribution of V_i can be controlled with the distribution of R. To get the desired collection of uniform random numbers, let $F_R(r) = 1 - \sqrt{(1 - 4r^2)})$, in which case it can be shown that V_i is uniform in the interval $[-0.5, 0.5]$. By taking $U_i = V_i + 0.5$, we have the desired collection of uniform [0,1] random variables. We could now, for example, take three CDF's from the list, let $X_{123} = {F_1}^{-1}(U_1) + {F_2}^{-1}(U_2) + {F_3}^{-1}(U_3)$, and replace the three distributions with the CDF of X_{123}. There are examples where this technique succeeds at playability while the two-at-a-time method fails, as well as vice versa.

The examples considered so far are consistent with the idea that playability is more of a problem in small Blotto games than in large ones. If one were interested only in the value of a large Blotto game, one might simply solve the relaxed game and assume playability. The analyst who has solved a relaxed game and is contemplating the playability question might also take some comfort from the Law of Large Numbers. If one were to simply assign the X_k sequentially and independently in accordance with the marginal distributions for the relaxed game, one would usually either run out of units before all areas had been considered or else have some units left over at the end. The number of such areas would be small as a proportion of n, however, if n is large.

Beale and Heselden (1962), who introduce the antithetic method described above, argue that relaxed games might very well be better models than Blotto games in the first place because of uncertainty in estimating f and g. They say, "...although one may be able to use an economic analysis to estimate the approximate strength of the enemy, it will often be unrealistic to suppose this strength known exactly; the same may apply to the number of units we can afford." If f and g are regarded as the mean number of units available rather than as a rigid constraint, the natural formulation is as a relaxed game.

6.3 The Asymmetric Case

In Sect. 6.1 we showed that relaxed games can be used to solve symmetric Blotto games, subject to playability reservations. The use of LP1 of Sect. 6.1 to solve relaxed games depends on a symmetry argument to conclude that the average number of units per area should be f/n. That symmetry argument fails if the areas have different payoff functions, so in this section we consider a Lagrangian approach that is suitable for solving the general case. In the Lagrangian approach, constraints on the amount of resource used are replaced by a cost of resource usage. We begin by considering linear program LP3, which applies to one generic target:

$$\begin{aligned}&\max_{v,\mathbf{x}\geq 0} v - c\sum\nolimits_i ix_i\\&\text{subject to} \sum\nolimits_i A(i,j)x_i \geq v - dj;\;\; j \geq 0,\\&\sum\nolimits_i x_i = 1,\end{aligned}$$

where $c > 0$ and $d > 0$ are input "prices" on units for players 1 and 2, respectively. LP3 has infinitely many variables and infinitely many constraints, but only a finite number need to be considered as long as $A(i, j)$ is bounded. We will assume in this section that $0 \leq A(i,j) \leq \overline{A}$ for all i, $j \geq 0$, where $\overline{A}$ is an upper bound on the payoff. In that case v cannot exceed $\overline{A}$, since otherwise the $j = 0$ constraint would be violated. Since $v \leq \overline{A}$, we know that $v - dj \leq 0$ for $j \geq N$, where N is the integer part of $\overline{A}/d$. Therefore, only the first $N + 1$ inequality constraints are necessary; intuitively, player 2 does not need to be constrained from spending more than the area is worth. Similarly, only the first $M + 1$ variables need to be considered, where M is the integer part of $\overline{A}/c$. Thus LP3 is effectively a Linear Program with $M + 2$ variables and $N + 2$ constraints. Let v* and $\mathbf{x}^* \equiv (x_0^*, \ldots, x_M^*)$ be optimal in LP3, and let $f^* \equiv \sum_i ix_i^*$ be the average number of units consumed by player 1. Like v* and $\mathbf{x}$*, f* depends on the Lagrangian prices (c, d), but to avoid overburdening the notation we are suppressing that fact for the moment.

Player 2's Linear Program is LP4:

$$\begin{aligned}&\min_{u,\mathbf{y}\geq 0} u + d\sum\nolimits_j jy_j\\&\text{subject to} \sum\nolimits_j A(i,j)y_j \leq u + ci;\;\; i \geq 0,\\&\sum\nolimits_j y_j = 1.\end{aligned}$$

Only the first $M + 2$ constraints (including the equality constraint) and $N + 2$ variables (including u) need to be considered. Let u* and $\mathbf{y}^* \equiv (y_0^*, \ldots, y_N^*)$ be the solution of LP4, and define g* to be the average number of units consumed by $\mathbf{y}$*.

Consider now a relaxed Blotto game where X and Y are the random numbers of units used by the two players, with mean-value constraints $E(X) \leq f^*$ and $E(Y) \leq g^*$. By employing $\mathbf{x}$* as the distribution of X, player 1 can guarantee

$$E(A(X,Y)) \geq E(v^* - dY) \geq v^* - dg^*, \tag{6.3-1}$$

where the first inequality follows from the constraints of LP3 and the second because $E(Y) \leq g^*$. By employing $\mathbf{y}$*, player 2 can guarantee

$$E(A(X,Y)) \leq E(u^* + cX) \leq u^* + cf^*. \tag{6.3-2}$$

Since LP3 and LP4 are dual linear programs, $v^* - cf^* = u^* + dg^*$. It follows that $v^* - dg^* = u^* + cf^*$, call the common value Π^*. All of these quantities depend on the as yet unspecified prices c and d.

Next consider an n-area Blotto game where the payoff function in area k depends on k. Solve LP3 and/or LP4 n times to obtain all of the quantities introduced above, but now subscripted by k. Thus we have $\mathbf{x}_k^*$, $f_k{}^*$, Π_k^*, etc. Let $\mathbf{x}(c,d) \equiv (\mathbf{x}_1^*, \ldots, \mathbf{x}_n^*)$ and define $\mathbf{y}(c,d)$ similarly. This notation emphasizes the previously suppressed dependence on (c,d). Also let $F(c,d) = \sum_k f_k^*$, $G(c,d) = \sum_k g_k^*$, and $\Pi(c,d) = \sum_k \Pi_k^*$. $F(c,d)$ is "the average total number of units used in employing $\mathbf{x}(c,d)$", and $G(c,d)$ has a similar meaning. Suppose that $\mathbf{x}(c,d)$ is feasible for player 1; that is, suppose that $F(c,d) \leq f$, where f is the given constraint on total units used by player 1. By summing (6.3-1) on k, we find that the average total payoff is bounded below by

$$\sum\nolimits_k \left(v_k^* - dE(Y_k)\right) = \sum\nolimits_k \left(\Pi_k^* + d(g_k - E(Y_k)\right) \geq \Pi(c,d) + d\left(G(c,d) - g\right), \tag{6.3-3}$$

where g is the total force level constraint for player 2. Thus, as long as (c,d) is such that $F(c,d) \leq f$, the value of the relaxed game must exceed the quantity (call it $a(c,d)$) on the right of (6.3-3). Note that $a(c,d)$ is $\Pi(c,d)$ if $G(c,d) = g$, but that otherwise a correction term is necessary. Similarly, player 2 can guarantee that the average total payoff will not exceed $b(c,d) \equiv \Pi(c,d) - c(F(c,d) - f)$ provided $G(c,d) \leq g$. We can conclude that

If (c,d) can be found such that $F(c,d) = f$ and $G(c,d) = g$, then the value of the relaxed game is $\Pi(c,d)$.

The above analysis offers no clue to how the (c,d) pair that makes both players "want" to use their constrained values is to be found, or even whether such a pair exists. This may be less of a problem than it appears if the real goal is to find how the value of the game depends on the force levels, or if the units are better priced than constrained in the first place. Nonetheless, the analyst who has made a sequence of bad guesses (c_t,d_t); $t = 1,\ldots,T$ can construct bounds on the game value that are determinable from the calculations already made. These bounds are obtainable by solving yet another Linear Program. To shorten the notation, let $a_t = a(c_t,d_t)$, $F_t = F(c_t,d_t)$, and similarly define b_t and G_t. Also let $a_0 = 0$ and $F_0 = 0$ be the effectiveness and cost of the null strategy of using no units at all for player 1, and let $b_0 = \sum_k \overline{A}_k$ and $G_0 = 0$ be the effectiveness and cost of using no units for player 2. Here $\overline{A}_k$ is an upper bound on the payoff in area k. Finally, consider LP5:

$$\max_{\mathbf{p}\geq 0} \sum_{t=0}^{T} p_t a_t$$

$$\text{subject to} \sum_{t=0}^{T} p_t F_t \leq f,$$

$$\sum_{t=0}^{T} p_t = 1.$$

The variables p_t can be interpreted as "the probability that player 1 uses $\mathbf{x}(c_t,d_t)$", with the first constraint requiring him to use no more than f on the average. There will always be a feasible solution on account of the precaution of including a null strategy, so the maximized objective function (call it $\Pi_{\min}$) will be a bound enforceable by player 1. Player 2's corresponding program is LP6:

$$\min_{\mathbf{q}\geq 0}\sum_{t=0}^{T} q_t b_t$$
$$\text{subject to}\sum_{t=0}^{T} q_t G_t \leq g,$$
$$\sum_{t=0}^{T} q_t = 1.$$

Let the minimized objective function be $\Pi_{\max}$. If the unknown value of the relaxed game is Π, we can say that $\Pi_{\min} \leq \Pi \leq \Pi_{\max}$. LP5 and LP6 are not duals, so there can be no certainty that $\Pi_{\min} = \Pi_{\max}$. If the two quantities are almost equal, however, they may be sufficiently good bounds on the value of the game.

Example Consider a very simple problem with only one target. Suppose $n = 1$, $f = 5$, $g = 5$, and that $A(i, j) = 1$ if $i \geq 10$, otherwise $A(i, j) = 0$. Player 1 wins if and only if he assigns 10 or more units, and player 2's units are useless. The solution of this trivial game is obvious: player 1 should flip a coin to use either 0 or 10, it doesn't matter what player 2 does, and the value is 0.5. Nonetheless, suppose the Lagrangian method were used with $(c, d) = (0.05, 0.05)$. We would find that $\Pi_1 = 1$, $F_1 = 10$, $G_1 = 0$, $a_1 = 1 - 0.05(5 - 0) = .75$, and $b_1 = 1 + 0.05(5 - 10) = 0.75$. The solution of LP5 is $\mathbf{p} = (0.5, 0.5)$, and $\Pi_{\min} = 0.375$. The solution of LP6 is $\mathbf{q} = (0,1)$, and $\Pi_{\max} = 0.75$, so we know that the value of the relaxed game is somewhere in the interval [0.375, 0.75]. In this example $F(c,d)$ is discontinuous at $c = 0.1$, with player 1 using nothing if c is larger than 0.1 or else 10 if c is smaller than 0.1. The relaxed game value will be better approximated if c in the vicinity of 0.1. If c is exactly 0.1, LP3 has alternate optimal solutions: (v^*, x_0^*, x_{10}^*) can be either (1,0,1) or (0,1,0). If LP5 includes the a_t and F_t values for both solutions, the optimized solution will be $\Pi_{\min} = 0.5$, exactly the value of the relaxed game.

6.4 Continuous Actions

In this section we omit the restriction that X_k and Y_k must be integers, while retaining the other constraints of a relaxed Blotto game. Linear programming is no longer applicable, but in some cases analytic solutions are still possible.

Consider a relaxed, symmetric Blotto game where the payoff is $A(x,y)$ in the typical area, and where the mixed strategies of the two players have CDFs $F()$ and

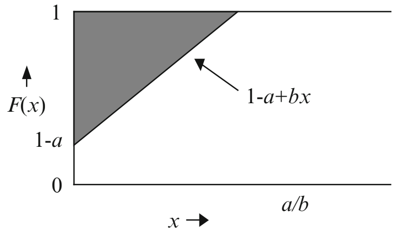


Fig. 6.4 A cumulative distribution function for player 1

$G()$, respectively, in each area. The CDF $F()$ is constrained to have a mean not exceeding μ_F, and $G()$ is constrained to have a mean not exceeding μ_G. Let $E(F,y) \equiv \int A(x,y)dF(x)$ be the average payoff when player 2 uses y against player 1's $F()$. This integral (like all integrals in this section) extends over the non-negative real line, and is written as a Stieltjes integral to account for the possibility that $F()$ may have jumps. Similarly let $E(x,G) \equiv \int A(x,y)dG(y)$ be the average payoff when player 1 uses x against $G()$, and note that the average payoff when $F()$ and $G()$ are each employed independently is

$$\int\int A(x,y)dF(x)dG(y) = \int E(F,y)dG(y) = \int E(x,G)dF(x) \tag{6.4-1}$$

An optimal strategy for player 1 can sometimes be obtained by hypothesizing that E(F,y) is bounded below by some linear function of y. Linear functions are appealing because if E(F,y) $\geq a - by$, it follows that

$$\int E(F,y)dG(y) \geq a - b\int ydG(y) \geq a - b\mu_G, \tag{6.4-2}$$

regardless of the unknown function $G()$. By carefully selecting a and b, player 1 can make $a - b\mu_G$ large without exceeding the mean constraint on $F()$.

Consider the game "majority rules" where $A(x, y) = 1$ if $x > y$, else 0. In that case (ignoring the possibility of ties, since allocations are continuous now), $E(F, y) = 1 - F(y)$ and $E(x, G) = G(x)$. If $F(x)$ is as shown in Fig. 6.4, then $1 - F(y) \geq a - by$ for all $y \geq 0$, and hence player 1 can guarantee a payoff of at least $a - b\mu_G$. The mean of player 1's allocation is $a^2/(2b)$, the area of the shaded region in Fig. 6.4, which must not exceed μ_F. Player 1 therefore desires to maximize $a - b\mu_G$ subject to the constraints $a^2 \leq 2b\mu_F$ and $0 \leq a \leq 1$. It is not difficult to show that the solution is $a = \mu_F/\mu_G$ if the ratio of relative strengths is smaller than 1, else $a = 1$, with $b = a^2/(2\mu_F)$ in either case. The corresponding lower bound on the game value is

$$v = \begin{cases} \mu_F/2\mu_G & \text{if } \mu_F/\mu_G \leq 1 \\ 1 - \mu_G/2\mu_F & \text{if } \mu_F/\mu_G \geq 1 \end{cases} \tag{6.4-3}$$

The demonstration that player 2 can enforce the same bound is left as an exercise. The value of the game is thus given by (6.4-3). When player 1 is weak relative to player 2, the probability of assigning 0 units is positive. Thus a weak player 1 sacrifices some areas in order to be strong enough in the others to compete effectively. Symmetric statements hold for player 2.

Washburn (2013) considers a Blotto model of the United States Electoral College where the resource for each side is money, and where the electors of each state go to the party that spends the most there. Formula (6.4-3) is generalized to include the possibility that one side or the other has a head start in each state. In an attack-defense scenario, the head start might consist of some defenders whose geographical location limits them to defending only a single target.

6.5 Attack and Defense of a Group of Targets

In most applications of Blotto games the "areas" are targets that are being attacked by player 1 and defended by player 2. Usually player 1 attacks with some kind of missile and player 2 defends by attacking player 1's missiles with countermissiles. The tactical mantra for both sides is "lose big and win small." If player 1 commits many missiles to a target undefended by player 2, he will regret having "wasted" the excess, and similarly player 2 might waste his defenders at targets that are not attacked. Each side would like to move last, making his allocation of units after observing the allocation of the other side and thereby avoiding waste. If both sides find achievement of this favorable position impossible for technological reasons, a Blotto game model results. In this section we review some of the possibilities. See also Eckler and Burr (1972).

If the targets are all identical, it is natural to take $A(i, j)$ to be the probability that a target is destroyed when attacked and defended by i and j units, respectively. The calculation of $A(i, j)$ might be based on assumptions of shot-to-shot independence, with the two inputs being

P = probability that an unintercepted attacker will destroy the target
P_I = probability that a defender will intercept its assigned attacker

To determine the probability of killing the target, we must also make an assumption about how the defenders are distributed over the attackers. The TPZS workbook on sheet "BlottoMin" contains a formulation of LP4 from Sect. 6.3 where $A(i, j)$ is computed under the assumption that the j defenders are distributed as evenly as possible over the i attackers, after which any surviving attackers attack the target. For example, if $(i, j) = (3, 4)$, then two attackers encounter one defender and one attacker encounters two defenders, so

$$A(3,4) = 1 - (1 - P(1 - P_I))^2 \left(1 - P(1 - P_I)^2\right)$$

Based on input values for P, P_I, the attacker cost c, and the defender cost d, sheet "BlottoMin" sets up LP4. If $(P, P_I, c, d) = (0.5, 0.8, 0.4, 0.3)$, the solution reveals that the defender makes $\mathbf{y}^* = (0.75, 0.25, 0, \ldots)$ and $u^* = 0$. The average number of defenders used per target is $g^* = 0.25$. The attacker's strategy can be obtained from LP3 (see sheet "BlottoMax") or the dual variables of LP4: $\mathbf{x}^* = (0.25, 0.75, \ldots)$ and $v^* = 0.375$. The average number of attackers used per target is $f^* = 0.75$. The probability that the target will be destroyed is $\Pi = u^* + cf^* = 0 + 0.4^*\, 0.75 = 0.3$ when both sides play their optimal mixed strategies. Both optimal mixed strategies are playable in a Blotto game with $n = 4$ targets, $f = 3$ attackers, and $g = 1$ defender, so the value of that Blotto game is $4(0.3) = 1.2$ targets destroyed. Blotto games with different values for (n, f, g) can be solved by varying c and d, as described in Sect. 6.3, or, as long as there is only one type of target, by the methods of Sect. 6.1.

Different rules are sometimes used for calculating $A(i, j)$. If for some reason at most one defender can be used to engage any given attacker (one-on-one defense), then the formula for $A(i, j)$ becomes

$$A(i,j) = 1 - \begin{cases} (1 - P(1 - P_I))^i & \text{if } i \leq j \\ (1 - P(1 - P_I))^j (1 - P)^{i-j} & \text{if } i \geq j. \end{cases} \tag{6.5-1}$$

Analytic results about optimal strategies and the game value are available for this case (Matheson (1967)).

If the defenders are perfect ($P_I = 1$), (6.5-1) simplifies to

$$A(i,j) = \begin{cases} 0 & \text{if } i \leq j \\ 1 - \exp\big(- \beta(i - j) & \text{if } i \geq j' \end{cases} \tag{6.5-2}$$

where $\beta \equiv -\ln(1 - P)$. If, in addition, the allocations of both sides are taken to be continuous as in Sect. 6.4, the value of the game depends only on the normalized numbers of attackers and defenders $A \equiv \beta\mu_F$ and $D \equiv \beta\mu_G$. A simple partial solution for the probability of destroying the target as a function of A and D is available for small numbers of attackers (Galiano (1969)). Specifically,

$$\Pi = A \Big/ \left(1 + D + \sqrt{D^2 + 2D}\right) \text{ for } A \leq \sqrt{D^2 + 2D} \tag{6.5-3}$$

Figure 6.5 shows the entire relationship for five values of D. The linear portions ("Defense Dominant") correspond to (6.5-3). The curve labeled "$D = 50$" is very close to (6.4-3) because the attackers are essentially perfect (β is large).

In all of the above formulations, each player is assumed to know the total resources of the other side. Bracken, Brooks, and Falk (1987) find robust strategies for the defense that are less sensitive to knowing the total attack level. These robust strategies are less effective than optimal strategies when the total attack level is as assumed, but suffer less degradation for other attack levels. If nothing whatever is known about the total attack level, it is sometimes advocated that the goal for the defense should be to minimize the maximum (over attack size) value destroyed *per attacker*, a philosophy that leads to the so-called Prim-Read defense (Eckler and Burr (1972), McGarvey (1987)).

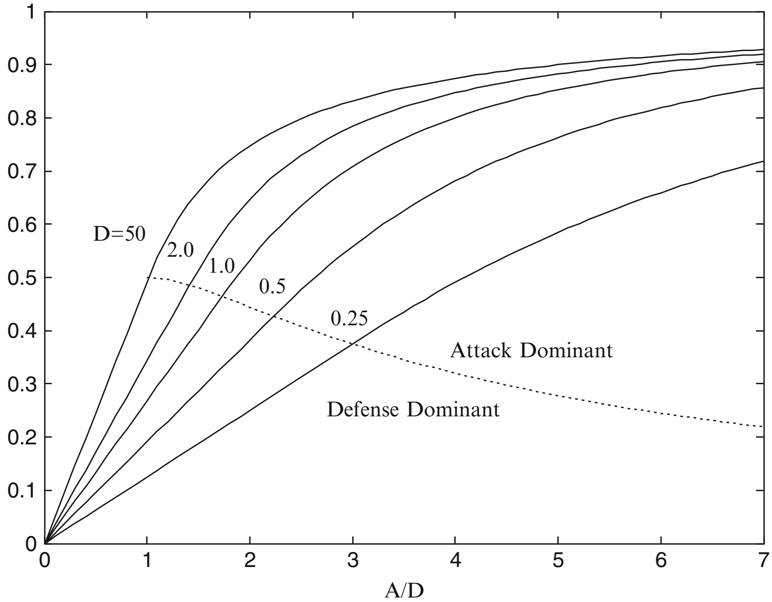


Fig. 6.5 Kill probability versus attack/defense ratio for various values of D

6.6 Concave-Convex Blotto Games

Mixed strategies are not required in this section, so **x** and **y** stand for vectors of allocations, rather than probability distributions. If $A_k(x,y)$ is a convex function of y for all k and x, and if player 2's allocations are continuous as in Sect. 6.4, then player 2 must have an optimal pure strategy (Sect. 5.2). Efficient methods of solution will exploit this feature, with Logistics games (Sect. 5.3) being a prime example. Similar comments hold if $A_k(x,y)$ is a concave function of x for all k and y. In the concave-convex case, the Blotto game must have a saddle point in pure strategies.

Example Croucher (1975) consider the game $A_k(x,y) = v_k(1 - \exp(-\alpha_k x))\exp(-\beta_k y)$. This concave-convex function is intended to represent "the value of the k^{th} target that is destroyed when x attackers and y defenders are assigned." Since the targets are all different, it is efficient to use a Lagrangian approach where input arsenals f and g are replaced by positive Lagrangian costs c and d. Let $B_k(x,y) \equiv A_k(x,y) - cx + dy$, and consider the modified game $\sum_k B_k(x_k,y_k)$ where **x** and **y** are required only to be non-negative. Since neither quantity is required to be an integer, the optimal allocations x_k and y_k can be found by equating derivatives with respect to x_k and y_k to 0. The simplicity of that optimization technique compensates for having to search for costs c and d such that $\sum_k x_k = f$ and $\sum_k y_k = g$. The search problem can be solved either by trial and error (see sheet "Croucher" of the TPZS workbook) or by Croucher's more systematic method.

6.7 Exercises

1. Using linear programming, verify the claims made in Sect. 6.2 about the solution of LP1 for the Blotto game of exercise 10 of Chap. 3. Note that x_i has a different meaning in Chap. 3 than in Sect. 6.2.

2. In Sect. 6.2 the antithetic method was successfully applied to a four-area problem where **x*** = (0.4, .015, 0.15, 0.15, 0.15). Consider instead the method of first selecting a set of allocations S, and then randomly selecting one of the permutations of S to obtain target assignments, where S is

$$\begin{array}{lll}\{0,0,2,4\} & \text{with probability} & 0.4,\\ \{0,1,1,4\} & \text{with probability} & 0.2,\\ \{0,1,2,3\} & \text{with probability} & 0.15.\\ \{1,1,2,2\} & \text{with probability} & 0.025, \quad \text{and}\\ \{0,0,3,3\} & \text{with probability} & 0.225.\end{array}$$

   Show that this method also succeeds in playing **x***. It is typical that mixed strategies are not only playable, but playable in many ways.

3. In Sect. 6.2 it was pointed out that **y*** = (1/6, 1/6, 1/6, 1/2) is not playable when $n = 2$.

   (a) Carry out the antithetic method for $n = 3$. After two iterations you should find that the final CDF has the total being 5 or 7 with probability 1/6, or 6 with probability 2/3. This is not the CDF of a constant, so the antithetic method is not successful at playing **y*** when $n = 3$.
   (b) Show how to play **y*** when $n = 3$. Hint: Do exercise 2 first.
   (c) Show that the antithetic method works when $n = 4$.

4. Section 6.5 includes an example of solving a relaxed game where the Lagrangian prices are $c = 0.4$ and $d = 0.3$.

   (a) Use pages "BlottoMax" and "BlottoMin" of the TPZS workbook to repeat the analysis for $c = 0.2$ and $d = 0.15$. What are the optimal marginal distributions **x*** and **y***, and what is the average kill probability Π?
   (b) Consider a relaxed game where there are 200 targets identical to those considered in part a, except that half of them have unit value and the other half have value 2. Let the Lagrangian prices be $c = 0.4$ and $d = 0.3$. Using the results from part a, determine the average consumption of units for each side, counting all 200 targets, and also the total average value captured (killed) by player 1.

5. Use pages "BlottoMax" and "BlottoMin" of the TPZS workbook to solve an attack-defense Blotto game where $P = 0.5$, $P_I = 1$, $c = 0.2$, and $d = 0.15$.

   (a) What are the average numbers of units used by player 1 (f*) and player 2 (g*), and what is Π?

(b) How does Π compare to the kill probability if the allocations are permitted to be continuous, with the same average allocations f^* and g^*? Use Fig. 6.5.

6. Solve Croucher's game (Sect. 6.6) with three targets where $\boldsymbol{\alpha} = (1,2,3)$, $\boldsymbol{\beta} = (1,1,1)$, $\mathbf{v} = (1,2,2)$, $c = 2$, and $d = 1$. What are the allocations of the two players to the three targets? You can solve this either by equating derivatives to 0, keeping in mind that both allocations must be nonnegative, or by appealing to sheet "Croucher" of the TPZS workbook, which incorporates the resulting formulas.

7. There are 100 identical targets, player 1 has 800 missiles available to attack them, and player 2 has 500 perfect interceptors available to intercept the attackers. Each side must assign attackers and interceptors to targets without knowing anything about the other side's allocations, except for the totals. If there are x attackers and y defenders at any target, then that target will certainly survive if $x < y$. Otherwise, each unintercepted attacker will independently kill its target with probability 0.2. Approximate the kill probability as in (6.5-2) and assume that allocations can be continuous. Given optimal play by both sides, what fraction of the targets can be expected to survive?

Chapter 7
Network Interdiction

Many are stubborn in pursuit of the path they have chosen, few in pursuit of the goal.

Friedrich Nietzsche

This chapter deals with a variety of competitive problems that occur on networks. Such problems deserve their own chapter because networks are becoming increasingly important in modern life. The Internet is a network, transportation systems are networks, power distribution systems are networks, communication systems are networks, social systems can be thought of as networks, and all of these are subject to competition between the intended users of the network and another player who wishes to interfere with that usage. In this chapter we will consistently refer to the two competing players as User and Breaker, rather than player 1 and player 2. Depending on the model, either player may be the maximizer. The models in Sect. 7.1 are maxmin formulations where Breaker's actions are known to User. In Sect. 7.2 we consider games where Breaker can keep his actions secret.

All of the models described in this chapter share the notation that N is the set of network nodes and A is the set of arcs that connect them. The number of nodes is $|N|$ and the number of arcs is $|A|$. The set N usually includes a special source node s and a special termination node t. In all models considered, User is for some reason compelled to move from s to t, and is thereby rendered vulnerable to actions by Breaker. An arc is denoted (i, j), the arc that connects node i (the tail of the arc) to node j (the head). All arcs are directed; that is, the existence of arc (i, j) does not always imply the existence of arc (j, i). Figure 7.1 shows a network with $|N| = 12$ nodes superimposed on a map of Iraq. The source is $s = 1$ and the termination (Baghdad) is $t = 12$. There are $|A| = 38$ arcs because each of the pictured 19 segments connecting two nodes can be traversed in either direction.

A. Washburn, *Two-Person Zero-Sum Games*, International Series in Operations Research & Management Science 201, DOI 10.1007/978-1-4614-9050-0_7,

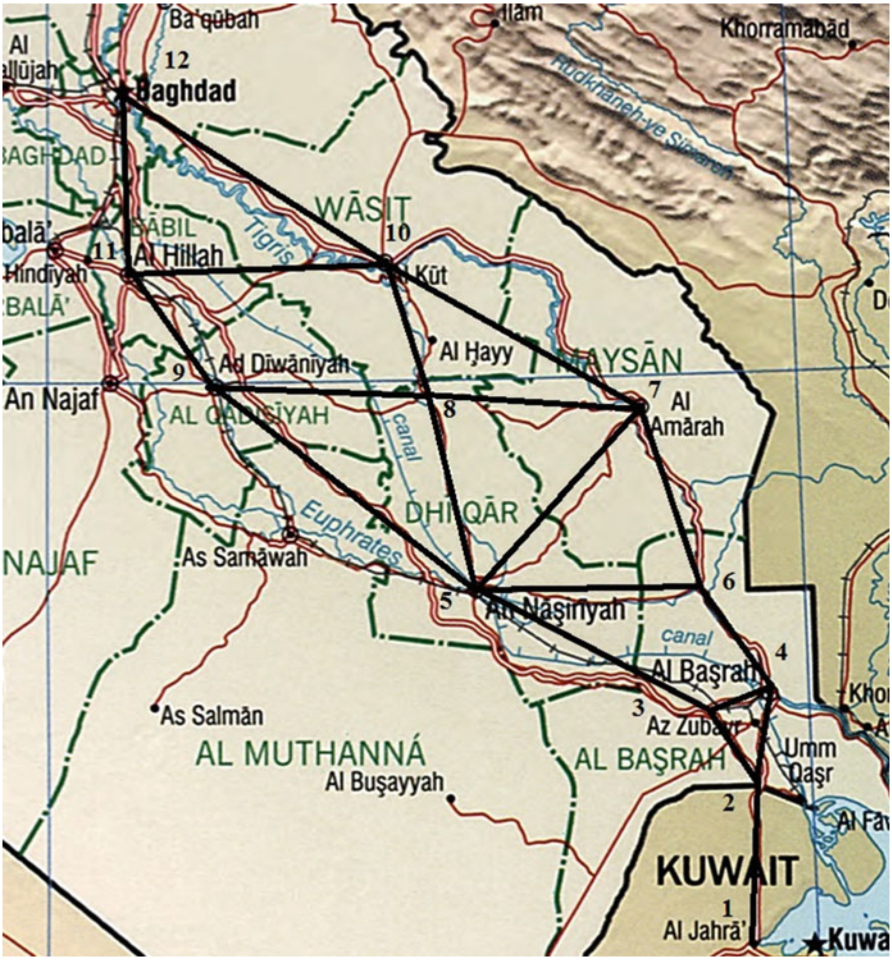


Fig. 7.1 A map showing the road segments connecting Al Jahra in Kuwait to Baghdad in Iraq, with a superimposed road network

The number of times User moves over an arc will be called the "flow" over the arc. A *path* is a sequence of arcs where the head of one arc is the same as the tail of the next. Arcs have associated data that depends on the problem being considered. No data will be associated with nodes, but bear in mind that a node can always be "split" into two nodes connected by an artificial arc, and that data can be associated with the artificial arc.

The reader may wish to review the notation for mathematical programs in Appendix A, since mathematical programs are heavily employed in this chapter.

7.1 Maxmin Network Problems

Here we consider problems where Breaker first takes an action that is observable by User, after which User observes the action and adjusts his use of the network accordingly. The resulting formulations are maxmin (or minmax) problems, with neither side resorting to the use of mixed strategies. Each of the subsections corresponds to a classic network optimization problem for User. The analytic goal in all cases is to express the maxmin problem as a single mathematical program, since this permits the use of optimization packages.

7.1.1 Maximizing the Length of the Shortest Path

Here the data associated with arc (i, j) is c_{ij}, the length of the arc, and User wishes to find the path from s to t that has the shortest total length. In practice (examples are routing packets over the internet and vehicles over roadways) "length" is usually the time to go from i to j, but any measure of User's inconvenience can be used for c_{ij} as long as lengths are additive over a path. If there are multiple arcs from i to j, then all but the shortest can be discarded. The seminal reference is Fulkerson and Harding (1977).

We can represent a path as a vector of binary indicator variables $\mathbf{y} = (y_{ij})$, with $y_{ij} = 1$ if arc (i, j) is included in the path, or otherwise 0. Except for s and t, a path must enter and exit each node the same number of times. This can be enforced by requiring that exits minus entrances must be 0. The problem of finding the shortest path can then be expressed as the following linear program LP:

$$
\begin{aligned}
& z = \min_{\mathbf{y}\geq 0} \sum_{(i,j)\in A} c_{ij}y_{ij} \\
& \text{subject to} \sum_{(k,i)\in A} y_{ki} - \sum_{(i,k)\in A} y_{ik} = \begin{cases} -1 & \text{if } i = s \quad [v_s] \\ 0 & \text{if } i \neq s,t \quad [v_i] \\ +1 & \text{if } i = t \quad [v_t] \end{cases}
\end{aligned}
\tag{7.1-1}
$$

In LP, the variables v_i shown in square brackets [] are dual variables that will be referred to later; the only controllable variables for User are the components of $\mathbf{y}$, as indicated by the subscript on min. The constraints say that, for each intermediate node i, the total flow over all arcs for which i is the head (the flow into i) must equal the total flow over all arcs for which i is the tail (the flow out of i). These are the flow balance constraints. Nodes s and t are exceptional, since a feasible path must exit s one more time than it enters, and must enter t one more time than it exits. LP expresses the problem of finding z, the length of the shortest path from s to t, as the sum of the lengths of all the arcs in the path. Note that $\mathbf{y}$ is not required to consist of integers, so LP is a linear program. This omission of the integer requirement is permissible because of a theorem to the effect that there is always an optimal solution **y** to problems like LP that consists entirely of integers, even when there is no explicit constraint to that effect (the constraints are "totally unimodular").

Breaker tries to make the shortest path as long as possible by adjusting the arc length data. A truly "broken" arc would have its length so long that User would surely avoid it, but Breaker is not necessarily restricted to such drastic actions. Suppose that Breaker's strategy amounts to the determination of the arc lengths in the vector **c**, and that **c** must lie in some specified set C. We might give him a certain budget and let the length of each arc increase from some default value as Breaker spends money on it, but the details of C are not important for the moment. We have a minimization problem embedded within a maximization problem, or in short a maxmin problem. This particular maxmin problem has a property that we can exploit.

The exploitable property is that LP is a linear program. Since every minimizing linear program has a dual that is maximizing, we can convert the overall problem from a maxmin problem to a max problem, a fundamentally simpler type, by employing the "dual trick" of Sect. 2.5.3. Breaker's overall problem is then mathematical program MP:

$$\begin{aligned} & z = \max_{\mathbf{c}\in C,\mathbf{v}} v_t - v_s \\ & \text{subject to } v_j - v_i \le c_{ij}; \quad (i,j) \in A \end{aligned} \tag{7.1-2}$$

With **c** fixed, MP is simply the dual of LP. It is permissible to add the constraint $v_s = 0$, since only differences of the components of **v** are involved in MP. In that case the meaning of v_i becomes "the shortest distance from node s to node i". User's flexibility is expressed in the constraints of MP, which state that the shortest distance to the head of an arc (j) cannot possibly exceed the shortest distance to the tail (i), plus the length of the arc.

Upon solving MP, Breaker's optimal choice **c*** is of course directly available. The dual variables of the arc constraints express User's optimal path **y*** in reaction to **c***, but should be of little interest. The reason is that User must not commit to choosing **y*** or any other specific path. The path chosen by User should depend on the cost vector **c** chosen by Breaker, lest User give up his advantage in moving last.

The computational difficulty of MP will depend on the nature of the set C. In the case where Breaker's budget is constrained to b, let x_{ij} be Breaker's spending on arc (i, j), and assume that the resulting arc length is $c_{ij} + d_{ij}x_{ij}$, where c_{ij} is now the default arc length and d_{ij} quantifies the effect of each dollar that Breaker spends on lengthening it. Then MP becomes LP1:

$$\begin{aligned} & z = \max_{\mathbf{x}\ge 0,\mathbf{v}} v_t - v_s \\ & \text{subject to } v_j - v_i - d_{ij}x_{ij} \le c_{ij}; \quad (i,j) \in A \\ & \sum_{(i,j)\in A} x_{ij} \le b \end{aligned} \tag{7.1-3}$$

LP1 is a linear program, and can therefore be solved even for large networks. Sheet "NetPath" of the TPZS workbook is a small example of this formulation where the reader can experiment with different budget levels or change the inputs in other ways. An additional observation here is that, since $\sum_{(i,j)\in A}(c_{ij}+d_{ij}x_{ij})y_{ij}$ is a concave function of $\mathbf{x}$ and the constraints on $\mathbf{x}$ meet the requirements of Theorems 5.1 and 5.2, Breaker's strategy would be optimal even in a game where he could keep his spending secret from User. This lack of importance of secrecy is not true in general, but it is true in this case.

We might even consider a three-stage problem where User first hardens certain arcs to make it difficult to attack them, then Breaker, knowing User's initial hardening actions, does his best to lengthen or entirely break arcs, and finally User, knowing the status of the damaged network, chooses a path. Alderson et al. (2011) consider several such three-stage problems related to infrastructure defense. In our case, since LP1 is a maximizing linear program, it has a dual that is minimizing. Therefore User's initial hardening problem can be cast as a mathematical program that will minimize the length of the path that is ultimately chosen. If that problem turns out to be a linear program, we could again incorporate the duality trick to introduce a fourth stage, etc.

There are other possibilities for C in MP. We might postulate the costs of removing the various arcs and give Breaker a budget for arc removal. This would not lead to as simple an optimization problem as LP1, but of course the choice of model structure should be mostly dictated by realism, rather than computational convenience.

In MP, the assumption that s and t are known to Breaker could well be false. Most networks, in fact, are useful to User because they permit multiple sources and terminations that depend on the needs of the moment. Instead of specifying a specific s and t, suppose we were to specify a probability distribution $p(\omega)$, where the probabilities sum to 1 over some set Ω of "scenarios". Each scenario ω determines the source node $s(\omega)$ and the termination node $t(\omega)$. Program MP would then become MP2:

$$\begin{aligned}&z=\max\nolimits_{\mathbf{c}\in C,\mathbf{v}}\sum_{\omega}p(\omega)\left(v_{t(\omega)}(\omega)-v_{s(\omega)}(\omega)\right)\\&\text{subject to } v_j(\omega)-v_i(\omega)\le c_{ij};\quad (i,j)\in A,\omega\in\Omega\end{aligned}\tag{7.1-4}$$

The objective function of MP2 is the average length of the shortest path. The constraints simply replicate the constraints of MP, one set for each scenario, so the number of variables in MP2 is $|A||\Omega|$. Program MP is thus a special case where there is only one scenario.

As in the case of MP, whether MP2 is a linear program or not depends on the nature of C. Here is a formulation where C is the same as in LP1, call it LP2:

$$\begin{aligned}&z=\max\nolimits_{\mathbf{x}\ge 0,\mathbf{v}}\sum_{\omega}p(\omega)\left(v_{t(\omega)}(\omega)-v_{s(\omega)}(\omega)\right)\\&\text{subject to } v_j(\omega)-v_i(\omega)-d_{ij}(\omega)x_{ij}\le c_{ij}(\omega);\quad (i,j)\in A,\omega\in\Omega\\&\sum_{(i,j)\in A}x_{ij}\le b\end{aligned}\tag{7.1-5}$$

Note that the default arc lengths $c_{ij}(\omega)$ and the arc length expansion rates $d_{ij}(\omega)$ can depend on the scenario, as do the source and termination. A use for this feature might be if the network is used by resupply trucks that must return to a base node after unloading. The actions of Breaker might be effective only for loaded trucks, in which case the expansion rates would all be 0 for scenarios where the termination is a base node. Hemmecke et al. (2003) consider this problem and others related to it, giving some benchmark computational results.

If the scenario affects only the source and termination, LP2 can be replaced with a linear program that will have fewer variables, at least when $|\Omega|$ is large. Let $q(s,t)$ be the probability that the source is s and the termination is t. Then LP2 can be replaced by LP3:

$$
\begin{aligned}
& z = \max\nolimits_{\mathbf{x}\geq 0,\mathbf{v}} \sum_{s,t} q(s,t)\left(v_{st} - v_{ss}\right) \\
& \text{subject to } v_{sj} - v_{si} - d_{ij}x_{ij} \leq c_{ij}; \quad (i,j)\in A, s\in N \qquad (7.1\text{-}6) \\
& \sum_{(i,j)\in A} x_{ij} \leq b
\end{aligned}
$$

As in MP, variable v_{ss} can be set to 0 for all s, or simply omitted from the equations of LP3. If this is done, the meaning of v_{si} is "the shortest distance from s to i", and LP3 has only $|A||N|$ variables, regardless of the number of scenarios. We are taking advantage of the fact that, for each source s, program MP determines the shortest distance from s to every other node, not just the shortest distance to t.

The shortest path problem is the most commonly solved network optimization problem, but there are other possibilities. In the rest of this section we consider two more of them.

7.1.2 Minimizing the Maximum Flow

Networks are often used to represent supply infrastructures, whether based on rails, roads, or trails. We will consider capacitated networks where c_{ij} is the capacity of arc (i, j). This capacity is generally in terms of stuff per unit time or sometimes just stuff, possibilities being tons/day or passengers or bits/second. If there are multiple arcs from i to j, their capacities can be summed to form a single equivalent arc. In this subsection "flow" over an arc refers to the movement of stuff from tail to head.

A classic problem for User is the maximum flow problem: what is the maximum rate at which stuff can be transferred from node s to node t, and how can that transfer be achieved? To answer that question, it is convenient to include an artificial arc from t to s with infinite capacity, and to imagine that all flow of stuff into node t simply goes back to s over that arc. With that understanding, User's problem is to maximize the flow on arc (t, s). Letting x_{ij} be the flow on arc (i, j), we can formulate a linear program LP to find User's maximum flow, with dual variables indicated in [·]:

$$
\begin{aligned}
&\max_{\mathbf{x}\geq 0} x_{ts} \\
&\text{subject to} \sum_{(i,k)\in A} x_{ik} - \sum_{(k,i)\in A} x_{ki} = 0; \quad i\in N \quad [w_i] \\
&x_{ij} \leq c_{ij}; \quad (i,j) \in A \quad \left[y_{ij}\right]
\end{aligned}
\tag{7.1-7}
$$

The first set of constraints requires flow balance, and the second set of constraints requires that no flow exceed the arc capacity.

Breaker's problem of minimizing the maximum flow is an old one. Harris and Ross (1955), for example, consider how to optimally reduce the capacity of a rail network, specifically having in mind the rail network connecting the Soviet Union to Eastern Europe. A more recent reference with a good bibliography is Altner et al. (2010). Breaker's options usually take the form of reducing the capacity of certain critical arcs. Rail lines might be bombed, airport operations might be slowed down by forcing delays for security checks, or internet servers might be slowed down by denial-of-service attacks.

In anticipation of another application of the dual trick in formulating a minimization problem for Breaker, we begin by stating the dual of LP, call it LP′:

$$
\begin{aligned}
&\min_{\mathbf{y}\geq 0,\mathbf{w}} \sum_{(i,j)\in A} c_{ij} y_{ij} \\
&\text{subject to } w_i - w_j + y_{ij} \geq 0; \quad (i,j) \in A \\
&w_t - w_s = 1
\end{aligned}
\tag{7.1-8}
$$

Here is an instructive way to construct a feasible solution of LP′. Consider any set S of nodes that includes t, but not s, and let w_i be 0 if node i is in S, or otherwise 1. Thus $w_t = 1$ and $w_s = 0$. Also let y_{ij} be 1 if arc (i, j) has its tail in S, but not its head, or otherwise let y_{ij} be 0. With these assignments, all of the constraints in LP′ are satisfied. The arcs where $y_{ij} = 1$ are a "cut" in the sense that any flow from s to t must pass over one of them, and the objective function is just the total capacity of the cut that corresponds to S. Since variables **w** and **y** together are a feasible solution of LP′, the minimized objective function cannot exceed the capacity of any cut. The famous "max-flow min-cut" Theorem (Ford and Fulkerson 1962) states that the maximum flow is in fact exactly equal to the minimum capacity of any cut, so there must be an optimal solution of LP′ of the proposed form. Even without knowing the possible actions for Breaker, the arcs in the minimizing cut are obvious candidates for interference because blocking one of them reduces the capacity of the min cut by exactly the capacity of the blocked arc.

We can formulate an overall minimization problem for Breaker by using the dual trick. If C represents the set of possible capacity vectors, the result is mathematical program MP:

$$\begin{aligned}&\min_{\mathbf{c}\in C,\,\mathbf{y}\geq 0,\,\mathbf{w}} \sum_{(i,j)\in A} c_{ij}y_{ij}\\&\text{subject to } w_i - w_j + y_{ij} \geq 0;\quad (i,j)\in A\\&w_t - w_s = 1\end{aligned} \tag{7.1-9}$$

If C is a small set, Breaker could of course simply solve a linear program for each **c** in C, and then select whichever **c** results in the smallest maximum flow. The solution of MP will otherwise be difficult because of the multiplication of variables in the objective function, which threatens to make MP a nonlinear program. Phillips (1993) and Wood (1993) consider several possibilities for C, including one where Breaker is given a budget for arc removal and can remove each arc (i, j) at a specified cost. Wood shows that Breaker's problem is fundamentally difficult, but nonetheless offers some useful methods for the solution of MP. See also Burch et al. (2003) and Cormican et al. (1998).

Sheet "NetFlow" of the TPZS workbook formulates a small problem of this type.

7.1.3 Minimizing the Maximum Survival Probability

In this subsection User wishes to go from s to t without being detected, or equivalently he wishes to survive all attempts at detection. A survival probability p_{ij} is given for each arc (i, j), and we assume that detections on all arcs are independent events. If there are multiple arcs from i to j, then all can be discarded except for the one with the highest survival probability. User's problem of selecting the most survivable path can be formulated as a maximizing linear program, but, anticipating another application of the dual trick, it is convenient to begin with the dual of that linear program, call it LP:

$$\begin{aligned}&z = \min_{\boldsymbol{\pi}\geq 0} \pi_t\\&\text{subject to } \pi_j - p_{ij}\pi_i \geq 0;\quad (i,j)\in A\\&\pi_s = 1\end{aligned} \tag{7.1-10}$$

LP can be motivated directly. The meaning of variable π_i is the probability that User survives on the safest path connecting s to i, so π_s is required to be 1 and Breaker's problem is to minimize π_t. User's freedom to choose a path is reflected in the constraints, which require that the probability of surviving up to the head (j) of any arc must be at least the probability of surviving to its tail (i), multiplied by the probability of surviving the arc. The multiplication is justified by the assumption of independence.

Now consider the problem of minimizing the maximum survival probability. Let $\mathbf{p} = (p_{ij})$ be the collection of arc survival probabilities, and suppose that Breaker can choose **p** within some set P. Since we have already formulated the dual of User's maximization problem, the overall problem for Breaker is program MP:

$$\begin{aligned} &z = \min_{\mathbf{p}\in P, \boldsymbol{\pi}\geq 0} \pi_t \\ &\text{subject to } \pi_j - p_{ij}\pi_i \geq 0; \quad (i,j) \in A \\ &\pi_s = 1 \end{aligned} \qquad (7.1\text{-}11)$$

The product $p_{ij}\pi_i$ is likely to be problematic because it threatens to turn MP into a nonlinear program, but it may be possible to avoid that difficulty by defining $z_t \equiv -\ln(\pi_t)$ and formulating MP in terms of $\mathbf{z}$, rather than $\boldsymbol{\pi}$. Mathematical program MIP of Sect. 7.2.2 is an example of this.

As in the case of the shortest route problem, we can take advantage of the fact that MP finds the survival probability to all nodes from s, not just the survivability to node t. If the source and termination are actually determined randomly, let $q(s, t)$ be the probability that s is the source and t the termination. Then Breaker can minimize the probability of safely getting from s to t by solving MP2:

$$\begin{aligned} &z = \min_{\mathbf{p}\in P, \boldsymbol{\pi}\geq 0} \sum_{s,t} q(s,t)\pi_{st} \\ &\text{subject to } \pi_{sj} - p_{ij}\pi_{si} \geq 0; \quad (i,j) \in A, s \in N \\ &\pi_{ss} = 1; \quad s \in N \end{aligned} \qquad (7.1\text{-}12)$$

MP2 has $|N|^2$ variables of the form π_{si}, plus whatever other variables and constraints are required to represent P. The difficulty of solution will of course depend on P. Morton et al. (2007) consider several tractable versions of MP2 in the process of investigating optimal deterrence of nuclear smuggling.

7.2 Network Detection Games

In this section Breaker takes his actions in secret, as does User, so we have a game instead of a maxmin problem. We consider games where User attempts to transit from s to t without being detected by Breaker. User knows what detection assets are available to Breaker, but does not know how they are deployed before choosing his path.

We might choose the payoff to be the average number of times User is detected, with Breaker maximizing, or we might choose it to be the probability that User is never detected, as in Sect. 7.1.3, with User maximizing. Either choice might be more appropriate, depending on circumstances, but mathematical necessity dictates the former here. We will briefly consider the latter in Sect. 7.2.2, but for the moment the payoff must be the average number of detections. We first define a general game G and show how to solve it efficiently. Then various special cases will be considered, depending on network details and the assets employed by Breaker.

In game G, User's pure strategies are paths from s to t, one of which he must choose without knowing Breaker's assignments of detection assets to arcs. Breaker is assumed to have several controllable interdiction assets available, at most one unit of which can be assigned to each arc, and in addition the uncontrollable asset 0 is always assigned. Asset 0 is intended to represent means for detection that Breaker can benefit from, but does not control, and which will not interfere with other assets. Examples of uncontrollable assets might be detections by shepherds or overflights by aircraft engaged in other business; each looks for User, but is not Breaker's to command. The "at most one" requirement may be omitted, but we include it for the moment because the assignment of multiple controllable assets might risk interference between them or even fratricide.

Let I_{ka} be a binary random variable that is 1 if asset k detects User on arc a, or otherwise 0. The total number of detections is then $D \equiv \sum_{a \in A} \sum_{k} I_{ka}$. The payoff in G is $E(D)$, the expected number of detections. It is sometimes convenient to refer to an arc by its index a, and sometimes as (i, j), the arc that connects tail i to head j. The former notation is used in the expression for D. Both notations are used below.

If asset k is assigned to arc a, and if a is in User's path, the conditional probability of detection by that asset is p_{ka}. This data is assumed known to both sides. Let y_a be the probability that User includes arc a in his path, and let x_{ka} be the probability that asset k is assigned to arc a, with $x_{0a} = 1$ for asset 0. Since Breaker and User choose their strategies independently, we have $E(I_{ka}) = x_{ka}p_{ka}y_a$. The indicator random variables I_{ka} are not necessarily independent of each other, but, nonetheless, since expected values and sums commute, we have

$$d \equiv E(D) = \sum_{a \in A} \sum_{k} E(I_{ka}) = \sum_{a \in A} \left(p_{0a} + \sum_{k \neq 0} x_{ka}p_{ka}\right) y_a \qquad (7.2\text{-}1)$$

User wants to minimize d and Breaker wants to maximize it.

In the rest of this section, the subscript k and the term "asset" will refer only to controllable assets, so it should be understood that k is not 0. Consider first Breaker's maximization problem. To formulate a problem that is free of y_a, which Breaker does not control, introduce some additional variables z_j, one for each node j. We will argue below that the meaning of $z_j - z_i$ is "for given $\mathbf{x}$, the minimal average number of detections on any path from i to j". Breaker's linear program LP, with dual variables indicated in [], is

$$\begin{aligned}
&d = \max_{\mathbf{x} \geq 0, \mathbf{z}} z_t - z_s \\
&\text{subject to (a) } z_j - z_i - \sum_{k} x_{ka}p_{ka} \leq p_{0a} \text{ for all arcs } a = (i,j), \quad [y_a] \\
&\text{(b) } \sum_{k} x_{ka} \leq 1 \text{ for all arcs } a, \text{and} \quad [w_a] \qquad (7.2\text{-}2) \\
&\text{(c) } \sum_{a} x_{ka} \leq 1 \text{ for all assets } k. \quad [u_k]
\end{aligned}$$

The dual variables will be used later in stating the dual of LP—the only variables in LP itself are **x** and **z**. If there are multiple arcs from i to j, then all must be retained and constraints (a) apply to each one of them separately.

To establish a meaning for constraints (a), observe that $v_a \equiv p_{0a} + \sum_k x_{ka} p_{ka}$ is the average number of detections on arc a, and consider any User path $i_0, i_1, \ldots, i_m$ from $i_0 = i$ to $i_m = j$. The average number of detections on this path is $v_{i_0,i_1} + \ldots + v_{i_{m-1},i_m}$, and constraints (a) imply that this sum will be at least $(z_{i_1} - z_{i_0}) + \ldots + (z_{i_m} - z_{i_{m-1}})$, a sum that collapses to $z_j - z_i$. No matter what path connects i to j, the average number of detections will be at least $z_j - z_i$. These considerations establish the meaning of $z_j - z_i$ as the minimal average number of detections among all paths that connect i to j. In particular, any path that connects s to t will suffer at least $z_t - z_s$ detections, so the objective function is the average number of detections that Breaker can guarantee in G. Constraints (b) of LP require that the average number of assets assigned to an arc cannot exceed 1, and constraints (c) require that the average number of times asset k can be assigned to an arc must not exceed 1. These constraints are relaxations of the constraints that bind Breaker's play in G, since each set requires something to be true on the average that must always be true in G. If there is a method of selecting strategies in G such that x_{ka} is indeed the probability that asset k is assigned to arc a, for all k and a, then we say that **x** is "playable" in G.

More needs to be said to establish LP as a solution of G, but first appreciate the simplicity of LP compared to the option of enumerating all allocations and paths to form a matrix game. There are $30 \times 29 \times 28 \times 27 \times 26$ ways of assigning five assets to a network with 30 arcs. That number is 657,720, and every one of those ways would have to be represented in the normal form of G. On the other hand **x** in LP would have only 150 components. The number of variables in LP will have to exceed 150 by the number of nodes, but LP will still be a remarkably small linear program.

We must establish the playability in G of Breaker's allocations **x**. Suppose for example that the solution of LP when there are three assets and five arcs is

$$\mathbf{x} = \begin{bmatrix} 0.5 & 0 & 0.5 & 0 & 0 \\ 0.2 & 0 & 0.4 & 0 & 0.4 \\ 0 & 1 & 0 & 0 & 0 \end{bmatrix} \tag{7.2-3}$$

The probability that asset 2 is assigned to arc 1 is 0.2, etc. All three rows sum to 1, since there is nothing to be gained by not assigning an asset, and all five column sums are bounded above by 1 because of constraints (b). The difficulty is that player 1's allocations in G must <u>never</u> utilize an asset more than once, and must <u>never</u> assign more than 1 asset to an arc. This would be assured if the components of **x** were all 0 or 1, but they are not. Is there any way to "play" **x** in the sense of finding an allocation policy that never violates any of the constraints of G? This question is similar to the playability question encountered in Blotto games, but the solution here is different.

The $\mathbf{x}$ specified in (7.2-3) can be expressed as the average of four matrices as follows:

$$0.4\begin{bmatrix}1&0&0&0&0\\0&0&1&0&0\\0&1&0&0&0\end{bmatrix}+0.2\begin{bmatrix}0&0&1&0&0\\1&0&0&0&0\\0&1&0&0&0\end{bmatrix}+0.3\begin{bmatrix}0&0&1&0&0\\0&0&0&0&1\\0&1&0&0&0\end{bmatrix}$$
$$+0.1\begin{bmatrix}1&0&0&0&0\\0&0&0&0&1\\0&1&0&0&0\end{bmatrix}$$

Note that each of the four matrices has the same properties as $\mathbf{x}$ (row sums are 1 and column sums are bounded above by 1) plus the property that all entries are either 0 or 1. The latter property means that each of the four matrices is playable in G, with the positions of the 1s determining how Breaker assigns assets to arcs. The four weights constitute a probability distribution (0.4, 0.2, 0.3, 0.1), so Breaker can play $\mathbf{x}$ by first sampling from that distribution to select a playable matrix, and then making assignments according to the selected matrix. If that is done, then the probability that asset k is assigned to arc a is indeed x_{ka}, while at the same time the constraints of G are always satisfied.

Although we have merely given an example here, it turns out that any solution of LP can be played in G by decomposing $\mathbf{x}$ in the same manner. The theorem required is a generalization of the Birkhoff-von Neumann Theorem (Gerards 1995). Since playability is guaranteed, we know that Breaker can guarantee d^* in G by playing $\mathbf{x}^*$, the optimal solution of LP.

It is instructive to consider the problem from User's viewpoint. Whatever strategy he employs for selecting a path from s to t will imply the arc occupancy probabilities $\mathbf{y}$. The flow balance constraints require that User must start in node s, end in node t, and that the net occupancy of all other nodes must be 0:

$$\sum_{(k,i)\in A} y_{ik} - \sum_{(i,k)\in A} y_{ki} = \begin{cases} -1 & \text{if } i = s \\ 0 & \text{if i} \neq \text{s, t} \\ +1 & \text{if } i = t \end{cases} \tag{7.2-4}$$

Equation (7.2-4) is a relaxation of the constraints that bind User in G, since the net occupancy must always be 0 in G, whereas (7.2-4) merely requires that the net occupancy be 0 on the average. The associated playability question has been settled by Ford and Fulkerson (1962), who show that any $\mathbf{y}$ satisfying (7.2-4) is playable in the sense that there always exists a mixed strategy over paths for which $\mathbf{y}$ is the occupation probabilities of the arcs. Thus playability is assured.

We can now consider the problem of finding User's best occupation probabilities. Consider linear program LP′, which involves auxiliary nonnegative variables $\mathbf{u}$ and $\mathbf{w}$ in addition to $\mathbf{y}$:

$$
\begin{aligned}
& d = \min_{\mathbf{y}\geq\mathbf{0},\,\mathbf{w}\geq\mathbf{0},\,\mathbf{u}\geq\mathbf{0}} \sum_a (w_a + p_{0a}y_a) + \sum_k u_k \\
& \text{subject to (7.2-3) and} \\
& -y_a p_{ka} + w_a + u_k \geq 0 \text{ for all } a, k
\end{aligned}
\tag{7.2-5}
$$

Let $\mathbf{x}$ be any strategy for Breaker that meets the requirements of LP. Upon multiplying the last set of constraints of (7.2-5) through by x_{ka} and summing, we have

$$\sum_{a,k} x_{ka}p_{ka}y_a \leq \sum_a w_a \sum_k x_{ka} + \sum_k u_k \sum_a x_{ka} \leq \sum_a w_a + \sum_k u_k,$$

where the second inequality employs constraints (b) and (c) of LP. It follows that

$$\sum_a \left(p_{0a} + \sum_k x_{ka}p_{ka} \right) y_a \leq \sum_a (p_{0a}y_a + w_a) + \sum_k u_k$$

This inequality states that the average number of detections is bounded above by the objective function of LP′. Thus LP′ provides an upper bound on the game value that User can guarantee by playing $\mathbf{y}$*, the optimal solution of LP′. User can guarantee that the payoff will not exceed d*.

The remaining question is how do we know that the optimized objective functions of LP and LP′ are equal, as we have implied by calling both of them d*? The answer is simple: LP and LP′ are dual linear programs, as the reader can verify by writing out the dual of LP. Therefore the objective functions have to be equal.

We thus have a complete solution ($\mathbf{x}$*, $\mathbf{y}$*, d*) of the game G. It suffices to solve either LP or LP′, since the dual variables of either linear program provide the optimal strategy for the other player. No further work is required if only the game value is of interest. A sampling method is required for actual play in G, but we are guaranteed that both $\mathbf{x}$* and $\mathbf{y}$* are playable.

We close this section with some comments:

1. As usual it is permissible to set $z_s = 0$ in LP, in which case z_i is the average number of detections up to node i.
2. We gave a 3×5 example of $\mathbf{x}$ where all of the Breaker's assets were employed, but such examples would be impossible if there were more assets than arcs. In that case every arc would have an assigned asset in the solution of LP, so the columns of $\mathbf{x}$ would all sum to 1, rather than the rows. Such solutions are still playable.
3. If Breaker is not restricted to have at most one resource on an arc, the only change needed is to omit constraints (b) from LP, or variables $\mathbf{w}$ from LP′. Recall that the objective function is the average number of detections, so it does not matter whether multiple resources act independently or not. The playability question for Breaker is now trivial, since each row of $\mathbf{x}$ can be sampled independently of the others.

4. If Breaker's assets are in some cases identical copies of some common type, let x_{ka} be the average number of assets of type k assigned to arc a, let m_k be the number of assets of type k available, and change constraint (c) of LP to $\sum_a x_{ka} \le m_k$ for all asset types k. In LP′, the corresponding change is to multiply u_k by m_k. No other changes are necessary.
5. Suppose that the (asset, arc) pairs are partitioned into two sets. For pairs $(k,a) \in H$, Breaker's allocation is observable by User before he chooses his path. An example of such an asset might be a permanently manned inspection post. For all other pairs, the allocations are made in secret, as before. There are thus three kinds of asset allocations for Breaker:

   a. uncontrollable and known to User (asset 0),
   b. controllable and known to User, as in H, and
   c. controllable and unknown to User.

   Since LP is a maximization, the only modification that needs to be made is to add the constraint that x_{ka} must be either 0 or 1 for $(k,a) \in H$. LP is then no longer a linear program, but large instances are nonetheless solvable.

7.2.1 Surveillance of a Region

The network detection problem has some special cases that are worth noting. This subsection describes one of them.

Suppose that a geographical region is partitioned into cells indexed by a, one of which User must hide in, and let p_{ka} be the probability that asset k will detect User if both asset k and User occupy cell a. No network is given, but an artificial network can be introduced by letting A consist of one artificial arc connecting artificial node s to artificial node t for every cell. Hiding in a cell is equivalent to finding a path from s to t or, equivalently, choosing a single arc. Take $z_s = 0$ and omit the subscript on z_t. Then (7.2-2) simplifies to LP:

$$
\begin{aligned}
& d = \max_{\mathbf{x}\ge 0, z} z \\
& \text{subject to} \quad \text{(a) } z - \sum_{k,a} x_{ka} p_{ka} \le p_{0a} \text{ for all cells } a \\
& \text{(b) } \sum_{k} x_{ka} \le 1 \text{ for all cells } a, \text{and} \\
& \text{(c) } \sum_{a} x_{ka} \le 1 \text{ for all assets } k.
\end{aligned}
\tag{7.2-6}
$$

The objective function d is the probability that User will be detected. Constraints (b) may be omitted if multiple assets are permitted in a cell.

Example Suppose that smugglers attempt to infiltrate a border area during each night over a long time period. Breaker has three asset types available to detect these attempts: (1) four policemen from station 1, each of whom can be assigned to one border segment on any night, (2) four policemen from station 2 with similar constraints, and (3) one unmanned aircraft. Multiple assets are permitted in a cell. The border consists of four segments, and each of the nine assets must be allocated to one segment on each night. The detection probabilities per asset are

$$(p_{ka}) = \begin{bmatrix} 1 & 0.5 & 0.5 & 0.2 \\ 0.1 & 0.8 & 0.2 & 0.6 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

These numbers might reflect the idea that police assignments require significant travel time, which depends on the police station, and that an aircraft is perfectly effective in any single border segment. Upon solving LP, we find that

$$\mathbf{x}^* = \begin{bmatrix} 1.66 & 0 & 2.34 & 0 \\ 0 & 2.08 & 0 & 1.92 \\ 0 & 0 & 0.5 & 0.5 \end{bmatrix}, \text{ with } d^* = 1.66.$$

No matter which segment User chooses, the average number of detections will be 1.66, a number probably high enough to discourage smuggling in this border area. The allocations **x*** would not change if all of the detection probabilities were multiplied by 0.001, but smugglers would be undeterred.

There is an alternative interpretation of LP where the presence of both asset k and User in cell a activates an independent Poisson process of detections at rate p_{ka}. In this interpretation constraints (b) are omitted and the time horizon is assumed to be indefinitely long, rather than just long enough for User to make one transit. Assets are assumed to move from cell to cell over this long period, movements that we assume are frequent and unpredictable to User. If x_{ka} represents the average number of assets of type k that are present in cell a, then $\sum_{k,a} x_{ka} p_{ka}$ is the total rate of detection in cell a, and d is the rate of detecting User when both sides play optimally. The reciprocal of d is User's mean lifetime once he enters the protected area. Such models have been used to estimate the lifetime of a submarine when various kinds of antisubmarine assets (aircraft, ships, hydrophone arrays, ...) are optimally arrayed to minimize that lifetime (Brown et al. 2011).

7.2.2 Survival Probability Payoff

It was mentioned at the beginning of this section that mathematical necessity dictated using the average number of detections as the payoff of game G. However, there are circumstances where detections after the first are irrelevant, and should not be counted. Here we consider the limited results that are available when the payoff is the probability that the number of detections is 0, with User maximizing this survival probability and Breaker minimizing. As often happens in analysis, the extreme cases are easy while the middle ground is difficult. The extreme cases here are when Breaker is poor in assets or rich in assets.

When Breaker is poor, the possibility of multiple detections is so remote that it can be ignored. We can solve game G of Sect. 7.2, find that the optimized value d^* is small compared to 1, and argue that the survival probability is approximately $1 - d^*$. The expression $1 - d^*$ is always a lower bound on the survival probability, even when Breaker is rich, but it should also be a good approximation when Breaker is poor. The ultimate poverty would be when Breaker has only one asset, in which case multiple detections are not possible and the "approximation" is exact. When Breaker is rich, the $1 - d^*$ lower bound becomes useless because d^* will become greater than 1 (a negative lower bound on a probability is nothing to brag about).

When Breaker is rich, survival probability can be approximated by a particular upper bound. That upper bound corresponds to revealing Breaker's allocations to User before User chooses his path, which converts the problem into a minmax problem instead of a game. Let $\alpha_{ka} = -\ln(1 - p_{ka})$ and consider mathematical program MIP where variable x_{ka} is 1 if asset k is assigned to arc a, or otherwise 0 (in stating MIP and in the rest of this subsection, k is not 0):

$$\begin{aligned}
&\max_{\mathbf{x}\geq 0,\mathbf{z}} z_t \\
&s.t.\ \ z_j - z_i - \sum_k \alpha_{ka} x_{ka} \leq \alpha_{0a} \text{ for all } \ a = (i,j) \\
&\sum_k x_{ka} \leq 1 \text{ for all } a \\
&\sum_a x_{ka} \leq m_k \text{ for all } k \\
&z_s = 0 \\
&x_{ka} = \text{integer}
\end{aligned} \tag{7.2-7}$$

For lack of a better word, call $\alpha_{0a} + \sum_k \alpha_{ka} x_{ka}$ the "pressure" on arc a. Pressure is analogous to the average number of detections in LP. Indeed, except for the substitution of α_{ka} for p_{ka}, the only significant structural difference between LP and MIP is that x_{ka} must be an integer in (7.2-7), but not in (7.2-2).

As in Sect. 7.2 we can argue that the total pressure on any path connecting i to j will be at least $z_j - z_i$, and it follows that the total pressure on any path S connecting s to t will be at least z_t Since adding logarithms is equivalent to multiplying factors, for any path S from s to t,

$$\begin{aligned}
\prod_{a\in S} \left((1 - p_{0a}) \prod_k (1 - p_{ka})^{x_{ka}}\right) &= \exp\left(-\sum_{a\in S} \left(\alpha_{0a} + \sum_k \alpha_{ka} x_{ka}\right)\right) \\
&\leq \exp(-z_t)
\end{aligned} \tag{7.2-8}$$

Under the assumption that all interdiction attempts are independent, the left-hand-side is the probability of User's survival on path S. Thus $\exp(-z_t)$ is an upper bound on the survival probability. Indeed, it is the smallest upper bound that Breaker can guarantee without considering mixed strategies. The requirement that x_{ka} must be an integer cannot be omitted in MIP because of the nonlinear dependence on x_{ka} in (7.2-8).

The upper bound $\exp(-z_t)$ is likely to be unity when Breaker is poor, since User will nearly always find a safe route from s to t once he knows where Breaker's assets are placed. But when Breaker is rich enough to guard all paths, it should make little difference whether User knows Breaker's allocations or not, so the upper bound should be a good approximation. The integer restriction on $\mathbf{x}$ turns MIP into a Mixed Integer Program, a comparatively difficult type compared to a linear program, but solving MIP is still much simpler than trying to express the problem as a matrix game by writing out all of Breaker's pure strategies.

We thus can approximate the survivability game well when Breaker is poor or rich in assets, in the former case through LP and in the latter case through MIP. Unfortunately this excludes many practical problems.

The case where $p_{ka} = 1$ for all arcs and assets is special. All assets become equivalent, and a pure strategy for Breaker is simply a set of arcs to which some asset is assigned. The question of User's survival is equivalent to the question of whether the set of arcs in Breaker's path and the set of arcs with some asset assigned are mutually exclusive. Jain et al. (2011) report success in computing optimal mixed strategies for both sides in this special case by using the strategy generation algorithm of Sect. 3.11.2.

7.3 Exercises

1. Consider the shortest path problem on sheet "NetPath" of the TPZS workbook, except vary Breaker's budget from 0 to 20 and sketch a graph of shortest path length versus budget. Over that range, is the graph convex (increasing slope), concave (decreasing slope) or neither?

2. Sheet "NetFlow" of the TPZS workbook is an example of Breaker minimizing the maximum flow from s to t by removing arcs from a network, subject to a budget limitation. The set C of (7.1-9) is implemented by including an additional vector of variables $\mathbf{x}$. Variable x_k is 1 if arc k is removed, else 0. The sheet shows a feasible solution to (7.1-9) where $\mathbf{x} = (0,0,0,1,0)$ (arc d is removed) and $\mathbf{y} = (1,1,0,0,0)$ (the cut consists of arcs a and b). The resulting maximum flow is 6.

   a) Open Solver on sheet "Netflow", confirm that it is set to solve (7.1-9), and run Solver to confirm that the "minimized" maximum flow is indeed 6. This answer is wrong to the point of almost being silly, since arc d is not even part of the cut. Solver (in Excel™ 2010, at least) does not always find the optimal solution in nonlinear problems such as this.

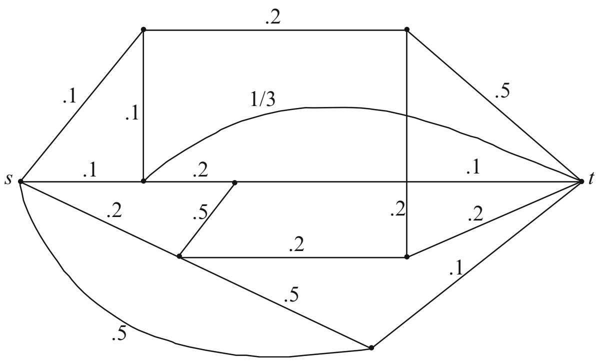
b) Guess a better set of arcs to remove and adjust the variables by hand to correspond to your guess. What is the true minimized maximum flow? If you run Solver again starting from your clever guess, does Solver repeat the mistake of removing arc *d*?

3. Sheet "NetDetect" of the TPZS workbook shows an example of a problem of the type considered in Sect. 7.2.1 where there is one asset of each of three distinct types.

   a) Run Solver to verify that the average number of detections is at least 1.1636 on every path from *s* to *t*, and then increase the number of assets of the second type to 2 and run Solver again. How much does the average number of detections increase?
   b) Round to integers the allocations of the three assets from part a in such a manner that each asset is still fully allocated. This will establish survival probabilities for User on each arc. What is User's survival probability on the most survivable path from *s* to *t*? There are only four paths, so just enumerate them. This is an upper bound on User's maximum survival probability, as discussed in Sect. 7.2.2.
   c) Play with it for a while. Change some of the green data and try to predict how allocations will change, then run Solver to see whether your guess is correct.

4. In the example of Sect. 7.2.1, find a way of playing **x***. It is relevant that there is no constraint prohibiting the assignment of multiple assets to an arc.

5. What is the solution of the network interdiction game shown below, where Breaker can select any single arc on which to ambush User? The interdiction probability for each arc is shown beside it. Your first action should be to subtract all the detection probabilities from 1, since the data required in Sect. 7.1.3 are survival probabilities.



6. Sheet "Iraq" of the TPZS workbook shows a formulation of (7.2-2) for the network shown in Fig. 7.1 and two units of the only asset.

a) Use Solver to find the solution when there is one unit of the only asset.
b) Use Solver to find the solution when there are ten units of the only asset.
c) Repeat part b, but first remove the constraints that require at most one unit on any arc. What is the relation between the solution of this problem and the solution in part a?

Chapter 8
Search Games

If Edison had a needle to find in a haystack, he would proceed at once with the diligence of the bee to examine straw after straw until he found the object of his search… I was a sorry witness of such doings, knowing that a little theory and calculation would have saved him ninety per cent of his labor

Nikola Tesla

In this chapter we consider searching for objects that don't want to be found. The searcher will be called Searcher, and the reluctant object of search will be called Evader. Search games are among the most common applications of game theory. A variety of problems can be posed, depending on whether Evader can move, the consequences of detection, whether the space to be searched is discrete or continuous, in the latter case how many dimensions there are, and so on. We have already considered several problems of this kind in previous chapters. These include

1. Search games where the two parties get information feedback as time goes by, as in Markov games. The inspection game of Sect. 4.1 is an example.
2. Sections 7.1.3 and 7.2 deal with search games where Evader is forced to move from s to t over a network while Searcher lies in wait for him. In this chapter Evader is not forced to move, although he sometimes does so to complicate life for Searcher. His only goal is to avoid detection.
3. Search games can be games on the unit square where the detection probability is a given function $f(x - y)$. The function might be bell-shaped, as in Sect. 5.1, or it might be as in exercise 18 of that chapter. Section 8.2.1 expands on this.

Our goal here is to review some search games that have known solutions, as well as some interesting games for which solutions are still unknown. Ruckle (1983) will receive frequent reference because he gives standard names to many of these problems. It is good to have names for things, even for mysteries.

A. Washburn, *Two-Person Zero-Sum Games*, International Series in Operations Research & Management Science 201, DOI 10.1007/978-1-4614-9050-0_8,

8.1 Search Games in Discrete Space

Here Evader simply chooses one of n cells to hide in. Possibilities for Searcher can be more complex.

8.1.1 Searcher Has a Single Look

Assume first that Searcher's single look will detect Evader with probability p_i if he and Evader choose the same cell i, or surely fail to detect Evader if Searcher chooses any other cell. The payoff is the probability that Searcher detects Evader, and the name of the game is One Shot. It is not hard to prove that both players should choose cell i with a probability that is inversely proportional to p_i, and that the value of One Shot is $p \equiv \left(\sum_{i=1}^{n} p_i^{-1} \right)^{-1}$. Section 3.9 includes an example. If any p_i is zero, then the value of the game is zero. It is counterintuitive that Searcher should be most likely to do what he is worst at. Intuition is perhaps happier with the statement that Searcher is most likely to do whatever Evader is most likely to do, which is also true.

Sakaguchi (1973) proposes a game closely related to One Shot where Searcher gets a fixed reward if he finds Evader, but must pay a search cost that depends on the cell searched. Such revisions of the payoff structure can strongly affect the nature of the game. The Scud Hunt game that occupies the rest of this subsection is another example of this dependence.

The Gulf War of 1990–1991 involved Iraq's use of Scud missiles. The launch of such a missile reveals the location of the launching site, at which can be found a nervous Transporter-Erector-Launcher (TEL) that itself becomes a target for retaliatory attacks by aircraft. The TEL must quickly find a place to hide after launching a Scud, perhaps under a bridge or in a grove of trees. There are only so many places to hide, and some are closer than others. If the TEL is tracked to a hiding place, it will be destroyed. It can even be destroyed if it is not tracked, but there are lots of hiding places and the aircraft cannot destroy them all. Where should the TEL hide, and (assuming the TEL is not tracked to a hiding place), which hiding place should the aircraft attack?

To abstract the situation, let there be n hiding places, all of which are known to both sides, and let q_i be the known probability that the launcher will get to the i^{th} hiding place without being tracked. If we assume that the aircraft will attack only if the TEL is tracked, then of course the TEL should select the hiding place for which q_i is largest, usually the closest hiding place. An interesting game results, however, if the disappointed aircraft attacks some hiding place in the hope of getting lucky, even if the TEL is not tracked. In that case it would be unwise for the TEL to always choose the closest hiding place, since that tactic leads to a predictable location.

Example Suppose $n = 4$ and $\mathbf{q} = (1/2, 1/3, 1/4, 1/5)$. We then have a 4×4 game where the payoff is the TEL's survival probability and the strategies for the two sides are the choice of a hiding place—in the case of the aircraft (Searcher) the choice is conditional on not tracking the TEL. The payoff matrix, with the TEL (Evader) playing the rows, is

$$\begin{bmatrix} 0 & 1/2 & 1/2 & 1/2 \\ 1/3 & 0 & 1/3 & 1/3 \\ 1/4 & 1/4 & 0 & 1/4 \\ 1/5 & 1/5 & 1/5 & 0 \end{bmatrix}$$

In row 2 column 3 for example, Evader chooses the second hiding place while Searcher conditionally chooses the third. Evader's survival probability is 1/3 (not 1) because the aircraft will attack the second hiding place if the TEL is tracked there, even though the aircraft's default choice is the third hiding place. It can be verified that the solution of this game is $v = 2/9$, $\mathbf{x}^* = (2,3,4,0)/9$, and $\mathbf{y}^* = (5,3,1,0)/9$. Note that Evader is more likely to choose hiding places 2 and 3 than hiding place 1, the place he is most likely to get to without being tracked. In spite of this oddity, Evader can guarantee to survive with probability 2/9 by playing $\mathbf{x}^*$, and it is pointless to hope for a larger payoff because Searcher can guarantee that the payoff does not exceed 2/9 by playing $\mathbf{y}^*$. Neither side ever chooses hiding place 4.

We can generalize this example by letting the survival probability when the correct hiding place is attacked be something larger than 0. Let that probability be p_i in hiding place i, assumed to be smaller than q_i. We then have a game where the payoff is

$$a_{ij} = \begin{cases} p_i & \text{if } i = j \\ q_i & \text{if } i \neq j \end{cases} \tag{8.1-1}$$

Without loss of generality (we can always rearrange the rows), assume q_i is nonincreasing in i, as in the example. Also let $d_i \equiv 1/(q_i - p_i)$, a positive quantity. Define J to be the smallest index such that

$$1 + q_j \left(\sum_{k=1}^{j} d_k \right) > \sum_{k=1}^{j} q_k d_k \text{ for } j \leq J \tag{8.1-2}$$

with the inequality being first false for $j = J + 1$. If the inequality is true for all j, take $J = n$, the number of hiding places. Let $A = \sum_{j=1}^{J} q_j d_j$ and $D = \sum_{j=1}^{J} d_j$. Then the solution of the game is

$$\begin{aligned} v &= (A-1)/D, \\ x_j^* &= d_j/D \quad \text{for} \quad j \le J, \text{else } 0, \text{and} \\ y_j^* &= d_j\left(q_j - v\right) \quad \text{for} \quad j \le J, \text{else } 0. \end{aligned} \tag{8.1-3}$$

We are using the bare hands technique for solving the Scud Hunt game—first guess the solution and then prove that the guess is correct. In this case the guess is essentially that both players will ignore hiding places where q_i is sufficiently small, and that otherwise the strategies of both players will be equalizing.

To prove that the offered solution is actually correct, we must show that the two optimal mixed strategies are indeed probability distributions, and that each guarantees v from its player's viewpoint.

Clearly Evader's $\mathbf{x}^*$ is nonnegative, and it should be clear from the definition of D that $\sum_{i=1}^{n} x_i^* = 1$.Therefore $\mathbf{x}^*$ is a probability distribution. Since $1 + Dq_J > A$ by definition of J, we must have $q_J > v$ by definition of v, and therefore $q_j > v$ for $j \le J$ because q_j is nonincreasing. Therefore y_j is nonnegative for $j \le J$. Since $\sum_{j=1}^{J} y_j^* = A - vD = 1$, $\mathbf{y}^*$ is also a probability distribution. Thus $\mathbf{x}^*$ and $\mathbf{y}^*$ are both probability distributions.

If Searcher uses $\mathbf{y}^*$, the payoff to Evader for any strategy j that does not exceed J is $q_j - y_j^*/d_j = v$. Thus Searcher's strategy is equalizing when Evader uses one of his active strategies (indeed, this equation was solved to get the expression for y_j^*). The payoff is q_{J+1} if Evader uses inactive strategy $J + 1$.We can use the definition of J to show that $1 + Dq_{J+1} \le A$, which is equivalent to $q_{J+1} \le v$, and it follows that $q_j \le v$ for $j > J$. Thus Searcher can guarantee v by using $\mathbf{y}^*$.

When Evader uses strategy $\mathbf{x}^*$, $\sum_{i=1}^{J} x_i^* q_i = A/D$, which exceeds v. This is the payoff when Searcher uses an inactive strategy. When Searcher uses active strategy i, the payoff is smaller because of the possibility that both players will choose the same strategy. Specifically, the payoff in that case is $A/D - x_i^*/d_i = A/D - 1/D$. But this is just v, so Evader can guarantee v, regardless of the strategy used by Searcher. The hypothesized solution is therefore correct.

Dresher (1961) gives an example with a different interpretation, but the same form of payoff matrix as Scud Hunt. In Dresher's example an attacker and defender must each choose one target out of n, with q_i being the value of the selected target. The payoff to the attacker is q_i if he attacks an undefended target, or else a smaller payoff p_i if the defender and the attacker both choose target i. The exploitable feature in both interpretations is that, except on the main diagonal, the payoff depends only on the maximizer's choice.

The game One Shot is "completely mixed" (meaning that all strategies are active for both sides), but Scud Hunt is usually not because both sides ignore the high-indexed strategies. Both games are square search games, but the solutions are qualitatively different.

8.1.2 Searcher Has Multiple Looks, No Overlook Probability

"Multiple looks" might mean several looks at one time or one look at several times. Our focus here is on the latter. Section 7.2 deals with the former.

We assume that Searcher moves from cell to cell, looking once in each cell, and the payoff is the average number of looks needed to finally encounter Evader. Ruckle (1983) refers to this as the Search on a Complete Graph (SCOM) game. Searcher has $n!$ strategies, since he can search the cells in any order. The factorial function grows very fast with n, so solving the game as a matrix game would not be attractive when n is large. Nonetheless, it is known that the value of this game is $v = (n + 1)/2$ for $n > 0$. An optimal strategy for Evader is to choose a cell at random. Searcher can choose any permutation of the cells, and then flip a coin to decide whether to examine them forward or backward.

Here is a proof that Searcher's strategy will guarantee v. Let random variable N be the number of looks required for detection. If Evader chooses to hide at the i^{th} cell in Searcher's chosen permutation, N will be i if Searcher goes forward, or $n + 1 - i$ if Searcher goes backward. Since Searcher is equally likely to go forward or backward, $E(N) = (n + 1)/2$, regardless of i. Thus Searcher can guarantee that the payoff will not exceed v (it will in fact equal v).

It is more difficult to prove that Evader's strategy will guarantee v, the difficulty being that all of Searcher's strategies must be considered, including those that visit a cell multiple times. Let m_k be the number of distinct cells visited by Searcher in the first k looks, so $m_k \leq k$. The event $N > k$ is the event that Evader's cell is not among the first k looks. Since Evader chooses a cell at random, $P(N > k) = (n - m_k)/n \geq (n - k)/n$. We next utilize the fact that $E(N) = \sum_{k=0}^{\infty} P(N > k)$, which is true for all nonnegative, integer-valued random variables. Thus $E(N) = \sum_{k=0}^{\infty} P(N > k) \geq \left(\sum_{k=0}^{n} (n - k) \right) / n$. But the last sum is the sum of the first n integers,[1] which is $n(n + 1)/2$, so $E(N) \geq (n + 1)/2 = v$. Since $E(N)$ is at least v no matter what Searcher does, the proof is complete.

So far we have assumed that Searcher can choose any of the $n!$ permutations of n cells, but the same proof works for a more restricted Searcher as long as he is free to examine some permutation both forward and backward. For example, the cells might be arranged in a line, with Searcher's next cell being allowed to be 2 from cell 1, $n - 1$ from cell n, or to be either neighbor from each of the interior cells. This is Ruckle's Search on a Linear Graph (SLIN) game. The permutation where the cells

[1] This is the formula that got Gauss into trouble as a boy. His teacher told him to add up the first 100 integers, thinking to keep Gauss busy for a while, but Gauss promptly announced that the sum is 5050!

are arranged in numerical order is still available, either forward or backward, so SLIN has the same solution as SCOM. If SLIN is modified so that Searcher cannot move backwards from cell n, we have the Search on a Directed Linear graph (SDL) game. In SDL, there is no permutation where Searcher can proceed in either direction. The value of SDL is n, and Searcher's optimal strategy is to always start in cell 1, since all other starting cells risk the possibility of an infinite payoff. Evader can guarantee that the payoff will exceed any number short of n, but cannot guarantee n, so this is an example where Evader does not have an optimal strategy (Theorem 3.2-1 does not apply because one of the payoffs in SDL is infinite).

Instead of counting looks, we might instead measure the total distance traveled by Searcher before finally encountering Evader. Let d_{ij} be the distance between cells i and j. Searcher can begin anywhere, and can visit the cells in any order, so he has the same $n!$ strategies that he does in SCOM. If his first cell happens to be the one selected by Evader, the payoff is 0; otherwise, until Evader is encountered, we begin adding up distances d_{ij} whenever Searcher moves from i to j. Exercise 21 of Chap. 3 is an example of this kind of game. It was presented at that early point because the author does not know how to solve it except by the brute force technique of enumerating all six of Searcher's strategies. There is as yet no general solution to games of this kind, although there are known solutions to special cases. Kikuta and Ruckle (1994), for example, give a solution for games where the cells are arranged in a tree, with Searcher starting at the root.

8.1.3 Searcher Has Multiple Looks, Nonzero Overlook Probability

Here we generalize the problems of 8.1.2 to include the possibility that Searcher looks in the correct cell, but doesn't detect (overlooks) Evader. Let p_i be the detection probability in cell i, assumed to be less than 1 in at least one cell, positive in all cells, and the same for every look. All looks are assumed to be independent, and the payoff is the average number of looks required for detection. Searcher is assumed to know whether each look succeeds or fails.

Much depends on the mobility of Evader. Suppose first that Searcher and Evader are both free to choose any cell on each look (Evader is thus very mobile). This game amounts to playing One Shot repeatedly until detection finally occurs. The number of One Shot plays until termination is a geometric random variable with success probability p, the value of the One Shot game, so the value of the overall game is $1/p$. This game is easily solved in spite of the fact that both sides have infinitely many strategies.

The game becomes more complicated if Evader must choose a cell and stay there, even though Evader now has only finitely many strategies. To illustrate the difficulty, suppose there are only two cells, with $p_1 = 1$. Searcher might reason that he should first search cell 1, and then, if the look in cell 1 fails, look in cell 2 until detection finally occurs. This strategy is simple because the very first look will settle the question of Evader's location (although Searcher must still continue looking in cell 2 until detection occurs), but it cannot be an optimal strategy. If it were optimal, then Evader would always hide in cell 2, which is not consistent with Searcher always beginning in cell 1. However, at least we can exploit the fact that Searcher should never look in cell 1 more than once. Let s_j be the strategy of placing the j^{th} look in cell 1 and all other looks in cell 2, so that the strategy posed above is s_1. Against s_j, the payoff is $a_{1j} \equiv j$ if Evader hides in cell 1, or $a_{2j} \equiv (1 - p_2)^{j-1} + 1/p_2$ if Evader hides in cell 2. The reasoning behind the second formula is that there will be $1/p_2$ looks in cell 2, on the average, plus one look in cell 1 if the first $j - 1$ looks in cell 2 fail. Since Evader (the maximizer) has only two strategies, we can solve any specific game graphically. If $p_2 = 1/2$, for example, the value of the game is 2.4, with Searcher's only active strategies being s_2 and s_3. More generally one can establish that Searcher will never have more than two active strategies, regardless of p_2, and that the two active strategies will always be sequentially numbered.

Bram (1963) considers the general game where there are n cells. He establishes that both sides have optimal strategies (this is not trivial, since the game is not finite) and that Searcher has an optimal strategy that is a mixture of at most n pure strategies. The latter fact might lead to a computational method based on strategy generation (Sect. 3.11.2), or see Neuts (1963). However, there does not seem to be a simple analytic formula for the value of this game. Apparently the "simplification" of forcing Evader to be stationary actually leads to difficult optimization problems.

Next suppose that the payoff is the total distance travelled by Searcher before detection occurs, rather than the number of looks. The worst case from Searcher's viewpoint is when Evader is not only mobile, but can use his knowledge of the location of past unsuccessful looks to choose his next location (Norris 1962). A practical example of this might have Searcher being a helicopter with a dipping sonar looking for a submarine in an enclosed bay. The submarine Evader can hear every ping (look) of the sonar, and can take advantage of this in choosing the next place to hide. Evader can make Searcher's problem difficult by moving away from the cell most recently searched, but should not always move to the cell that is farthest away, since doing so would permit Searcher to predict Evader's next position. What is the right compromise for Evader, and should Searcher emphasize cells that are close to his most recent ping, which are easy to get to, or cells that are far away, which are more likely to include Evader? To answer these questions we can formulate a Markov game G as in Chap. 4. In game G each element G_i is subscripted by the cell most recently searched. Evader cannot force infinite average play, so it suffices to solve the value equation (4.2-3).

The distinguishing feature of G is that, except for the possibility of detection, Searcher's column choice alone determines the succeeding game element, thus making G a "single controller" game (Raghavan and Filar 1991). This fact simplifies the solution of the value equation, and in some cases even permits an analytic solution. Consider any example where symmetry dictates that all game elements have a common value v, so that solution of any game element suffices. For example we might have four cells arranged cyclically in a unit square where d_{ij} is the shortest distance (clockwise or counterclockwise) from cell i to cell j, plus 1 (we are adding 1 to make the point that the time required to search a cell can be included, as well as travel time between cells). Then, with rows and columns labeled 1, 2, 3, 4, we have for game element G_1

$$v = val\left(\begin{bmatrix} 1 & 2+v & 3+v & 2+v \\ 1+v & 2 & 3+v & 2+v \\ 1+v & 2+v & 3 & 2+v \\ 1+v & 2+v & 3+v & 2 \end{bmatrix}\right)$$

The payoff in row 2, column 4 is 2 + v because it takes one time unit to go from cell 1 to its cyclic neighbor cell 4, the search of cell 4 takes an additional time unit, the search fails because row and column do not agree, and game element 4 with value v must be played next. Searcher's strategy **y** will make the payoff in every row equal to v, and so, letting $\overline{v} \equiv y_1 + 2y_2 + 3y_3 + 2y_4$, we must have $v = \overline{v} + v(1 - y_i)$ in row i. Evidently y_i must be independent of i, and consequently equal to 0.25. Therefore $\overline{v}$ must be 2, and v must be 8. As long as the game starts with Searcher's location known to Evader, Searcher will have to move a distance of 8, on the average, before detecting Evader. More generally, as long as $\sum_{j=1}^{n} d_{ij}$ is the same for every i, then that common number is also the value of the game, and the optimal strategy for Searcher is uniform. The optimal strategy for Evader is not uniform—we leave its derivation to exercise 8. Washburn (1980) has further detail on this kind of game.

8.1.4 Searcher Has Continuous "Looks"

In this case we assume that Searcher has some resource that he must divide among the cells, y_i in cell i, with the conditional nondetection probability in cell i being $f_i(y_i)$ if Evader is located in cell i, or otherwise 1. These functions are all assumed to be decreasing and convex. In the case of "random search" (Stone 1975) the resource is time and we have $f_i(y_i) \equiv \exp(-\alpha_i y_i)$, where the nonnegative parameters α_i depend on the speed and sweepwidth of Searcher in cell i. We take the nondetection probability to be the payoff, so Searcher is the minimizer.

Let x_i be the probability that Evader hides in cell i; $i = 1,\ldots, n$. The nondetection probability is then $A(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{n} x_i f_i(y_i)$, to be minimized by Searcher under the constraint that $\sum_{i=1}^{n} y_i \le c$, where c is the total amount of resource available. This is a Logistics game as defined in Sect. 5.3, and can be solved using the techniques described there. Exercise 16 of Chap. 5 is an example. A curiosity here is that solving this game is simpler than solving the corresponding minimization problem where $\mathbf{x}$ is given, since the latter requires the solution of a nonlinear mathematical program.

Hohzaki and Iida (1998) formulate a similar problem in which Evader chooses a path $\omega \in \Omega$, where Ω is some arbitrary nonempty set of feasible paths among the cells. For $t = 1, \ldots, T$, $\omega(t)$ is the cell occupied by Evader at time t, while $c(t)$ is the known amount of resource available at that time for Searcher. The problem considered in the previous paragraph is thus the special case where $T = 1$.

8.2 Search Games in Continuous Space

Here Evader is free to choose a hiding point in some nonempty convex set that is a subset of either the real line (one dimension) or the plane (two dimensions). Since space is continuous, these games usually include specification of an additional parameter—the distance d at which Searcher detects Evader.

8.2.1 One Dimension, Including Certain Networks

Consider a game where both players choose a point in the unit interval, x for Searcher and y for Evader. The payoff to Searcher is 1 if $|x - y| \leq d$, or otherwise 0. This game on the unit square can be thought of as one where Evader hides in the unit interval, while Searcher attempts to find him using a sensor with detection radius d. It generalizes exercise 18 of Chap. 5, where d was taken to be 0.2. One might expect the value of the game to be $2d$, the length of the interval that Searcher can cover, and that both players should uniformly choose a point in the unit interval. That would be true if the unit interval were made into a circle by connecting the endpoints, but it is not true here because of end effects. If Searcher were to uniformly choose a point in the unit interval, Evader could hold the payoff to only d by choosing one of the endpoints.

The solution of this game has both players using discrete mixed strategies. Specifically, let n be the largest integer such that $2nd \leq 1$. An optimal strategy for Evader is to randomly hide at one of $n + 1$ evenly spaced points in the unit interval, including points 0 and 1. Since the points are separated by more than $2d$, Searcher cannot cover more than one of them with a subinterval of length $2d$, so the payoff cannot exceed $1/(n + 1)$. On the other hand Searcher can completely cover the unit interval with $n + 1$ subintervals of length $2d$, and then randomly choose the center of one of those subintervals. If he does so, every point in the unit interval is included in the chosen subinterval with probability at least $1/(n + 1)$. Therefore the value of the game is $1/(n + 1)$. This value is smaller than $2d$; when d is 0.2, for example, the value is 1/3, rather than 0.4.

Ruckle (1983) refers to the game just described as the Interval Hider Game, or IHG. The IHG has a known solution even if Evader is required to choose a subinterval of specified length, rather than a point. Arnold (1962) generalizes to cases where the coverage probability decreases gradually as separation distance increases, rather than abruptly as in the IHG. The solutions of these generalized games still involve only a finite number of strategies, as do the solutions of the bell-shaped games mentioned in Sect. 5.1.

If Searcher can move, we dispense with the parameter d and require Searcher to move at unit speed until his position is coincident with the stationary Evader's. Searcher can start his path at any point he chooses. The time until coincidence happens is now the payoff, so Searcher is the minimizer. When played on an interval of length L, the value of the game is $L/2$, with Evader choosing a point at random and Searcher moving from one randomly selected end of the interval to the other, much as in the SLIN. Essentially the same solution applies if the game is played on any undirected network that includes at least one Eulerian path (a path that includes every arc exactly once). Evader should choose a point at random on one of the arcs, and Searcher should choose any Eulerian path, flipping a coin to decide whether to go from start to end or vice versa. If the total length of all the arcs in the network is μ, the value of the game is $\mu/2$. The proof of this statement closely resembles the proof for the SCOM game, as do the optimal strategies. Another solved case is where the network is a tree, and Searcher is required to start at its root Gal (1979) shows that the value of the game in this case is μ, and also considers some games where Evader is mobile. There is no known general solution that applies to all networks.

8.2.2 Stationary Search in Two Dimensions, HDG

As a basic example of stationary search in two dimensions, consider Ruckle's Hiding in a Disk Game, or HDG. Both players select a point within the unit circle, and Searcher wins if and only if his point is within Euclidean distance d of Evader. Let $HDG(d)$ be the value of the game. The solution of HDG is known only for certain values of d. For example $HDG(1) = 1$, with Searcher always occupying the center of the circle, and $HDG(0.5) = 1/7$. An optimal strategy for Searcher when $d = 0.5$ is to randomly choose one of seven points, those being the center of the disk or the center of one of the six edges of a hexagon inscribed in the disk. Seven circles with radius 0.5 centered on those points completely cover the disk, so any location of the Evader will be included in at least one of them. An optimal strategy for Evader is to choose the center of the disk with probability 1/7, or else uniformly choose a point on the circumference.

HDG has so far resisted attempts to solve it in general. Danskin (1990) shows that, when d is small, there is no solution where the mixed strategies involve only finitely many points, noting that "virtually nothing is known" about such games. However, we can at least show that the value of HDG is essentially d^2 when d is small. By "essentially", we mean that $\lim_{d \to 0} HDG(d)/d^2 = 1$. To prove this, first observe that Evader can guarantee an upper bound of d^2 by simply hiding at a point chosen uniformly at random in the disc, call this strategy A. Evader will be discovered only if a circle with radius d with Evader at its center contains the point chosen by Searcher, and this probability cannot exceed d^2, the ratio of the area of Evader's circle to the area of the unit disc. Therefore Evader's use of strategy A guarantees a detection probability of at most d^2.

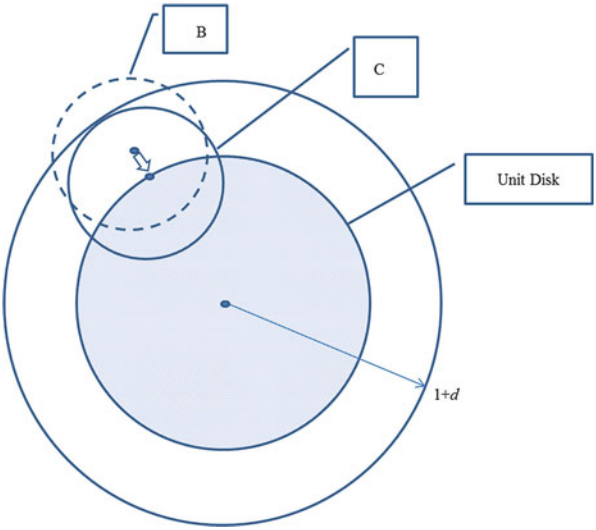


Fig. 8.2-1 The center of the dashed circle (*B*) is shown outside of the unit disk. As the center of *B* moves radially inward to rest on the unit disk edge (as at *C*), the covered part of the unit disk increases

Strategy A will not work for Searcher because Evader has access to the edge of the disk where much of the area covered by Searcher's circle is wasted. Consider strategy B where Searcher covers a circle whose center X is uniformly located inside an expanded disk with radius $1 + d$ (see Fig. 8.2-1). Since a circle centered at any point in the unit disk is entirely contained in the expanded disc, and since the area of the expanded disk is $\pi(1 + d)^2$, strategy B guarantees a detection probability of at least $d^2/(1 + d)^2$, even if Evader hides on the boundary of the unit disc. Strategy B is illegal, since Searcher is required to choose a point within the unit disc, but strategy B is easily modified to strategy C, which simply moves any point X that is not located inside the unit disk radially inward until it is on the boundary (again see Fig. 8.2-1). The moved point will always cover more of the unit disk than the unmoved point, so strategy C has at least as good a detection probability as strategy B, in addition to being legal. Considering both the upper and lower bounds, we therefore have

$$\frac{d^2}{(1+d)^2} \le HDG(d) \le d^2 \tag{8.2-1}$$

For $d = 0.5$, this leads to bounds of 1/9 and 1/4 on a game value that we already know to be 1/7. For small d, it leads to our contention that the value of the game is essentially d^2. When the detection radius is very small, edge effects become negligible.

There is a relationship between the HDG and the ancient geometric problem of packing a given number of unit circles into the smallest possible larger circle. For example Fodor (1999) has proved that 19 unit circles can be packed into a circle of radius $1 + \sqrt{2} + \sqrt{6}$, but not into any smaller circle. The centers of those unit circles lie within a circle of radius $\sqrt{2} + \sqrt{6} = 3.86$. It follows that $HDG(1/3.86) \leq 1/19 = 0.053$. Evader can randomly choose the center of one of 19 circles at which to hide, assured that Searcher cannot cover more than one of those points. This is a better upper bound than the one provided by (8.2-1), which is substantially larger at 0.067. Lower bounds could be obtained from the solutions of the equally ancient circular coverage problem where a given number of unit circles must completely cover a larger circle.

As in the IHG, the HDG simplifies if edge effects are eliminated. If S is the surface of a sphere, rather than a disk, both players simply choose a point at random on the sphere.

8.2.3 Moving Search in Two Dimensions, P&M

The natural world contains some hints about searching when both parties are capable of motion. Often Searcher is a predator and Evader is prey that is compelled to move by the necessity of browsing. Searcher is conflicted. If he moves about, the noise he makes himself may obscure the noise made by his prey, thereby causing him to miss a possible detection, or the prey may hear Searcher first and go into hiding. But if Searcher remains stationary, he will cover so little territory that he is unlikely to encounter any prey, either aurally or through his other senses. Some searchers resolve the conflict by staying in motion most of the time in order to cover territory, but with occasional pauses where the detection range for prey is increased. Military searchers sometimes "sprint and drift" for the same reasons, carefully judging the timing of switches between the two modes. Other searchers (spiders, certain cats) instead remain stationary, relying on prey motion to bring prey within range. We can again see a military parallel, since this is the tactic of booby-traps and mines. The prey's dual motivation is crucial—remaining stationary would best avoid detection by predators, but motion is required to mate or secure sustenance.

There are very few search games of this type that have been solved, or even partially solved. Two games on which some progress has been made are reviewed in this subsection. In both of them the detection range is taken to be a given quantity that does not depend on the speed of either party, and Evader's singular object is to avoid detection. As we shall see, Evader will choose to move about in both games in spite of having only one objective.

In the Princess and Monster (P&M) game, Princess is trapped within the unit circle, pursued by a blind Monster who will catch her if the distance between them ever gets as small as d, a given small but positive capture distance. The P&M game was introduced by Isaacs (1965) as a prototype where both parties are capable of motion. Monster can move at any speed up to v, a given constant. Princess is also blind, but can move in any continuous manner within the disk. The payoff is the time for Monster to capture Princess, so Princess is the maximizer.

It is perhaps intuitive, and in any case correct, that Monster should dash about randomly at top speed. However, intuition fails for Princess. If she remains stationary, Monster can exhaustively search the disk at area coverage rate $\lambda = 2vd$, and will find her after at most $\tau \equiv \pi/\lambda$, the time required for Monster to completely cover the unit disk. However, the *average* time to find her will be only half of that, and seemingly Princess can hope to hold out for longer than $\tau/2$. If she could manage to make her positions uniformly distributed over the disk and independent at all times, the time to detection would be an exponential random variable with mean τ, which is twice as large as she can guarantee by remaining stationary. But moving about opens up the possibility that she will run into Monster, which is just as bad as vice versa. What should she do?

Gal (1979) gives asymptotically optimal strategies for the two sides and an approximate value for the P&M game. An optimal strategy for Princess has her connecting a sequence of dots that are selected independently and uniformly in the region. At each dot Princess pauses for a carefully selected amount of time before moving on to the next dot. Thus Princess moves slowly, but still fast enough to introduce some unpredictable independence into her sequence of positions.

Lalley and Robbins (1988) give an appealing optimal strategy for Monster that works in any convex region. The strategy is a diffuse reflection wherein Monster constantly moves at top speed, reflecting randomly from the boundary of the region in the same manner as light reflects from a diffuse surface. That strategy has the property of equalizing the amount of time spent in all parts of the region, and is therefore an attractive candidate for motion that covers a region "randomly and uniformly" at constant speed. Regardless of what Princess does, this strategy guarantees a detection time of at most τ.

The net result of all this maneuvering by Princess and Monster is that the time to detection is approximately an exponential random variable with mean τ, which is what Princess hoped to accomplish by moving. This is as close as game theory gets to making fairy tales come true. Capture is inevitable, unfortunately, so all Princess can do is delay it as much as possible.

Capture is inevitable even in a revised game where Princess is always aware of Monster's location. There are no analytic solutions of such games, but Washburn (2002) describes some operational gaming experiments where Princess is always aware of the direction to Monster as the game proceeds. This information opens up the possibility of moving directly away from Monster, significantly increasing her mean survival time.

The P&M game is not without its practical applications. Someone trying to avoid an aircraft that he cannot see has a similar problem.

The Flaming Datum problem is related in an odd way to the P&M game. The name comes from a situation where a submarine has just torpedoed a merchant ship, and retaliatory forces arrive after a time delay. The torpedoed ship marks the place where the submarine once was, and thus acts as a “flaming datum” for the subsequent search for the submerged Evader. Evader might also be a burglar who has triggered an alarm and expects pursuit by police. In general, Evader commits an act that marks his position at time 0. After a time delay τ, Searcher arrives and begins searching while Evader tries to leave the area at a speed that cannot exceed U. At time t, Evader must be somewhere within a farthest-on-circle (FOC) with radius Ut. This expanding circle takes the place of the unit disk that confines Princess in the P&M game. Like Princess, Evader is assumed to know nothing about the location of Searcher.

Princess’s speed is not a P&M parameter—she uses her mobility only to foil Monstrous attempts to search the unit disk exhaustively. However, Evader’s top speed is a crucial input because it affects the size of the FOC. Like Princess, Evader would like to make his positions uniform within the FOC at all times, as well as independent from time to time. Since Princess succeeds at approximately doing that, we can approximate the Flaming Datum problem by assuming that Evader can do likewise.

Assume that Searcher can examine area at some fixed rate R. This rate might be the product of Searcher’s speed and the width of the strip that he can effectively examine. The ratio of R to the area of the FOC is a detection rate $\lambda(t) \equiv R/(\pi U^2 t^2)$. The detection rate $\lambda(t)$ is constant in the P&M game, but a rapidly decreasing function of time in the Flaming Datum problem because of FOC expansion. The average number of detections over the time interval $[\tau, T]$ can be obtained by integrating the detection rate:

$$n(T) \equiv \int_{\tau}^{T} \lambda(t)dt = \frac{R}{\pi U^2}\left(\frac{1}{\tau} - \frac{1}{T}\right)$$

Finally, if Evader is as successful as Princess at achieving his independence goal, $n(T)$ will be the mean of a Poisson random variable, so the probability of at least one detection will be $Pd(T) \equiv 1 - \exp(-n(T))$. This detection probability function is shown in Fig. 8.2-2, where all parameters are 1 except that $\tau = 0.5$. The detection probability rises quickly as soon as Searcher arrives, but then flattens out as the FOC expands. The detection probability does not approach 1 in the limit as T approaches infinity.

There is very little of game theory in this analysis of the Flaming Datum problem, and no tactical guidance for the players other than the vague admonition for Evader to “make your position uniform within the FOC at all times, and independent from time to time”. It is nonetheless useful to have an approximation for the value of the game, and our confidence in the quality of that approximation is improved by knowledge that the same kind of analysis works in the similar P&M game. See Danskin (1968) for a different look at the Flaming Datum problem as a game, Hohzaki and Washburn (2001) for an analysis similar to this where the

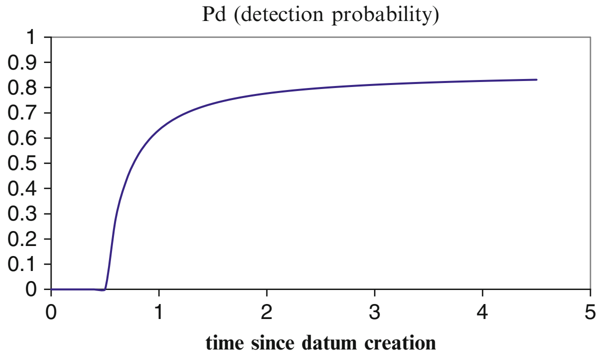


Fig. 8.2-2 Detection probability versus time T for the Flaming Datum problem

evader has an energy constraint, rather than a speed constraint, or Thomas and Washburn (1991) for a generalization. Exercise 22 of Chap. 3 deals with a discrete approximation in one dimension.

8.3 Exercises

1. Evader can hide in one of three cells, and Searcher must correctly guess Evader's cell to have any hope of finding him. If Searcher guesses correctly, the probability of finding Evader is 0.1, 0.2, and 0.5 in the three cells. The payoff is 1 if Evader is found, or otherwise 0. What is the solution of the game?

2. Use formulas (8.1-1)–(8.1-3) to solve the specific 4×4 game given in Sect. 8.1.1.

3. The last part of verifying the solution of Scud Hunt given in Sect. 8.1.1 is to show that the payoff is v when Searcher uses active strategy i against Evader's $\mathbf{x}^*$. Complete the details of the argument.

4. It was shown in Sect. 8.1.2 that the value of the SLIN game is $(n + 1)/2$. The Search on a Directed Cyclic graph game (SDC) is similar to the SLIN, except that there is good news and bad news for Searcher. The good news is that Searcher is allowed to move from cell n to cell 1, which he cannot do in SLIN. The bad news is that he is not allowed to go backward. He can start anywhere he wants, but must move cyclically through the cells. Solve SDC by the bare hands technique.

5. In Sect. 8.1.3, a game whose value is claimed to be 2.4 is formulated. Prove that the value is indeed 2.4, and find the optimal strategies for the two sides.
6. Same as exercise 5, except change p_2 to 1/3. The graphical technique of Sect. 3.6 can be used to solve the game.

7. In solving the value equation for the Markov game of Sect. 8.1.3, it is claimed that the optimal strategy for Searcher must be equalizing, which is equivalent to

arguing that all of Evader's strategies must be active in an optimal solution. Why must that be true? Assume instead that the Evader has a 0 optimal probability of moving to some cell, and find a contradiction.

8. What is Evader's optimal mixed strategy in the 4×4 Markov game considered in Sect. 8.1.3?
9. Modify the Markov game of Sect. 8.1.3 so that there are only three cells. The distance between cells is 2 in all cases, except that the distance from a cell to itself is 1. Find the value of the game and the optimal strategies for both sides.
10. Let $\boldsymbol{\alpha} = (1, 2, 4, 4)$ and $c = 8$ in a four-cell version of the random search game considered in Sect. 8.1.4. What is the solution of the game?
11. Consider an undirected network with four nodes at the corners of a square and an arc from each node to both of its neighbors. All four of these arcs have unit length. There is also a fifth diagonal arc from node 1 to node 3 that has length 6. Searcher and Evader each select a point somewhere on one of the arcs. Evader does not move, but Searcher moves at speed 1 until he finds Evader. The payoff is the length of time until this happens. Is this game covered in Chap. 8? If so, what is its solution?
12. A strategy for Evader in the game HDG(0.5) is given in Sect. 8.2.2, but a proof that it guarantees a detection probability of at most 0.5 is missing. Provide one.
13. Consider a one-dimensional P&M game played on the interval $[-1, 1]$, with Monster having unit speed and Princess having any speed she wants. Each player is free to start at any point, and capture occurs at the first time when the two positions are equal. Strategy A for Monster is to flip a coin to decide at which end to start, and then proceed at unit speed to the other end. Strategy B for Princess is to always start at the origin, stay there for slightly less than one unit of time, and then flip a coin to decide which end to go to at unit speed. If Princess is lucky, she will move in the same direction as Monster and capture will occur at time 2. If she is unlucky, capture will occur at (actually just before) time 1.The average capture time when A is employed against B is 1.5. Is this the value of the game?

Chapter 9
Miscellaneous Games

And now for something completely different

Monty Python

In this last chapter we collect some games that have little to do with one another, but which are individually interesting.

9.1 Ratio Games

In a Ratio game, **x** and **y** as usual represent mixed strategies for players 1 and 2, but the payoff is

$$R(\mathbf{x},\mathbf{y}) \equiv \frac{\sum_{i,j} x_i y_j a_{ij}}{\sum_{i,j} x_i y_j b_{ij}} \tag{9.1-1}$$

We assume that **a** and **b** are both finite matrices of equal dimensions. We also assume that $b_{ij} > 0$ for all i, j, so there is no danger of dividing by 0. One direct motivation is that the numerator and denominator represent "payoffs" for players 1 and 2, but what the players really care about is the ratio. Player 1 judges his own payoff in relation to player 2's, and vice versa. We have all known people who think that way.

Ratio games can arise in a variety of circumstances, some of them surprising. Stochastic games with only a single game element turn out to be equivalent to Ratio games. For such games the value equation (4.2-3) reduces to $v = val(a_{ij} + vp_{ij})$. Here we have omitted unnecessary superscripts, so a_{ij} is the immediate reward and

A. Washburn, *Two-Person Zero-Sum Games*, International Series in Operations Research & Management Science 201, DOI 10.1007/978-1-4614-9050-0_9,

p_{ij} is the probability of continuation if the two players choose strategies i and j. Equivalently, letting $b_{ij} = 1 - p_{ij}$ and employing Theorem 3.2-2, we must have

$$val\left(a_{ij} - vb_{ij}\right) = 0 \tag{9.1-2}$$

Solving the Stochastic game thus amounts to finding a number v such that the value of the difference game $(a_{ij} - vb_{ij})$ is 0, after which v is the value of the Stochastic game. If $\mathbf{x}^*$ is an optimal strategy for player 1 in the Stochastic game, then it must be true that

$$\sum_i \left(a_{ij} - vb_{ij}\right)x_i^* \geq 0 \text{ for all } j, \tag{9.1-3}$$

and it follows that $R(\mathbf{x}^*, \mathbf{y}) \geq v$ for all $\mathbf{y}$. Similarly $R(\mathbf{x}, \mathbf{y}^*) \leq v$ for all $\mathbf{x}$ if $\mathbf{y}^*$ is optimal for player 2, so if $(\mathbf{x}^*,\mathbf{y}^*)$ solves the Stochastic game, then it also solves the Ratio game. In the context of Stochastic games, the numerator of the ratio in (9.1-1) is the average payoff per stage, while the denominator is the termination probability. The reciprocal of the termination probability is the average number of stages before termination, so $R(\mathbf{x},\mathbf{y})$ is the product of the payoff per stage and the number of stages.

Even in Ratio games having no relation to Stochastic games, finding a solution is equivalent to solving (9.1-2), which will always have a solution because of the continuity of the $val()$ function (Theorem 3.2-2). Thus, even though $R(\mathbf{x},\mathbf{y})$ is a nonlinear function of $\mathbf{x}$ and $\mathbf{y}$, Ratio games always have saddle points for the same reason that finite matrix games always have mixed strategy solutions.

To solve the Ratio game, first guess that the value is s and solve the difference game with payoff matrix $(a_{ij} - sb_{ij})$. If the value of the difference game is 0, then $v = s$ and you are finished. Otherwise try a larger (smaller) guess if the value of the difference game is positive (negative). Schroeder (1970) offers an iterated linear programming procedure for making a sequence of guesses that converges to the game value v, but no general finite procedure involving only the four fundamental operations of arithmetic can exist for calculating v exactly—the example at the end of this section establishes this because calculation of v requires a square root. A more direct method of finding a solution of the Ratio game would be to solve mathematical program NLP:

$$\begin{aligned}
&\max_{x \geq 0, v} v \\
&\text{subject to} \sum_{i=1}^{m} \left(a_{ij} - vb_{ij}\right)x_i \geq 0 \text{ for all } j \\
&\sum_{i=1}^{m} x_i = 1
\end{aligned}$$

The constraints of NLP require that the average payoff to player 1 exceed the average payoff to player 2 by a factor of at least v, no matter what player 2 does.

NLP is not a linear program because of the product of variables v and x_i that occurs in the constraints.

Equation (9.1-2) is central to von Neumann's (1937) model of an expanding economy. Economies have expansion rates that are usually measured as percentage changes in gross national product. The question here is whether there is any way to predict the expansion rate from fundamental data about production processes.

Consider first a very simple economy where there are no exports, no imports and only one good that we can think of as women. The expansion of that good over a generation depends on the number of female children that the average woman gives birth to over her lifetime, and which survive to an age where they can bear children themselves, an age that we take to be our time period. If that number of children is 1.5, then the population expansion rate is 50 % per time period, or alternatively we can say that the expansion factor is 1.5.

Now generalize that example so that instead of one process (childbirth) and one good (women) there are m processes, n goods, and two fundamental $m \times n$ matrices **a** and **b**:

b_{ij} = amount of good j consumed immediately by process i operated at unit level
a_{ij} = amount of good j produced one time period later by process i operated at unit level

In the simple example we would have $m = n = 1, b_{11} = 1, a_{11} = 1.5$, and it would be obvious that the expansion factor is 1.5. No expansion factor is given in the generalization, but it turns out that it has to be the value v in 9.1-2. Player 1's optimal mixed strategy can be interpreted as the relative levels at which the various processes operate, and player 2's optimal mixed strategy can be interpreted as the normalized prices (weights) of the goods. In NLP, we are requiring that the output of every good must exceed the input by a factor of at least v. Thompson (1989) includes a comprehensive economic interpretation of this result.

Example Suppose $\mathbf{b} = \begin{bmatrix} 2 & 0 & 2 \\ 1 & 1 & 0 \end{bmatrix}$ and $\mathbf{a} = \begin{bmatrix} 3 & 0 & 0 \\ 0 & 1.5 & 1 \end{bmatrix}$. One might think of this as an economy where the two processes are "farming" and "home maintenance", while the three resources are "food", "women", and "men". Men are needed for farming, but not in the home, women are needed at home, but not for farming, and food is needed for both processes. Solving (9.1-2) requires one to guess whether column 2 or column 3 should be active for player 2. It turns out to be column 3. The solution is $\mathbf{x}^* = (0.378, 0.622)$, $\mathbf{y}^* = (0.341, 0, 0.659)$, and $v^* = 0.5\left(\sqrt{7} - 1\right) = 0.823$. The price of women is 0 because the economy, in the process of trying to produce enough men and food to keep the farms going, overproduces women. This economy is doomed because the expansion factor is less than 1. Change a_{11} from 3 to 5 (exercise 1) and the expansion factor becomes 1.158, which is consistent with an expanding economy.

9.2 Racetime Games

This is another class of games that, surprisingly, have saddle points. They are useful in modeling prolonged battles of attrition (Sect. 4.4.2), although that will not be obvious from our abstract description here. Player 1 is given a set X of feasible strategies, each of which is a nonnegative vector $\mathbf{x}$ with m components, and a fixed reward vector $\mathbf{a}$ that also has m components. Player 2 is similarly given a set Y of nonnegative vectors $\mathbf{y}$ with n components, and a reward vector $\mathbf{b}$. The payoff is

$$A(\mathbf{x},\mathbf{y}) \equiv \frac{\sum_{i=1}^{m} x_i a_i + \sum_{j=1}^{n} y_j b_j}{\sum_{i=1}^{m} x_i + \sum_{j=1}^{n} y_j} \tag{9.2-1}$$

We assume that the $\mathbf{0}$ vector is in neither X nor Y, so division by 0 is impossible. Note that $\mathbf{x}$ and $\mathbf{y}$ are pure strategies in this section, not mixed strategies as in Ratio games. There is no restriction on the sums of the components of these vectors.

The game is completely defined mathematically at this point, but it is useful to have an interpretation. Imagine that one of $m + n$ mutually exclusive events will eventually occur, with the payoff to player 1 depending entirely on which event happens first. Payoff vectors $\mathbf{a}$ and $\mathbf{b}$ specify the possible payoffs. The game is played in continuous time until an event occurs. The probability that event i occurs in a small time interval of length d is assumed to be dx_i, so think of x_i as the rate at which event i happens. Player 2's strategy is interpreted similarly, so the total probability that some event happens in the next small time interval is $d\left(\sum_{i=1}^{m} x_i + \sum_{j=1}^{n} y_j\right) \equiv d\lambda$. The total rate at which events happen is λ, the denominator of (9.2-1). The conditional probability that the event that happens is i, given that some event happens, is $(dx_i)/(d\lambda)$, with d cancelling. Likewise the conditional probability that the event is j is y_j/λ. With this interpretation, (9.2-1) simply calculates the average payoff to player 1 per event. In a sense the two players are having a race against time, with each player hoping that one of his events happens first.

The main use of Racetime games is in modeling battles between player 1, with n units on his side, and player 2 with m units, as exhaustive Markov games. The events correspond to the $m + n$ possibilities for someone losing a unit, and the payoffs are conditional probabilities of victory for player 1. Since losing a unit is usually not a good thing, we would expect to have $a_i \geq b_j$ in such a battle, for all i and j. The value of the game is in this case also a probability of winning, and an iterative scheme for solving the game can be based on this observation (Kikuta 1986). Section 4.4.2 includes an example.

We now turn to finding an efficient solution method. Let $a^* \equiv \max_{\mathbf{x}\in X} \sum_i x_i a_i$ and $b^* \equiv \min_{\mathbf{y}\in Y} \sum_j y_j b_j$. If $a^* + b^* = 0$, then the value of the game is 0 because player 1 can guarantee that the numerator is nonnegative by using his maximizing strategy, and player 2 can guarantee that it is nonpositive by using his minimizing strategy. But subtracting v from all $m + n$ payoffs is equivalent to subtracting v from $A(\mathbf{x},\mathbf{y})$, so our object can be to modify the payoffs in that subtractive manner until the value of the modified game is 0. If the value of the modified game is 0, then the value of the original game is v.

Consider the following linear program LP with two variables and two groups of constraints (a) and (b):

$$\begin{aligned} &\min_{v,w} w \\ &\text{subject to (a)} \quad w + v\sum_{i=1}^{m} x_i \geq \sum_{i=1}^{m} x_i a_i; \;\; \mathbf{x}\in X, \\ &\qquad\qquad\;\; \text{(b)} \quad -w + v\sum_{j=1}^{n} y_j \leq \sum_{j=1}^{n} y_j b_j; \;\; \mathbf{y}\in Y. \end{aligned} \tag{9.2-2}$$

At least one constraint in each group must be an equality in any optimal solution, since otherwise v and w could be adjusted to make w smaller. To prove this, first observe that if there is no equality in group (a), then all of the constraints will still hold if w is decreased by some small but positive amount δ. This might violate a constraint in group (b), but let $\mu \equiv \min_{\mathbf{y} \in Y} \sum_j y_j$ and note that μ is by assumption positive. If w is decreased by δ and v is decreased by δ/μ, all of the constraints in both (a) and (b) will still hold, so the (w,v) solution can be improved. This is a contradiction, since the solution was assumed to be optimal. A similar contradiction can be reached if there is no equality in group (b), since in that case w can be decreased while v is increased. Therefore each group of constraints must have at least one equality in an optimal solution. Let $\mathbf{x}^*$ be an equality in group (a), so that $w = \sum_{i=1}^{m} x_i^*(a_i - v) \equiv a^*(v)$. Similarly let $\mathbf{y}^*$ be an equality in group (b), so that $-w = \sum_{j=1}^{n} y_j^*\left(b_j - v\right) \equiv b^*(v)$. Since $a^*(v) + b^*(v) = 0$, either player can guarantee a payoff of 0 in the modified game, and therefore either player can guarantee v in the original, unmodified game. In other words, the solution of LP directly provides a solution of the unmodified game—there is a saddle point with value v at $(\mathbf{x}^*,\mathbf{y}^*)$. Sheet "RaceTime" of the TPZS workbook includes a small example (exercise 2).

9.3 Lopside Games and Signaling Games

Except for our aversion to the name, we might title this section "games of incomplete information". It was pointed out in Chap. 3 that we always assume that both players know the rules when studying TPZS games, but that the restriction does not prevent the examination of the effects of uncertainty. In this section we assume that the uncertainty comes in the form of multiple matrices (called "game elements") $\mathbf{a}^1, \ldots, \mathbf{a}^K$, with uncertainty about which element determines the payoff (the superscript does not denote a power here, but only the reservation of the subscript space for strategies). A probability distribution $\mathbf{p} = (p_1, \ldots, p_K)$ over the game elements is given, with p_k being the probability that the payoff is determined by $\mathbf{a}^k$. This distribution $\mathbf{p}$ is known to both sides, although the operative game element may not be. In Lopside games the players choose their strategies independently, as usual. In Signaling games one player observes the other's strategy choice before making his own.

If Lopside or Signaling games are repeated, one must distinguish whether the same element is shared by all repetitions, or alternatively determined independently for each repetition. In the latter case the number of repetitions is not important, since each repetition can be played independently. In the former case we have multistage versions where the number of repetitions is an important parameter. Here we deal only with the single-stage version.

9.3.1 Lopside Games

There are several possibilities for what the players know about the game element at the crucial moment when they have to choose their strategies. In this subsection we will explore three of them, especially the one for which the subsection is named.

Case 1 If the game element is determined before either player chooses a strategy, then the players end up playing a randomly chosen game in which they both know the payoff matrix. If v_k is the value of game element $\mathbf{a}^k$, then the value of the overall game is simply $pay_1 \equiv \sum_{k=1}^{K} p_k v_k$.

Case 2 If the game element is determined <u>after</u> both players have chosen their strategies, then the overall payoff for strategy i versus strategy j is $a_{ij} = \sum_{k=1}^{K} p_k a_{ij}^k$; that is, we first average all the payoffs before solving the game. In order for this to make sense, all game elements must have the same number of rows and columns; indeed, the rows and columns must mean the same thing in all elements because the players must choose their strategies before the element is known. Let pay_2 be the value of this "average game". Either pay_1 or pay_2 might be the larger number, since the difference between the first two cases does not automatically favor either player.

Case 3 Here one player knows the applicable game element, but the other does not. For lack of a better term call these "Lopside" games, since the rules are clearly lopsided in favoring the player who knows which game element determines the payoff. To be specific, assume that player 1 is the one who does not know which game element applies. Every game element must have the same number of rows because of Player 1's uncertainty, but the number of columns can vary between elements. If there are m rows in every game element, then player 1's mixed strategy $\mathbf{x}$ has m components. If there are n_k columns in the k^{th} game element, n in total, then player 2's mixed strategy is $\mathbf{y} = (y_{jk})$, where y_{jk} is the probability that the game matrix is k and that player 2 chooses that game's column j. Note that variable y_{jk} is the joint probability that the game element is k and the column choice is j. It might be more intuitive to make y_{jk} the conditional probability that the column choice is j given that the game matrix is k, with the sum on j being 1 for every k. However, the joint probabilities are better suited to our current purpose. The joint probability that the game element is k, the column choice is j, and the row choice is i is $x_i y_{jk}$, so the average payoff is $A(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{k} \sum_{i=1}^{m} \sum_{j=1}^{n_k} x_i y_{jk} a_{ij}^k$. The vector $\mathbf{p}$ occurs indirectly in this expression because $\mathbf{y}$ must be chosen so that $\sum_{j=1}^{n_k} y_{jk} = p_k;\ \ k = 1, \ldots, K$. This is an example of a bilinear game as introduced in Sect. 5.5.

To find his optimal strategy, player 1 can solve linear program LP:

$$
\begin{aligned}
&\max_{\mathbf{x} \geq 0, \mathbf{v}} \sum_{k=1}^{K} p_k v_k \\
&\text{subject to } \sum_{i=1}^{m} x_i = 1 \\
&\sum_{j=1}^{n_k} a_{ij}^k x_i - v_k \geq 0;\ \ k = 1, \ldots, K;\ \ j = 1, \ldots, n_k
\end{aligned}
\tag{9.3-1}
$$

As long as the constraints of LP hold, and bearing in mind that $\mathbf{y}$ must also meet its constraints,

$$
A(\mathbf{x}, \mathbf{y}) = \sum_{k=1}^{K} \sum_{j=1}^{n_k} \sum_{i=1}^{m} y_{jk} x_i a_{ij}^k \geq \sum_{k=1}^{K} \sum_{j=1}^{n_k} y_{jk} v_k = \sum_{k=1}^{K} p_k v_k. \tag{9.3-2}
$$

In other words, the objective function of LP is a value that player 1 can guarantee.

To find his optimal strategy, player 2 can solve linear program LP′:

$$
\begin{aligned}
&\min_{\mathbf{y} \geq 0, u} u \\
&\text{subject to } \sum_{j=1}^{n_k} y_{jk} = p_k;\ \ k = 1, \ldots, K \\
&\sum_{k=1}^{K} \sum_{j=1}^{n_k} a_{ij}^k y_{jk} - u \leq 0;\ \ i = 1, \ldots, m.
\end{aligned}
\tag{9.3-3}
$$

As long as the constraints of LP′ hold, $A(\mathbf{x},\mathbf{y}) \leq u$, so the objective function of LP′ is a value that player 2 can guarantee. Since LP and LP′ are duals, the optimized objective functions must be equal to some common value pay_3. The Lopside value pay_3 will be smaller than both pay_1 and pay_2 because player 1 is at a disadvantage in the Lopside game compared to either of the other two cases.

In principle we could also solve a Lopside game as an ordinary matrix game, but each strategy for player 2 would have to specify his column choice in all game elements. There are $\prod_{k=1}^{K} n_k$ such strategies, a number usually far greater than $1 + \sum_{k=1}^{K} n_k$, the number of variables in LP′. In other words, Lopside games are another kind of game where a special solution technique makes it computationally feasible to solve games that would otherwise be problematic.

A generalization of Lopside games would have the game element be doubly superscripted, with player 1 knowing one superscript and player 2 the other. Ponssard (1980) shows how to solve all such finite games using linear programming.

Lopside example Suppose $\mathbf{p} = (0.5, 0.5)$, $\mathbf{a}^1 = \begin{bmatrix} 2 & 0 \\ 4 & 8 \end{bmatrix}$, and $\mathbf{a}^2 = \begin{bmatrix} 8 & 4 \\ 0 & 2 \end{bmatrix}$. The value of each game element is 4 (each has a saddle point), so the average of the game values is $pay_1 = 4$. The average of the two payoff matrices is $\begin{bmatrix} 5 & 2 \\ 2 & 5 \end{bmatrix}$, and this average game has value $pay_2 = 3.5$. The value of the Lopside game can be found by solving a small linear program, but the solution is almost obvious. Player 1 should use $\mathbf{x}^* = (0.5, 0.5)$. If he does so, the average payoff will be at least 3 regardless of the payoff matrix. Player 2 can guarantee the same value by always using column 1 in game 1 or column 2 in game 2, which corresponds to $\left(y_{jk}{}^*\right) = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$. Therefore $pay_3 = 3$.

A second Lopside example can be found on sheet "Lopside" of the TPZS workbook. That example has two game elements, one 3×2 and the other 3×4. The sheet offers the opportunity to adjust the data and see the effect on the solution.

One realistic situation where Lopside games arise is when player 1 is defending infrastructure against attacks by multiple uncoordinated enemies such as terrorist groups. Payoff matrix $\mathbf{a}^k$ is for group k, which may be differently financed or located compared to other groups and therefore have different strategies available. The entries in each game element are the costs of paying for damages and repairing the infrastructure after an attack, or rather the negations of those costs because player 1 is the maximizer. The probabilities p_k might be estimated by observing the historic rate at which group k has conducted attacks, call it λ_k. The total rate of attacks would then be $\lambda \equiv \sum_k \lambda_k$, and the probability that the next attack will be of type k would be $p_k = \lambda_k/\lambda$. Even though player 1 does not know which group will conduct the next attack, he must know the probabilities if the situation is to be treated as a TPZS game. Even though the various terrorist groups do not coordinate attacks (the groups are implicitly assumed to do their planning independently), each must understand that the existence of the others influences player 1's tactics. Pita et al. (2008) describe such an application to security patrols at Los Angeles International Airport.

9.3.2 Signaling Games

Here we continue to assume that player 2 knows the game element, while player 1 does not, but now player 1 has the compensating advantage of seeing player 2's strategy choice before making his own. Player 2 is conflicted upon learning the game element, since his column choice carries information about which game element is being played, and this signal might be of use to player 1. In the previous Lopside example player 2 might be attracted to the first column of the first game or the second column of the second, always choosing the column with the smallest sum. However, always choosing that column is now dangerous because player 1 can adopt the strategy of choosing row 2 when player 2 chooses column 1 or row 1 when player 2 chooses column 2. The payoff would then consistently be 4. Player 2 can do better, as we shall see. Except for the reversal of roles, Basic Poker (Sect. 3.4) is another example of a signaling game.

For a general analysis, first note that all game elements must have identical row names for the same reason as in Lopside games. The game elements can have different numbers of columns, as in Lopside games, but it is no longer just a matter of counting columns because player 1 can observe the column choice. Let there be n columns in total, considering all columns in all game elements, and let S_j be the nonempty set of game elements in which strategy j is feasible for player 2. Let y_{jk} be player 2's probability that the game element is k and the column choice is j, and let x_{ij} be the probability that player 1 chooses row i, given that the observed column is j. Player 1's variables are subject to the constraint that $\sum_{i=1}^{m} x_{ij} = 1;\ \ j = 1, \ldots, n$, while player 2's are subject to the same constraints as in Lopside games. The joint probability that the game element is k, the column choice is j, and the row choice is i is now $x_{ij}y_{jk}$. Therefore the average payoff to player 1 is

$$A(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{m} \sum_{j=1}^{n} \sum_{k \in S_j} a_{ij}^k x_{ij} y_{jk} = \sum_{i=1}^{m} \sum_{j=1}^{n} x_{ij} \sum_{k \in S_j} a_{ij}^k y_{jk}. \tag{9.3-4}$$

Now consider linear program LP′ for player 2:

$$\begin{aligned}
&\min_{\mathbf{y}\geq 0,\,\mathbf{u}} \sum_{j=1}^{n} u_j \\
&\text{subject to} \sum_{j \mid k \in S_j} y_{jk} = p_k; \quad k = 1, \ldots, K \\
&\sum_{k \in S_j} a_{ij}^{k} y_{jk} - u_j \leq 0; \quad i = 1, \ldots, m \text{ and } j = 1, \ldots, n
\end{aligned} \tag{9.3-5}$$

As long as the last set of constraints holds, we have

$$A(\mathbf{x},\mathbf{y}) = \sum_{i=1}^{m}\sum_{j=1}^{n} x_{ij} \sum_{k \in S_j} a_{ij}^{k} y_{jk} \leq \sum_{j=1}^{n}\sum_{i=1}^{m} x_{ij} u_j = \sum_{j=1}^{n} u_j, \tag{9.3-6}$$

where the last equality is because the sum on i of x_{ij} is always 1. The objective function of LP′ is therefore an upper bound on the average payoff that player 2 can enforce.

The dual of LP′ is LP, a linear program for player 1:

$$\begin{aligned}
&\max_{\mathbf{x}\geq 0,\,\mathbf{v}} \sum_{k=1}^{K} p_k v_k \\
&\text{subject to} \sum_{i=1}^{m} x_{ij} = 1; \quad j = 1, \ldots, n \\
&\sum_{i=1}^{m} a_{ij}^{k} x_{ij} - v_k \geq 0; \quad k \in S_j, \quad j = 1, \ldots, n
\end{aligned} \tag{9.3-7}$$

It can be shown that the objective function of LP provides a lower bound on the average payoff that player 1 can enforce (exercise 5). Since LP and LP′ are duals, we have solved the Signaling game. Its value is the optimized objective function of either LP or LP′.

Signaling example Using the same data as in the Lopside example, we can solve this subsection's LP′ to find that player 2 can do better than an average value of 4. By using $\mathbf{y} = \left(y_{jk}\right) = \begin{bmatrix} 0.4 & 0.1 \\ 0.1 & 0.4 \end{bmatrix}$, he can guarantee that the average payoff will not exceed 3.2. Player 1 can guarantee the same thing by using $\mathbf{x} = \left(x_{ij}\right) = \begin{bmatrix} 0.4 & 0.6 \\ 0.6 & 0.4 \end{bmatrix}$. The Signaling game value is of course larger than 3, the value of the corresponding Lopside game.

Sheet "Signal" of the TPZS workbook includes a somewhat larger example (exercise 6).

Multistage Signaling games with $K = 2$ sometimes occur in a protracted war when player 2 may or may not have broken the communications code used by player 1. If player 2 has indeed broken the code, he may be in a favorable position for defeating any given attack by player 1, but must carefully choose his defenses so that he doesn't reveal that the code has been broken. An extreme example of this would be the purported British decision not to defend Coventry against a German bombing raid in World War II for fear of giving away the fact that the German Enigma code had been broken (Winterbotham 1974). Whether that assertion is true or not, there is definitely a multistage Signaling game going on when the breaking of secret codes is an issue. Any user of a secret code must constantly be examining enemy tactics for clues as to whether the code has been broken. Owen (1995) discusses multistage Signaling game solution methods, or see Kohlberg (1975).

9.4 The Bomber Versus Battleship Game

Figure 9.4-1 is a World War II picture of an attack by B-17's from high altitude on the Japanese aircraft carrier Hiryu, taken at the beginning of the Battle of Midway. It should be evident that the undamaged Hiryu is threading her way through a lot of bomb impacts. In spite of all the bombs dropped, Hiryu was never damaged by B-17's in that battle. A fundamental reason for the lack of damage is that it takes the better part of a minute for a bomb to fall from high altitude, and in that time the bomb's target can move a distance that exceeds its own dimensions. Even if the aim of the Bombers is perfect, they can never be sure where to aim because of the time delay between dropping and impact. The Bomber-versus-Battleship (BvB) game is an abstraction where the aim of the Bombers is perfect, but there is a time delay

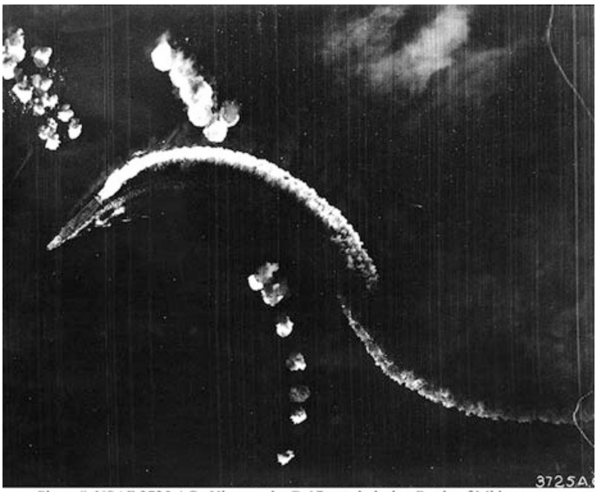


Photo # USAF 3725 AC Hiryu under B-17 attack during Battle of Midway

Fig. 9.4-1 The splashes are bomb impacts. A sharp turn on Hiryu's part can be seen from the wake discontinuity before impact. Hiryu was later sunk by Navy dive bombers, by the way, a mode of attack that minimizes the time delay at the cost of first exposing the bombers to defensive fire.

Rufus Isaacs and others confronted the BvB game at RAND in the years following World War Two. The game as pictured in Fig. 9.4-1 involves two physical dimensions and Hiryu's ship kinematics, so Isaacs' approach was to first construct a simpler model in one dimension where the two players are Ship and Bomber. Ship's only choice in each time interval is to move one step right or left. Bomber's single bomb will kill Ship if and only if it hits Ship's position, and Bomber has to forecast Ship's position n steps ahead. Ship does not know when the bomb is dropped and cannot see it falling. Both Bomber and Ship are unlimited in endurance.

BvB is not a Markov game because there is no way to define "game element" in such a manner that Ship's fate is not influenced by Bomber's decisions in past elements, as well as the current one. The sole exception to this is when $n = 1$, in which case there is a single element that is repeated until Bomber finally drops the bomb. When $n > 1$, we could make the game Markov by changing the rules so that Ship knows when the bomb is dropped, but that is not the rule in BvB. We therefore have no theorem that guarantees that the value of BvB even exists.

When $n = 1$, the value of BvB is 0.5. Ship can guarantee this by flipping a coin to go right or left, and Bomber can guarantee it by at any time dropping the bomb right or left of Ship's current position. This solution is so easily achieved that considerable generalization is possible as long as $n = 1$—Garnaev (2000) in his "Infiltration Game with Bunker" gives Bomber several imperfect bombs, introduces a special position where Ship is safe from bombs, and is still able to find an analytic solution.

The case $n = 2$ turns out to be qualitatively different. Here is a quote from Isaacs (1955):

> *Our intention was now to take up* $n = 2, 3, 4, \ldots$, *and, from the knowledge gained, proceed to the continuous case. Thence we hoped to restore planarity to the ocean and approach practicality by more realistic assumptions about the ship's kinematics, accuracy of the Bomber, number of bombs, etc.*

However, the case $n = 2$ proved to be unexpectedly difficult.

When $n = 2$, there are only three places where Ship can be after two steps, so Bomber can guarantee a hit probability of 1/3 at any time by choosing one of them at random. However, Ship can only guarantee 1/2 by flipping coins because that is the probability that two successive moves will cancel each other and leave Ship unmoved. We therefore know from these elementary considerations only that the game value is somewhere between 1/3 and 1/2. The difficulty is in finding exactly where the value of the game lies in that interval.

The optimal strategy for Ship turned out to be easier to find than the optimal strategy for Bomber. Ship should have a constant probability of turning, which requires Ship to have a one move memory for the past. Let X be the net change of position, in the same direction as the previous step, after two more steps with a

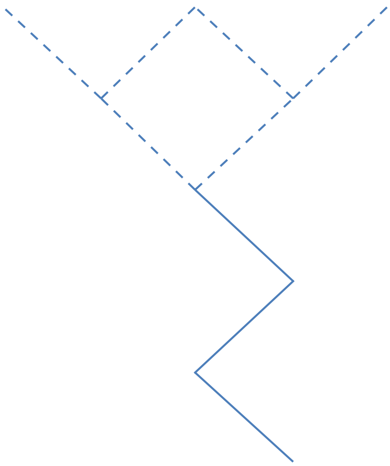


Fig. 9.4-2 The bomb is dropped when the solid track ends. The dashed lines show future possibilities in the 2-step game

constant turn probability x (see Fig. 9.4-2). The three possibilities have probabilities:

$$\begin{aligned} &P(X=-2)=x(1-x)\\ &P(X=0)=x^2+(1-x)x=x\\ &P(X=+2)=(1-x)^2. \end{aligned}$$

Therefore Ship can guarantee that the hit probability will not exceed

$$v=\max_{0\le x\le 1}\left(x(1-x),x,(1-x)^2\right)=\left(3-\sqrt{5}\right)/2=0.382,$$

with x^* also being 0.382. This value solves the equation $x=(1-x)^2$.

It took a long time to prove that v is actually the value of the game, the problem being that, while Bomber can guarantee $v-\varepsilon$ for any positive ε, he cannot quite guarantee v. This should make intuitive sense, since there is always something to be gained by studying Ship's movements a little longer before acting when your endurance is unlimited, but of course one cannot wait forever. The near-optimal Bomber strategy is complex, requiring long memory for past Ship movements. Ferguson (1967) includes a description of that strategy.

The 2-step game is disappointing in the complexity of Bomber's optimal strategy, but it does have the potentially useful feature of requiring only a finite memory on Ship's part. Perhaps the n-step game requires Ship to remember only the previous $n-1$ steps in deciding whether to turn or not. This "Markov" hypothesis leads to an optimal solution of the 2-step game, and to an upper bound of 0.294 on the value of the 3-step game. However, Bram (1973) shows that the value of the 3-step game is somewhere in the interval [0.28423, 0.28903], so the Markov hypothesis is false. Lee and Lee (1990) improve the upper bound to 0.28837, but only by using a mixed strategy for Ship that employs some 400 distinct conditional probabilities. The exact solution of the 3-step game is still unknown.

The complexity of the 3-step BvB game has led to the consideration of other, similar games that are easier to solve. Ferguson (1967) considers a class of 2-step games with different rules of motion for Ship, including motion in two dimensions, and succeeds in finding solutions. The phenomenon that Ship needs to remember only its previous position persists.

Matula (1966) considers emission-prediction (EP) games where Emitter produces a binary sequence that is closely watched by Predictor, who at some point marks the sequence. Predictor wins if and only if the digits following the mark are not exactly *s*, where *s* is some finite sequence known to both sides. EP games are simpler than BvB games because only one sequence *s* is involved, rather than several. If *s* is 11, Emitter can win all the time by emitting nothing but 1s. However, Emitter cannot win all the time if *s* is 10—if he simply emits a sequence where 0s and 1s alternate, Predictor will notice the pattern and win by marking the sequence right after a 1 is emitted. The solution of EP1100, the game where *s* is 1100, is particularly interesting. The optimal strategy for Emitter involves a Markov chain with four states, call them *a*, *b*, *c*, and *d*. Emitter first emits a sequence of letters, and then replaces letters by digits using the rule that *a* and *b* map to 1, while *c* and *d* map to 0. For example aabaccbda… would map to 111100101… The interesting feature is that, although the letters are a stationary Markov chain, the digits are not!

Matula's solution of EP1100 opens up the possibility that the Markov hypothesis for the 3-step BvB game simply needs to be modified to permit memory for something other than binary digits. However, nobody as yet has figured that out. Half a century later, the simplicity that Isaacs hoped for when making the quote above still awaits discovery.

The difficulty of BvB and related games is unfortunate, since military situations involving a time delay have not gone away. It is true that a modernized Hiryu would not survive an attack by modernized B-17s, since the bombers would employ missiles that could distinguish an aircraft carrier from its background and home on it. However, many modern weapons are not of the homing type, and arrive at their intended detonation points only after a time delay. The targets of such weapons will always find it advantageous keep moving.

9.5 Exercises

1. The economic example in Sect. 9.1 contains a claim that the value of a modified game is 1.158. Verify it, and also find the optimal strategy for player 1 in the modified game.

2. Sheet "RaceTime" of the TPZS workbook includes a small example of a Racetime game whose value is 1.5. Convince yourself that the problem being solved by Solver corresponds to the LP formulation in Sect. 9.2. If player 2's first vector is changed from (1,4,3) to (1,5,3), how does the value of the game change, and does either player's strategy change?

3. Sheet "Lopside" of the TPZS workbook includes a small example of a Lopside game whose value is 1.65. A Lopside game should reduce to an ordinary matrix game if the probability vector **p** has a 1 in it. Test whether Solver gets the correct Lopside solution by changing **p** to include only one of the game elements. Use Solver to solve the modified Lopside game, and then solve the ordinary matrix game based on that single element. Are the solutions identical?

4. Copy sheet "Lopside" to a new sheet of the TPZS workbook, delete all the formulas needed to formulate player 1's linear program LP, insert the formulas needed for player 2's linear program LP′, and modify Solver to solve player 2's problem. You should find that Solver gets the same game value as on the "Lopside" sheet. Does it?

5. In Sect. 9.3.2 it is claimed that the objective function of LP is a lower bound on the average value of the Signaling game that player 1 can enforce. Prove it.

6. Sheet "Signal" of the TPZS workbook is set up to solve a Signaling game with two game elements, one 3×2 and one 3×4, using Solver. The example in Sect. 9.3.2 also has two game elements, but they are both 2×2. Change the inputs on sheet Signal to solve the example of Sect. 9.3.2. You will have to input the excess data in a manner that makes the excess rows and columns unattractive to the players.

7. Consider a Signaling game where the columns are all uniquely labeled, so that player 2's column choice reveals exactly which game element is operative. Describe how to find the solution of such a game.

8. Try solving exercise 1 by using NLP of Sect. 9.1. Copy both 2×3 matrices to a new sheet of the TPZS workbook and set up Solver to find a solution using Solver's "GRG Nonlinear" method (that will be the name if you are using Excel™ 2010, at least). Does Solver get the right solution?

Appendix A: Math Programming

In spite of the name, a mathematical program is a concise statement of a mathematical optimization problem, not a computer program. It specifies a vector of real variables $\mathbf{u}$, together with a scalar objective function $f(\mathbf{u})$ that is to be optimized (either maximized or minimized), and also possibly some constraints on $\mathbf{u}$. The most compact statement of the constraints would be to say $\mathbf{u} \in U$; that is, only members of the set U are eligible. If the problem is to maximize $f(\mathbf{u})$, the notation that we will use for specifying the associated mathematical program is

$$\begin{aligned}&\max_{\mathbf{u}} f(\mathbf{u})\\ &\text{subject to } \mathbf{u} \in U\end{aligned}$$

Note that the variables are named in the subscript on max or min; if a symbol is not named there, then it is not a variable. If the set U is empty, the program is said to be "infeasible". If the objective function can be made arbitrarily large, the program is said to be "unbounded". An example of an unbounded problem is to maximize u subject to $u \geq 0$. In practice most mathematical programs are feasible and bounded.

For brevity we will also permit two elaborations of the subscript. One is to state the constraining set in the subscript, so that the above program is simply $\max_{\mathbf{u} \in U} f(\mathbf{u})$. The other is to state that all components of $\mathbf{u}$ must be nonnegative. Consider the following example

$$\begin{aligned}&\min_{v, \mathbf{u} \geq 0} v^2 + u_1 + w{u_2}^4\\ &\text{subject to } u_1 + \sin(v) \leq 5\\ &u_1 + v = -3\end{aligned}$$

There are three variables in this mathematical program, the two components of $\mathbf{u}$ and the scalar v. The symbol w is not a variable, since it is not among the subscripts, so its value must somehow be determined before the problem is solved. The subscript on min requires both components of $\mathbf{u}$ to be nonnegative. The first explicit constraint is that a particular function of the variables cannot exceed 5,

A. Washburn, *Two-Person Zero-Sum Games*, International Series in Operations Research & Management Science 201, DOI 10.1007/978-1-4614-9050-0,

and the second is that a different function must equal −3. The relationship "≥" is permitted in constraints in addition to "≤" and "=", but only those three. Mathematical programming packages are prepared to find the solutions of such problems, and some of them are also prepared to accept constraints to the effect that certain variables must be integer-valued.

To distinguish variables that are optimal, we will simply affix an asterisk to the variable name. Thus, if the mathematical program is $\min_x x^2$, we say $x^* = 0$ because the quadratic function is minimized when x is set to 0.

The most important special case of a mathematical program is a linear program, since software exists that will solve large instances of linear programs quickly and reliably. A linear program is the problem of optimizing a linear function subject to constraints that are also linear. An example is LP1 of Chap. 3, which finds the optimal mixed strategy $\mathbf{x}$ for player 1 in a game whose payoff matrix is $\mathbf{a} = (a_{ij})$:

$$
\begin{aligned}
&\max_{v,\mathbf{x}\geq 0} v \\
&\text{subject to} \sum_{i=1}^{m} x_i a_{ij} - v \geq 0;\;\; j = 1, \ldots, n \\
&\sum_{i=1}^{m} x_i = 1
\end{aligned}
$$

LP1 has $m + 1$ variables, only one of which (v, the game value) is involved in the objective function. In stating LP1 we have observed the convention that the right-hand-side of every constraint is a constant, and that there are no constant terms on the left-hand-side. Linear program LP2 of Chap. 2 similarly expresses player 2's problem of minimizing the payoff, instead of maximizing it, using a linear program with $n + 1$ variables and $m + 1$ constraints. LP1 and LP2 are duals of each other.

Every linear program has a dual linear program with the opposite objective. If the original "primal" program LP is:

$$
\begin{aligned}
&\max_{\mathbf{x}\geq 0} \sum_{i=1}^{m} c_i x_i \\
&\text{subject to} \sum_{i=1}^{m} a_{ij} x_i \leq b_j;\;\; j = 1, \ldots, n,
\end{aligned}
$$

then the dual linear program is LP′:

$$
\begin{aligned}
&\min_{\mathbf{y}\geq 0} \sum_{j=1}^{n} b_j y_j \\
&\text{subject to} \sum_{j=1}^{n} a_{ij} y_j \geq c_i;\;\; i = 1, \ldots, m.
\end{aligned}
$$

LP is stated in "canonical form",[1] meaning that it is a maximization with nonnegative variables and upperbounding constraints, but any linear program can be put into canonical form. Consider LP1, for example. LP1 has an unconstrained variable v, but v can be expressed as $v_1 - v_2$, the difference of two nonnegative variables. LP1 also has an equality constraint, but any constraint of the form $z = b$ is equivalent to two upperbounding constraints $z \leq b$ and $(-1)z \leq -b$. Thus LP1 could be put into canonical form using $m + 2$ variables and $n + 2$ constraints.

The two important facts about duality are

1. $(\mathrm{LP}')'$ is LP; that is, the dual of the dual is the primal
2. If LP and LP′ have solutions $\mathbf{x}^*$ and $\mathbf{y}^*$, then $\sum_{i=1}^{m} c_i x_i^* = \sum_{j=1}^{n} b_j y_j^*$; that is, the dual and the primal both have the same optimized objective.

The first fact is not difficult to prove. If the reader wishes to try his hand at it, remember that any minimization problem can be turned into a maximization problem by changing the sign of the objective function, so even LP′ can be put into canonical form. The second fact implies Theorem 3.2-1, the fundamental theorem that underlies most of the results in TPZS game theory.

Many linear programming packages are available for download, or the reader may have access to one already. Two possible package sources are Solver in Microsoft's Excel ™ and the linprog() function in MATLAB™. These packages return dual variables that have a useful sensitivity interpretation, as well as primal variables. INFORMS periodically conducts a survey of available linear programming packages (Fourer 2013).

To solve LP1 using Solver, input the matrix **a** in rows and columns, designate $m + 1$ cells for the variables, do the math to form the required linear expressions, start Solver and tell it about the variables (Solver's term is "adjustable cells") and constraints, and then maximize. The constraints should say that all components of **x** must be nonnegative, but not (since games are allowed to have negative values) v. Solver will adjust the variables to be maximizing. If you prefer, solve LP2 instead, as in Sheet "MorraLP" of the TPZS workbook. The other player's optimal strategy is available through the dual variables (Solver's term is "shadow prices") on the "sensitivity report", a new sheet that Solver offers after solving the problem.

In MATLAB, the following LP2-based script will take an input matrix **A** (note upper case, and that MATLAB is case sensitive) and produce the game value and optimal strategies for both players. The script uses the linprog() function that is available (as of this writing in 2013) in the Optimization toolbox. The script can be copied from sheet "MATLAB" in the TPZS workbook.

[1] Linear programming computer packages do not require that inputs be expressed in canonical form, but use of the form makes it easy to express the dual.

```
% gamesolver.mat
% Solves TPZS (2-person, zero sum) games using LINPROG command
% Payoff matrix must be called A

[m,n] = size(A);
% Create the various inputs for the LINPROG function
B = [P −1*ones(m,1)];       % creates the B matrix
b = zeros(m,1);             % creates the goal
f = [zeros(n,1); 1];        % want the value of the game to be minimized
Beq = [ones(1,n) 0];        % Beq and beq set up criterion that the sum of
the solution probabilities = 1
beq = 1;                    %
lb = [zeros(n,1); -Inf];    % lower bound on solution
% Using the LINPROG command to solve this
[y,fval,exitflag,output,lambda] = linprog(f,B,b,Beq,beq,lb);
% Output
v = fval;
y = y(1:n);
x = lambda.ineqlin;
% Formatted, output

Answer = char('For the payoff matrix ', int2str(P),'','The value of the
game is ', num2str(v),'', 'Player I''s strategy is', num2str(x),'',
'Player II''s strategy is', num2str(y'))
```

As mentioned above, solving LP1 or LP2 is a matter of convenience, since the optimal strategy for the other player is available through dual variables. The call to linprog() in the above script puts the dual variables in the output array lambda, from which player 1's optimal mixed strategy **x** is retrieved. Different LP packages have different conventions about whether variables are assumed by default to be non-negative. They are not so assumed in MATLAB, so the statement defining array **lb** in the above script states explicitly that the first n variables (the first n components of **y**) must not be negative because they all have a lower bound of 0, whereas the last variable (v is named y_{n+1}) is effectively unconstrained.

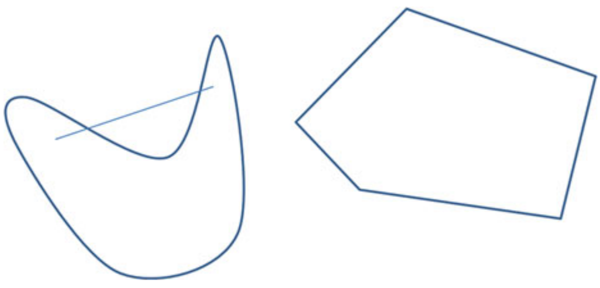
Appendix B: Convexity

Convexity is a property of both functions and sets, and plays a heavy role in game theory. This appendix is an introduction to terms and fundamental theorems.

B.1 Convex Sets and Convex Hulls

Figure B.1 shows two subsets of 2-dimensional Euclidean space. Set A on the left is bounded, but not convex because the illustrated line segment does not lie entirely within A. The definition of a convex set in any number of dimensions is that every line segment connecting two points of the set must lie entirely within the set, and A fails that test. Set B on the right is convex, as would be a circle. Set B is a convex polygon with five extreme points (corners).

Fig. B.1 A set A that is not convex on the left and a convex polygon B on the right



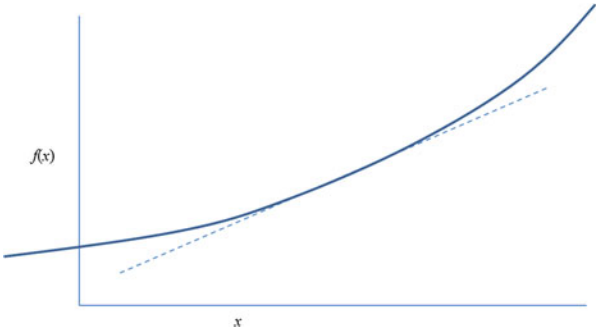
Every bounded set such as A has a convex hull that is by definition the intersection of all convex sets that contain it, or more simply the smallest convex set that contains it. In two dimensions, put a fence on the edge of A, surround the fence with a rubber band and let go. The rubber band will settle at the convex hull. A convex set is its own convex hull, so the convex hull of B is already pictured. Set B is the convex hull of just five extreme points, so the extreme points themselves are a compact way of describing it.

A. Washburn, *Two-Person Zero-Sum Games*, International Series in Operations Research & Management Science 201, DOI 10.1007/978-1-4614-9050-0,

B.2 Convex Functions and Jensen's Inequality

Figure B.2 is a graph of a convex function, together with a tangent line. The function lies entirely above the tangent line no matter where the tangent is drawn, a property that characterizes convex functions.

Fig. B.2 A convex function (solid) lying above one of its tangents (dashed)



More generally, suppose that $f(x)$ maps some convex set S in n-dimensional Euclidean space onto the real line. We say that the function is convex if and only if, at every point $y \in S$, there exists a gradient vector **s** such that

$$f(\mathbf{x}) \geq f(\mathbf{y}) + \sum_{i=1}^{n} s_i(x_i - y_i) \ \text{ for all } \ x \in S$$

The linear expression on the right-hand-side is a hyperplane that is tangent to the function at point **y**. The gradient **s** of that hyperplane need not be unique. For example $|x|$ is a convex function on the whole real line even though it is not differentiable at 0. At point 0 the function lies above many different tangent lines.

A function is concave if its negation is convex. Linear functions are both concave and convex.

One of the important theorems about convex functions is Jensen's inequality, which states that, if $f(x)$ is a convex function defined on convex set S and if $\mathbf{X}$ is a vector of random variables that takes values in S, then $E(f(\mathbf{X})) \geq f(E(\mathbf{X}))$. To prove this, let $\mathbf{y} \equiv E(\mathbf{X})$. Point **y** is in S because S is convex, so there is a gradient **s** such that always

$$f(\mathbf{X}) \geq f(\mathbf{y}) + \sum_{i=1}^{n} s_i(X_i - y_i)$$

Upon taking expected values of both sides, the sum on the right vanishes and we are left with Jensen's inequality.

The difference between the expected value of a function and the function of the expected value is actually intuitive, and is even the source of a certain kind of perverted humor. For example, if half of your tennis balls go too far and the other half go into the net, you opponent might say "You're doing great, on the average", thinking himself clever. Even though all your shots are losers, a shot with the average length, if only you could hit one, would be a winner.

Appendix C: Exercise Solutions

Chapter 1

There are no exercises in chapter 1.

Chapter 2

1. $v_1 = 3$ and $v_2 = 4$
2. S should first include $\{1, 2\}$, then $\{3, 4, 5\}$ in response to player 1's choice of row 2, and then any subset of two columns in response to player 1's choice of row 3. Since $\{1, 2\}$ is among those subsets and already in S, the final S includes only two subsets.

Chapter 3

1.

	x*	**y***	v
a	(0, 0.5, 0.5)	(0, 0, 0.75, 0.25)	3.5
b	(0.25, 0.75)	(0.5, 0.5)	2.5
c	(1/3, 2/3)	(1/6, 5/6, 0)	4/3
d	(0, 1 ,0)	(0, 1, 0)	2
e	(0, 0, 1, 0)	(1, 0, 0)	1
f	(0, 9/16, 0, 0, 7/16)	(0.5, 0.5, 0)	4.5
g	(1/3, 2/3)	(1/3, 0, 2/3, 0)	8/3
h	(0, 1/4, 0, 0, 0, 3/4)	(0, 7/8, 0, 1/8)	15/4
i	(0.5, 0.5, 0)	(0, 0, 0, 0.5, 0.5)	2.5

A. Washburn, *Two-Person Zero-Sum Games*, International Series in Operations Research & Management Science 201, DOI 10.1007/978-1-4614-9050-0,

6. maximize $qx_1{}^*y_1{}^*$ subject to $v = 0$ to find that the maximum is $1.5 - \sqrt{2} = 0.0858$, slightly larger than 1/12, the probability of a called bluff in Basic Poker.
8. Each side has four strategies in the normal form.
10. a 4 × 5 matrix with value 0.5
11. The modification of 1c is: $\begin{bmatrix} 4 & 0 & 2 \\ -2 & 0 & 1 \end{bmatrix}$

    For example, the third column is the strategy "choose column 3 of the original game if there is no leak, else choose the best response to player 1's choice".
12b. $\mathbf{y}^* = (0.5, 0, 0.5)$ is also optimal
13. 18 rows and 2 columns
14. The payoff matrix is $\begin{bmatrix} 7 & 0 \\ 3 & 4 \end{bmatrix}$, with the first strategy for both players being "up". The solution is $\mathbf{x}^* = (1/8, 7/8)$, $\mathbf{y}^* = (1/2, 1/2)$, and $v = 3.5$. I say go with the mixed strategy, but Aumann and Machsler (1972) would like a chance to talk you out of it.
15. The solution of the game is that $v = \frac{6}{\pi^2}$, and that $x_j = y_j = v/j^2$ for all positive integers j.
16. Correct to two decimal places, you should find that $v = 2.57$, $\mathbf{x^*} = (0.43, 0.57)$, and $\mathbf{y^*} = (0.86, 0, 0.14)$
17. (a) The upper bound is 5/9 and the lower bound is −5/9.

    (b) $\mathbf{x}^* = \mathbf{y}^* = (0, 0, 0.433, 0, 0.324, 0, 0.243, 0, 0)$, and $v = 0$
18. (a) $\mathbf{x}^* = \mathbf{y}^* = (10/66, 26/66, 13/66, 16/66, 1/66)$

    (b) $v = -0.28$, so player 2 is favored
20. $v = 1.8$. (1/3, 1/3, 1/3) is not optimal for player 1.
21. The payoff matrix will depend on the order in which you list player 2's strategies, but the game value should not. One version of the matrix is $\begin{bmatrix} 0 & 0 & 3 & 9 & 5 & 7 \\ 3 & 9 & 0 & 0 & 8 & 4 \\ 7 & 5 & 8 & 4 & 0 & 0 \end{bmatrix}$, where the last column corresponds to searching cells in the order CBA. There is no dominance. The value of the game is 7/2. Player 2 is equally likely to use ABC or CBA, the first and last columns. Player 1's optimal probabilities are (14,7,15)/36.
23. $\mathbf{x}^* = \mathbf{y}^* = (1,1,1,1,1,0,0,\ldots)/5$, $v^* = 1/5$

Chapter 4

2. You should find $\mathbf{x}^* = (0.238, 0.191, 0.190, 0.381)$, $\mathbf{y}^* = (0.619, 0.238, 0.095, 0.048)$, $v = 0.381$
3. $v(2,1,2,1) = 0.5$, with player 1 entering a woman with probability 0.5 and player 2 entering a mouse with probability 0.75.

5. In part a, the answer is the same as whatever the value of 4 & 10 is when punts are short, which is about 67 yards. In part b, the answer is that more additional yards can be gained from 2 & 2 than from 1 & 10.
7. $v(3) = (0.200, 0.333, 0.600)$
8. $u_1 = 0$, $u_2 = -0.5$, $g = v_1 = v_2 = 0.5$.
9. (a) The last row is (2, 2, 2) and the last column is (1.5, 2, 2, 2).
   (b) The first row is (2, 3.625, 3.75) and the first column is (2, 0.75, 1, 2), so there is a saddle point where $v(3, 2, 3) = 2$ and both sides use pure CA.
   (c) $\lim_{n\to\infty} V(2,1,n) = 3$.
10. Player 1 has a strategy that will always trap the nickel, but it is hard to find and very unforgiving of mistakes. As a result, actual play of the game often leads to the incorrect conclusion that an expert nickel is impossible to trap.
11. You should find that $v(2, 7) = v(7, -2) = v(0, -5) = 11$, with all other (finite) values being smaller.
12. There are many solutions of (4.2-8). $(u_1, u_2) = (3, 4)$ is the smallest.
13. If the submarine survives the current encounter, the payoff for all future encounters is vq, so $v = \min_r \max\{P_1(r) + vq, P_2(r)(1 + vq)\}$. The two expressions within { } must be equal, so there are two equations to solve for the two unknowns r and v.
15. $\mathbf{x}^* = (k/n, 1 - k/n)$ and $\mathbf{y}^* = (1/n, 1 - 1/n)$ in game element G_{nk}.
17c. Follow the NimSum strategy with two exceptions. If there is only one pile remaining, with that pile having at least two stones, remove all stones but one. If there are two piles remaining, with one of them having a single stone, remove the other pile.
18. Player 2 can win by always leaving player 1 with a sum that is a multiple of 11. If player 1 adds x, player 2 adds $11 - x$.

Chapter 5

2. By equating $(d/dx)A(x,y)$ and $(d/dy)A(x,y)$ to 0 and solving the resulting pair of equations, one obtains $y^* = b/(b + 2a)$, $x^* = (y^*)^2$, and $v = A(x^*,y^*) = -ax^*$. Since $x^* < y^*$, the assumption that the saddle point occurs in the region $x < y$ is confirmed.
3. The curve should go through the points $(v, c) = (1, 3)$ and $(2, 0.5)$.
4. $\mathbf{x}^* = (15/26, 10/26, 1/26)$.
10. The optimal CDF for player 1 is 0 for $x \leq 1/3$. For $x > 1/3$ it has derivative (probability density) $F'(x) = 1/(4x)^3$.
11. $\mathbf{y}^* = (0, 1, 2)$
12. $v = -0.2$
14. $v = 0.25$. Player 1's strategy is mixed, but player 2's is not.
15. $\mathbf{y}^* = (0, 0.5, 0.75, 0.9)$, and $c = 2.15$.
16a. $\mathbf{x}^* = \mathbf{y}^* = (0.358, 0.642)$, $v = 0.328$.
17. In the revised game $\mathbf{x}^* = (0, 16/21, 5/21)$, $\mathbf{y}^* = (6/21, 15/21)$, and $v = 0.305$.

18. (a) no (b) 0.2 (c) 0.4 (d) The value of the game is 1/3.

    The edge effect is central to the game. In the equivalent game played on a circle (imagine connecting the 0 and 1 ends of the unit interval), the optimal mixed strategies are both uniform.
20. Player 2 makes $y = \pi$, in which case player 1 sees a payoff of $-\cos(x)$ and can do no better than make $x = \pm 0.2$. Player 1 flips a coin between 0.2 and -0.2, in which case player 2 sees $0.5(\cos(y - 0.2) + \cos(y + 0.2)) = \cos(y)\cos(0.2)$ and can do no better than make $y = \pi$.
21. When two constraints are added for each player, LP of Sect. 5.5 becomes

$$\max_{\mathbf{x}\geq 0,\mathbf{u}\geq 0} u_1 - u_2$$
$$\text{subject to } \sum_i x_i \leq 1,$$
$$-\sum_i x_i \leq -1,$$
$$u_1 - u_2 - \sum_i x_i a_{ij} \leq 0;\ \ j = 1, \ldots, n$$

The difference $u_1 - u_2$ is simply an unconstrained variable, so this is equivalent to LP1 of Sect. 3.10.

23. The **b** matrix is $\begin{bmatrix} -0.5 & 1 \\ 0 & -0.5 \\ -0.5 & 0 \\ -1 & 2 \end{bmatrix}$ and the value of the game is $-1/8$. Player 1 uses mixed strategy (1,3,0,0)/4 in game **b**, and player 2 uses mixed strategy (3,1)/4.
25. In part a, the most sweepers should go on segment 4, not segment 5. In part b, you would need 76 sweepers.

Chapter 6

3. In part b, choose one of the three permutations of (0, 3, 3) with probability 1/6, or one of the six permutations of (1, 2, 3) with probability 1/12. In part c after two steps, X_{12} and X_{34} are both equally likely to be 3, 4, or 5. Combining the two random variables antithetically produces a random variable that is always 8.
4. In part a, you should find that $f^* = 2.825$, $g^* = 1.528$, and $\Pi = 0.565$. In part b, since doubling the target value is tactically equivalent to halving the two costs, the average number of units consumed per target is as in part a. Considering both classes of target, the total consumptions are therefore 100(0.250 + 1.528) = 177.8 interceptors and 100(0.750 + 2.825) = 357.5 missiles. Since type 2 targets have value 2 each, the average value captured is 100(0.3 + 2(0.565)) = 143.0.
5. $f^* = 2.1$, $g^* = 1.2$, and $\Pi = u^* + cf^* = 0 + (0.2)(2.1) = 0.42$. With continuous allocations, using Fig. 6.5-2 with $\beta = 0.693$, the same average resources

result in a kill probability of about 0.44. The attacker benefits most from the continuity assumption because he loses ties in the discrete model.

6. Consider the generic target with parameters α, β, and v. The two cost-effectiveness ratios α/λ and β/μ are crucial to finding the saddle point. If $v \leq \alpha/\lambda$, player 1 will not attack and of course player 2 will not defend. If $\lambda/\alpha \leq v \leq \lambda/\alpha + \mu/\beta$, player 1 attacks, but player 2 does not defend. It is only when $\lambda/\alpha + \mu/\beta \leq v$ that both players have positive allocations. The first, second and third cases apply to targets 1, 2 and 3, respectively The solution is $\mathbf{x} = (0, 0.3466, 0.3054)$ and $\mathbf{y} = (0, 0, 0.1823)$.
7. Use formula 6.5-3 with $\beta = 0.223$, $A = 1.785$, and $D = 1.116$. Π is 0.45, so 55 % of the targets should survive.

Chapter 7

1. Use Solver to solve a succession of linear programs with varying budgets. The graph of shortest path length versus budget starts at 4 when the budget is 0 and ends at 17.5 when the budget is 20. It is concave.

2b. You can reduce the maximum flow to 1 by removing arcs c and e, which is within the budget of 2, and setting the cut to $\{a, c, e\}$. If you give Solver that starting point, it will not go back to spending all of Breaker's budget on arc d. This answer is for Solver in Excel 2010—who knows what future versions of Excel may bring?

3. The average number of detections increases to 1.5 when you increase the number of asset type 2 from 1 to 2. In part b, Breaker assigns something to all arcs except for arc c. Arcs b and d will have a zero survival probability, so the only hope for User is to travel arcs a, c, and e. The product of the three survival probabilities on those arcs is $(0.5)(1)(0.2) = 0.1$, so User would at best have one chance in ten of surviving.
4. $(1.66, 2.34) = 0.66(2, 2) + 0.34(1, 3)$, where (m, n) means m units of type 1 to arc 1 and n units of type 1 to arc 3. Therefore we can play (2, 2) with probability 0.66 and (1, 3) with probability 0.34 to achieve the desired result for units of type 1. The other two types can be treated similarly and independently, since there is no prohibition on multiple assets on an arc. Each play requires three random numbers.
5. $v = 0.05$. The min cut has five arcs in it.
6. The optimized objective function in part a) is 0.12, with arcs (3, 5) and (4, 6) receiving attention from Breaker. It increases to 0.7 in part b), with 12 arcs receiving attention. In part c), the solution is ten times the solution of part a).

Chapter 8

1. $v = 1/17$, $\mathbf{x}^* = \mathbf{y}^* = (10/17, 5/17, 2/17)$.
2. You should find $J = 3$, $A = 3$, $D = 9$.
3. If Evader uses column j, the payoff is $\sum_{i \neq j} x_i^* q_i + x_j^* p_j = \sum_{i=1}^{J} x_i^* q_i - x_j^* (q_j - p_j)$, which is equivalent to the expression given in the text.
4. Evader hides at a random point, and Searcher starts at a random point and proceeds through the cells until detection. The value is n, the number of cells. See Ruckle (1983).
5. $\mathbf{x}^* = (0, 0.2, 0.8, 0, 0, \ldots)$ $\mathbf{y}^* = (0.6, 0.4)$.
6. Searcher's active strategies are s_3 and s_4, and his optimal mixed strategy is $\mathbf{x}^* = (0, 0, 4/31, 27/31, \ldots)$. The value of the game is $105/31 = 3.387$, and $\mathbf{y}^* = (19/31, 12/31)$.
8. $x^* = (1,2,3,2)/8$. You can get this result either by linear programming, since you know v and therefore the payoff matrix, or by using the equation that, in game element 1, we must have $x_i v = d_{1i}$.
9. $\mathbf{x}^* = (1,2,2)/5$, $\mathbf{y}^* = (1,1,1)/3$, and $v = 5$.
10. $\sum_{i=1}^{4} \alpha_i^{-1} = 2$, so $\mathbf{y} = (4,2,1,1)$ and $v = \exp(-4) = 0.0183$. Detection is almost certain.
11. The game is covered in Sect. 8.2.1 because the network has an Eulerian path from 1 to 3 that includes all five arcs exactly once, namely 123413. That path has a length of 10. The value of the game is therefore 5.
12. If Searcher looks within 0.5 of the center, he will cover the center and at most one point on the circumference. Since all points on the circumference have probability 0 of including Evader, such looks will detect Evader with probability 1/7. If Searcher looks farther away from the center he can cover at most 1/6 of the circumference, but not the center, so the detection probability cannot exceed (1/6) (6/7).
13. Strategies A and B are not in equilibrium. If Monster follows A, it is true that Princess can do no better than B. But if Princess follows B, Monster can start at the origin and detect Princess immediately. The value of this game is unknown, although Alpern et al. (2008) establish bounds for it.

Chapter 9

1. Assume that column 2 is not active and solve the resulting 2×2 game. If the value is to be 0, then we must have $v^2 + v - 2.5 = 0$. Solve for $v = 1.158$, verify that column 2 is indeed not active, and then solve for $\mathbf{x}^* = (0.302, 0.698)$. If instead you mistakenly assume that column 3 is not active, you will find that $v = 1.5$. However, that game value makes column 3 become active, a contradiction.

2. Changing player 2's first vector reduces the game value to 1.19, and the saddle point changes to one where both players use their first vector.
3. If **p** is changed to (1, 0), Solver's solution of the Lopside game has $\mathbf{x}^* = (2,7)/9$ and $v = 4/3$, which agrees with the solution of the first game element.
6. Make the last row of the 3×2 element all 0's, or any negative number. That will keep player 1 from using it. Also make the last two columns of the 3×4 element all 9's, or any large number, since doing so will make columns 3 and 4 unattractive to player 2.
7. Both players know which game element is operative, but player 2 has to move first. Let m_k be the minmax value of game element k. Then the value of the game is $\sum_k p_k m_k$.
8. If the initial guess is $\mathbf{x} = (0.5, 0.5)$ and $v = 1$, Solver gets the correct solution. In fact, Solver in Excel 2010 always seems to get the correct solution regardless of the intial guess.

References

Alderson, D., Brown, G., Carlyle, M., & Wood, K. (2011). Solving defender-attacker-defender models for infrastructure defense. In R. K. Wood & R. F. Dell (Eds.), *Operations research, computing, and homeland defense* (pp. 28–49). Hanover, MD: INFORMS.

Alpern, S., Fokkink, R., Lindelauf, R., & van Oldser, G. (2008). The princess and monster game on an interval. *SIAM Journal on Control and Optimization, 47*(3), 1178–1190.

Altner, D., Ergun, O., & Uhan, N. (2010). The maximum flow network interdiction problem: Valid inequalities, integrality gaps, and approximability. *Operations Research Letters, 38*(1), 33–38.

Arnold, R. (1962). Avoidance in one dimension: a continuous matrix game. *Operations Evaluation Group report OEG IRM-10*, Chief of Naval Operations, Washington.

Aumann, R., & Machsler, M. (1972). Some thoughts on the minimax principle. *Management Science, 18*(5-Part 2), 54–63.

Baston, V., & Bostok, F. (1993). Infinite deterministic graphical games. *Society for Industrial and Applied Mathematics, 31*(6), 1623–1629.

Beale, E. (1964). Blotto Games and the decomposition principle. In A. Mensch (Ed.) *Proceedings of a conference under the aegis of the NATO Scientific Affairs Committee*, *Toulon*, *France* (pp.64–84), American Elsevier.

Beale, E., & Heselden, G. (1962). An approximate method of solving Blotto Games. *Naval Research Logistics, 9*(2), 65–79.

Berkovitz, L., & Dresher, M. (1959). A game theory analysis of tactical air war. *Operations Research, 7*(5), 599–620.

Bewley, T., & Kohlberg, E. (1976). The asymptotic theory of stochastic games. *Math of Operations Research, 1*(3), 197–208.

Blackett, D. (1954). Some Blotto games. *Naval Research Logistics, 1*(1), 55–60.

Blackett, D. (1958). Pure strategy solutions of Blotto games. *Naval Research Logistics, 5*(2), 107–9.

Blackwell, D. (1954). On multi-component attrition games. *Naval Research Logistics, 1*(3), 210–16.

Blackwell, D., & Ferguson, T. (1968). The big match. *Annals of Mathematical Statistics, 39*(1), 159–63.

Bracken, J., Brooks, P., & Falk, J. (1987). Robust preallocated preferential defense. *Naval Research Logistics, 34*(1), 1–22.

Bram, J. (1963). A 2-player N-region search game. *Operations Evaluation Group report OEG IRM-31 (AD 402914)*, Chief of Naval Operations, Washington.

Bram, J. (1973). *The discrete evasion game*. Arlington: CNA 244cbr, Center for Naval Analyses.

Brown, G., Kline, J., Thomas, A., Washburn, A., & Wood, K. (2011). A game-theoretic model for defense of an oceanic bastion against submarines. *Military Operations Research, 16*(4), 25–40.

A. Washburn, *Two-Person Zero-Sum Games*, International Series in Operations Research & Management Science 201, DOI 10.1007/978-1-4614-9050-0,

Burch, C., Carr, R., Krumke, S., Marathe, M., Phillips, C., & Sundberg, E. (2003). A decomposition-based pseudoapproximation algorithm for network flow inhibition. in *Network Interdiction and Stochastic Programming*, D.L. Woodruff (Ed.), Kluwer.

Burger, E. (1963). *Introduction to the Theory of Games*. Prentice-Hall.

Charnes, A. (1953). Constrained games and linear programming. *Proceedings of the National Academy of Sciences, 39*(7), 639–41.

Coleman, A. (1982). *Game Theory and Experimental Games*. Pergamon.

Cormican, K., Morton, D., & Wood, K. (1998). Stochastic network interdiction. *Operations Research, 46*(2), 184–197.

Croucher, J. (1975). Application of the fundamental theory of games to an example concerning antiballistic missile defense. *Naval Research Logistics, 22*(1), 197–203.

Danskin, J. (1967). *The Theory of Max-Min*. Springer-Verlag.

Danskin, J. (1968). A helicopter submarine search game. *Operations Research, 16*(3), 509–517.

Danskin, J. (1990). On the cookie cutter game: search and evasion on a Disc. *Mathematics of Operations Research, 15*(4), 573–596.

Dresher, M., Karlin, S., Shapley, L. (1950). Polynomial games. In H. Kuhn, & A. Tucker (Eds.), *Contributions to the theory of games I* (*Annals of Mathematics Studies*, Vol. 24, pp. 161–180), Princeton University Press.

Dresher, M. (1961). *Games of strategy theory and applications*. Prentice-Hall, subsequently republished in 1980 as *The Mathematics of Games of Strategy Theory and Applications*, Dover.

Dresher, M. (1962) *A sampling inspection problem in arms control agreements: a game theory approach*. Memorandum RM-2972-ARPA, RAND, Santa Monica, CA.

Eckler, A., & Burr, S. (1972). *Mathematical models of target coverage and missile allocation*. Arlington, VA: Military Operations Research Society.

Everett, H. (1957) Recursive games. In M. Dresher, A. Tucker, & P. Wolfe (Eds.), *Contributions to the theory of games III* (*Annals of Mathematical Studies*, Vol. 39, pp. 47–78), Princeton University Press.

Ferguson, T. (1967). On discrete evasion games with a two-move information lag, *Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability* (Vol. 1, pp. 453–462), University of California Press.

Ferguson, T., & Melolidakis, C. (1998). On the inspection game. *Naval Research Logistics, 45*(3), 327–334.

Filar, J., & Schultz, T. (1987). Bilinear programming and structured stochastic games. *Journal of Optimization Theory and Applications, 53*, 85–104.

Fodor, F. (1999). The densest packing of 19 congruent circles in a circle. *Geometriae Dedicata, 74*, 139–145.

Ford, L. & Fulkerson D. (1962). *Flows in Networks*. Princeton University Press.

Fréchet, M. (1953). Émile Borel, initiator of the theory of psychological games and its application. *Econometrica, 21*, 95–96.

Fourer, R. (2013). Linear programming. *ORMS Today, 40*(3), 40–53.

Friedman, Y. (1977). Optimal strategy for the one-against-many battle. *Operations Research, 25*(5), 884–888.

Fulkerson, D., & Hardin, G. (1977). Maximizing the minimum source-sink path subject to a budget constraint. *Mathematical Programming, 13*(1), 116–118.

Gal, S. (1979). Search games with mobile and immobile hider. *SIAM Journal on Control and Optimization, 17*(1), 99–122.

Gale, D., & Stewart, F. (1953). Infinite games with perfect information. In H. Kuhn, A. Tucker (Eds.), *Contributions to the theory of games II* (*Annals of Mathematics Studies,* Vol. 28), Princeton University Press.

Galiano, R. (1969). *A simple analytic model for preallocated defense*. Paper 30, Lambda Corporation, Arlington, VA.

Gallo, G., & Ulkucu, A. (1977). Bilinear programming: an exact algorithm. *Mathematical Programming, 12*(1), 173–194.

Garnaev, A. (2000). *Search games and other applications of game theory*. Springer.

Gerards, A. (1995). Theorem 14 in chapter 3 of *Handbooks in Operations Research and Management Science*, in M. Ball, T. Magnanti, C. Monma, & G. Nemhauser (Eds.), North Holland.

Guy, R. (Ed.). (1991). *Combinatorial games*. Rhode Island: American Mathematical Society.

Harris, T., & Ross, F. (1955). *Fundamentals of a method for evaluating rail network capacities*. Memorandum RM-1573, RAND Corporation, Santa Monica, CA.

Harsani, J. (1967). Games with incomplete information. *Management Science, 14*(3), 159–182.

Haywood, O. (1954). Military decision and game theory. *Operations Research, 2*(4), 365–385.

Hemmecke, R., Schultz, R., & Woodruff, D. (2003). Interdicting stochastic networks with binary interdiction effort. In D. L. Woodruff (Ed.) *Network interdiction and stochastic programming*, Kluwer.

Ho, Y., Bryson, A., & Baron, S. (1965). Differential games and optimal pursuit-evasion strategies. *IEEE Transactions on Automatic Control, 10*(4), 385–389.

Hofstadter, D. (1982). Mathemagical themas. *Scientific American* 16–18, August.

Hohzaki, R., & Iida, K. (1998). A search game with reward criterion. *Journal of the Operations Research Society of Japan, 41*(4), 629–642.

Hohzaki, R., & Washburn, A. (2001). The diesel submarine flaming datum problem. *Military Operations Research, 6*(4), 19–30.

Isaacs, R. (1955). *The problem of aiming and evasion*. Memorandum P-642, RAND Corporation, Santa Monica, CA.

Isaacs, R. (1965). *Differential Games*, Wiley.

Isbell, J., & Marlow, W. (1956). Attrition games. *Naval Research Logistics, 3*(1–2), 71–94.

Isler, V., Kannan, S., & Khanna, S. (2005). Randomized pursuit-evasion in a polygonal environment. *IEEE Transactions on Robotics, 5*(21), 864–875.

Jain, M., Korzhyk, D., Vanek, O., Conitzer, V., Pechouochk, M., & Tambe, M. (2011). A double oracle algorithm for zero-sum security games on graphs. In *The 10th International Conference on Autonomous Agents and Multiagent Systems* (Vol. 1). International Foundation for Autonomous Agents and Multiagent Systems, Richland.

Johnson, S. (1964). A search game. *Advances in Game Theory* (Vol. 52, p.39), Princeton University Press.

Kallenberg, L. (1983). Linear programming and finite Markovian control problems. Mathematical Centre Tracts (Vol. 148, pp. 1–245), Amsterdam.

Karlin, S. (1959). *Mathematical Methods and Theory in Games*, *Programming*, *and Economics* (Vols. 1 and 2). Addison-Wesley.

Kaufman, H., & Lamb, J. (1967). An empirical test of game theory as a descriptive model. *Perceptual and Motor Skills, 24*(3), 951–960.

Kikuta, K. (1983). A note on the one-against-many battle. *Operations Research, 31*(5), 952–956.

Kikuta, K. (1986). The matrix game derived from the many-against-many battle. *Naval Research Logistics, 33*(4), 603–612.

Kikuta, K., & Ruckle, W. (1994). Initial point search on weighted trees. *Naval Research Logistics, 41*(6), 821–831.

Kohlberg, E. (1975). Optimal strategies in repeated games with incomplete information. *International Journal of Game Theory, 4*(1), 7–24.

Kuhn, H. (1950). A simplified two-person poker. *Contributions to the theory of games I*. In H. Kuhn and A. Tucker (Eds.) *Ann. Math. Studies, 24*, 97–103, Princeton U. Press.

Ladany, S., & Machol R. (Eds.) (1977). *Optimal Strategies in Sports*. North Holland.

Lalley, S., & Robbins, H. (1988). Stochastic search in a convex region. *Probability Theory and Related Fields, 77*(1), 99–116.

Lieberman, B. (1960). Human behavior in a strictly determined 3×3 matrix game. *Behavioural Science, 5*(4), 317–322.

Lee, L. Y. & Lee, K. T. (1990). The discrete evasion game with three-move lag. *Naval Research Logistics, 37*(6), 867–874.

Luce, R., & Raiffa, H. (1957). *Games and Decisions*. Wiley.
Matheson, J. (1967). *Preferential strategies with imperfect weapons*. Report AR67-1, Analytical Services Corporation, Falls Church, VA.
Matula, D. (1966). *Games of sequence prediction*. Report No. ORC 66–3, Operations Research Center, University of California, Berkeley.
McGarvey, D. (1987). *Mixed strategy solutions to integer Prim-Read defense problems*. Report P-7302, RAND Corporation, Santa Monica, CA.
McKinsey, J. (1952). *Introduction to the Theory of Games*. McGraw Hill.
Mertens, J., & Neyman, A. (1981). Stochastic games have a value. *International Journal of Game Theory, 10*(1), 53–66.
Messick, D. (1967). Interdependent decision strategies in zero-sum games: a computer-controlled study. *Behavioural Science, 12*(1), 33–48.
Meinardi, J. (1964). A sequentially compounded search game. In A. Mensch, (Ed.), *Proceedings of a conference under the aegis of the NATO Scientific Affairs Committee*, *Toulon*, *France* (pp. 285–299), American Elsevier.
Morin, R. (1960). Strategies in games with saddle points. *Psychological Reports, 7*(3), 479–483.
Morse, P., & Kimball, G. (1950). *Methods of Operations Research*. Joint publishers Technology Press of MIT and Wiley, Section 5.4.5.
Morton, D., Pan, F., & Saeger, L. (2007). Models for nuclear smuggling interdiction. *IIE Transactions, 39*(1), 3–14.
Neuts, M. (1963). A multistage search game. *Society for Industrial and Applied Mathematics, 11*(2), 502–507.
Newman, D. (1959). A model for 'real' poker. *Operations Research, 7*(5), 557–560.
Norris, R. (1962). Studies in search for a conscious evader. *Report 279 (AD 294832)*, Lincoln Lab, MIT, Cambridge, MA.
Owen, G. (1995). *Game Theory* (3rd ed.). Academic Press.
Phillips, C. (1993). The network inhibition problem, *Proceedings of the 25th Annual ACM Symposium on the Theory of Computing*, (pp. 776–785), ACM.
Pita, J., Jain, M., Marecki, J., Ordóñez, F., Portway, C., Tambe, M., Western, C., Paruchuri, P., & Kraus, S. (2008). Deployed ARMOR Protection: the application of a game theoretic model for security at the Los Angeles International Airport. *Proceedings of the 7th international joint conference on Autonomous agents and multiagent systems* (pp. 125–132), International Foundation for Autonomous Agents and Multiagent Systems.
Ponssard, J. (1980). The L-P formulation of finite zero-sum games with incomplete information. *International Journal of Game Theory, 9*(2), 99–105.
Pultr, A., & Morris, F. (1984). Prohibiting repetition makes playing games substantially harder. *International Journal of Game Theory, 13*, 27–40.
Raghavan, T., & Filar, J. (1991). Algorithms for stochastic games–a survey. *Methods and Models of Operations Research, 35*(6), 437–472.
Robinson, J. (1951). An iterative method of solving a game. *Annals of Mathematics, 54*(2), 296–301.
Ruckle, W. (1983). *Geometric games and their applications*. Pitman, Boston.
Sakaguchi, M. (1973). Two-sided search games. *Journal of the Operations Research Society of Japan, 16*, 207–225.
Schroeder, R. (1970). Linear programming solutions to ratio games. *Operations Research, 18*(2), 300–305.
Schwartz, E. (1979). An improved computational procedure for optimal allocation of aircraft sorties. *Operations Research, 27*(3), 621–627.
Shapley, L. (1953). Stochastic games. *Proceedings of the National Academy of Sciences, 39*(10), 1095–1100.
Shilov, G. (1965). *Mathematical analysis* (pp. 44–45). Pergamon.
Sherali, H., & Shetty, C. (1980). A finitely convergent algorithm for bilinear programming problems. *Mathematical Programming, 19*, 14–31.

Stone, L. (1975). *Theory of Optimal Search*. Academic Press.

Thomas, C., & Deemer, W. (1957). The role of operational gaming in operations research. *Operations Research, 5*(1), 1–27.

Thomas, C. (1964). Some past applications of game theory to problems of the United States Air Force. In A. Mensch (Ed.), *Proceedings of a conference under the aegis of the NATO Scientific Affairs Committee*, *Toulon*, *France* 250–267, American Elsevier.

Thomas, L., Hartley, R. & Lavercombe, A. (1983). Computational comparison of value iteration algorithms for discounted Markov decision processes. *Operations Research Letters, 2*(2), 72–76.

Thomas, L., & Washburn, A. (1991). Dynamic search games. *Operations Research, 39*(3), 415–422.

Thompson, G. (1989). John von Neumann's contributions to mathematical programming economics. In M.Dore, S. Chakravarty, & R.Goodwin (Eds.), *John von Neumann and Modern Economics*, Clarendon Press.

U.S. Air Force (Studies and Analysis). (1971). Methodology for use in measuring the effectiveness of general purpose forces (an algorithm for approximating the game theoretical value of N-staged games). *Report Saber Grand (Alpha).*

Van Creveld, M. L. (1985). *Command in war*. Harvard University Press.

Washburn, A. (1980). Search evasion game in a fixed region. *Operations Research, 28*(6), 1290–1298.

Washburn, A. (1990). Deterministic graphical games. *Journal of Mathematical Analysis and Applications, 153*(1), 84–96.

Washburn, A., & Wood, K. (1995). Two person zero-sum games for network interdiction. *Operations Research, 43*(2), 243–251.

Washburn, A. (2002). *Search and detection* (pp. 2–14). MD: INFORMS, Linthicum.

Washburn, A., & Ewing, L. (2011). Allocation of clearance assets in IED warfare. *Naval Research Logistics, 58*(3), 180–187.

Washburn, A. (2013). Blotto politics. *Operations Research, 61*(3), 532–543.

Williams, J. (1954). *The Compleat Strategyst*. McGraw-Hill.

Winston, W., & Cabot, A. (1984). *A stochastic game model of football play selection*. Indiana University mimeograph.

Winston, W. (2009). *Mathletics*. Princeton University Press.

Winterbotham, F. (1974). *The ULTRA Secret*. London: Weidenfeld and Nicolson.

Wood, K. (1993). Deterministic network interdiction. *Mathematical and Computer Modelling, 17*, 1–18.

von Neumann, J. (1928). Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen* **100** 295–320. English translation in *Contributions to the Theory of Games* **4** 13–42, Princeton University Press.

von Neumann, J. (1937). Über ein Ökonomisches Gleichungssystem und eine Verallgemeinerung des Brouwerschen Fixpunktsatzes. *Ergebnisse eines Mathematischen Kolloquiums, 8*, 73–83.

von Neumann, J., & Morgenstern, O. (1944). *Theory of games and economic behaviour*, Princeton University Press.

Zermelo, E. (1912). Über eine Anwendung der Mengenlehre auf die Theorie des Schachspiels. *Proceedings of the 5th International Congress of Mathematicians* (pp. 501–510). Cambridge University Press.

Index

A. Washburn, *Two-Person Zero-Sum Games*, International Series in Operations Research & Management Science 201, DOI 10.1007/978-1-4614-9050-0,