Colonel Blotto: Searching for Patterns in a Game of Allocation

By
Jeremiah Jones

A SENIOR RESEARCH PAPER PRESENTED TO THE DEPARTMENT OF MATHEMATICS AND COMPUTER SCIENCE OF STETSON UNIVERSITY IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF BACHELOR OF SCIENCE

STETSON UNIVERSITY
2015

Acknowledgments

I'd like to thank my advisor, Dr. Erich Friedman, for his support throughout the year. Without him, I would never have pursued this research topic. I'd also like to thank Dr. Will Miles for his constant support throughout my time at Stetson. For his "developmental advising" and conscientiousness as a professor, I am truly in his debt.

My thanks extend to Professor Mollie Rich. She is responsible for much of the personal and musical growth I have experienced at Stetson. For her radiating confidence, her ability to simplify the complex, and the countless lessons she has taught—vocal and otherwise—I am forever grateful.

Last, and certainly not least, I thank my family: my parents, for unceasing love that defies understanding; my sisters, for being both friends and mentors. Without them seeing me through both the best and worst of times, I don't know where I would be.

Oh, and thanks to John von Neumann. His theorem gave rise to Game Theory.

Contents

List of Figures

Abstract

Colonel Blotto: Searching for Patterns in a Game of Allocation
By
Jeremiah Jones
May 2015

Advisor: Erich Friedman
Department: Mathematics and Computer Science

In this paper, we analyze Colonel Blotto games, a type of allocation strategy game played by two players. We discuss the history of the game and build a framework of game-theoretic and linear programming knowledge that will allow us to solve these games. MATLAB code is presented that will automatically generate and solve Colonel Blotto games for any parameters. The solutions to these games are presented for a wide range of parameters. We prove several results, and conjecture many other patterns. Finally, we discuss ways in which the research, and the search for patterns among the solutions, may continue in the future.

1 History and Previous Research

1.1 What is Colonel Blotto?

To begin our discussion of the history of the problem, we will develop a complete and tidy description of the standard Colonel Blotto game. The game is played between two players, an Attacker and a Defender. The Attacker has a certain number of armies A, and the Defender has a number of armies D. There are a certain number of objectives or bases B that the Defender wishes to keep the Attacker from capturing. To do this, the Defender must allocate his D armies among the B bases according to some partition he deems fit. Conversely, the Attacker must allocate his A armies among the B bases according to some partition that he believes to be optimum. After both players have made their choices in ignorance of the other player's allocations, the results are revealed. The player who allocates the greater number of armies to each base gains possession of it. In cases where both players assign the same amount of force to a base, the base is divided equally among the players. The Colonel Blotto game at its most basic level is an experiment in resource allocation. Given a certain number of possible objectives, the task of a Colonel Blotto player is to reach the greatest number of these given some finite level of available resources. However, the aspect that makes Colonel Blotto a particularly powerful model is the competitive aspect of the game. The Blotto player must reach as many of the objectives as possible while knowing that the enemy is attempting the same. Immediately, the reader might think of a few applications to which the Blotto game might apply. In fact, applications abound in diverse fields such as "international politics, electoral politics, business, the law, biology, [and] sports," and make their way into areas such as auction strategy and, perhaps obviously, wartime operations [10, 4].

1.2 Previous Formulations and Results

By all accounts, the Colonel Blotto game was first introduced by Émile Borel in 1921 [7]. However, the paper most commonly cited for introducing the game makes no mention of the

name "Colonel Blotto," nor does it describe any rules of play. Instead, it deals with integrals of skew symmetric payoff functions, which may have led to one formulation of the Colonel Blotto game as we know it today. Regardless, the formulation of the game was met with interested mathematicians in the early years of game theory. Subsequently, interest declined. Some attribute this to the game's complexity, and the lack of "clean, precise... results" from its analysis [10]. This has led to an odd state of affairs; knowledge of the Colonel Blotto game is widespread, as it is presented in most introductory texts in the field of game theory, yet the literature on the subject is fairly sparse. The papers which have been published on the subject generally introduce alternative formulations of the Colonel Blotto game.

Gross and Wagner examine certain cases of the game in which the Attacker and Defender's forces are not given integer values, but continuously divisible ones. They find analytic solutions for any number of attacking and defending armies when there are 2 bases, and geometric solutions for symmetric games when the number of bases greater than 2 [11]. Adamo and Matros also analyze a Blotto game with continuously divisible resources. They frame this game as an auction with N players competing for K prizes, where each player has an independently and identically distributed budget and each prize has some positive value. They prove that this game has a symmetric monotonic Bayesian equilibrium — one where every player adopts the same strategy. The strategy basically requires that each player bid an amount on every prize that is proportional to their individual budget. The player with the highest budget thus wins all prizes [4].

Kovenock and Roberson analyze quite a different formulation of the Colonel Blotto game. In this paper, a player A competes in two separate Colonel Blotto games against two opponents 1 and 2. Before competing in their respective games, players 1 and 2 may form an alliance to trade their resources. After trading resources, they must compete in their games simultaneously with no other trading of resources allowed. The paper finds that there are ranges of parameters for which one player gives away resources to the other player. This occurs because this manner of trade leads to a strategic shift of enemy troops

being allocated away from the player making the transfer and toward the player receiving the transfer. This demonstrates how such transfers which may be seen as only benefitting one player may actually have direct and strategic benefits for both players [12].

Roberson and Kvasov take a different approach to the Colonel Blotto game. Instead of formulating Colonel Blotto as a zero- or constant-sum game, they formulate a non-constant-sum game. In the standard formulation of Colonel Blotto, the allocated resource is "use it or lose it" in that unallocated resources have no value to the players. Thus, the non-constant-sum game adds a small value to not using all of the resources available to a player. This makes for a much more realistic and interesting auction. One rarely expects to pay an exact amount of money when entering an auction. Valuations are made on individual items, and the bidder compares their own valuation of the items to the opportunity cost of having resources to spend on other items, within the auction or outside of it. As the writers point out, in the constant-sum game, a certain level of asymmetry in the players' budgets makes the game trivial: one player has sufficient resources to outbid the other player in every auction. In the non-constant-sum game, no amount of asymmetry makes the game trivial. The value of holding on to a certain amount of resources thus keeps one player from unequivocally overwhelming the other player. The writers find that a certain level of asymmetry in the players' budgets allows a one-to-one mapping from the set of solutions in the constant-sum game to the set of solutions in the non-constant-sum game. When the asymmetry reaches a certain threshold, this mapping breaks down due to the aforementioned differences in the games. Entirely different equilibrium solutions are then constructed by the authors [13].

Golman and Page generalize the Colonel Blotto game, invoking a witticism in the process. The class of "General Blotto" games allows for different valuations of objectives. That is, the payoffs of different strategies depend not only on the number of objectives captured, but on the valuation of the objectives as well. They show also that pure strategy equilibria rarely exist. Instead, the equilibria of the General Blotto games usually involve mixed strategies [10].

In these papers, all variants of the Colonel Blotto game are symmetric in the game's rules; they are "fair" in that no player is favored over any other players. In particular, when players assign an equal amount of resources to the same objective, the standard rules would split the objective evenly among the players. However, there are situations in which the rules of the game should not be so symmetric. If we formulate the game with its military underpinnings and suppose that one player acts as an Attacker and another as a Defender. One would expect that a military unit attacking a fortified unit of equivalent size would be at a slight disadvantage, and thus we might expect that if players assigned an equal number of armies to the same base, the Defender would win the conflict. Thus, the version of Colonel Blotto studied in this paper will carry this alteration in the rules described in the previous section: in case of ties, the Defender wins the objective. A rule of this type, asymmetrically benefitting one player, has not been involved in the formulation of a Blotto game studied with any frequency. Therefore, the author of this paper hopes to discover some patterns of interest in the study of this particular game.

2 Game Theory

Before beginning the analysis of our altered Colonel Blotto game, we must build the mathematical framework for analyzing games of this type. In this section, we will present an overview of game theory and methods of solution, showing applications to the Colonel Blotto game as the theory is developed.

2.1 What's in a Game? Terminology and the Basics

In building mathematical theory it is often useful to define when the tools developed may be used. In the English vernacular a "game" can encompass many different activities: sports competitions, cards, hide-and-seek; the list goes on. Mathematically, we abstract from these events to arrive at some generalized concept of a game, which we hope may be used to describe a vast array of situations that may or may not be colloquially called games.

We define a **game** to be a situation in which multiple parties (whom we call **players**) have conflicting goals. Furthermore, in a game, the players have some ability to achieve their goals by taking actions that affect the outcome of a game. For example, in *Rock, Paper, Scissors*, each player presumably desires to win the contest and chooses to play either Rock, Paper, or Scissors each turn in order to achieve this goal.

A player influences the outcome of a game by employing **strategies**. In Game Theory, a **strategy** is a comprehensive way of playing a game that takes into account all possible actions on the part of the opposition. In other words, a strategy is a method for playing a game that works against any strategy the opposition chooses. This doesn't mean that the strategy is required to work well against any opposing strategy, just that it yields some outcome for the game, favorable or not. Let us look at the *Rock, Paper, Scissors* example. Each party's decision of which item to play is made without knowledge of the other's decision. So "always play Rock" is a valid strategy for a player. It may be a poor strategy, unless the opposition is equally misguided and employs the strategy "always play Paper," but our

definition of a strategy does not require it to be good, only that it work for every enemy strategy.

We see that for *Rock, Paper, Scissors*, the decision to always play one of the three items in the title is a strategy. A reader well versed in this particular game might experientially submit that each of these three strategies is poor when the opposition has all but the most dismal pattern recognition skills. Instead of employing these **pure strategies**, a player might wish to use a **mixed strategy** — that is, sometimes use one strategy and sometimes another. The strategies may be mixed to any set of proportions and chosen at random by any chance device that a player deems suitable. A player might decide that he should flip a coin and play Rock if it turns up heads, Paper if it turns up tails, and Scissors only if it hits the enemy player on the forehead while being flipped. While this would be one method for mixing strategies, we will see later that there are good reasons for mixing strategies according to specific proportions.

In the *Rock, Paper, Scissors* example, two players are in direct conflict: when one player wins, the other player must lose. When the winnings of one player are equivalently the losses of the other, we call this a zero-sum game. Most parlor games fall into this category, and the theory behind them is very well-developed. However, many conflicts between parties are not so black-and-white. Sometimes, some amount of coordination between players can benefit both more than they might obtain through open conflict. If we abstract the Theory of Games to the dynamics of nations, we might see how cooperation can benefit nations more than open war can in many cases. Through a great wealth of imagination, we refer to these types of games as nonzero-sum games.

Since the Colonel Blotto game as we have defined it is a two-player zero-sum game, the theory that will be developed in the following section will pertain solely to games of this type. However, it is important to realize that nonzero-sum games provide a model for applications in many different fields. We might consider that war is not a zero-sum game: wealth and human lives are lost on both sides. Thus the Colonel Blotto game might be cast

in terms of a nonzero-sum game to achieve greater modelling power within the context of wartime operations. Following this small aside on a rather lengthy subject, we move on to building the theory of two-player zero-sum games.

2.2 Game Matrices

We have seen that the outcome of a game depends on the strategies enacted by both players. Thus it is useful to represent as a game as a matrix, in which the columns represent the different strategies that one player (referred to as the Column Player) may use, and the rows represent the strategies that the other player (referred to as the Row Player) may use. The entries in this matrix represent the **payoff** from the Column Player to the Row Player that occurs when the each player enacts the strategies of that entry's row and column. In other words, for a game with m strategies for the Row Player and n strategies for the Column Player, the **payoff matrix** is given by

$$\begin{pmatrix} a_{1,1} & \cdots & a_{1,n} \\ \vdots & \ddots & \vdots \\ a_{m,1} & \cdots & a_{m,n} \end{pmatrix}$$

where $a_{i,j}$ is the payoff of the Column Player to the Row Player when the Row Player uses strategy i and the Column Player uses strategy j.

The payoffs used in a matrix may be any quantitative units of measurement. In the case of Poker, the payoff may be in dollars. In *Rock, Paper, Scissors*, there might be some fiscal component, or the payoff may simply be pride in winning and shame in defeat. We could then use some quantitative measurements to represent this; let 1 represent a win for the Row Player, 0 represent a tie, and -1 represent a defeat for the Row Player. Then the *Rock, Paper, Scissors* game would have the following payoff matrix given by Figure 1.

		C		
		Rock	Paper	Scissors
	Rock	0	-1	1
R	Paper	1	0	-1
	Scissors	-1	1	0

Figure 1: A payoff matrix for *Rock, Paper, Scissors.*

2.3 Solutions to Game Matrices

In the analysis of a game, we are searching for the best strategies for each player to employ so that the payoff to each player is at a maximum. For the Row Player, this means maximizing the winnings; for the Column Player, this means minimizing the losses. The methods for finding optimal solutions are diverse and appropriate in various contexts. The following sections will explain techniques used in different situations. For a nearly-universal solution method that will be indispensable in solving Colonel Blotto games, refer to Section 3.

2.3.1 Saddle Point Solutions

We begin our discussion of solutions with a straightforward example. Let us analyze a game where the Row Player has m different strategies and the Column Player has n strategies, and let the game be described by the matrix $\mathbf{A} = (a_{i,j})$ where $i = 1 \ldots m$ and $j = 1 \ldots n$.

The Row Player, who we will call R, wishes to choose a strategy to maximize his winnings. In analyzing the game, R makes the assumption that the Column Player, C, is trying to minimize his losses, and thus has made the following calculation for any row i:

$$\text{Minimize } a_{i,j} \text{ over } j = 1 \ldots n.$$

Then R desires to maximize his profit even with C attempting this minimization. So R calculates a lower bound for his profit given by

$$v^- = \max_{i=1\ldots m} \min_{j=1\ldots n} a_{i,j}$$

where v^- is called both the **maximin** and the **lower value of the game**. With this calculation, R has taken into account the worst-case scenarios for each of his strategies, and chosen the best among them.

This calculation is similar for C. We assume that R is playing to maximize his winnings for any column j and is thus attempting to

$$\text{Maximize } a_{i,j} \text{ over } i = 1 \ldots m.$$

Then C desires to minimize his loss even with R attempting this maximization. So C calculates a lower bound for his loss given by

$$v^+ = \min_{j=1\ldots n} \max_{i=1\ldots m} a_{i,j}$$

where v^+ is called both the **minimax** and the **upper value of the game**. Again, C has taken into account the worst-case scenarios for each of his strategies and chosen the best among them (remember that C wants low values, so this minimization is how C chooses the best worst-case scenario).

If $v^+ = v^-$, then there is at least one entry in the payoff matrix with this value, and it corresponds to the most cautious strategies of both players. This is because v^- has been calculated as a best worst-case scenario for R and v^+ has been calculated as a best worst-case scenario for C. Therefore, if both players utilize these cautious strategies, they assure that the opponent can do no better than $v^- = v^+$. We call this payoff the **value** of the game, and denote it by v. The entries in the payoff matrix that are equal to v are called **saddle**

points.

Definition 1. A row i^* and column j^* constitute a **saddle point** if for all rows i and columns j,

$$a_{i^*,j} \leq a_{i^*,j^*} \leq a_{i,j^*}.$$

This means that if R plays i^*, C's best response would be to play j^*; if C plays j^*, R's best response would be to play i^*. For this reason, we make the following prescription:

Definition 2 (Saddle Point Principle)**.** If a game has a saddle point, both players should play the strategies corresponding to that entry.

To demonstrate saddle point solutions and the logic behind their use, consider the following game.

			C	
		I	II	III
	I	5	10	6
R	II	4	7	2
	II	2	1	9

Figure 2: A game demonstrating a saddle point solution.

To find the maximin, we first minimize among the columns $j = 1 \ldots n$ for each row $i = 1 \ldots m$. Then, we find the maximum of these values over the rows $i = 1 \ldots m$. The minimax can be found by attempting this process in reverse: first maximize over all the rows for each column, then find the minimum of these values.

We can carry out this process more intuitively by approaching it visually. We always minimize horizontally across the columns and maximize vertically down the rows. To find the maximin, find the minimum values of each row and write these values to the right of the row. Then find the maximum of these values to yield v^-. To find the minimax, find the

maximum values of each column and write these values below each column. Then find the minimum of these values to yield v^+. This process is shown in the following figure.

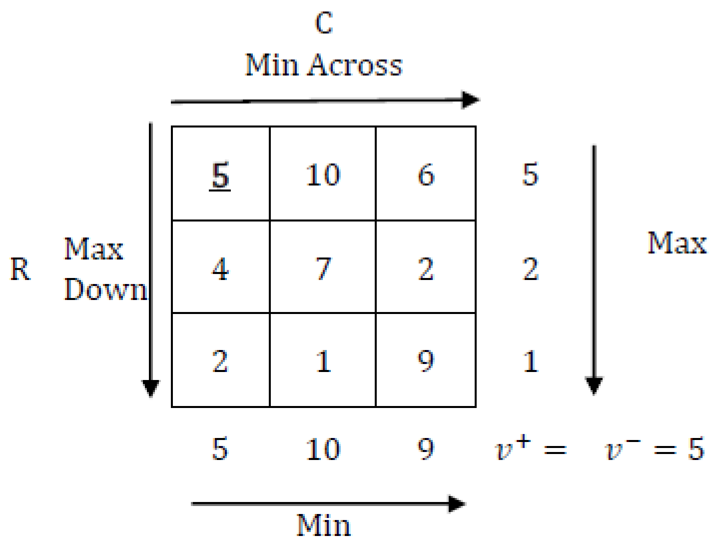

Figure 3: A demonstration of finding the minimax and the maximin. In this case there is a saddle point, which is bolded and underlined.

Notice that in this case $v^+ = v^-$, which implies that (I,I) is a saddle point. We can see that it is beneficial for both players to play this strategy by considering the consequences of deviating from the saddle point. For example, suppose C plays the saddle point strategy I. Then it would be in R's best interest to also play the saddle point strategy I, as any other strategy would lead to a lower payoff to R. Now conversely, suppose R plays the saddle point strategy I and that C is considering which strategy to play. It would be in C's best interest to play I as well, because any other strategy would lead to a higher loss for C. If either player is enticed by greed to play other strategies, the opposition may take advantage. If C desires a loss of 1 and moves to play strategy II, then R can improve the winnings of the game by simply sticking with the saddle point strategy I. If a player moves away from the saddle, then that player is gifting the opposition with a better payoff. This is why rational players obey the Saddle Point Principle.

2.3.2 The von Neumann Minimax Theorem

What methods of solution are available to us when $v^- < v^+$? We can't find saddle point solutions as described in the previous chapter, but we hope we might find ways of playing that are optimal. Luckily, we have a theorem that is of use in this search. This theorem applies to games of the finite, two-player zero-sum variety. The first qualifier, finite, simply means that the number of pure strategies for each player is finite. Since Colonel Blotto is a finite game, we may use the following theorem and all of its important results.

Theorem 3 (The Minimax Theorem). *For every finite two-player zero-sum game:*

- *there is a number v, called the value of the game;*
- *there is a mixed strategy for R such that R's average winnings are at least v no matter what C does; and*
- *there is a mixed strategy for C such that C's average loss is at most v no matter what R does.*

The Minimax Theorem essentially states that even if a game does not have a pure strategy saddle point, a saddle point will exist in *mixed* strategies. Thus, when searching for a solution to a game, we can carry out the search more generally by searching for mixed strategies, as pure strategies are simply a special case of mixed strategies (i.e. the case when the probability of using all other strategies is zero).

A rigorous proof of this theorem is meticulous, involved, and ultimately beyond the scope of this paper. However, Section 3.1 will essentially prove the Minimax Theorem using linear programming. A brief mention will be made in that section as to the specific theorem in linear programming that may be used to arrive at the results of the Minimax Theorem.

2.3.3 The Oddment Method

We continue with a method for producing odds for mixing pure strategy that would give us a saddle point in mixed strategies. While this method is limited to 2 by 2 game matrices, it

is simple and efficient enough to garner attention. Consider the game shown in Figure 4.

		C	
		I	II
R	I	1	6
	II	5	4

Figure 4: A 2x2 matrix game.

A search for a saddle point in pure strategies yields no result, so we must try something different.

The oddment method of finding mixed strategies proceeds as follows. First consider the game from R's perspective. Subtract across the rows, and write the absolute value of this quantity next to the row (shown in Figure 5). Then, switch the numbers in each row. The numbers next to each row now give the odds for using that row in a mixed strategy. If we prefer to deal with probabilities instead of odds, then divide each of the oddments by their sum. This process is shown in Figure 6.

R	I	1	6	5
	II	5	4	1

Figure 5: The positive difference across the rows.

					Odds	Probability
R	I	1	6	5	1	1/6
	II	5	4	1	5	5/6

Figure 6: A demonstration of the Oddment Method for R. This yields the odds in R's optimal mixed strategy.

This process proceeds similarly for C. Subtract down the columns, writing the absolute value of this quantity at the bottom of each column. Switch the numbers to produce the odds for mixing these strategies, and divide each oddment by their sum to produce the probabilities for mixing the strategies.

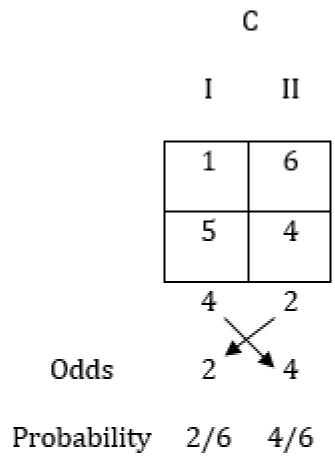

Figure 7: A demonstration of the Oddment Method for C. This yields the odds in C's optimal mixed strategy.

According to the Oddment Method, a wise player in R's position would mix his strategies randomly, playing strategies I and II with odds 1:5. On the other hand, a wise player in C's position would mix her strategies randomly, playing strategies I and II with odds 1:2.

Since we now have a method for finding optimal mixed strategies, we now discuss briefly how to find the value of such a game. If $\mathbf{p}$ is C's optimal mixed strategy, and $\mathbf{s}$ is a row of the payoff matrix corresponding to one of the strategies in R's optimal mix, simply compute the vector product $\mathbf{p} \cdot \mathbf{s}$. This proceeds similarly for the other player: given R's optimal strategy $\mathbf{q}$ and a column of the payoff matrix $\mathbf{t}$ that corresponds to one of the strategies used in C's optimal mix (i.e. if $\mathbf{p} = (p_j)$ for $j = 1 \ldots n$ is C's optimal strategy, then $\mathbf{t}$ is some column k where $p_k \neq 0$), the value of the game is the vector product $\mathbf{q} \cdot \mathbf{t}$. Thus, in the above example, the game's value is $\frac{13}{3}$.

2.3.4 Domination

Another fairly simple method of solution relies on the following logic: a player should not use patently bad strategies. As obvious as this seems, this gives the player a way of shaving the number of strategies off of a very large game to make it much more manageable. In particular the logic works as follows: if every entry in a strategy is worse or equivalent to the entries in another strategy, then the former is called a dominated strategy and the latter is called a dominating strategy. We summarize these definitions more precisely in the following paragraph.

Let $\mathbf{S_1}$ and $\mathbf{S_2}$ be vectors that correspond to strategies of a player who wishes to maximize their profit (respectively, minimize their loss). If $\mathbf{S_1} \leq \mathbf{S_2}$ (respectively, $\mathbf{S_2} \leq \mathbf{S_1}$) then S_1 is called a dominated strategy and S_2 is called a dominating strategy.

For example, if $\mathbf{S_1} = \left(\begin{smallmatrix} 5 \\ 0 \\ 3 \end{smallmatrix}\right)$ and $\mathbf{S_2} = \left(\begin{smallmatrix} 7 \\ 2 \\ 3 \end{smallmatrix}\right)$, then $S_1 \leq S_2$. Thus $\mathbf{S_2}$ is a dominating strategy for the maximizing player, while $\mathbf{S_1}$ is a dominating strategy for the minimizing player.

Definition 4 (Domination Principle)**.** A player should never play dominated strategies.

The Domination Principle allows us to collapse games with domination into smaller forms. If rational players never use dominated strategies, then a game matrix with dominated strategies is equivalent to that same matrix with such inferior strategies removed. Thus when analyzed for dominance, the matrix given by Figure 8 can be amended by removing inferior strategies. Notice that removing dominated strategies yields a matrix equivalent to the one displayed in Figure 5. For rational players, or rather, those who obey the Dominance Principle, these two games are equivalent. Optimal strategies for both players are found by following the methods described in Section 2.3.3 (The Oddment Method) and Figures 6 and 7. The optimal strategy for R is to play I with probability $\frac{1}{6}$ and II with probability $\frac{5}{6}$. For C, the optimal strategy is to play I with probability $\frac{2}{6}$ and II with probability $\frac{4}{6}$.

		C			
		I	II	III	IV
R	I	1	6	2	6
	II	5	4	6	5

Figure 8: A 2×4 matrix with dominance. Notice that both III and IV are dominated by I.

3 Linear Programming: A More Powerful Solution

The methods of solution presented in Section 2.3 are simple and elegant. They permit the finding of optimal methods of play with a relatively modest investment of time and a reasonable tax on one's mental fortitude. Unfortunately, each of the three methods presented thus far are restricted to very specific circumstances. If we wish to solve games of a much broader scope, we must utilize more sophisticated mathematical tools. The field of *Operations Research* has developed the theory behind many methods of optimizing processes. Such methods are relevant to our study because in "solving" games, we are really looking to extremize the payoff associated with adopting certain strategies. One topic within Pperations Research, linear programming, is of particular relevance to the study of games. Linear programming is a way to model different processes or systems, and search for optimal solutions. The words "optimal" or "best" may be difficult to define, but in linear programming we define these terms not in subjective judgments but in the value of an objective function. When looking for optimal solutions, we are interested in solutions where the objective function is maximized or minimized, depending on which goal is desired.

A linear program as a model contains some standard characteristics. We state an objective function that depends on some number of variables. We also state whether we wish to maximize or minimize the objective function, thereby obtaining values for these variables that will constitute an optimal solution. To create a bound on the values within the solution, another important aspect is the set of constraints. These are generally a set of inequalities,

although equality constraints are not uncommon. Finally, as one might guess, it is required that the objective function and all constraints be linear in the variables. Violations of this final requirement do not necessarily make the program unsolvable. However, the techniques that we will describe will not avail the solver. You may recognize all of these aspects in this example of a linear program.

$$\begin{aligned} \text{Maximize} \quad f = \quad & 3x_1 + 2x_2 \\ \text{Subject to:} \qquad & 2x_1 + x_2 \leq 5 \\ & 5x_1 + 7x_2 \leq 13 \\ & x_1, x_2 \geq 0 \end{aligned}$$

Knowing the form and goals of a linear program, we may notice a relation between this model and our game-theoretic model. In both we are looking to choose strategies that maximize or minimize the payoff to a player. The optimal solution would thus be the mixed strategy that extremizes our objective function, the value of the game.

There are a few methods for solving a linear program, but one of the most useful is the Simplex Method. Given only two constraints we would be able to find an optimal value by graphing the functions and searching the corners of the geometric figure created by the inequalities. The Simplex Method is an algorithm for finding optimal solutions of linear programs that works by considering the set of constraints to be equations that describe the faces of a hypergeometric set. By searching the corner-points of this set, we can find an optimal solution.

Using the Simplex Method, there is some fundamental terminology to grasp. A **solution** is any set of values for the variables involved in the program. A **feasible solution** is a solution that satisfies all of the constraints, while an **infeasible solution** is one that does not. Within a solution, the variables that are given nonzero values are called **basic variables**, while the variables that are equivalently zero are called **nonbasic variables**. If we consider a program with n constraints, we shall see that the Simplex Method works by

choosing n basic variables, and setting these basic variables equal to their maximum feasible values. In this way, the Simplex Method is finding corner-points of the set in n-space.

We start by defining the standard form of a linear program that can be solved by Simplex. If $\mathbf{A}$ is an $m \times n$ matrix, $\mathbf{x}$ is the solution column vector of size n, $\mathbf{b}$ is the column vector of constraint values with size m, and $\mathbf{c}$ is the row vector describing the objective function with size n, then a linear program may be described by the system given by (1)

$$\begin{aligned} \text{Maximize} \quad & \mathbf{cx} \\ \text{Subject to} \quad & \mathbf{Ax} \leq \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned} \tag{1}$$

where we make two assumptions:

Nonnegativity: $b \geq 0$; and

Nondegeneracy: The rows of A represent exactly m linearly independent equations.

Before beginning with the method, we make one alteration to the program. Since the inequalities of b can be a bit problematic in finding solutions, we introduce a set of **slack variables** that take up the "slack" in the inequalities. These slack variables are introduced so that the above system (1) may be equivalently written as the following system (2),

$$\begin{aligned} \text{Maximize} \quad & \mathbf{cx} \\ \text{Subject to} \quad & \mathbf{Ax} - \mathbf{y} = \mathbf{b} \\ & \mathbf{x} \geq 0 \end{aligned} \tag{2}$$

where the vector $\mathbf{y}$ contains the slack variable for each constraint row.

A complete walkthrough of the Simplex algorithm is far beyond the scope of this paper. However, a few of the salient features will be discussed to give an idea of the ways in which the method works. First we set up a **tableau** of values that shall describe the

program. This is accomplished by creating the concatenated matrix

$$\begin{pmatrix} \mathbf{A} & \mathbf{b} \\ \mathbf{c} & 0 \end{pmatrix}$$

which would be a size $(m+1) \times (n+1)$ matrix. This tableau gives solutions to the program by listing out m basic variables. In this initial tableau, all of the slack variables are basic. Using a complex sequence of logic, the method works by choosing a nonbasic variable that would increase the objective function, then choosing a basic variable to leave the basis. The nonbasic variable enters the basis by pivoting on a the column of the entering nonbasic variable and the row of the leaving basic variable. This new matrix now has a different set of basic variables, with a different objective function value. The algorithm continues until one of many conditions is reached that will indicate that either the program is unbounded or that an optimal solution has been reached.

3.1 A Linear Program for Games

We begin by analyzing the game from the Row Player's perspective. Since the payoff matrix represents the payoffs to the Row Player, the goal of this player is to maximize their payoff. Thus, we are searching for a strategy that maximizes the value of the game for R. If we can find a set of constraints on this value, then we will have a linear program.

Let $\mathbf{A} = (a_{i,j})$ be the payoff matrix for the game, and suppose R has m, and C has n strategies to choose from. In searching for an optimal strategy for R, we are searching for a set of probabilities $p_1, \ldots p_m$ such that the mixed strategy column vector $\mathbf{p} = (p_1, \ldots, p_m)^T$ in some way maximizes R's expected winnings. Remember that we assume C is rational; thus C has already attempted to minimize the average loss across R's mixed strategy $\mathbf{p}$. In

other words C has attempted the following calculation:

$$\min_{j=1\dots n}\sum_{i=1}^{m}(p_i\cdot a_{i,j}) = \min_{j=1\dots n}(\mathbf{p^T}\cdot\mathbf{A})_j$$

where $(\mathbf{p^T}\cdot\mathbf{A})_j$ is the j^{th} element of the row vector $(\mathbf{p^T}\cdot\mathbf{A})$, which gives the expected payoff of using the Row Player's mixed strategy $\mathbf{p}$ against the Column player's strategy j. Thus R's task is the following program.

$$\begin{aligned} \text{Maximize} \quad & \min\nolimits_{j=1\dots n}(\mathbf{p^T}\cdot\mathbf{A})_j \\ \text{Subject to} \quad & \textstyle\sum_{i=1}^{m} p_i = 1 \\ & p_i \geq 0 \end{aligned}$$

Unfortunately, this is not a linear program due to the inclusion of the minimization operator. However, we can turn it into one by introducing a new variable v, restricting it to be less than the objective function, and then maximizing it. Notice that

$$v \leq \min_{j=1\dots n}(\mathbf{p^T}\cdot\mathbf{A})_j$$

is equivalent to

$$v \leq (\mathbf{p^T}\cdot\mathbf{A})_j \text{ for all } \; j = 1\dots n.$$

which allows us to create the following program.

Program 5. *The Row Player's linear program is given by*

$$\begin{aligned} \textit{Maximize} \quad & v \\ \textit{Subject to} \quad & \mathbf{v} - \mathbf{p^T}\cdot\mathbf{A} \leq \mathbf{e} \\ & \sum_{i=1}^{m} p_i = 1 \\ & p_i \geq 0 \textit{ for } \; i = 1\dots m \end{aligned}$$

where $\mathbf{v} = (v, \ldots, v)$ *and* $\mathbf{e} = (0, \ldots, 0)$ *are row vectors of size* n*,* $\mathbf{A}$ *is the game matrix, and* $\mathbf{p}$ *is the Row Player's optimal mixed strategy expressed as a column vector of size* m*.*

We may approach this problem similarly to find an optimal strategy for C. If this strategy is given by the column vector $\mathbf{q} = (q_1, \ldots, q_n)^T$, we assume that the following calculation has been made by R

$$\max_{i=1\ldots m} (\mathbf{Aq})_i.$$

In spite of R's efforts, C still wishes to minimize the losses, so much like the previous section we define w which is to be minimized, which is constrained from below by each element of $\mathbf{Aq}$. Thus C's linear program is given as follows.

Program 6. *The Column Player's linear program is given by*

$$\begin{aligned} &\textit{Minimize} && w \\ &\textit{Subject to} && \mathbf{Aq} - \mathbf{w} \leq \mathbf{f} \\ & && \sum_{i=j}^{n} q_i = 1 \\ & && q_j \geq 0 \textit{ for } \ j = 1 \ldots n \end{aligned}$$

where $\mathbf{w} = (w, \ldots, w)^T$ *and* $\mathbf{f} = (0, \ldots, 0)^T$ *are column vectors of size* m*,* $\mathbf{A}$ *is the game matrix, and* $\mathbf{q}$ *is the Column Player's optimal mixed strategy expressed as a column vector of size* n*.*

The programs Program 5 and Program 6 are similar, and yet markedly different. They are similar in that they make use of the same matrix, $\mathbf{A}$, to create constraints. Yet notice that in one program, the goal is to maximize the variable v, while in the other the goal is to minimize w. There is also some opposition in the constraints, featuring a $-\mathbf{p^T A}$ in Program 5 and $\mathbf{Aq}$ in Program 6. In linear programming, these are called **dual programs**. The following theorem gives an important property about dual programs.

Theorem 7 (The Duality Theorem of Linear Programming)**.** *If both of a pair of dual linear programs have feasible solutions, then both programs have optimal solutions with equal values for their objective functions. If either program is not feasible, then neither program has an optimal solution.*

Specifically, this implies that if both Program 5 and Program 6 have optimal solutions, then $v = w$. Note that this implies the results of Theorem 3, the Minimax Theorem.

3.2 An Example of a Game as a Linear Program

These linear programs are essential to our study of Colonel Blotto games, but their precise meaning can be lost in the vector and matrix notation. A linear program will be developed for a Colonel Blotto game, one that would not be easily solvable without such methods. Consider the Colonel Blotto game for $A = 4, D = 5, B = 2$ given in Figure 9. Attempts at

		D		
		5-0	4-1	3-2
	4-0	1/2	1/2	1
A	3-1	1	1/2	1/2
	2-2	1	1	0

Figure 9: The payoff matrix of a Colonel Blotto game for $A = 4, D = 5, B = 2$.

using any of the previous methods of solutions are bound to fail. However, formulating this game as a linear program allows the finding of a solution.

We proceed to find an optimal solution for the Attacker. This player wishes to utilize the different row strategies according to some probabilities $\mathbf{p^T} = (p_1, p_2, p_3)$. In choosing these probabilities, the player wishes to maximize the expected outcome of the game v. Letting $\mathbf{A}$ equal the game matrix given in Figure 9, we arrive at the linear program given

in Program 5. To make the algebra and reasoning clearer, we will rewrite Program 5 as the following system.

Program 8.

$$\begin{aligned}
&\textit{Maximize} \qquad && v && (3)\\
&\textit{Subject to} \qquad && v - \frac{1}{2}p_1 - p_2 - p_3 \leq 0 && (4)\\
& && v - \frac{1}{2}p_1 - \frac{1}{2}p_2 - p_3 \leq 0 && (5)\\
& && v - p_1 - \frac{1}{2}p_2 \leq 0 && (6)\\
& && p_1 + p_2 + p_3 = 1 && (7)\\
& && p_i \geq 0 \ \textit{for} \ \ i = 1, 2, 3 && (8)
\end{aligned}$$

These inequalities ensure that the Attacker's winnings will be at least v under each of the Defender's pure strategies. If the Defender plays the first column, then Equation (4) ensures this fact. If the Defender plays the second or third, then Equations (5) and (6) respectively ensure that the Attacker will win at least v. Because each of these equations has been enforced, all linear combinations of them are ensured as well. That is, no matter what mixed strategy the Defender employs, that linear combination of Equations (4) to (6) will also ensure that the Attacker wins at least v, a value that is maximized under the linear program.

The linear program for the Defender proceeds similarly. By choosing a mixed strategy according to probabilities $\mathbf{q^T} = (q_1, q_2, q_3)$, the Defender seeks to minimize the expected number of bases lost w. Letting $\mathbf{A}$ equal the game matrix given in Figure 9, we arrive at the following system.

Program 9.

$$
\begin{aligned}
&\textit{Minimize} && w && (9)\\
&\textit{Subject to} && \frac{1}{2}q_1 + \frac{1}{2}q_2 + q_3 - w \leq 0 && (10)\\
& && q_1 + \frac{1}{2}q_2 + \frac{1}{2}q_3 - w \leq 0 && (11)\\
& && q_1 + q_2 - w \leq 0 && (12)\\
& && q_1 + q_2 + q_3 = 1 && (13)\\
& && q_j \geq 0 \textit{ for } \; j = 1, 2, 3 && (14)
\end{aligned}
$$

Equations (10) to (12) ensure that the Defender's mixed strategy will lose no more than w bases for any of the Attacker's pure strategies, and therefore any Attacking mixed strategy will capture no more than w bases. Since the program minimizes w, we know that this Defending mixed strategy $\mathbf{q}$ is optimal.

3.3 Linear Programming in MATLAB

After obtaining linear programs for the Attacking and Defending players, the next step is to solve them. Colonel Blotto games will be solved using the MATLAB coding language, as there are packages that utilize the Simplex Algorithm to solve linear programs. Within the MATLAB Optimization Toolbox, the *linprog* function solves linear programs in the following form. Comparing Program 10 to both Program 5 and Program 6, we see that our statement of each player's program needs some manipulation before it can be solved using MATLAB.

Program 10. *The* linprog *function solves linear programs that appear in the following form:*

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{f^T x} \\ \textit{such that} \quad & \mathbf{Ax} \leq \mathbf{b} \\ & \mathbf{A_{eq} x} = \mathbf{b_{eq}} \\ & \mathbf{lb} \leq \mathbf{x} \end{aligned}$$

where $\mathbf{f}, \mathbf{x}, \mathbf{b}, \mathbf{b_{eq}}$, *and* $\mathbf{lb}$ *are column vectors and* $\mathbf{A}$ *and* $\mathbf{A_{eq}}$ *are matrices.*

We will begin with A's program given by Program 5. First, note that $\mathbf{x}$ represents the set of all variables. Thus $\mathbf{x^T} = (p_1, \ldots, p_m, v)$. To maximize v, we note that this is equivalent to minimizing $-v$, which implies that $\mathbf{f^T} = (0, \ldots, 0, -1)$. The first condition of Program 5 is $\mathbf{v} - \mathbf{p^T G} \leq \mathbf{e}$, where $\mathbf{G} = (g_{i,j})$ is the payoff matrix and $\mathbf{e}$ is a vector of zeros. We may create a condition equivalent to this in the form of Program 10 by creating the transposed and concatenated matrix

$$\mathbf{A} = \begin{pmatrix} -g_{1,1} & \cdots & -g_{m,1} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ -g_{1,n} & \cdots & -g_{m,n} & 1 \end{pmatrix}.$$

Then, if $\mathbf{b}$ is a size n vector of zeros, the first condition of Program 10 is met by

$$\mathbf{A} \cdot \mathbf{x} \leq \mathbf{b}.$$

The other conditions are fairly simple: $\mathbf{A_{eq}} = (1, \ldots, 1, 0)$, $\mathbf{lb} = (0, \ldots, 0, -\infty)$, and $\mathbf{b_{eq}} = 1$, where the first two vectors are of size $m+1$. These give the MATLAB program the last two

conditions of Program 5:

$$\sum_{i=1}^{m} p_i = 1$$

$$p_i \geq 0 \text{ for } \; i = 1 \ldots m.$$

With these definitions, the linear program in the form of Program 10 will give the Attacker's optimal strategy and the value of the game. The conditions will be restated for clarity.

Program 11. *The* linprog *function will optimize a game* $\mathbf{G} = (g_{i,j})$ *for the Attacker by putting it in the following form:*

$$
\begin{aligned}
\min_{\mathbf{x}} \quad & \mathbf{f^T x} \\
\textit{such that} \quad & \mathbf{Ax} \leq \mathbf{b} \\
& \mathbf{A_{eq} x} = \mathbf{b_{eq}} \\
& \mathbf{lb} \leq \mathbf{x}
\end{aligned}
$$

$$
\textit{where} \qquad \mathbf{A} = \begin{pmatrix} -g_{1,1} & \cdots & -g_{m,1} & 1 \\ \vdots & \ddots & \vdots & \vdots \\ -g_{1,n} & \cdots & -g_{m,n} & 1 \end{pmatrix}, \qquad \mathbf{x} = \begin{pmatrix} p_1 \\ \vdots \\ p_m \\ v \end{pmatrix} \qquad \mathbf{f} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ -1 \end{pmatrix}
$$

$$
\mathbf{A_{eq}} = (1, \ldots, 1, 0) \qquad \mathbf{lb^T} = (0, \ldots, 0, -\infty)
$$

$$
\mathbf{b_{eq}} = 1 \qquad \mathbf{b^T} = (0, \ldots, 0)
$$

Now we will derive D's program Program 6 so that it matches the form of Program 10. Much of this proceeds as A's program did. Since $\mathbf{x}$ represents all the variables, we have $\mathbf{x} = (q_1, \ldots, q_n, w)$ and $\mathbf{f^T} = (0, \ldots, 0, 1)$ with $\mathbf{x}$ and $\mathbf{f}$ both of size $n+1$. Because we want

$$\mathbf{Gq} - \mathbf{w} \leq \mathbf{f}$$

where $\mathbf{G} = (g_{i,j})$ is the payoff matrix and $\mathbf{f}$ is a vector of zeros, we create the concatenated matrix

$$\mathbf{A} = \begin{pmatrix} g_{1,1} & \cdots & g_{1,n} & -1 \\ \vdots & \ddots & \vdots & -1 \\ g_{m,1} & \cdots & g_{m,n} & -1 \end{pmatrix}.$$

Then, if $\mathbf{b}$ is a size m vector of zeros, the first condition of Program 6 is met by

$$\mathbf{Ax} \leq \mathbf{b}.$$

The other conditions are exactly the same as in A's program.

Program 12. *The* linprog *function will optimize a game* $\mathbf{G} = (g_{i,j})$ *for the Defender by putting it in the following form:*

$$\begin{aligned} \min_{\mathbf{x}} \quad & \mathbf{f^T x} \\ \textit{such that} \quad & \mathbf{Ax} \leq \mathbf{b} \\ & \mathbf{A_{eq}x} = \mathbf{b_{eq}} \\ & \mathbf{lb} \leq \mathbf{x} \end{aligned}$$

$$\textit{where} \qquad \mathbf{A} = \begin{pmatrix} g_{1,1} & \cdots & g_{1,n} & -1 \\ \vdots & \ddots & \vdots & -1 \\ g_{m,1} & \cdots & g_{m,n} & -1 \end{pmatrix} \qquad \mathbf{x} = \begin{pmatrix} q_1 \\ \vdots \\ q_n \\ w \end{pmatrix} \qquad \mathbf{f} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

$$\mathbf{A_{eq}} = (1, \ldots, 1, 0) \qquad \mathbf{lb^T} = (0, \ldots, 0, -\infty)$$

$$\mathbf{b_{eq}} = 1 \qquad \mathbf{b^T} = (0, \ldots, 0)$$

with $\mathbf{f}$, $\mathbf{A_{eq}}$ *and* $\mathbf{lb}$ *of size* $n+1$.

3.4 An Example of a Linear Program in MATLAB

Once again, an example of a particular Colonel Blotto game should be useful in understanding the process above. Consider once more the Colonel Blotto game of Section 3.2.

To create the Attacker's program, notice that we are attempting to minimize $\mathbf{f^T x} = -v$. The first condition described in the Attacker's MATLAB program is

$$\begin{pmatrix} -\frac{1}{2} & -1 & -1 & 1 \\ -\frac{1}{2} & -\frac{1}{2} & -1 & 1 \\ -1 & -\frac{1}{2} & 0 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ v \end{pmatrix} \geq \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}.$$

The other conditions are

$$\begin{pmatrix} 1 & 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ v \end{pmatrix} = 1$$

$$\begin{pmatrix} 0 \\ 0 \\ 0 \\ -\infty \end{pmatrix} \leq \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ v \end{pmatrix}$$

Notice the equivalence to Program 8. The Defender's program follows in a similar manner.

4 Our Model and Solutions

4.1 Full Statement of the Colonel Blotto Game

For convenience, we will restate the particulars of the Colonel Blotto game we will consider. Two players, an attacker A and defender D, are each interested in capturing a certain number of bases B. To do this, they have a certain number of armies at their disposal, which they must allocate among the bases. We shall denote the number of armies for A and D by A and D respectively. If A_i and D_i represent the number of armies allocated to base i by the attacker and defender respectively, with $i = 1 \ldots B$, then the attacker gains possession of base i exactly when $A_i > D_i$. Otherwise, when $A_i \leq D_i$, the defender gains possession of the base. The goal of each player is to maximize the number of bases that they capture.

The possible strategies that confront the players are the possible ways they may split their armies and send them among the bases. Since we assume that armies are discrete units, this means that the strategies for A and D consist of the different partitions of the integers A and D. Furthermore, because each base is valued equally by each player (the payoff for capturing any base is 1) the ordering of the troops between the bases can be neglected in enumerating the strategies. For example, if $A = 3$ and $B = 2$ then there are two different strategies for A: sending all armies to one base and none to another (the 3-0 strategy) or sending two armies to one base and one to the other (the 2-1 strategy). We do not need to include a 1-2 strategy, as we have counted this strategy in the group of "sending all armies to one base." We account for the possibility of permuting the armies among the bases by enumerating not the payoffs of using each strategy in the game matrix, but the expected payoffs. Treating each permutation as an equally likely random event, we average together all strategies that are equivalently reordered.

There are good reasons for doing this. For even small parameter values, the game matrices are fairly large. Even the simple game with 2 attackers, 2 defenders, and 2 bases contains a 3×3 game matrix with ordered strategies, as illustrated in Figure 10. Furthermore,

we would not be able to solve this matrix without linear programming.

		D			
		2-0	1-1	0-2	Solution
A	2-0	0	1	1	1/3
	1-1	1	0	1	1/3
	0-2	1	1	0	1/3
	Solution	1/3	1/3	1/3	

Figure 10: The matrix for a Colonel Blotto game with ordered strategies. Note the lack of saddle points or dominance and the size which prohibits the use of the Oddment Method.

On the other hand, the matrix with averaged strategies is easily solvable using the Oddment Method.

		D		
		2-0	1-1	Solution
A	2-0	1/2	1	2/3
	1-1	1	0	1/3
	Solution	2/3	1/3	

Figure 11: The matrix game using the same parameters as Figure 10, using averaged strategies. Notice the equivalence of the solutions in both figures.

Even with linear programming in our toolbox, formulating the game as being represented by smaller matrices cuts down on computational time — which would be of use when the parameters approach large values.

4.2 Calculating the Payoff Matrix

When searching for solutions, the first step involves the creation of the model of the game, the payoff matrix. To standardize the game matrices we will adopt the following conventions: payoffs will be given in terms of the expected number of bases captured by the Attacker, the strategies of whom will comprise the rows of the matrix. The strategies of the Defender will comprise the columns of the matrix.

In the Colonel Blotto game, we will be evaluating outcomes in terms of bases captured by each player. Therefore the payoff matrix for a Blotto Game will be populated by the expected number of bases captured by the Attacker. The entries also represent the losses by the Defender, as it is assumed that the bases belong to the Defender and are captured by the Attacker. The strategies for each player are the different ways of splitting the Attacking and Defending Armies respectively. To find the expected number of bases captured, we compare each partition of the Attacker's armies to each partition of the Defender's armies. If the number of attackers is strictly greater than the number of defenders, then the base is captured. We then add up the number of captured bases and divide this quantity by the number of bases to account for the different ways of permuting the armies.

Why does this algorithm work for calculating the average, or expected number of bases captured? If the partitions of both armies are assigned to bases in a uniformly random way, then each partition is assigned to a particular base with probability $\frac{1}{B}$. The expected value is the outcome of each partition being assigned to a base multiplied by the probability of that assignment occurring, summed over every possible permutation. That is, if X_i^k represents the outcome of the Attacker's i^{th} army being sent to the base where the Defender's k^{th} was allocated, then the expected outcome is

$$\sum_{i=1}^{B}\sum_{k=1}^{B}\frac{1}{B}X_i^k.$$

Thus, by comparing each of the Attacker's armies to each of the Defender's, summing the

outcomes of such pairings, and dividing by the number of bases, we are calculating the following expression,

$$\frac{1}{B}\sum_{i=1}^{B}\sum_{i=1}^{B} X_i^k \tag{15}$$

which is equivalent to the previous expression.

An example of a particular Colonel Blotto game might prove illuminating. Suppose there are 2 bases, 2 attackers, and 3 defenders. We see that there are two strategies for the Defender: send 3 armies to one base and 0 to another (3-0 strategy) or send 2 armies to one base and 1 to another (2-1 strategy). The Attacker faces a similar choice: 2 to one and 0 to another (2-0 strategy) or 1 to each (1-1 strategy). Thus we expect a 2 by 2 payoff matrix for this game. When the Attacker plays 2-0 and the Defender plays 3-0, the Attacker would win 1 base with the 2 armies and no bases with the 0 armies. The average for this set of strategies is thus $\frac{1+0}{2} = \frac{1}{2}$. This logic is similar if the Attacker played 2-0 against the Defender's 2-1. If the Attacker played 1-1 vs. the Defender's 3-0, then the each of the Attacker's armies would win only if it attacked the undefended base. The average for this set of strategies is $\frac{1+1}{2} = 1$. When the Attacker uses 1-1 against 2-1, there is no way for him to gain a base. The outcome for this set of strategies is 0. Thus our payoff matrix for this game is given by Figure 12, where the numbers along the top and left edge represent the different partition strategies for the Column and Row players respectively.

		D	
		3-0	2-1
A	2-0	1/2	1/2
	1-1	1	0

Figure 12: The payoff matrix for a Colonel Blotto game with $A = 2, D = 3, B = 2$.

Game matrices can be created by hand, using the this method and Equation (15).

However, this process, involving simple arithmetic and inequality logic can be easily done by a computer as well. While the game matrices were created by hand for small parameter values (that is, small numbers of Attacking and Defending forces), the matrices for larger parameters are populated automatically through coding.

A *ListStrats* function was created that takes the number of armies as an argument, and outputs the different strategies for each player as a matrix. The rows of the matrix give the different strategies with the different partitions ordered from largest to smallest. For example, a strategy output by the *ListStrats* function might be 4-2-1-0 but not 4-1-2-0. The function works by starting from an unpartitioned strategy (sending all armies to one base) and decrementing the numbers sent to that base. At each decrement, the different ways of sending the remaining armies to the remaining bases are enumerated through a recursive call to the function. The algorithm terminates when the strategy reaches an even split—when the decrementing number is at most the number of armies divided by the number of bases, rounded up. Interested readers may refer to Appendix C.1. This generates the different splitting strategies for any number of bases, without regard to which bases will be attacked or defended.

Using the MATLAB language, a *CreateGameMatrix* function was created to return a game matrix when the parameters of a particular game are input: the number of Attacking forces, the number of Defending forces, and the number of bases. The *ListStrats* function is called to enumerate the different strategies for each player. Once the strategies are enumerated, a game matrix of the appropriate size is initialized and populated by performing the summation (15) to find the expected number of bases captured by the Attacker for each pair of strategies. This process yields a game matrix given any number of parameters.

4.3 Solving the Colonel Blotto Games

Before linear programming was implemented, many games were solvable by hand using the methods described in Chapter 2. Using tools like Saddle Points, Dominance, and the

		D	
		2-0	1-1
	4-0	1	1
A	3-1	2	1
	2-2	1	2

Figure 13: The payoff matrix for a Colonel Blotto game with $A = 4, D = 2, B = 2$.

		D		
		2-0	1-1	Strategy
A	3-1	2	1	1/2
	2-2	1	2	1/2
	Strategy	1/2	1/2	

Figure 14: The game matrix equivalent to Figure 13, with the dominated row removed. This matrix is easily solved by the Oddment Method to arrive at the strategies shown.

Oddment Method, optimal strategies and values for the games were found. As the parameter values became larger however, the game matrices not only became larger, but irreducible by dominance as well. For example, the game in Figure 13 appears fairly large, but is reducible via the Dominance Principle into a more manageable form. Notice that the Attacker's 4-0 strategy is dominated by the other two strategies. Thus we can discard this strategy to arrive at a matrix solvable with the Oddment Method, shown in Figure 14.

However as the parameters increased, these solution methods became increasingly unhelpful. Consider the game with $A = 4$, $D = 4$, and $B = 2$, shown in Figure 15. An attempt to find saddle points ends only in dismay, as do futile searches for dominated strategies. The only way to solve this matrix is through linear programming.

A more robust method of finding solutions involved the use of linear programming.

		D		
		4-0	3-1	2-2
	4-0	1/2	1	1
A	3-1	1	1/2	1
	2-2	1	1	0

Figure 15: The payoff matrix for the Colonel Blotto game with $A = 4, D = 4, B = 2$. This game is unsolvable without linear programming.

Another function called *GameSolve* implemented the Programs 11 and 12 defined in Section 3.3. This *GameSolve* function takes as its argument the parameters of a Colonel Blotto game, passes these parameters to *CreateGameMatrix* to create the matrix, uses the *linprog* function to solve that matrix, and then outputs the optimal strategies of the Attacker and Defender along with the game's value.

4.4 Generating a Table

Large amounts of data were needed to search for patterns among these games and their solutions. The *GameSolve* function could solve a game for any given parameters, but it would be tedious and time-consuming to input these parameters directly into the function for a wide range of numbers. A much more efficient method involved the creation of an array that stored the solutions of Colonel Blotto games throughout a range of parameters. This array was called the BlottoTable, and the *BlottoTableCreator* function used for loops to populate this table for A and D parameters from 1 to 30 and for a B parameter of 2 to 10.

Once the BlottoTable was created, another function called the *BlottoTableWriter* took the data stored in BlottoTable and output different excel spreadsheets. These spreadsheets showed the values of Blotto games as cross sections of particular values of B, as they varied

with different parameters of A and D. Examples of these tables are printed in Appendix A.

4.5 Conversion of Values

These spreadsheets contain the values of Colonel Blotto games for different parameters. Unfortunately their formatting as decimal values can obscure certain patterns. For example, patterns involving rational numbers are easily seen moving diagonally downward with a slope of 1 in $B = 2$ when using these rational numbers. Compare Figures A.1 and A.9.

Many of the patterns present in these games are much more easily seen using rational numbers instead of decimals. The conversion of these decimal values into fractions was accomplished by using the *Rats* function supplied in MATLAB. A separate book of Excel spreadsheets was generated using these rational values.

Another transformation of these values was attempted to make these values more easily discernable. One method of analyzing the Colonel Blotto game is by examining it not through the numbers of troops deployed and bases captured, but through *ratios* of these numbers. Taking as an example the game shown in Figure 9, we may write the game as the following matrix:

		D		
		5-0	4-1	3-2
	4-0	1/4	1/4	1/2
A	3-1	1/2	1/4	1/4
	2-2	1/2	1/2	0

Figure 16: The payoff matrix of a Colonel Blotto game for $A = 4, D = 5, B = 2$ with payoffs given as ratios.

where the outcomes are not the expected number of bases captured, but the ratio of the bases captured to the total number of bases. That is, each element in the matrix is

divided by B. We can relate the values for different parameters in this way since the ratio, or if one prefers the "percentage," of bases captured can be compared across different B parameters. We can also divide the A and D parameters by the number of bases B. This allows us to compare across different levels of B the patterns that are affected by varying numbers of armies.

5 Results and Patterns

A number of patterns have been found in the Colonel Blotto game. Some of these patterns have been proved; others are conjectured. We will distinguish between these patterns as being either proven or non-proven in the following chapter.

For the following results we will be looking at patterns in the occurrence of values. These patterns depend upon the parameters of the Colonel Blotto game. Therefore, we shall use $V(A, D, B)$ to represent the value of the Blotto game with A attackers, D defenders, and B bases.

5.1 Results

5.1.1 Overwhelming Force

Theorem 13 (Overwhelming Attacking Force)**.** *When $A \geq B(D+1)$ the value of the Blotto game is B.*

Proof. Suppose the parameters of the Blotto game are such that the above inequality holds, and let $\mathcal{A}_i$ and $\mathcal{D}_i$ with $i = 1...B$ represent the number of armies assigned to the i-th base by the Attacker and Defender respectively. Since the parameters A, D, and B are integers, the above inequality implies that

$$\left\lfloor \frac{A}{B} \right\rfloor \geq D + 1.$$

Notice that the Attacker has at least one strategy that assigns at least $\lfloor \frac{A}{B} \rfloor$ armies to every base i. That is, $\mathcal{A}_i \geq \lfloor \frac{A}{B} \rfloor$. However, the Defending player can assign at most D armies to any base. That is, $\mathcal{D}_i \leq D$. Thus we have that

$$\mathcal{A}_i \geq \left\lfloor \frac{A}{B} \right\rfloor \geq D + 1 > D \geq \mathcal{D}_i,$$

which implies that

$$\mathcal{A}_i > \mathcal{D}_i \ \forall i = 1...B.$$

Therefore the Attacker will capture all bases, making the value of the Blotto game B. ■

Theorem 14 (Overwhelming Defending Force)**.** *When $D \geq AB$ the value of the Blotto game is 0.*

Proof. Suppose the parameters of the Blotto game are such that the above inequality holds, and following the notation and logic of the previous proof, we see that

$$\left\lfloor \frac{D}{B} \right\rfloor \geq A.$$

We see that the Defender has a strategy such that

$$\mathcal{D}_i \geq \left\lfloor \frac{D}{B} \right\rfloor$$

for all bases i, and that the Attacker's strategies are such that $\mathcal{A}_i \leq A$ for all bases i. This gives the inequality

$$\mathcal{D}_i \geq \left\lfloor \frac{D}{B} \right\rfloor \geq A \geq \mathcal{A}_i$$

which shows that $\mathcal{D}_i \geq \mathcal{A}_i \; \forall i = 1...B$. Therefore the Defender will capture all bases, making the value of the Blotto game 0. ■

5.1.2 Monotonicity

In the following section, we will prove that game values are monotonic with respect to the number of attacking and defending armies. That is, the game value is non-decreasing as the number of attacking armies increase, and non-increasing as the number of defending armies increase. This will be shown in two different steps: first, we will show that if the entries in a game matrix are non-decreasing, then the value is also non-decreasing; second we will show that adding a new row to a matrix will not decrease the value of the game; finally, we will show that increasing the number of armies of one player affects the game matrix in these ways, and will affect the value of the game monotonically.

Lemma 15. *Let two game matrices $G^* = (g^*_{ij})$ and $G = (g_{ij})$ be matrices of equal dimension, with values V^* and V respectively. If $g^*_{ij} \geq g_{ij}$ for all i and j, then $V^* \geq V$.*

Proof. Let $\mathbf{p^T}$ be the attacker's optimal strategy for G. Then from the given inequality, we have that

$$\mathbf{p^T} G^* \geq \mathbf{p^T} G.$$

If $\mathbf{p}^{*T}$ represent the Attacker's optimal strategy for G^*, then by definition of optimality we have that

$$\mathbf{p}^{*\mathbf{T}} G^* \geq \mathbf{p^T} G^*$$

and by combining these inequalities, we see that

$$\mathbf{p}^{*\mathbf{T}} G^* \geq \mathbf{p^T} G.$$

Similarly, we can multiply both sides of this last inequality by the Defender's optimal strategies $\mathbf{q}$ and $\mathbf{q}^*$. We arrive at

$$\mathbf{p}^{*\mathbf{T}} G^* \mathbf{q}^* \geq \mathbf{p^T} G \mathbf{q},$$

or equivalently $V^* \geq V$. ∎

Essentially, this shows that bounding a matrix by another matrix will bound that game's value by the value of the bounding matrix. Next we will show that adding an additional strategy for a player will not negatively affect the game's value from their perspective.

Lemma 16. *Adding a row to a matrix will not decrease the game's value, and adding a column to a matrix will not increase the game's value.*

Proof. Let G be a $m \times n$ game matrix, and G' be a $(m+1) \times n$ matrix where the first m rows are equivalently G.

Case 1. Suppose this additional strategy of G' is dominated by some other pure or mixed strategy. Then it should be used with probability 0, and the value of G' is equivalent

to the value of G.

Case 2. Suppose this additional strategy of G' is not dominated and should be used with some nonzero probability. Then using this row must ensure that $V' \geq V$, or else it would not be included in the optimal mixed strategy.

Therefore, adding a strategy for the Row Player will not decrease a game's value. The proof for the Column Player follows similarly. ■

Now we are able to make inferences about the values of games when their matrices behave in certain ways. In the following proposition, we will show that increasing the number of armies of either player will affect the matrices in these previously defined ways. This makes it possible to make inferences about the values of Colonel Blotto games, knowing only the parameters involved and not the underlying matrices that are generated by these parameters.

Lemma 17. *Suppose G is a $m \times n$ matrix generated from A attacking armies and D defending armies, and G' is generated from $A+1$ attacking and D defending armies. Then either:*
(1) G' is a $m \times n$ matrix with $G' \geq G$; or
(2) G' is a $(m+k) \times n$ matrix containing a $m \times n$ submatrix G'' where $G'' \geq G$.

Proof. Whether the matrix in question behaves according to (1) or (2) depends on the strategy sets generated by A and $A+1$. That is, if increasing the number of attacking armies by 1 does not change the number of partitions, then the matrix will behave according to (1). If this extra army gives the Attacker k more strategies, then the matrix will behave according to (2).

Case 1. Suppose the strategy set generated from $A+1$ armies is the same size as that generated from A armies. The number of armies allocated to base 1 will be higher in the former strategy set. This can be seen by comparing the strategies generated by 4 and 5 armies when $B = 2$, seen in Figure 17.

The algorithm that generates the game matrix involves an inequality comparison between each of the Attacking and Defending partitions. Having an equal or larger number

Allocation Strategies	
4 Armies	5 Armies
40	50
31	41
22	32

Figure 17: A table showing the different strategies for allocating 4 and 5 armies. Notice how the first number is always larger in the 5-army set compared to the 4-army set.

of forces in each partition would then increase the expected outcome of each strategy against each of the Defender's strategies. That is, the outcomes of the strategies using $A+1$ armies will be greater than or equal to the strategies using A armies. Thus, we arrive at the inequality $G' \geq G$.

Case 2. Suppose the strategy set $\mathbf{S}^{A+1}$ generated from $A+1$ armies will have k more strategies than the set $\mathbf{S}^{A}$ generated from A, and G' will have k more rows than G. However, comparing a subset of these partitioning strategies of G' to those of G yields a similar situation to that of part (1). For each strategy in $\mathbf{S}^{A}$, there is a corresponding strategy in $\mathbf{S}^{A+1}$ where the number of armies assigned to each base are equivalent, except for the first base in the list, which is greater by exactly 1.

An example should illuminate this final point. Compare the strategies for 5 and 6 armies when $B = 3$.

```
>> ListStrats(5,3)

ans =

     5     0     0

     4     1     0

     3     2     0

     3     1     1

     2     2     1

>> ListStrats(6,3)

ans =
```

```
    6    0    0
    5    1    0
    4    2    0
    4    1    1
    3    3    0
    3    2    1
    2    2    2
```

Notice that for every strategy in the 5 army set, there is a corresponding strategy in the 6 army set that is equivalent, with the only exception being that the first number in the strategy is larger. For example, we can find the following strategies where the strategies in the second set have only the first number in the set increased by 1.

```
From ListStrats(5,3):
    5    0    0
    4    1    0
    3    2    0
    3    1    1
    2    2    1


From ListStrats(6,3):
    6    0    0
    5    1    0
    4    2    0
    4    1    1
    3    2    1
```

By the logic of part (1), this means the outcomes in these corresponding rows of G' will be at least as large as the outcomes in G. If we define these rows of G' to be G'', then

we arrive at our desired result.

■

Lemma 18. *Suppose G is a $m \times n$ matrix generated from A attacking armies and D defending armies, and G' is generated from A attacking and $D+1$ defending armies. Then either:*
(1) G' is a $m \times n$ matrix with $G' \leq G$; or
(2) G' is a $m \times (n+k)$ matrix containing a $m \times n$ submatrix G'' where $G'' \leq G$.

Proof. Notice that Lemma 17 may be applied to increasing the number of Defending armies: simply change all of the $\geq$ signs to $\leq$. The algorithmic argument of (1) will then assert that a higher number of Defending armies at each base will ensure a smaller value in case (1). In the case of (2), a subset of strategies can be found in the same way, which by the previous argument would lead to a $G'' \leq G$. ■

Theorem 19 (Monotonicity of Values)**.** *Increasing the number of Attacking armies will not decrease the game's value. Increasing the number of Defending armies will not increase the game's value.*

Proof. From Lemma 17 we see that comparing the games with A and $A+1$ attacking armies will either result in (1) a matrix of the same size where each entry is at least as large as the previous matrix, or (2) a matrix with extra rows in which each entry of some submatrix is at least as large as those in the original matrix.

Case 1. From Lemma 15 we see that the matrix generated from the larger attacking force will have a value no smaller than the matrix generated from the smaller attacking force.

Case 2. From Lemma 15 we see that this submatrix has a value at least as large as the game with A attackers. From Lemma 16, we know that adding the extra row will not negatively affect the value for the attacker. Therefore, the game with the larger attacking force will have a value at least as large as the game with the smaller attacking force.

Therefore for every game with A attackers, D defenders, and B bases, the game with $A+1$ attackers, D defenders, and B bases will not have a smaller value. Therefore, the value of Colonel Blotto games are increasing with respect to the number of attackers.

These arguments may also apply to increasing the number of Defending armies by 1. The previous propositions ensure that the value of Colonel Blotto games are decreasing with respect to the number of defenders.

■

5.1.3 Special Cases

An interesting phenomenon occurs when $A = D+1$ and $B = 2$, that is, when the Attackers outnumber the Defenders by 1 in a 2-base game. In this case, the matrix for the game is populated entirely by ones. This means that no matter which strategy the Attacker or Defender chooses, the Attacker will always capture 1 base and the Defender will always retain 1 base. We may choose to call these **non-games**, as they are not influenced by the players' strategies at all but instead behave much more like a deterministic system. These non-games represent a point of "fairness" in our formulation of Colonel Blotto. The rules we have adopted regarding ties have favored the Defender. Thus, this pattern is a re-calibration of where equality exists between the two players. After all, if the Attacker is expected to win 1 base, the Defender is expected to win 1 base as well. For the skeptic reader, a proof is given after this phenomenon is formally stated.

Proposition 20 (Appearance of Non-Games)**.** *When $A = D+1$ and $B = 2$, the game matrix is populated entirely by ones. This is a non-game, as every possible strategy leads to an outcome of 1. Therefore for all integers n, $V(n+1, n, 2) = 1$.*

Proof. Suppose $A = D+1$ and $B = 2$. Let $\mathcal{B}_i = [a_i, d_i]$ represent the number of attacking and defending armies assigned to the ith base.

Case 1. Suppose $a_1 > d_1$, and notice that $a_1 = D+1-a_2$ and $d_1 = D - d_2$. From these equalities, we can see that $D+1-a_2 > D-d_1$ which implies that $1+d_2 > a_2$. Since

these values are all integers, this also implies that $d_2 \geq a_2$. Thus, the Attacker captures Base 1 and the Defender captures Base 2. The outcome of this game is 1.

Case 2. Suppose $a_1 \leq d_1$. Then $D+1-a_2 \leq D-d_2$, which implies that $d_2+1 \leq a_2$. Once again, this implies that $d_2 < a_2$. Thus the Defender captures Base 1 and the Attacker captures Base 2. The outcome of this game is 1.

Therefore, every entry in the game matrix will be populated by ones.

■

These non-games appear on a diagonal line given by $A = D + 1$ in the value table. Because the game values with 2 bases seem to interact about this diagonal of non-games, we will refer to it as the **main diagonal**.

5.2 Conjectures

The bulk of the search for patterns was focused on Blotto game values. The methods of the previous semester allowed us to input parameters into MATLAB and obtain optimal solutions to the corresponding Blotto game. This functionality was extended to generate a large array which would store this information, which could then be printed to an Excel book to aid in visualization of patterns. Some of these tables are printed in Appendix A.

The following patterns have been recognized from this endeavor.

5.2.1 The Diagonal Value Function for 2 Bases

Looking at the table of values for Blotto games using 2 bases and up to 30 attacking and defending armies, there are some noticable patterns. These tables are printed in Appendix A. The reader may notice patterns occurring above and below the main diagonal. By reading directly above the main diagonal for example, we find the pattern

$$\frac{1}{2}, \frac{2}{3}, \frac{3}{4}, \ldots$$

while reading the diagonal above this will reveal a related pattern:

$$\frac{1}{2}, \frac{1}{2}, \frac{2}{3}, \frac{2}{3}, \frac{3}{4}, \frac{3}{4}, \ldots$$

In general, the sequence of values d diagonals above the main diagonal contains the same elements as the sequence directly above it, with each element in the sequence repeated c times. This type of behavior also occurs below the main diagonal, the sequences 1 and 2 diagonals below respectively being:

$$\frac{3}{2}, \frac{4}{3}, \frac{5}{4}, \ldots$$

$$\frac{3}{2}, \frac{3}{2}, \frac{4}{3}, \frac{4}{3}, \frac{5}{4}, \frac{5}{4}, \ldots$$

This behavior is stated formally as follows.

Conjecture 21. *Let $V(A, D, B)$ be the value of the Colonel Blotto game with A attackers, D defenders, and B bases. Let $d = D + 1 - A$ be the number of diagonals above the main diagonal that the game $(A, D, 2)$ lies on. When the game lies below the main diagonal, d is a negative integer. Then*

$$V(n, n+d-1, 2) = 1 - \frac{1}{\left\lceil \frac{n+sgn(d)}{d} \right\rceil}$$

where $sgn(x)$ is the sign function defined by

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

We call this the ***Diagonal Value Function for 2 Bases****.*

A proof of this conjecture is elusive—by increasing or decreasing d while keeping A constant, the power differential between A and D increases. Each partition of D is either weakened or strengthened, which makes the game matrix behave according to complex patterns, which is exacerbated by the removal or addition of columns in the game matrix whenever D increases by 2. When $d = 1$ or $d = -1$, the matrices have a predictable structure that lends itself to finding an optimal strategy. However when $|d| > 1$, these structures become much harder to optimize in a general sense. The following output shows some of these palatable and not-so-palatable matrices and their corresponding solutions.

```
>> CreateGameMatrix(4,5,2)

ans =


    0.5000    0.5000    1.0000

    1.0000    0.5000    0.5000

    1.0000    1.0000         0

>> CreateGameMatrix(6,3,2)

ans =

    1.0000    1.0000

    1.5000    1.0000

    1.5000    1.5000

    1.0000    2.0000

>> CreateGameMatrix(6,8,2)

ans =


    0.5000    0.5000    0.5000    1.0000    1.0000

    1.0000    0.5000    0.5000    0.5000    1.0000

    1.0000    1.0000    0.5000    0.5000         0

    1.0000    1.0000    1.0000         0         0
```

```
>> CreateGameMatrix(6,4,2)

ans =

    1.0000    1.0000    1.0000
    1.5000    1.0000    1.0000
    1.0000    1.5000    1.0000
    1.0000    1.0000    2.0000

>> CreateGameMatrix(4,4,2)

ans =

    0.5000    1.0000    1.0000
    1.0000    0.5000    1.0000
    1.0000    1.0000         0
```

These last two matrices for $|d| = 1$ might be generalized in both structure and solution. A proof of the Diagonal Value Function for these cases follows. We will begin by proposing the structure of the matrices along each diagonal and find a general optimal solution. Then the Diagonal Value Function will be proved for each of these diagonals.

Proposition 22. *When $d = -1$, the game matrix will be of size $\lceil \frac{A+1}{2} \rceil \times \lceil \frac{A-1}{2} \rceil$ and have the following structure:*

$$\begin{pmatrix} 1 & 1 & 1 & \cdots & 1 \\ \frac{3}{2} & 1 & 1 & \cdots & 1 \\ 1 & \frac{3}{2} & 1 & \cdots & 1 \\ 1 & 1 & \ddots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \ddots & 1 \\ 1 & 1 & \cdots & 1 & x \end{pmatrix} \tag{16}$$

where $x = \begin{cases} \frac{3}{2} & \text{if } A \text{ is odd,} \\ 2 & \text{if } A \text{ is even.} \end{cases}$

Proof. Since $d = -1$, we have that $A = D + 2$. Let $\mathcal{A} = (a_1, a_2)$ represent an Attacking

strategy, where $a_1 \geq a_2$. Similarly, $\mathcal{D} = (d_1, d_2)$ represents a Defending strategy where $d_1 \geq d_2$. We will now find the entries in the game matrix given by A, D, and B. We will break this proof into four cases: $a_1 \leq d_1$, $a_1 = d_1 + 1$ and $d_1 > d_2$, $a_1 = d_1 + 1$ and $d_1 = d_2$, and $a_1 \geq d_1 + 2$.

Case 1. Suppose $a_1 \leq d_1$. This implies that $a_2 \leq a_1 \leq d_1$. Furthermore since $A > D$, we know that $a_1 - A < d_1 - D$, which implies

$$a_2 > d_2$$
$$a_1 \geq a_2 > d_2.$$

From these inequalities we see that the outcome for the pair of strategies that follow $a_1 \leq d_1$ will be $\frac{2}{2} = 1$.

Case 2. Suppose $a_1 = d_1 + 1$ and that A and D are both odd. Then $D + 2 - a_2 = D - d_2 + 1$, which implies that $a_2 = d_2 + 1$. Because these numbers of armies are both odd, then there does not exist a strategy where $d_1 = d_2$ i.e. there is no even split of an odd number. Then we may say that $d_1 > d_2$ which allows us to make the following inferences:

$$a_1 > d_2$$
$$d_1 \geq a_2$$

which gives an outcome of $\frac{3}{2}$.

Case 3. Suppose $a_1 = d_1 + 1$ and that A and D are both even. Then following the logic of the previous case, we have that $a_2 = d_2 + 1$. For any strategy obeying this first equality where the Defending armies are not evenly split, the outcome will be the same as in Case 2 since $d_1 > d_2$. However, when $a_1 = d_1 + 1$ and $d_1 = d_2$, then each of the Attacking

partitions will outnumber both of the Defending partitions by 1. That is,

$$a_1 = d_2 + 1$$

$$a_2 = d_1 + 1$$

which means that the expected outcome of this pair of strategies is 0.

Case 4. Suppose $a_1 \geq d_1 + 2 > d_1$. Then $a_1 - A \geq d_1 + 2 - A = d_1 - D$, which yields $a_2 \leq d_2$. Because $a_1 \geq a_2$ and $d_1 \geq d_2$ we have the following inequalities:

$$a_1 \leq a_2 \leq d_2$$

$$a_1 \geq d_1 > d_2$$

which yields an outcome of $\frac{2}{2} = 1$.

This defines the structure of a matrix when $d = -1$, based on the pairs of strategies of each entry. When the number of Attacking armies is A, the number of strategies available to that player is $\lceil \frac{A+1}{2} \rceil$. Because of this, and because D is exactly $A - 2$, the Defender will have exactly 1 less strategy than the Attacker. The Defender's number of strategies then is $\lceil \frac{A-1}{2} \rceil$. We may see this by comparing the strategy sets $\mathbf{S^A}$ and $\mathbf{S^D}$.

$$\mathbf{S^A} = \begin{pmatrix} A & 0 \\ A-1 & 1 \\ \vdots & \vdots \\ A - \lceil \frac{A-1}{2} \rceil & \lceil \frac{A-1}{2} \rceil \end{pmatrix} \qquad \mathbf{S^D} = \begin{pmatrix} A-2 & 0 \\ A-3 & 1 \\ \vdots & \vdots \\ A - \lceil \frac{A-1}{2} \rceil - 1 & \lceil \frac{A-1}{2} \rceil - 1 \end{pmatrix}$$

Thus by comparing each of these strategies, we arrive at a matrix given by Equation (16). ■

Proposition 23. *When a matrix is in the form of* (16)*, the value of this game is*

$$1 - \frac{1}{1-A}.$$

Proof. When the game matrix is in the form of Equation (16) we can find an optimal solution for both players. This solution is presented for both values of x.

Case 1. Suppose A and D are odd, with $x = \frac{3}{2}$. Then the first row of Equation (16) is dominated by every other strategy. By the Domination Principle, this means that this row should never be played. Instead, we deal with an $n \times n$ matrix with a diagonal of $\frac{3}{2}$ and 1 otherwise. The optimal strategy for both players is to play every strategy in this set with equal probability, yielding the strategy vectors

$$\mathbf{p^T} = \mathbf{q^T} = \left(\frac{1}{n}, \ldots, \frac{1}{n}\right).$$

If this is an optimal strategy, then the Attacker's maximin should equal the Defender's minimax. We see that it does, because the arithmetic for both players becomes

$$\frac{1}{n}\left(\frac{3}{2}\right) + \frac{n-1}{n}(1) = \frac{2n+1}{2n}.$$

Therefore both players assure themselves of a value no worse than the above expression, which upon substitution for $n = \lceil \frac{A-1}{2} \rceil$ yields the desired value:

$$\frac{2\lceil \frac{A-1}{2} \rceil + 1}{2\lceil \frac{A-1}{2} \rceil} = \frac{A-1+1}{A-1} = 1 + \frac{1}{A-1} = 1 - \frac{1}{1-A}.$$

Therefore the desired value is obtained when A is odd.

Case 2. Suppose A and D are even, with $x = 2$. From examining the solutions of matrices following these conditions, we can see a pattern emerging. Observe the following solution for $(A, D, B) = (4, 2, 2)$.

```
>>A=CreateGameMatrix(4,2,2)

    1.0000    1.0000

    1.5000    1.0000

    1.0000    2.0000

>>AtkStrat=BlottoTable(4,2,2,1)

AtkStrat =

         0

    0.6667

    0.3333

>>DefStrat=BlottoTable(4,2,2,2)

DefStrat =

    0.6667

    0.3333
```

We see the domination of the first row represented in the Attacker's strategy. Of this smaller matrix we see that the row and column with 1.5 in it should be used with twice the probability as the row or column with 2 in it. We may guess that this optimal strategy should share these characteristics with all matrices of this form, a hypothesis which will be tested as follows.

Suppose a game matrix is of the form given by Equation (16). Once again the first row is dominated and is therefore discarded. This smaller matrix has $n = \lceil \frac{A-1}{2} \rceil$ rows and columns. We hazard that the first $n-1$ rows and columns should each be used with twice the probability of the n^{th} row and column. Thus, we propose that the optimal strategies of both players are

$$\mathbf{p^T} = \mathbf{q^T} = \left(\frac{2}{2n-1}, \frac{2}{2n-1}, \ldots, \frac{1}{2n-1} \right).$$

The expected outcome for each player is found by multiplying this mixed strategy against any of the other player's good pure strategies. The arithmetic for both players then is one

of the following sums.

$$\frac{2(n-1)}{2n-1}(1) + \frac{1}{2n-1}(2) = \frac{2n}{2n-1}$$
$$\frac{2}{2n-1}(1.5) + \frac{2(n-2)}{2n-1}(1) + \frac{1}{2n-1}(1) = \frac{2n}{2n-1} \tag{17}$$

As in the last case, substituting in $n = \lceil \frac{A-1}{2} \rceil$ yields the desired value. Therefore the desired value is attained when A is even.

We have found optimal strategies for matrices along the diagonal $d = -1$ and shown the structure of their optimal solutions. These solutions guarantee that the value of the Blotto game will be $1 - \dfrac{1}{1-A}$. ■

Corollary 24 (Diagonal Value Function for $d = -1$)**.** *The Colonel Blotto game with $B = 2$ and $d = D + 1 - A = -1$, has a value V given by*

$$V(n, n-2, 2) = 1 - \frac{1}{\left\lceil \frac{n+sgn(-1)}{-1} \right\rceil}.$$

Proof. This follows from Propositions 22 and 23. The value of games along this diagonal is

$$1 - \frac{1}{1-A} = 1 - \frac{1}{\frac{A-1}{-1}} = 1 - \frac{1}{\lceil \frac{A-1}{-1} \rceil}$$

which is the desired result. ■

We will next examine the diagonal $d = 1$ in a similar fashion.

Proposition 25. *When $d = 1$, the game matrix will be of size $\lceil \frac{A+1}{2} \rceil \times \lceil \frac{A+1}{2} \rceil$ and have the*

following structure:

$$\begin{pmatrix} \frac{1}{2} & 1 & \cdots & 1 \\ 1 & \frac{1}{2} & \cdots & \vdots \\ \vdots & \vdots & \ddots & 1 \\ 1 & 1 & \cdots & x \end{pmatrix} \tag{18}$$

where $x = \begin{cases} \frac{1}{2} & \textit{if } A \textit{ is odd,} \\ 0 & \textit{if } A \textit{ is even.} \end{cases}$

Proof. Because $d = 1$, games along this diagonal will have that $A = D$. Let $\mathcal{A} = (a_1, a_2)$ and $\mathcal{D} = (d_1, d_2)$ represent the Attacker and Defender strategies, with $a_1 \geq a_2$ and $d_1 \geq d_2$.

Case 1. Suppose $a_1 = d_1$ and the strategy $\mathcal{A}$ is not evenly split. Then we have that $a_2 = d_2$ because $A = D$, and the uneven split implies that $a_1 > a_2$. We thus arrive at the following inequalities

$$a_1 > d_2$$
$$d_1 > a_2$$

which give an outcome of $\frac{1}{2}$.

Case 2. Suppose $a_1 = d_1$ and the strategy $\mathcal{A}$ is evenly split, implying that $a_1 = a_2$. Then by subtracting both sides from $A = D$, we obtain $a_2 = d_2$, which yields

$$a_2 = d_1$$
$$a_1 = d_2.$$

Thus the Attacker will win no bases and the outcome is 0.

Case 3. Suppose $a_1 > d_1$. Subtracting each side from $A = D$ yields $a_2 < d_2$. Because the first elements in each strategy vector are no smaller than the second elements, we arrive

at the following inequalities

$$a_1 > d_2$$
$$a_2 < d_1$$

which yields an outcome of 1.

Case 4. Suppose $a_1 < d_1$. Following the logic of the previous case, we subtract each side from the parameters A and D to yield $a_2 > d_2$ and obtain

$$a_1 > d_2$$
$$a_2 < d_1$$

yielding an outcome of 1.

Therefore, we obtain the matrix given by (18). ■

Proposition 26. *When a matrix is in the form of* (18)*, the value of this game is*

$$1 - \frac{1}{A+1}.$$

Proof. Notice that the matrix given by Equation (18) is very similar to the matrix given by Equation (16). In fact, these two matrices represent the same game, given from different perspectives. Since Colonel Blotto is a zero-sum game, subtracting each element of (16) from 2 yields the outcomes as payoffs to the Defender. Because the matrix with payoffs to the Defender is equivalent to (18), the solutions listed in the proof of Corollary 24 therefore apply to (18). The skeptic soul may verify this fact by performing the necessary arithmetic.

Case 1. When $A = D$ are odd, the matrix is symmetrical and the solution is

$$\mathbf{p^T} = \mathbf{q^T} = \left(\frac{1}{n}, \ldots, \frac{1}{n}\right)$$

where $n = \lceil \frac{A+1}{2} \rceil$ is the number of columns. Performing the arithmetic yields a value of

$$\frac{n-1}{n}(1) + \frac{1}{n}\left(\frac{1}{2}\right) = \frac{2n-1}{2n} = \frac{A+1-1}{A+1} = 1 - \frac{1}{A+1}.$$

These ceiling functions may be disregarded when simplifying because A is odd and the fraction within is a whole number. From this last expression we arrive at the desired result.

Case 2. When $A = D$ is even then $x = 0$ the solution is

$$\mathbf{p^T} = \mathbf{q^T} = \left(\frac{2}{2n-1}, \frac{2}{2n-1}, \ldots, \frac{1}{2n-1}\right).$$

The value is found using arithmetic similar to Equation (17), yielding

$$\frac{2(n-1)}{2n-1}(1) + \frac{1}{2n-1}(0) = \frac{2n-2}{2n-1} = 1 - \frac{1}{2n-1}$$

$$\frac{2}{2n-1}\left(\frac{1}{2}\right) + \frac{2(n-2)}{2n-1}(1) + \frac{1}{2n-1}(1) = \frac{2n-2}{2n-1} = 1 - \frac{1}{2n-1}.$$

Substituting in for our number of columns n, we obtain a value of

$$1 - \frac{1}{2\lceil \frac{A+1}{2} \rceil - 1} = 1 - \frac{1}{A+1}$$

which is our desired result.

We have found optimal strategies for matrices along the diagonal $d = 1$ and shown the structure of their optimal solutions. These solutions guarantee that the value of the Blotto game will be $1 - \dfrac{1}{A+1}$. ■

Corollary 27. *The Colonel Blotto game with $B = 2$ and $d = D + 1 - A = 1$, has a value V given by*

$$V(A, D, 2) = 1 - \frac{1}{\left\lceil \frac{A+sgn(1)}{1} \right\rceil}.$$

Proof. This follows from Propositions 25 and 26. The value of games along this diagonal is

$$1 - \frac{1}{A+1} = 1 - \frac{1}{\frac{A+1}{1}} = 1 - \frac{1}{\lceil \frac{A+1}{1} \rceil}$$

which is the desired result. ■

From the proof of these two Corollaries of the Diagonal Value Function, we see that the symmetry along the main diagonal in $B = 2$, arising from the zero-sum nature of the Colonel Blotto game, might simplify a proof of Conjecture 21. That is, a proof for any $d > 0$ might imply the results for $d < 0$ or vice versa.

5.2.2 Value Contours

When $B > 2$ very different types of patterns occur. Diagonal patterns occur, though these patterns are not the sequences of 2 bases. Instead, optimal values seem to occur along diagonals of varying slope. That is, instead of looking along various diagonals with a slope of 1, we must turn our attention to diagonals with different slopes. The value table for 4 bases should give the reader an indication of these patterns, though similar patterns exist for other numbers of bases. Readers may easily see these patterns by looking at Figure B.1.

The slopes of integer optimal values are easily recognized. In fact for the values of 0 and 4, these slopes were predicted by Theorem 14 (Overwhelming Defending Force), and Theorem 13 (Overwhelming Attacking Force) as $\frac{1}{B}$ and B respectively. The slopes of the other integer values have not been completely predicted. However, the symmetry of these slopes is noticeable especially in the contour plot of Figure B.1. The following conjectures should give an indication of this structure.

If n is a positive integer, we have the following conjectures:

Conjecture 28. $V(n, 3n-1, 4) = \frac{1}{2}$

Conjecture 29. $V(n, 2n-2, 4) = 1$

Conjecture 30. $V(3n, 4n-6, 4) = \frac{3}{2}$.

Conjecture 31. $V(4n+6, 3n, 4) = \frac{5}{2}$.

Conjecture 32. $V(2n+2, n, 4) = 3$

Conjecture 33. $V(3n+3, n, 4) = \frac{7}{2}$

Notice the symmetry in the value contours between Conjectures 28 to 30 and Conjectures 31 to 33. While there are definite patterns, a generalized formula for these patterns is yet to be found.

The symmetry of these value contours lies around a main diagonal. Recall that a main diagonal is a contour of values that represent "fairness" in the game, a contour where $V = \frac{B}{2}$. In higher numbers of bases, this main diagonal is different. Instead of having $V = \frac{B}{2}$ occur at every set of parameters along this diagonal, we see from the following conjecture that "fair" games do not occur as frequently as in 2 bases.

Conjecture 34. $V(2n+2, 2n, 4) = 2$.

This essentially means that values of 2 occur at every other game along this main diagonal. Furthermore in higher numbers of bases, a pattern similar to this occurs. The BlottoTable was populated up to 9 bases, and the following pattern holds. This would give the parameters needed for each player to win half of the bases.

Conjecture 35 (Parameters of Fairness)**.** *If λ and n are integers, then for $B = 2\lambda$, we can find the values $V[\lambda(n+1), \lambda n, B] = \lambda$.*

One implication of this conjecture is that games with odd numbers of bases cannot be fair. There appear to be no optimal strategies for both players to capture half of the possible bases, on average. This is supported by all of the BlottoTable data that has been created: there are no occurrences of $V = \frac{B}{2}$ in odd numbers of bases. The conjecture does seem to predict some asymptotically close values. For example when $B = 3$, we would expect

$V[\frac{3n+2}{2}, \frac{3n}{2}, 3] = \frac{3}{2}$ even though such parameters of A and D are not integers. The diagonal of $A = D + 2$ asymptotically approaches this value from below while $A = D + 3$ approaches it asymptotically above, since these two diagonals lie directly around the one predicted by Conjecture 35.

In Conjecture 31 we see a sufficient condition for the value of $\frac{5}{2}$ occurring in $B = 4$. However, this condition is not necessary; the value $V(5, 2, 4) = \frac{5}{2}$ which lies outside this diagonal. Thus it is unclear if these slopes break down in predictable ways.

5.2.3 Colonel Blotto as a Game of Ratios

Systematically comparing the values of Colonel Blotto games across many different numbers of bases B presents a challenge. For example, a value $V = 2$ has very different implications when $B = 2$ than when $B = 4$. The first case implies overwhelming force—the Attacker has captured all of the bases. In the second case, the Attacker has only captured half of the bases. Thinking about the values in this way deals not with the "expected number of bases captured," but the "expected ratio of bases captured." We might abstract this method of comparison to present a solution to this challenge. That is, we express the payoffs not as values V but as ratios $R = \frac{V}{B}$. When comparing outcomes as ratios across different numbers of bases, the results are much more meaningful.

This principal also applies to the A and D parameters as well; it would seem only natural to quantify the amount of force given to each player by considering the objectives. This is partially informed from the Overwhelming Force Theorems 13 and 14: the ratios of 1 and 0 depend on certain levels of Attacking or Defending forces being proportional to the number of bases. To account for this, we can also analyze the ratio R as a function of ratios of A and D. We may define these as $A_r = \frac{A}{B}$ and $D_r = \frac{D}{B}$, and search for the payoffs of the Colonel Blotto game $R(A_r, D_r, B)$.

This method presents some trends, though their analysis was not able to be completed in depth. A table of values may be found in [insert appendix]. Using ratios reveals some

structure in the patterns of value contours. That is, the slopes are symmetric about the ratio contour $R = \frac{1}{2}$. For example, comparing Conjectures 28 and 33 we see that the slopes obey a reciprocal relationship. Converting these value contours into ratio contours, we find that $R(\frac{n}{4}, \frac{3n-1}{4}, 4) = \frac{1}{8}$ and $R(\frac{3n+3}{4}, \frac{n}{4}, 4) = \frac{7}{8}$. From Conjectures 28 to 33 we can make the following conjecture.

Conjecture 36. *For some set of ratios r_i for $i = 1, \ldots, m$, the slope of the ratio contours of r_i and $1 - r_i$ have reciprocal values. That is, if*

$$r_i = R(an + c, bn + d, B)$$

then

$$1 - r_i = R(bn + j, an + k, B) \text{ for } i = 1, \ldots, m.$$

Finding the ratios r_i for which this is true could potentially yield important results in the study of Blotto games.

6 Future Work

There is still plenty to be done in searching for patterns among Colonel Blotto games. Finding a proof for the Diagonal Value Function would be important. Unfortunately, the matrices begin to look less palatable and the solutions are not as easily generalized as $|d|$ increases. A proof for this function might not be straightforward in the manner of Corollaries 24 and 27.

Instead, the proof of this function might come from examining other aspects of the Blotto game. For example, by looking at the value ratios across the different numbers of bases, enduring patterns may reveal themselves. In proving these patterns the Diagonal Value Function might be implied. Even if this endeavor does not yield a proof for this conjecture, the resulting patterns must be of some interest. Formulating the outcomes in this way might reveal more obscure patterns within the values.

Using value ratios might also benefit the study of these value contours as well. The slopes of these value contours might depend in part on the ratios that these value contours represent. This is partially informed from the symmetry of these slopes about the main diagonals for each number of bases, as seen in Conjectures 28 to 33. Finding instances of ratios that violate this symmetry might reveal other systematic patterns within the solutions.

This search for patterns has dealt mainly with values. For the most part, all optimal solutions have been treated equally—mainly out of necessity. The presence and possibility of multiple optimal solutions has been for the most part ignored, because the presence of multiple optimal solutions implies the presence of infinitely-many optimal solutions. There are many cases in applications where this may not be the case. For simplicity's sake, and because no definite application has been defined for this formulation of the game, we were not discriminate in selecting the form of the optimal solutions. Future work might take this into account.

A BlottoTables

A/D	1	2	3	4	5	6	7	8	9	10	11	12
1	0.5	0	0	0	0	0	0	0	0	0	0	0
2	1	0.666667	0.5	0	0	0	0	0	0	0	0	0
3	1.5	1	0.75	0.5	0.5	0	0	0	0	0	0	0
4	2	1.333333	1	0.8	0.666667	0.5	0.5	0	0	0	0	0
5	2	1.5	1.25	1	0.833333	0.666667	0.5	0.5	0.5	0	0	0
6	2	2	1.5	1.2	1	0.857143	0.75	0.666667	0.5	0.5	0.5	0
7	2	2	1.5	1.333333	1.166667	1	0.875	0.75	0.666667	0.5	0.5	0.5
8	2	2	2	1.5	1.333333	1.142857	1	0.888889	0.8	0.666667	0.666667	0.5
9	2	2	2	1.5	1.5	1.25	1.125	1	0.9	0.8	0.75	0.666667
10	2	2	2	2	1.5	1.333333	1.25	1.111111	1	0.909091	0.833333	0.75
11	2	2	2	2	1.5	1.5	1.333333	1.2	1.1	1	0.916667	0.833333
12	2	2	2	2	2	1.5	1.5	1.333333	1.2	1.090909	1	0.923077
13	2	2	2	2	2	1.5	1.5	1.333333	1.25	1.166667	1.083333	1
14	2	2	2	2	2	2	1.5	1.5	1.333333	1.25	1.166667	1.076923
15	2	2	2	2	2	2	1.5	1.5	1.5	1.333333	1.25	1.142857
16	2	2	2	2	2	2	2	1.5	1.5	1.333333	1.333333	1.2
17	2	2	2	2	2	2	2	1.5	1.5	1.5	1.333333	1.25
18	2	2	2	2	2	2	2	2	1.5	1.5	1.5	1.333333
19	2	2	2	2	2	2	2	2	1.5	1.5	1.5	1.333333
20	2	2	2	2	2	2	2	2	2	1.5	1.5	1.5
21	2	2	2	2	2	2	2	2	2	1.5	1.5	1.5
22	2	2	2	2	2	2	2	2	2	2	1.5	1.5
23	2	2	2	2	2	2	2	2	2	2	1.5	1.5
24	2	2	2	2	2	2	2	2	2	2	2	1.5
25	2	2	2	2	2	2	2	2	2	2	2	1.5
26	2	2	2	2	2	2	2	2	2	2	2	2
27	2	2	2	2	2	2	2	2	2	2	2	2
28	2	2	2	2	2	2	2	2	2	2	2	2
29	2	2	2	2	2	2	2	2	2	2	2	2
30	2	2	2	2	2	2	2	2	2	2	2	2

Figure A.1: Table of values for $B = 2$, given in decimal format.

A/D	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	0.5	0	0	0	0	0	0	0	0	0	0	0
8	0.5	0.5	0.5	0	0	0	0	0	0	0	0	0
9	0.5	0.5	0.5	0.5	0.5	0	0	0	0	0	0	0
10	0.666667	0.666667	0.5	0.5	0.5	0.5	0.5	0	0	0	0	0
11	0.75	0.666667	0.666667	0.5	0.5	0.5	0.5	0.5	0.5	0	0	0
12	0.857143	0.8	0.75	0.666667	0.666667	0.5	0.5	0.5	0.5	0.5	0.5	0
13	0.928571	0.857143	0.8	0.75	0.666667	0.666667	0.5	0.5	0.5	0.5	0.5	0.5
14	1	0.933333	0.875	0.8	0.75	0.666667	0.666667	0.666667	0.5	0.5	0.5	0.5
15	1.071429	1	0.9375	0.875	0.833333	0.75	0.75	0.666667	0.666667	0.5	0.5	0.5
16	1.142857	1.066667	1	0.941176	0.888889	0.833333	0.8	0.75	0.666667	0.666667	0.666667	0.5
17	1.2	1.125	1.0625	1	0.944444	0.888889	0.833333	0.8	0.75	0.666667	0.666667	0.666667
18	1.25	1.2	1.125	1.058824	1	0.947368	0.9	0.857143	0.8	0.75	0.75	0.666667
19	1.333333	1.25	1.166667	1.111111	1.055556	1	0.95	0.9	0.857143	0.8	0.75	0.75
20	1.333333	1.333333	1.25	1.166667	1.111111	1.052632	1	0.952381	0.909091	0.857143	0.833333	0.8
21	1.5	1.333333	1.25	1.2	1.166667	1.1	1.05	1	0.954545	0.909091	0.875	0.833333
22	1.5	1.333333	1.333333	1.25	1.2	1.142857	1.1	1.047619	1	0.956522	0.916667	0.875
23	1.5	1.5	1.333333	1.333333	1.25	1.2	1.142857	1.090909	1.045455	1	0.958333	0.916667
24	1.5	1.5	1.5	1.333333	1.333333	1.25	1.2	1.142857	1.090909	1.043478	1	0.96
25	1.5	1.5	1.5	1.333333	1.333333	1.25	1.25	1.166667	1.125	1.083333	1.041667	1
26	1.5	1.5	1.5	1.5	1.333333	1.333333	1.25	1.2	1.166667	1.125	1.083333	1.04
27	1.5	1.5	1.5	1.5	1.5	1.333333	1.333333	1.25	1.2	1.166667	1.125	1.076923
28	2	1.5	1.5	1.5	1.5	1.333333	1.333333	1.333333	1.25	1.2	1.166667	1.111111
29	2	1.5	1.5	1.5	1.5	1.5	1.333333	1.333333	1.25	1.25	1.2	1.142857
30	2	2	1.5	1.5	1.5	1.5	1.5	1.333333	1.333333	1.25	1.25	1.2

Figure A.2: Continuation of Figure A.1.

A/D	1	2	3	4	5	6	7	8	9	10	11	12
1	0.666667	0.333333	0	0	0	0	0	0	0	0	0	0
2	1.333333	0.888889	0.666667	0.444444	0.333333	0	0	0	0	0	0	0
3	2	1.333333	1	0.833333	0.666667	0.5	0.333333	0.333333	0	0	0	0
4	2.333333	1.777778	1.333333	1.111111	0.933333	0.8	0.666667	0.533333	0.444444	0.333333	0.333333	0
5	2.666667	2.111111	1.666667	1.388889	1.166667	1	0.888889	0.777778	0.666667	0.555556	0.444444	0.333333
6	3	2.333333	2	1.666667	1.4	1.2	1.066667	0.952381	0.857143	0.761905	0.666667	0.571429
7	3	2.555556	2.166667	1.888889	1.633333	1.4	1.244444	1.111111	1	0.916667	0.833333	0.75
8	3	2.666667	2.333333	2.066667	1.833333	1.6	1.422222	1.269841	1.142857	1.047619	0.962963	0.888889
9	3	3	2.5	2.2	2	1.8	1.6	1.428571	1.285714	1.178571	1.083333	1
10	3	3	2.666667	2.333333	2.111111	1.933333	1.755556	1.587302	1.428571	1.309524	1.203704	1.111111
11	3	3	2.666667	2.466667	2.222222	2.047619	1.888889	1.730159	1.571429	1.440476	1.324074	1.222222
12	3	3	3	2.555556	2.333333	2.142857	2	1.857143	1.714286	1.571429	1.444444	1.333333
13	3	3	3	2.666667	2.444444	2.238095	2.083333	1.952381	1.821429	1.690476	1.564815	1.444444
14	3	3	3	2.666667	2.555556	2.333333	2.166667	2.037037	1.916667	1.796296	1.675926	1.555556
15	3	3	3	3	2.666667	2.428571	2.25	2.111111	2	1.888889	1.777778	1.666667
16	3	3	3	3	2.666667	2.5	2.333333	2.185185	2.066667	1.962963	1.859259	1.755556
17	3	3	3	3	2.666667	2.555556	2.416667	2.259259	2.133333	2.030303	1.933333	1.836364
18	3	3	3	3	3	2.666667	2.5	2.333333	2.2	2.090909	2	1.909091
19	3	3	3	3	3	2.666667	2.555556	2.407407	2.266667	2.151515	2.055556	1.969697
20	3	3	3	3	3	2.666667	2.666667	2.466667	2.333333	2.212121	2.111111	2.025641
21	3	3	3	3	3	3	2.666667	2.555556	2.4	2.272727	2.166667	2.076923
22	3	3	3	3	3	3	2.666667	2.555556	2.466667	2.333333	2.222222	2.128205
23	3	3	3	3	3	3	2.666667	2.666667	2.5	2.393939	2.277778	2.179487
24	3	3	3	3	3	3	3	2.666667	2.555556	2.444444	2.333333	2.230769
25	3	3	3	3	3	3	3	2.666667	2.666667	2.5	2.388889	2.282051
26	3	3	3	3	3	3	3	2.666667	2.666667	2.555556	2.444444	2.333333
27	3	3	3	3	3	3	3	3	2.666667	2.555556	2.5	2.384615
28	3	3	3	3	3	3	3	3	2.666667	2.666667	2.555556	2.428571
29	3	3	3	3	3	3	3	3	2.666667	2.666667	2.555556	2.466667
30	3	3	3	3	3	3	3	3	3	2.666667	2.666667	2.5

Figure A.3: Table of values for $B = 3$, given in decimal format.

A/D	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0.333333	0.333333	0	0	0	0	0	0	0	0	0	0
6	0.5	0.444444	0.333333	0.333333	0.333333	0	0	0	0	0	0	0
7	0.666667	0.583333	0.5	0.444444	0.333333	0.333333	0.333333	0.333333	0	0	0	0
8	0.814815	0.740741	0.666667	0.592593	0.533333	0.444444	0.444444	0.333333	0.333333	0.333333	0.333333	0
9	0.933333	0.866667	0.8	0.733333	0.666667	0.6	0.533333	0.5	0.444444	0.333333	0.333333	0.333333
10	1.037037	0.969697	0.909091	0.848485	0.787879	0.727273	0.666667	0.606061	0.555556	0.5	0.444444	0.444444
11	1.140741	1.066667	1	0.944444	0.888889	0.833333	0.777778	0.722222	0.666667	0.611111	0.555556	0.5
12	1.244444	1.163636	0.75	1.030303	0.974359	0.923077	0.871795	0.820513	0.769231	0.717949	0.666667	0.615385
13	1.348148	1.260606	1.181818	1.116162	1.055556	1	0.952381	0.904762	0.857143	0.809524	0.761905	0.714286
14	1.451852	1.357576	1.272727	1.20202	1.136752	1.076923	1.025641	0.977778	0.933333	0.888889	0.844444	0.8
15	1.555556	1.454545	1.363636	1.287879	1.217949	1.153846	1.098901	1.047619	1	0.958333	0.916667	0.875
16	1.651852	1.551515	1.454545	1.373737	1.299145	1.230769	1.172161	1.11746	1.066667	1.022222	0.980392	0.941176
17	1.739394	1.642424	1.545455	1.459596	1.380342	1.307692	1.245421	1.187302	1.133333	1.086111	1.041667	1
18	1.818182	1.727273	1.636364	1.545455	1.461538	1.384615	1.318681	1.257143	1.2	1.15	1.102941	1.058824
19	1.883838	1.79798	1.712121	1.626263	1.542735	1.461538	1.391941	1.326984	1.266667	1.213889	1.164216	1.117647
20	1.944444	1.863248	1.782051	1.700855	1.619658	1.538462	1.465201	1.396825	1.333333	1.277778	1.22549	1.176471
21	2	1.923077	1.846154	1.769231	1.692308	1.615385	1.538462	1.466667	1.4	1.341667	1.286765	1.235294
22	2.047619	1.974359	1.901099	1.827839	1.754579	1.681319	1.608059	1.536508	1.466667	1.405556	1.348039	1.294118
23	2.095238	2.022222	1.952381	1.88254	1.812698	1.742857	1.673016	1.603175	1.533333	1.469444	1.409314	1.352941
24	2.142857	2.066667	2	1.933333	1.866667	1.8	1.733333	1.666667	1.6	1.533333	1.470588	1.411765
25	2.190476	2.111111	2.041667	1.977778	1.913889	1.85	1.786111	1.722222	1.658333	1.594444	1.531863	1.470588
26	2.238095	2.155556	2.083333	2.019608	1.958333	1.897059	1.835784	1.77451	1.713235	1.651961	1.590686	1.529412
27	2.285714	2.2	2.125	2.058824	2	1.941176	1.882353	1.823529	1.764706	1.705882	1.647059	1.588235
28	2.333333	2.244444	2.166667	2.098039	2.037037	1.980392	1.923747	1.867102	1.810458	1.753813	1.697168	1.640523
29	2.380952	2.288889	2.208333	2.137255	2.074074	2.017544	1.962963	1.908382	1.853801	1.79922	1.744639	1.690058
30	2.428571	2.333333	2.25	2.176471	2.111111	2.052632	2	1.947368	1.894737	1.842105	1.789474	1.736842

Figure A.4: Continuation of Figure A.3.

A/D	1	2	3	4	5	6	7	8	9	10	11	12
1	0.75	0.5	0.25	0	0	0	0	0	0	0	0	0
2	1.5	1	0.833333	0.666667	0.5	0.333333	0.25	0	0	0	0	0
3	2.25	1.5	1.25	1	0.875	0.75	0.625	0.5	0.375	0.25	0.25	0
4	3	2	1.666667	1.333333	1.166667	1	0.9	0.8	0.7	0.6	0.5	0.4
5	3.25	2.5	2.083333	1.666667	1.458333	1.25	1.125	1	0.916667	0.833333	0.75	0.666667
6	3.5	3	2.5	2	1.75	1.5	1.35	1.2	1.1	1	0.928571	0.857143
7	3.75	3.166667	2.75	2.333333	2.041667	1.75	1.575	1.4	1.283333	1.166667	1.083333	1
8	4	3.333333	3	2.666667	2.333333	2	1.8	1.6	1.466667	1.333333	1.238095	1.142857
9	4	3.5	3.125	2.833333	2.541667	2.25	2.025	1.8	1.65	1.5	1.392857	1.285714
10	4	3.666667	3.25	3	2.75	2.5	2.25	2	1.833333	1.666667	1.547619	1.428571
11	4	3.75	3.375	3.1	2.875	2.65	2.425	2.2	2.016667	1.833333	1.702381	1.571429
12	4	4	3.5	3.2	3	2.8	2.6	2.4	2.2	2	1.857143	1.714286
13	4	4	3.625	3.3	3.083333	2.9	2.716667	2.533333	2.35	2.166667	2.011905	1.857143
14	4	4	3.75	3.4	3.166667	3	2.833333	2.666667	2.5	2.333333	2.166667	2
15	4	4	3.75	3.5	3.25	3.071429	2.916667	2.761905	2.607143	2.452381	2.297619	2.142857
16	4	4	4	3.6	3.333333	3.142857	3	2.857143	2.714286	2.571429	2.428571	2.285714
17	4	4	4	3.666667	3.416667	3.214286	3.0625	2.928571	2.794643	2.660714	2.526786	2.392857
18	4	4	4	3.75	3.5	3.285714	3.125	3	2.875	2.75	2.625	2.5
19	4	4	4	3.75	3.583333	3.357143	3.1875	3.055556	2.9375	2.819444	2.701389	2.583333
20	4	4	4	4	3.666667	3.428571	3.25	3.111111	3	2.888889	2.777778	2.666667
21	4	4	4	4	3.75	3.5	3.3125	3.166667	3.05	2.944444	2.838889	2.733333
22	4	4	4	4	3.75	3.571429	3.375	3.222222	3.1	3	2.9	2.8
23	4	4	4	4	3.75	3.625	3.4375	3.277778	3.15	3.045455	2.95	2.854545
24	4	4	4	4	4	3.666667	3.5	3.333333	3.2	3.090909	3	2.909091
25	4	4	4	4	4	3.75	3.5625	3.388889	3.25	3.136364	3.041667	2.954545
26	4	4	4	4	4	3.75	3.625	3.444444	3.3	3.181818	3.083333	3
27	4	4	4	4	4	3.75	3.666667	3.5	3.35	3.227273	3.125	3.038462
28	4	4	4	4	4	4	3.75	3.555556	3.4	3.272727	3.166667	3.076923
29	4	4	4	4	4	4	3.75	3.6	3.45	3.318182	3.208333	3.115385
30	4	4	4	4	4	4	3.75	3.666667	3.5	3.363636	3.25	3.153846

Figure A.5: Table of values for $B = 4$, given in decimal format.

A/D	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0.333333	0.25	0.25	0	0	0	0	0	0	0	0	0
5	0.583333	0.5	0.416667	0.333333	0.25	0.25	0.25	0	0	0	0	0
6	0.785714	0.714286	0.642857	0.571429	0.5	0.428571	0.375	0.333333	0.25	0.25	0.25	0
7	0.9375	0.875	0.8125	0.75	0.6875	0.625	0.5625	0.5	0.4375	0.375	0.333333	0.25
8	1.071429	1	0.944444	0.888889	0.833333	0.777778	0.722222	0.666667	0.611111	0.555556	0.5	0.444444
9	1.205357	1.125	1.0625	1	0.95	0.9	0.85	0.8	0.75	0.7	0.65	0.6
10	1.339286	1.25	1.180556	1.111111	1.055556	1	0.954545	0.909091	0.863636	0.818182	0.772727	0.727273
11	1.473214	1.375	1.298611	1.222222	1.161111	1.1	1.05	1	0.958333	0.916667	0.875	0.833333
12	1.607143	1.5	0.75	1.333333	1.266667	1.2	1.145455	1.090909	1.045455	1	0.961538	0.923077
13	1.741071	1.625	1.534722	1.444444	1.372222	1.3	1.240909	1.181818	1.132576	1.083333	1.041667	1
14	1.875	1.75	1.652778	1.555556	1.477778	1.4	1.336364	1.272727	1.219697	1.166667	1.121795	1.076923
15	2.008929	1.875	1.770833	1.666667	1.583333	1.5	1.431818	1.363636	1.306818	1.25	1.201923	1.153846
16	2.142857	2	1.888889	1.777778	1.688889	1.6	1.527273	1.454545	1.393939	1.333333	1.282051	1.230769
17	2.258929	2.125	2.006944	1.888889	1.794444	1.7	1.622727	1.545455	1.481061	1.416667	1.362179	1.307692
18	2.375	2.25	2.125	2	1.9	1.8	1.718182	1.636364	1.568182	1.5	1.442308	1.384615
19	2.465278	2.347222	2.229167	2.111111	2.005556	1.9	1.813636	1.727273	1.655303	1.583333	1.522436	1.461538
20	2.555556	2.444444	2.333333	2.222222	2.111111	2	1.909091	1.818182	1.742424	1.666667	1.602564	1.538462
21	2.627778	2.522222	2.416667	2.311111	2.205556	2.1	2.004545	1.909091	1.829545	1.75	1.682692	1.615385
22	2.7	2.6	2.5	2.4	2.3	2.2	2.1	2	1.916667	1.833333	1.762821	1.692308
23	2.759091	2.663636	2.568182	2.472727	2.377273	2.281818	2.186364	2.090909	2.003788	1.916667	1.842949	1.769231
24	2.818182	2.727273	2.636364	2.545455	2.454545	2.363636	2.272727	2.181818	2.090909	2	1.923077	1.846154
25	2.867424	2.780303	2.693182	2.606061	2.518939	2.431818	2.344697	2.257576	2.170455	2.083333	2.003205	1.923077
26	2.916667	2.833333	2.75	2.666667	2.583333	2.5	2.416667	2.333333	2.25	2.166667	2.083333	2
27	2.958333	2.878205	2.798077	2.717949	2.637821	2.557692	2.477564	2.397436	2.317308	2.237179	2.157051	2.076923
28	3	2.923077	2.846154	2.769231	2.692308	2.615385	2.538462	2.461538	2.384615	2.307692	2.230769	2.153846
29	3.035714	2.961538	2.887363	2.813187	2.739011	2.664835	2.590659	2.516484	2.442308	2.368132	2.293956	2.21978
30	3.071429	3	2.928571	2.857143	2.785714	2.714286	2.642857	2.571429	2.5	2.428571	2.357143	2.285714

Figure A.6: Continuation of Figure A.5.

A/D	1	2	3	4	5	6	7	8	9	10	11	12
1	0.8	0.6	0.4	0.2	0	0	0	0	0	0	0	0
2	1.6	1.2	0.933333	0.8	0.666667	0.533333	0.4	0.266667	0.2	0	0	0
3	2.4	1.8	1.4	1.2	1	0.9	0.8	0.7	0.6	0.5	0.4	0.3
4	3.2	2.4	1.866667	1.6	1.333333	1.2	1.066667	0.96	0.88	0.8	0.72	0.64
5	4	3	2.333333	2	1.666667	1.5	1.333333	1.2	1.1	1	0.933333	0.866667
6	4.2	3.4	2.8	2.4	2	1.8	1.6	1.44	1.32	1.2	1.12	1.04
7	4.4	3.8	3.266667	2.8	2.333333	2.1	1.866667	1.68	1.54	1.4	1.306667	1.213333
8	4.6	4.066667	3.6	3.133333	2.666667	2.4	2.133333	1.92	1.76	1.6	1.493333	1.386667
9	4.8	4.2	3.8	3.4	3	2.7	2.4	2.16	1.98	1.8	1.68	1.56
10	5	4.333333	4	3.666667	3.333333	3	2.666667	2.4	2.2	2	1.866667	1.733333
11	5	4.466667	4.1	3.8	3.5	3.2	2.9	2.64	2.42	2.2	2.053333	1.906667
12	5	4.6	4.2	3.933333	3.666667	3.4	3.133333	2.88	2.64	2.4	2.24	2.08
13	5	4.733333	4.3	4.04	3.8	3.56	3.32	3.08	2.84	2.6	2.426667	2.253333
14	5	4.8	4.4	4.12	3.9	3.68	3.46	3.24	3.02	2.8	2.613333	2.426667
15	5	5	4.5	4.2	4	3.8	3.6	3.4	3.2	3	2.8	2.6
16	5	5	4.6	4.28	4.066667	3.88	3.693333	3.506667	3.32	3.133333	2.946667	2.76
17	5	5	4.7	4.36	4.133333	3.96	3.786667	3.613333	3.44	3.266667	3.093333	2.92
18	5	5	4.8	4.44	4.2	4.028571	3.866667	3.704762	3.542857	3.380952	3.219048	3.057143
19	5	5	4.8	4.52	4.266667	4.085714	3.933333	3.780952	3.628571	3.47619	3.32381	3.171429
20	5	5	5	4.6	4.333333	4.142857	4	3.857143	3.714286	3.571429	3.428571	3.285714
21	5	5	5	4.68	4.4	4.2	4.05	3.914286	3.778571	3.642857	3.507143	3.371429
22	5	5	5	4.733333	4.466667	4.257143	4.1	3.971429	3.842857	3.714286	3.585714	3.457143
23	5	5	5	4.8	4.533333	4.314286	4.15	4.022222	3.9	3.777778	3.655556	3.533333
24	5	5	5	4.8	4.6	4.371429	4.2	4.066667	3.95	3.833333	3.716667	3.6
25	5	5	5	5	4.666667	4.428571	4.25	4.111111	4	3.888889	3.777778	3.666667
26	5	5	5	5	4.733333	4.485714	4.3	4.155556	4.04	3.933333	3.826667	3.72
27	5	5	5	5	4.8	4.542857	4.35	4.2	4.08	3.977778	3.875556	3.773333
28	5	5	5	5	4.8	4.6	4.4	4.244444	4.12	4.018182	3.92	3.821818
29	5	5	5	5	4.8	4.657143	4.45	4.288889	4.16	4.054545	3.96	3.865455
30	5	5	5	5	5	4.7	4.5	4.333333	4.2	4.090909	4	3.909091

Figure A.7: Table of values for $B = 5$, given in decimal format.

A/D	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0.2	0.2	0	0	0	0	0	0	0	0	0	0
4	0.56	0.48	0.4	0.32	0.266667	0.2	0.2	0	0	0	0	0
5	0.8	0.733333	0.666667	0.6	0.533333	0.466667	0.4	0.333333	0.266667	0.2	0.2	0.2
6	0.971429	0.914286	0.857143	0.8	0.742857	0.685714	0.628571	0.571429	0.514286	0.457143	0.4	0.342857
7	1.133333	1.066667	1	0.95	0.9	0.85	0.8	0.75	0.7	0.65	0.6	0.55
8	1.295238	1.219048	1.142857	1.085714	1.028571	0.977778	0.933333	0.888889	0.844444	0.8	0.755556	0.711111
9	1.457143	1.371429	1.285714	1.221429	1.157143	1.1	1.05	1	0.96	0.92	0.88	0.84
10	1.619048	1.52381	1.428571	1.357143	1.285714	1.222222	1.166667	1.111111	1.066667	1.022222	0.981818	0.945455
11	1.780952	1.67619	1.571429	1.492857	1.414286	1.344444	1.283333	1.222222	1.173333	1.124444	1.08	1.04
12	1.942857	1.828571	0.75	1.628571	1.542857	1.466667	1.4	1.333333	1.28	1.226667	1.178182	1.134545
13	2.104762	1.980952	1.857143	1.764286	1.671429	1.588889	1.516667	1.444444	1.386667	1.328889	1.276364	1.229091
14	2.266667	2.133333	2	1.9	1.8	1.711111	1.633333	1.555556	1.493333	1.431111	1.374545	1.323636
15	2.428571	2.285714	2.142857	2.035714	1.928571	1.833333	1.75	1.666667	1.6	1.533333	1.472727	1.418182
16	2.590476	2.438095	2.285714	2.171429	2.057143	1.955556	1.866667	1.777778	1.706667	1.635556	1.570909	1.512727
17	2.752381	2.590476	2.428571	2.307143	2.185714	2.077778	1.983333	1.888889	1.813333	1.737778	1.669091	1.607273
18	2.895238	2.733333	2.571429	2.442857	2.314286	2.2	2.1	2	1.92	1.84	1.767273	1.701818
19	3.019048	2.866667	2.714286	2.578571	2.442857	2.322222	2.216667	2.111111	2.026667	1.942222	1.865455	1.796364
20	3.142857	3	2.857143	2.714286	2.571429	2.444444	2.333333	2.222222	2.133333	2.044444	1.963636	1.890909
21	3.235714	3.1	2.964286	2.828571	2.692857	2.566667	2.45	2.333333	2.24	2.146667	2.061818	1.985455
22	3.328571	3.2	3.071429	2.942857	2.814286	2.688889	2.566667	2.444444	2.346667	2.248889	2.16	2.08
23	3.411111	3.288889	3.166667	3.044444	2.922222	2.8	2.677778	2.555556	2.453333	2.351111	2.258182	2.174545
24	3.483333	3.366667	3.25	3.133333	3.016667	2.9	2.783333	2.666667	2.56	2.453333	2.356364	2.269091
25	3.555556	3.444444	3.333333	3.222222	3.111111	3	2.888889	2.777778	2.666667	2.555556	2.454545	2.363636
26	3.613333	3.506667	3.4	3.293333	3.186667	3.08	2.973333	2.866667	2.76	2.653333	2.552727	2.458182
27	3.671111	3.568889	3.466667	3.364444	3.262222	3.16	3.057778	2.955556	2.853333	2.751111	2.650909	2.552727
28	3.723636	3.625455	3.527273	3.429091	3.330909	3.232727	3.134545	3.036364	2.938182	2.84	2.741818	2.643636
29	3.770909	3.676364	3.581818	3.487273	3.392727	3.298182	3.203636	3.109091	3.014545	2.92	2.825455	2.730909
30	3.818182	3.727273	3.636364	3.545455	3.454545	3.363636	3.272727	3.181818	3.090909	3	2.909091	2.818182

Figure A.8: Continuation of Figure A.7.

A/D	1	2	3	4	5	6	7	8	9	10	11	12
1	1/2	0	0	0	0	0	0	0	0	0	0	0
2	1	2/3	1/2	0	0	0	0	0	0	0	0	0
3	3/2	1	3/4	1/2	1/2	0	0	0	0	0	0	0
4	2	4/3	1	4/5	2/3	1/2	1/2	0	0	0	0	0
5	2	3/2	5/4	1	5/6	2/3	1/2	1/2	1/2	0	0	0
6	2	2	3/2	6/5	1	6/7	3/4	2/3	1/2	1/2	1/2	0
7	2	2	3/2	4/3	7/6	1	7/8	3/4	2/3	1/2	1/2	1/2
8	2	2	2	3/2	4/3	8/7	1	8/9	4/5	2/3	2/3	1/2
9	2	2	2	3/2	3/2	5/4	9/8	1	9/10	4/5	3/4	2/3
10	2	2	2	2	3/2	4/3	5/4	10/9	1	10/11	5/6	3/4
11	2	2	2	2	3/2	3/2	4/3	6/5	11/10	1	11/12	5/6
12	2	2	2	2	2	3/2	3/2	4/3	6/5	12/11	1	12/13
13	2	2	2	2	2	3/2	3/2	4/3	5/4	7/6	13/12	1
14	2	2	2	2	2	2	3/2	3/2	4/3	5/4	7/6	14/13
15	2	2	2	2	2	2	3/2	3/2	3/2	4/3	5/4	8/7
16	2	2	2	2	2	2	2	3/2	3/2	4/3	4/3	6/5
17	2	2	2	2	2	2	2	3/2	3/2	3/2	4/3	5/4
18	2	2	2	2	2	2	2	2	3/2	3/2	3/2	4/3
19	2	2	2	2	2	2	2	2	3/2	3/2	3/2	4/3
20	2	2	2	2	2	2	2	2	2	3/2	3/2	3/2
21	2	2	2	2	2	2	2	2	2	3/2	3/2	3/2
22	2	2	2	2	2	2	2	2	2	2	3/2	3/2
23	2	2	2	2	2	2	2	2	2	2	3/2	3/2
24	2	2	2	2	2	2	2	2	2	2	2	3/2
25	2	2	2	2	2	2	2	2	2	2	2	3/2
26	2	2	2	2	2	2	2	2	2	2	2	2
27	2	2	2	2	2	2	2	2	2	2	2	2
28	2	2	2	2	2	2	2	2	2	2	2	2
29	2	2	2	2	2	2	2	2	2	2	2	2
30	2	2	2	2	2	2	2	2	2	2	2	2

Figure A.9: Table of values for $B = 2$, given as rational numbers.

A/D	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0	0	0
7	1/2	0	0	0	0	0	0	0	0	0	0	0
8	1/2	1/2	1/2	0	0	0	0	0	0	0	0	0
9	1/2	1/2	1/2	1/2	1/2	0	0	0	0	0	0	0
10	2/3	2/3	1/2	1/2	1/2	1/2	1/2	0	0	0	0	0
11	3/4	2/3	2/3	1/2	1/2	1/2	1/2	1/2	1/2	0	0	0
12	6/7	4/5	3/4	2/3	2/3	1/2	1/2	1/2	1/2	1/2	1/2	0
13	13/14	6/7	4/5	3/4	2/3	2/3	1/2	1/2	1/2	1/2	1/2	1/2
14	1	14/15	7/8	4/5	3/4	2/3	2/3	2/3	1/2	1/2	1/2	1/2
15	15/14	1	15/16	7/8	5/6	3/4	3/4	2/3	2/3	1/2	1/2	1/2
16	8/7	16/15	1	16/17	8/9	5/6	4/5	3/4	2/3	2/3	2/3	1/2
17	6/5	9/8	17/16	1	17/18	8/9	5/6	4/5	3/4	2/3	2/3	2/3
18	5/4	6/5	9/8	18/17	1	18/19	9/10	6/7	4/5	3/4	3/4	2/3
19	4/3	5/4	7/6	10/9	19/18	1	19/20	9/10	6/7	4/5	3/4	3/4
20	4/3	4/3	5/4	7/6	10/9	20/19	1	20/21	10/11	6/7	5/6	4/5
21	3/2	4/3	5/4	6/5	7/6	11/10	21/20	1	21/22	10/11	7/8	5/6
22	3/2	4/3	4/3	5/4	6/5	8/7	11/10	22/21	1	22/23	11/12	7/8
23	3/2	3/2	4/3	4/3	5/4	6/5	8/7	12/11	23/22	1	23/24	11/12
24	3/2	3/2	3/2	4/3	4/3	5/4	6/5	8/7	12/11	24/23	1	24/25
25	3/2	3/2	3/2	4/3	4/3	5/4	5/4	7/6	9/8	13/12	25/24	1
26	3/2	3/2	3/2	3/2	4/3	4/3	5/4	6/5	7/6	9/8	13/12	26/25
27	3/2	3/2	3/2	3/2	3/2	4/3	4/3	5/4	6/5	7/6	9/8	14/13
28	2	3/2	3/2	3/2	3/2	4/3	4/3	4/3	5/4	6/5	7/6	10/9
29	2	3/2	3/2	3/2	3/2	3/2	4/3	4/3	5/4	5/4	6/5	8/7
30	2	2	3/2	3/2	3/2	3/2	3/2	4/3	4/3	5/4	5/4	6/5

Figure A.10: Continuation of Figure A.9.

A/D	1	2	3	4	5	6	7	8	9	10	11	12
1	2/3	1/3	0	0	0	0	0	0	0	0	0	0
2	4/3	8/9	2/3	4/9	1/3	0	0	0	0	0	0	0
3	2	4/3	1	5/6	2/3	1/2	1/3	1/3	0	0	0	0
4	7/3	16/9	4/3	10/9	14/15	4/5	2/3	8/15	4/9	1/3	1/3	0
5	8/3	19/9	5/3	25/18	7/6	1	8/9	7/9	2/3	5/9	4/9	1/3
6	3	7/3	2	5/3	7/5	6/5	16/15	20/21	6/7	16/21	2/3	4/7
7	3	23/9	13/6	17/9	49/30	7/5	56/45	10/9	1	11/12	5/6	3/4
8	3	8/3	7/3	31/15	11/6	8/5	64/45	80/63	8/7	22/21	26/27	8/9
9	3	3	5/2	11/5	2	9/5	8/5	10/7	9/7	33/28	13/12	1
10	3	3	8/3	7/3	19/9	29/15	79/45	100/63	10/7	55/42	65/54	10/9
11	3	3	8/3	37/15	20/9	43/21	17/9	109/63	11/7	121/84	143/108	11/9
12	3	3	3	23/9	7/3	15/7	2	13/7	12/7	11/7	13/9	4/3
13	3	3	3	8/3	22/9	47/21	25/12	41/21	51/28	71/42	169/108	13/9
14	3	3	3	8/3	23/9	7/3	13/6	55/27	23/12	97/54	181/108	14/9
15	3	3	3	3	8/3	17/7	9/4	19/9	2	17/9	16/9	5/3
16	3	3	3	3	8/3	5/2	7/3	59/27	31/15	53/27	251/135	79/45
17	3	3	3	3	8/3	23/9	29/12	61/27	32/15	67/33	29/15	101/55
18	3	3	3	3	3	8/3	5/2	7/3	11/5	23/11	2	21/11
19	3	3	3	3	3	8/3	23/9	65/27	34/15	71/33	37/18	65/33
20	3	3	3	3	3	8/3	8/3	37/15	7/3	73/33	19/9	79/39
21	3	3	3	3	3	3	8/3	23/9	12/5	25/11	13/6	27/13
22	3	3	3	3	3	3	8/3	23/9	37/15	7/3	20/9	83/39
23	3	3	3	3	3	3	8/3	8/3	5/2	79/33	41/18	85/39
24	3	3	3	3	3	3	3	8/3	23/9	22/9	7/3	29/13
25	3	3	3	3	3	3	3	8/3	8/3	5/2	43/18	89/39
26	3	3	3	3	3	3	3	8/3	8/3	23/9	22/9	7/3
27	3	3	3	3	3	3	3	3	8/3	23/9	5/2	31/13
28	3	3	3	3	3	3	3	3	8/3	8/3	23/9	17/7
29	3	3	3	3	3	3	3	3	8/3	8/3	23/9	37/15
30	3	3	3	3	3	3	3	3	3	8/3	8/3	5/2

Figure A.11: Table of values for $B = 3$, given as rational numbers.

A/D	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0
5	1/3	1/3	0	0	0	0	0	0	0	0	0	0
6	1/2	4/9	1/3	1/3	1/3	0	0	0	0	0	0	0
7	2/3	7/12	1/2	4/9	1/3	1/3	1/3	1/3	0	0	0	0
8	22/27	20/27	2/3	16/27	8/15	4/9	4/9	1/3	1/3	1/3	1/3	0
9	14/15	13/15	4/5	11/15	2/3	3/5	8/15	1/2	4/9	1/3	1/3	1/3
10	28/27	32/33	10/11	28/33	26/33	8/11	2/3	20/33	5/9	1/2	4/9	4/9
11	154/135	16/15	1	17/18	8/9	5/6	7/9	13/18	2/3	11/18	5/9	1/2
12	56/45	64/55	12/11	34/33	38/39	12/13	34/39	32/39	10/13	28/39	2/3	8/13
13	182/135	208/165	13/11	221/198	19/18	1	20/21	19/21	6/7	17/21	16/21	5/7
14	196/135	224/165	14/11	119/99	133/117	14/13	40/39	44/45	14/15	8/9	38/45	4/5
15	14/9	16/11	15/11	85/66	95/78	15/13	100/91	22/21	1	23/24	11/12	7/8
16	223/135	256/165	16/11	136/99	152/117	16/13	320/273	352/315	16/15	46/45	50/51	16/17
17	287/165	271/165	17/11	289/198	323/234	17/13	340/273	374/315	17/15	391/360	25/24	1
18	20/11	19/11	18/11	17/11	19/13	18/13	120/91	44/35	6/5	23/20	75/68	18/17
19	373/198	178/99	113/66	161/99	361/234	19/13	380/273	418/315	19/15	437/360	475/408	19/17
20	35/18	218/117	139/78	199/117	379/234	20/13	400/273	88/63	4/3	23/18	125/102	20/17
21	2	25/13	24/13	23/13	22/13	21/13	20/13	22/15	7/5	161/120	175/136	21/17
22	43/21	77/39	173/91	499/273	479/273	153/91	439/273	484/315	22/15	253/180	275/204	22/17
23	44/21	91/45	41/21	593/315	571/315	61/35	527/315	101/63	23/15	529/360	575/408	23/17
24	15/7	31/15	2	29/15	28/15	9/5	26/15	5/3	8/5	23/15	25/17	24/17
25	46/21	19/9	49/24	89/45	689/360	37/20	643/360	31/18	199/120	287/180	625/408	25/17
26	47/21	97/45	25/12	103/51	47/24	129/68	749/408	181/102	233/136	337/204	649/408	26/17
27	16/7	11/5	17/8	35/17	2	33/17	32/17	31/17	30/17	29/17	28/17	27/17
28	7/3	101/45	13/6	107/51	55/27	101/51	883/459	857/459	277/153	805/459	779/459	251/153
29	50/21	103/45	53/24	109/51	56/27	115/57	53/27	979/513	317/171	923/513	895/513	289/171
30	17/7	7/3	9/4	37/17	19/9	39/19	2	37/19	36/19	35/19	34/19	33/19

Figure A.12: Continuation of Figure A.11.

A/D	1	2	3	4	5	6	7	8	9	10	11	12
1	3/4	1/2	1/4	0	0	0	0	0	0	0	0	0
2	3/2	1	5/6	2/3	1/2	1/3	1/4	0	0	0	0	0
3	9/4	3/2	5/4	1	7/8	3/4	5/8	1/2	3/8	1/4	1/4	0
4	3	2	5/3	4/3	7/6	1	9/10	4/5	7/10	3/5	1/2	2/5
5	13/4	5/2	25/12	5/3	35/24	5/4	9/8	1	11/12	5/6	3/4	2/3
6	7/2	3	5/2	2	7/4	3/2	27/20	6/5	11/10	1	13/14	6/7
7	15/4	19/6	11/4	7/3	49/24	7/4	63/40	7/5	77/60	7/6	13/12	1
8	4	10/3	3	8/3	7/3	2	9/5	8/5	22/15	4/3	26/21	8/7
9	4	7/2	25/8	17/6	61/24	9/4	81/40	9/5	33/20	3/2	39/28	9/7
10	4	11/3	13/4	3	11/4	5/2	9/4	2	11/6	5/3	65/42	10/7
11	4	15/4	27/8	31/10	23/8	53/20	97/40	11/5	121/60	11/6	143/84	11/7
12	4	4	7/2	16/5	3	14/5	13/5	12/5	11/5	2	13/7	12/7
13	4	4	29/8	33/10	37/12	29/10	163/60	38/15	47/20	13/6	169/84	13/7
14	4	4	15/4	17/5	19/6	3	17/6	8/3	5/2	7/3	13/6	2
15	4	4	15/4	7/2	13/4	43/14	35/12	58/21	73/28	103/42	193/84	15/7
16	4	4	4	18/5	10/3	22/7	3	20/7	19/7	18/7	17/7	16/7
17	4	4	4	11/3	41/12	45/14	49/16	41/14	313/112	149/56	283/112	67/28
18	4	4	4	15/4	7/2	23/7	25/8	3	23/8	11/4	21/8	5/2
19	4	4	4	15/4	43/12	47/14	51/16	55/18	47/16	203/72	389/144	31/12
20	4	4	4	4	11/3	24/7	13/4	28/9	3	26/9	25/9	8/3
21	4	4	4	4	15/4	7/2	53/16	19/6	61/20	53/18	511/180	41/15
22	4	4	4	4	15/4	25/7	27/8	29/9	31/10	3	29/10	14/5
23	4	4	4	4	15/4	29/8	55/16	59/18	63/20	67/22	59/20	157/55
24	4	4	4	4	4	11/3	7/2	10/3	16/5	34/11	3	32/11
25	4	4	4	4	4	15/4	57/16	61/18	13/4	69/22	73/24	65/22
26	4	4	4	4	4	15/4	29/8	31/9	33/10	35/11	37/12	3
27	4	4	4	4	4	15/4	11/3	7/2	67/20	71/22	25/8	79/26
28	4	4	4	4	4	4	15/4	32/9	17/5	36/11	19/6	40/13
29	4	4	4	4	4	4	15/4	18/5	69/20	73/22	77/24	81/26
30	4	4	4	4	4	4	15/4	11/3	7/2	37/11	13/4	41/13

Figure A.13: Table of values for $B = 4$, given as rational numbers.

A/D	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0
4	1/3	1/4	1/4	0	0	0	0	0	0	0	0	0
5	7/12	1/2	5/12	1/3	1/4	1/4	1/4	0	0	0	0	0
6	11/14	5/7	9/14	4/7	1/2	3/7	3/8	1/3	1/4	1/4	1/4	0
7	15/16	7/8	13/16	3/4	11/16	5/8	9/16	1/2	7/16	3/8	1/3	1/4
8	15/14	1	17/18	8/9	5/6	7/9	13/18	2/3	11/18	5/9	1/2	4/9
9	135/112	9/8	17/16	1	19/20	9/10	17/20	4/5	3/4	7/10	13/20	3/5
10	75/56	5/4	85/72	10/9	19/18	1	21/22	10/11	19/22	9/11	17/22	8/11
11	165/112	11/8	187/144	11/9	209/180	11/10	21/20	1	23/24	11/12	7/8	5/6
12	45/28	3/2	17/12	4/3	19/15	6/5	63/55	12/11	23/22	1	25/26	12/13
13	195/112	13/8	221/144	13/9	247/180	13/10	273/220	13/11	299/264	13/12	25/24	1
14	15/8	7/4	119/72	14/9	133/90	7/5	147/110	14/11	161/132	7/6	175/156	14/13
15	225/112	15/8	85/48	5/3	19/12	3/2	63/44	15/11	115/88	5/4	125/104	15/13
16	15/7	2	17/9	16/9	76/45	8/5	84/55	16/11	46/33	4/3	50/39	16/13
17	253/112	17/8	289/144	17/9	323/180	17/10	357/220	17/11	391/264	17/12	425/312	17/13
18	19/8	9/4	17/8	2	19/10	9/5	189/110	18/11	69/44	3/2	75/52	18/13
19	355/144	169/72	107/48	19/9	361/180	19/10	399/220	19/11	437/264	19/12	475/312	19/13
20	23/9	22/9	7/3	20/9	19/9	2	21/11	20/11	115/66	5/3	125/78	20/13
21	473/180	227/90	29/12	104/45	397/180	21/10	441/220	21/11	161/88	7/4	175/104	21/13
22	27/10	13/5	5/2	12/5	23/10	11/5	21/10	2	23/12	11/6	275/156	22/13
23	607/220	293/110	113/44	136/55	523/220	251/110	481/220	23/11	529/264	23/12	575/312	23/13
24	31/11	30/11	29/11	28/11	27/11	26/11	25/11	24/11	23/11	2	25/13	24/13
25	757/264	367/132	237/88	86/33	665/264	107/44	619/264	149/66	191/88	25/12	625/312	25/13
26	35/12	17/6	11/4	8/3	31/12	5/2	29/12	7/3	9/4	13/6	25/12	2
27	71/24	449/156	291/104	106/39	823/312	133/52	773/312	187/78	241/104	349/156	673/312	27/13
28	3	38/13	37/13	36/13	35/13	34/13	33/13	32/13	31/13	30/13	29/13	28/13
29	85/28	77/26	1051/364	256/91	997/364	485/182	943/364	229/91	127/52	431/182	835/364	202/91
30	43/14	3	41/14	20/7	39/14	19/7	37/14	18/7	5/2	17/7	33/14	16/7

Figure A.14: Continuation of Figure A.13.

A/D	1	2	3	4	5	6	7	8	9	10	11	12
1	4/5	3/5	2/5	1/5	0	0	0	0	0	0	0	0
2	8/5	6/5	14/15	4/5	2/3	8/15	2/5	4/15	1/5	0	0	0
3	12/5	9/5	7/5	6/5	1	9/10	4/5	7/10	3/5	1/2	2/5	3/10
4	16/5	12/5	28/15	8/5	4/3	6/5	16/15	24/25	22/25	4/5	18/25	16/25
5	4	3	7/3	2	5/3	3/2	4/3	6/5	11/10	1	14/15	13/15
6	21/5	17/5	14/5	12/5	2	9/5	8/5	36/25	33/25	6/5	28/25	26/25
7	22/5	19/5	49/15	14/5	7/3	21/10	28/15	42/25	77/50	7/5	98/75	91/75
8	23/5	61/15	18/5	47/15	8/3	12/5	32/15	48/25	44/25	8/5	112/75	104/75
9	24/5	21/5	19/5	17/5	3	27/10	12/5	54/25	99/50	9/5	42/25	39/25
10	5	13/3	4	11/3	10/3	3	8/3	12/5	11/5	2	28/15	26/15
11	5	67/15	41/10	19/5	7/2	16/5	29/10	66/25	121/50	11/5	154/75	143/75
12	5	23/5	21/5	59/15	11/3	17/5	47/15	72/25	66/25	12/5	56/25	52/25
13	5	71/15	43/10	101/25	19/5	89/25	83/25	77/25	71/25	13/5	182/75	169/75
14	5	24/5	22/5	103/25	39/10	92/25	173/50	81/25	151/50	14/5	196/75	182/75
15	5	5	9/2	21/5	4	19/5	18/5	17/5	16/5	3	14/5	13/5
16	5	5	23/5	107/25	61/15	97/25	277/75	263/75	83/25	47/15	221/75	69/25
17	5	5	47/10	109/25	62/15	99/25	284/75	271/75	86/25	49/15	232/75	73/25
18	5	5	24/5	111/25	21/5	141/35	58/15	389/105	124/35	71/21	338/105	107/35
19	5	5	24/5	113/25	64/15	143/35	59/15	397/105	127/35	73/21	349/105	111/35
20	5	5	5	23/5	13/3	29/7	4	27/7	26/7	25/7	24/7	23/7
21	5	5	5	117/25	22/5	21/5	81/20	137/35	529/140	51/14	491/140	118/35
22	5	5	5	71/15	67/15	149/35	41/10	139/35	269/70	26/7	251/70	121/35
23	5	5	5	24/5	68/15	151/35	83/20	181/45	39/10	34/9	329/90	53/15
24	5	5	5	24/5	23/5	153/35	21/5	61/15	79/20	23/6	223/60	18/5
25	5	5	5	5	14/3	31/7	17/4	37/9	4	35/9	34/9	11/3
26	5	5	5	5	71/15	157/35	43/10	187/45	101/25	59/15	287/75	93/25
27	5	5	5	5	24/5	159/35	87/20	21/5	102/25	179/45	872/225	283/75
28	5	5	5	5	24/5	23/5	22/5	191/45	103/25	221/55	98/25	1051/275
29	5	5	5	5	24/5	163/35	89/20	193/45	104/25	223/55	99/25	1063/275
30	5	5	5	5	5	47/10	9/2	13/3	21/5	45/11	4	43/11

Figure A.15: Table of values for $B = 5$, given as rational numbers.

A/D	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	1/5	1/5	0	0	0	0	0	0	0	0	0	0
4	14/25	12/25	2/5	8/25	4/15	1/5	1/5	0	0	0	0	0
5	4/5	11/15	2/3	3/5	8/15	7/15	2/5	1/3	4/15	1/5	1/5	1/5
6	34/35	32/35	6/7	4/5	26/35	24/35	22/35	4/7	18/35	16/35	2/5	12/35
7	17/15	16/15	1	19/20	9/10	17/20	4/5	3/4	7/10	13/20	3/5	11/20
8	136/105	128/105	8/7	38/35	36/35	44/45	14/15	8/9	38/45	4/5	34/45	32/45
9	51/35	48/35	9/7	171/140	81/70	11/10	21/20	1	24/25	23/25	22/25	21/25
10	34/21	32/21	10/7	19/14	9/7	11/9	7/6	10/9	16/15	46/45	54/55	52/55
11	187/105	176/105	11/7	209/140	99/70	121/90	77/60	11/9	88/75	253/225	27/25	26/25
12	68/35	64/35	12/7	57/35	54/35	22/15	7/5	4/3	32/25	92/75	324/275	312/275
13	221/105	208/105	13/7	247/140	117/70	143/90	91/60	13/9	104/75	299/225	351/275	338/275
14	34/15	32/15	2	19/10	9/5	77/45	49/30	14/9	112/75	322/225	378/275	364/275
15	17/7	16/7	15/7	57/28	27/14	11/6	7/4	5/3	8/5	23/15	81/55	78/55
16	272/105	256/105	16/7	76/35	72/35	88/45	28/15	16/9	128/75	368/225	432/275	416/275
17	289/105	272/105	17/7	323/140	153/70	187/90	119/60	17/9	136/75	391/225	459/275	442/275
18	304/105	41/15	18/7	171/70	81/35	11/5	21/10	2	48/25	46/25	486/275	468/275
19	317/105	43/15	19/7	361/140	171/70	209/90	133/60	19/9	152/75	437/225	513/275	494/275
20	22/7	3	20/7	19/7	18/7	22/9	7/3	20/9	32/15	92/45	108/55	104/55
21	453/140	31/10	83/28	99/35	377/140	77/30	49/20	7/3	56/25	161/75	567/275	546/275
22	233/70	16/5	43/14	103/35	197/70	121/45	77/30	22/9	176/75	506/225	54/25	52/25
23	307/90	148/45	19/6	137/45	263/90	14/5	241/90	23/9	184/75	529/225	621/275	598/275
24	209/60	101/30	13/4	47/15	181/60	29/10	167/60	8/3	64/25	184/75	648/275	624/275
25	32/9	31/9	10/3	29/9	28/9	3	26/9	25/9	8/3	23/9	27/11	26/11
26	271/75	263/75	17/5	247/75	239/75	77/25	223/75	43/15	69/25	199/75	702/275	676/275
27	826/225	803/225	52/15	757/225	734/225	79/25	688/225	133/45	214/75	619/225	729/275	702/275
28	1024/275	997/275	194/55	943/275	916/275	889/275	862/275	167/55	808/275	71/25	754/275	727/275
29	1037/275	1011/275	197/55	959/275	933/275	907/275	881/275	171/55	829/275	73/25	777/275	751/275
30	42/11	41/11	40/11	39/11	38/11	37/11	36/11	35/11	34/11	3	32/11	31/11

Figure A.16: Continuation of Figure A.15.

A/D	1	2	3	4	5	6	7	8	9	10	11	12
1	5/6	2/3	1/2	1/3	1/6	0	0	0	0	0	0	0
2	5/3	4/3	1	8/9	7/9	2/3	5/9	4/9	1/3	2/9	1/6	0
3	5/2	2	3/2	4/3	7/6	1	11/12	5/6	3/4	2/3	7/12	1/2
4	10/3	8/3	2	16/9	14/9	4/3	11/9	10/9	1	14/15	13/15	4/5
5	25/6	10/3	5/2	20/9	35/18	5/3	55/36	25/18	5/4	7/6	13/12	1
6	5	4	3	8/3	7/3	2	11/6	5/3	3/2	7/5	13/10	6/5
7	31/6	13/3	7/2	28/9	49/18	7/3	77/36	35/18	7/4	49/30	91/60	7/5
8	16/3	14/3	4	32/9	28/9	8/3	22/9	20/9	2	28/15	26/15	8/5
9	11/2	5	9/2	4	7/2	3	11/4	5/2	9/4	21/10	39/20	9/5
10	17/3	46/9	14/3	38/9	34/9	10/3	55/18	25/9	5/2	7/3	13/6	2
11	35/6	47/9	29/6	40/9	73/18	11/3	121/36	55/18	11/4	77/30	143/60	11/5
12	6	16/3	5	14/3	13/3	4	11/3	10/3	3	14/5	13/5	12/5
13	6	49/9	61/12	43/9	161/36	25/6	139/36	32/9	13/4	91/30	169/60	13/5
14	6	50/9	31/6	44/9	83/18	13/3	73/18	34/9	7/2	49/15	91/30	14/5
15	6	17/3	21/4	5	19/4	9/2	17/4	4	15/4	7/2	13/4	3
16	6	52/9	16/3	76/15	29/6	23/5	131/30	62/15	39/10	11/3	103/30	16/5
17	6	35/6	65/12	77/15	59/12	47/10	269/60	64/15	81/20	23/6	217/60	17/5
18	6	6	11/2	26/5	5	24/5	23/5	22/5	21/5	4	19/5	18/5
19	6	6	67/12	79/15	91/18	73/15	421/90	202/45	43/10	37/9	353/90	56/15
20	6	6	17/3	16/3	46/9	74/15	214/45	206/45	22/5	38/9	182/45	58/15
21	6	6	23/4	27/5	31/6	5	29/6	14/3	9/2	13/3	25/6	4
22	6	6	35/6	82/15	47/9	106/21	44/9	298/63	32/7	278/63	268/63	86/21
23	6	6	35/6	83/15	95/18	107/21	89/18	302/63	65/14	283/63	547/126	88/21
24	6	6	6	28/5	16/3	36/7	5	34/7	33/7	32/7	31/7	30/7
25	6	6	6	17/3	97/18	109/21	121/24	103/21	267/56	389/84	755/168	61/14
26	6	6	6	86/15	49/9	110/21	61/12	104/21	135/28	197/42	383/84	31/7
27	6	6	6	52/9	11/2	37/7	41/8	5	39/8	19/4	37/8	9/2
28	6	6	6	35/6	50/9	16/3	31/6	136/27	59/12	259/54	505/108	41/9
29	6	6	6	35/6	101/18	113/21	125/24	137/27	119/24	523/108	1021/216	83/18
30	6	6	6	6	17/3	38/7	21/4	46/9	5	44/9	43/9	14/3

Figure A.17: Table of values for $B = 6$, given as rational numbers.

A/D	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0	0	0	0	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0	0	0	0	0	0
3	5/12	1/3	1/4	1/6	1/6	0	0	0	0	0	0	0
4	11/15	2/3	3/5	8/15	7/15	2/5	1/3	4/15	2/9	1/6	1/6	0
5	17/18	8/9	5/6	7/9	13/18	2/3	11/18	5/9	1/2	4/9	7/18	1/3
6	17/15	16/15	1	20/21	19/21	6/7	17/21	16/21	5/7	2/3	13/21	4/7
7	119/90	56/45	7/6	10/9	19/18	1	23/24	11/12	7/8	5/6	19/24	3/4
8	68/45	64/45	4/3	80/63	76/63	8/7	23/21	22/21	1	26/27	25/27	8/9
9	17/10	8/5	3/2	10/7	19/14	9/7	69/56	33/28	9/8	13/12	25/24	1
10	17/9	16/9	5/3	100/63	95/63	10/7	115/84	55/42	5/4	65/54	125/108	10/9
11	187/90	88/45	11/6	110/63	209/126	11/7	253/168	121/84	11/8	143/108	275/216	11/9
12	34/15	32/15	2	40/21	38/21	12/7	23/14	11/7	3/2	13/9	25/18	4/3
13	221/90	104/45	13/6	130/63	247/126	13/7	299/168	143/84	13/8	169/108	325/216	13/9
14	119/45	112/45	7/3	20/9	19/9	2	23/12	11/6	7/4	91/54	175/108	14/9
15	17/6	8/3	5/2	50/21	95/42	15/7	115/56	55/28	15/8	65/36	125/72	5/3
16	136/45	128/45	8/3	160/63	152/63	16/7	46/21	44/21	2	52/27	50/27	16/9
17	289/90	136/45	17/6	170/63	323/126	17/7	391/168	187/84	17/8	221/108	425/216	17/9
18	17/5	16/5	3	20/7	19/7	18/7	69/28	33/14	9/4	13/6	25/12	2
19	319/90	151/45	19/6	190/63	361/126	19/7	437/168	209/84	19/8	247/108	475/216	19/9
20	166/45	158/45	10/3	200/63	190/63	20/7	115/42	55/21	5/2	65/27	125/54	20/9
21	23/6	11/3	7/2	10/3	19/6	3	23/8	11/4	21/8	91/36	175/72	7/3
22	248/63	34/9	76/21	218/63	208/63	22/7	253/84	121/42	11/4	143/54	275/108	22/9
23	509/126	35/9	157/42	226/63	433/126	23/7	529/168	253/84	23/8	299/108	575/216	23/9
24	29/7	4	27/7	26/7	25/7	24/7	23/7	22/7	3	26/9	25/9	8/3
25	709/168	49/12	221/56	80/21	617/168	99/28	571/168	137/42	25/8	325/108	625/216	25/9
26	361/84	25/6	113/28	82/21	317/84	51/14	295/84	71/21	13/4	169/54	325/108	26/9
27	35/8	17/4	33/8	4	31/8	15/4	29/8	7/2	27/8	13/4	25/8	3
28	479/108	233/54	151/36	110/27	427/108	23/6	401/108	97/27	125/36	181/54	349/108	28/9
29	971/216	473/108	307/72	112/27	871/216	47/12	821/216	199/54	257/72	373/108	721/216	29/9
30	41/9	40/9	13/3	38/9	37/9	4	35/9	34/9	11/3	32/9	31/9	10/3

Figure A.18: Continuation of Figure A.17.

A/D	1	2	3	4	5	6	7	8	9	10	11	12
1	6/7	5/7	4/7	3/7	2/7	1/7	0	0	0	0	0	0
2	12/7	10/7	8/7	20/21	6/7	16/21	2/3	4/7	10/21	8/21	2/7	4/21
3	18/7	15/7	12/7	10/7	9/7	8/7	1	13/14	6/7	11/14	5/7	9/14
4	24/7	20/7	16/7	40/21	12/7	32/21	4/3	26/21	8/7	22/21	34/35	32/35
5	30/7	25/7	20/7	50/21	15/7	40/21	5/3	65/42	10/7	55/42	17/14	8/7
6	36/7	30/7	24/7	20/7	18/7	16/7	2	13/7	12/7	11/7	51/35	48/35
7	6	5	4	10/3	3	8/3	7/3	13/6	2	11/6	17/10	8/5
8	43/7	37/7	31/7	80/21	24/7	64/21	8/3	52/21	16/7	44/21	68/35	64/35
9	44/7	39/7	34/7	30/7	27/7	24/7	3	39/14	18/7	33/14	153/70	72/35
10	45/7	41/7	37/7	100/21	30/7	80/21	10/3	65/21	20/7	55/21	17/7	16/7
11	46/7	127/21	39/7	107/21	97/21	29/7	11/3	143/42	22/7	121/42	187/70	88/35
12	47/7	43/7	40/7	37/7	34/7	31/7	4	26/7	24/7	22/7	102/35	96/35
13	48/7	131/21	41/7	115/21	107/21	33/7	13/3	169/42	26/7	143/42	221/70	104/35
14	7	19/3	6	17/3	16/3	5	14/3	13/3	4	11/3	17/5	16/5
15	7	45/7	85/14	121/21	229/42	36/7	29/6	95/21	59/14	82/21	51/14	24/7
16	7	137/21	43/7	41/7	39/7	37/7	5	33/7	31/7	29/7	136/35	128/35
17	7	139/21	87/14	125/21	239/42	38/7	31/6	103/21	65/14	92/21	289/70	136/35
18	7	47/7	44/7	211/35	81/14	194/35	53/10	177/35	337/70	32/7	303/70	143/35
19	7	143/21	89/14	213/35	41/7	197/35	27/5	181/35	173/35	33/7	157/35	149/35
20	7	48/7	45/7	43/7	83/14	40/7	11/2	37/7	71/14	34/7	65/14	31/7
21	7	7	13/2	31/5	6	29/5	28/5	27/5	26/5	5	24/5	23/5
22	7	7	46/7	219/35	127/21	41/7	17/3	115/21	37/7	107/21	103/21	33/7
23	7	7	93/14	221/35	128/21	207/35	86/15	583/105	188/35	109/21	526/105	169/35
24	7	7	47/7	223/35	43/7	209/35	29/5	197/35	191/35	37/7	179/35	173/35
25	7	7	95/14	45/7	130/21	295/49	41/7	279/49	271/49	263/49	255/49	247/49
26	7	7	48/7	227/35	131/21	297/49	124/21	845/147	274/49	799/147	776/147	251/49
27	7	7	48/7	229/35	44/7	299/49	125/21	853/147	277/49	809/147	787/147	255/49
28	7	7	7	33/5	19/3	43/7	6	41/7	40/7	39/7	38/7	37/7
29	7	7	7	233/35	134/21	303/49	169/28	289/49	1129/196	551/98	1075/196	262/49
30	7	7	7	47/7	45/7	305/49	85/14	291/49	569/98	278/49	543/98	265/49

Figure A.19: Table of values for $B = 7$, given as rational numbers.

A/D	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0	0	0	0	0	0	0	0	0	0	0
2	1/7	0	0	0	0	0	0	0	0	0	0	0
3	4/7	1/2	3/7	5/14	2/7	3/14	1/7	1/7	0	0	0	0
4	6/7	4/5	26/35	24/35	22/35	4/7	18/35	16/35	2/5	12/35	2/7	8/35
5	15/14	1	20/21	19/21	6/7	17/21	16/21	5/7	2/3	13/21	4/7	11/21
6	9/7	6/5	8/7	38/35	36/35	48/49	46/49	44/49	6/7	40/49	38/49	36/49
7	3/2	7/5	4/3	19/15	6/5	8/7	23/21	22/21	1	27/28	13/14	25/28
8	12/7	8/5	32/21	152/105	48/35	64/49	184/147	176/147	8/7	54/49	52/49	50/49
9	27/14	9/5	12/7	57/35	54/35	72/49	69/49	66/49	9/7	243/196	117/98	225/196
10	15/7	2	40/21	38/21	12/7	80/49	230/147	220/147	10/7	135/98	65/49	125/98
11	33/14	11/5	44/21	209/105	66/35	88/49	253/147	242/147	11/7	297/196	143/98	275/196
12	18/7	12/5	16/7	76/35	72/35	96/49	92/49	88/49	12/7	81/49	78/49	75/49
13	39/14	13/5	52/21	247/105	78/35	104/49	299/147	286/147	13/7	351/196	169/98	325/196
14	3	14/5	8/3	38/15	12/5	16/7	46/21	44/21	2	27/14	13/7	25/14
15	45/14	3	20/7	19/7	18/7	120/49	115/49	110/49	15/7	405/196	195/98	375/196
16	24/7	16/5	64/21	304/105	96/35	128/49	368/147	352/147	16/7	108/49	104/49	100/49
17	51/14	17/5	68/21	323/105	102/35	136/49	391/147	374/147	17/7	459/196	221/98	425/196
18	269/70	18/5	24/7	114/35	108/35	144/49	138/49	132/49	18/7	243/98	117/49	225/98
19	141/35	19/5	76/21	361/105	114/35	152/49	437/147	418/147	19/7	513/196	247/98	475/196
20	59/14	4	80/21	76/21	24/7	160/49	460/147	440/147	20/7	135/49	130/49	125/49
21	22/5	21/5	4	19/5	18/5	24/7	23/7	22/7	3	81/28	39/14	75/28
22	95/21	13/3	29/7	83/21	79/21	176/49	506/147	484/147	22/7	297/98	143/49	275/98
23	488/105	67/15	30/7	431/105	412/105	184/49	529/147	506/147	23/7	621/196	299/98	575/196
24	167/35	23/5	31/7	149/35	143/35	192/49	184/49	176/49	24/7	162/49	156/49	150/49
25	239/49	33/7	223/49	215/49	207/49	199/49	191/49	183/49	25/7	675/196	325/98	625/196
26	730/147	101/21	228/49	661/147	638/147	205/49	592/147	569/147	26/7	351/98	169/49	325/98
27	743/147	103/21	233/49	677/147	655/147	211/49	611/147	589/147	27/7	729/196	351/98	675/196
28	36/7	5	34/7	33/7	32/7	31/7	30/7	29/7	4	27/7	26/7	25/7
29	1021/196	71/14	967/196	235/49	913/196	443/98	859/196	208/49	115/28	389/98	751/196	181/49
30	517/98	36/7	491/98	239/49	465/98	226/49	439/98	213/49	59/14	200/49	387/98	187/49

Figure A.20: Continuation of Figure A.19.

A/D	1	2	3	4	5	6	7	8	9	10	11	12
1	7/8	3/4	5/8	1/2	3/8	1/4	1/8	0	0	0	0	0
2	7/4	3/2	5/4	1	11/12	5/6	3/4	2/3	7/12	1/2	5/12	1/3
3	21/8	9/4	15/8	3/2	11/8	5/4	9/8	1	15/16	7/8	13/16	3/4
4	7/2	3	5/2	2	11/6	5/3	3/2	4/3	5/4	7/6	13/12	1
5	35/8	15/4	25/8	5/2	55/24	25/12	15/8	5/3	25/16	35/24	65/48	5/4
6	21/4	9/2	15/4	3	11/4	5/2	9/4	2	15/8	7/4	13/8	3/2
7	49/8	21/4	35/8	7/2	77/24	35/12	21/8	7/3	35/16	49/24	91/48	7/4
8	7	6	5	4	11/3	10/3	3	8/3	5/2	7/3	13/6	2
9	57/8	25/4	43/8	9/2	33/8	15/4	27/8	3	45/16	21/8	39/16	9/4
10	29/4	13/2	23/4	5	55/12	25/6	15/4	10/3	25/8	35/12	65/24	5/2
11	59/8	27/4	49/8	11/2	121/24	55/12	33/8	11/3	55/16	77/24	143/48	11/4
12	15/2	7	13/2	6	11/2	5	9/2	4	15/4	7/2	13/4	3
13	61/8	85/12	53/8	37/6	137/24	21/4	115/24	13/3	65/16	91/24	169/48	13/4
14	31/4	43/6	27/4	19/3	71/12	11/2	61/12	14/3	35/8	49/12	91/24	7/2
15	63/8	29/4	55/8	13/2	49/8	23/4	43/8	5	75/16	35/8	65/16	15/4
16	8	22/3	7	20/3	19/3	6	17/3	16/3	5	14/3	13/3	4
17	8	89/12	113/16	27/4	103/16	49/8	93/16	11/2	83/16	39/8	73/16	17/4
18	8	15/2	57/8	41/6	157/24	25/4	143/24	17/3	43/8	61/12	115/24	9/2
19	8	91/12	115/16	83/12	319/48	51/8	293/48	35/6	89/16	127/24	241/48	19/4
20	8	23/3	29/4	7	27/4	13/2	25/4	6	23/4	11/2	21/4	5
21	8	31/4	117/16	141/20	109/16	263/40	507/80	61/10	469/80	45/8	431/80	103/20
22	8	47/6	59/8	71/10	55/8	133/20	257/40	31/5	239/40	23/4	221/40	53/10
23	8	63/8	119/16	143/20	111/16	269/40	521/80	63/10	487/80	47/8	453/80	109/20
24	8	8	15/2	36/5	7	34/5	33/5	32/5	31/5	6	29/5	28/5
25	8	8	121/16	29/4	169/24	137/20	799/120	97/15	251/40	73/12	707/120	57/10
26	8	8	61/8	73/10	85/12	69/10	403/60	98/15	127/20	37/6	359/60	29/5
27	8	8	123/16	147/20	57/8	139/20	271/40	33/5	257/40	25/4	243/40	59/10
28	8	8	31/4	37/5	43/6	7	41/6	20/3	13/2	19/3	37/6	6
29	8	8	125/16	149/20	173/24	197/28	55/8	47/7	367/56	179/28	349/56	85/14
30	8	8	63/8	15/2	29/4	99/14	83/12	142/21	185/28	271/42	529/84	43/7

Figure A.21: Table of values for $B = 8$, given as rational numbers.

A/D	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0	0	0	0	0	0	0	0	0	0	0
2	1/4	1/6	1/8	0	0	0	0	0	0	0	0	0
3	11/16	5/8	9/16	1/2	7/16	3/8	5/16	1/4	3/16	1/8	1/8	0
4	19/20	9/10	17/20	4/5	3/4	7/10	13/20	3/5	11/20	1/2	9/20	2/5
5	19/16	9/8	17/16	1	23/24	11/12	7/8	5/6	19/24	3/4	17/24	2/3
6	57/40	27/20	51/40	6/5	23/20	11/10	21/20	1	27/28	13/14	25/28	6/7
7	133/80	63/40	119/80	7/5	161/120	77/60	49/40	7/6	9/8	13/12	25/24	1
8	19/10	9/5	17/10	8/5	23/15	22/15	7/5	4/3	9/7	26/21	25/21	8/7
9	171/80	81/40	153/80	9/5	69/40	33/20	63/40	3/2	81/56	39/28	75/56	9/7
10	19/8	9/4	17/8	2	23/12	11/6	7/4	5/3	45/28	65/42	125/84	10/7
11	209/80	99/40	187/80	11/5	253/120	121/60	77/40	11/6	99/56	143/84	275/168	11/7
12	57/20	27/10	51/20	12/5	23/10	11/5	21/10	2	27/14	13/7	25/14	12/7
13	247/80	117/40	221/80	13/5	299/120	143/60	91/40	13/6	117/56	169/84	325/168	13/7
14	133/40	63/20	119/40	14/5	161/60	77/30	49/20	7/3	9/4	13/6	25/12	2
15	57/16	27/8	51/16	3	23/8	11/4	21/8	5/2	135/56	65/28	125/56	15/7
16	19/5	18/5	17/5	16/5	46/15	44/15	14/5	8/3	18/7	52/21	50/21	16/7
17	323/80	153/40	289/80	17/5	391/120	187/60	119/40	17/6	153/56	221/84	425/168	17/7
18	171/40	81/20	153/40	18/5	69/20	33/10	63/20	3	81/28	39/14	75/28	18/7
19	361/80	171/40	323/80	19/5	437/120	209/60	133/40	19/6	171/56	247/84	475/168	19/7
20	19/4	9/2	17/4	4	23/6	11/3	7/2	10/3	45/14	65/21	125/42	20/7
21	393/80	187/40	71/16	21/5	161/40	77/20	147/40	7/2	27/8	13/4	25/8	3
22	203/40	97/20	37/8	22/5	253/60	121/30	77/20	11/3	99/28	143/42	275/84	22/7
23	419/80	201/40	77/16	23/5	529/120	253/60	161/40	23/6	207/56	299/84	575/168	23/7
24	27/5	26/5	5	24/5	23/5	22/5	21/5	4	27/7	26/7	25/7	24/7
25	661/120	319/60	41/8	74/15	569/120	91/20	523/120	25/6	225/56	325/84	625/168	25/7
26	337/60	163/30	21/4	76/15	293/60	47/10	271/60	13/3	117/28	169/42	325/84	26/7
27	229/40	111/20	43/8	26/5	201/40	97/20	187/40	9/2	243/56	117/28	225/56	27/7
28	35/6	17/3	11/2	16/3	31/6	5	29/6	14/3	9/2	13/3	25/6	4
29	331/56	23/4	313/56	38/7	295/56	143/28	277/56	67/14	37/8	125/28	241/56	29/7
30	503/84	35/6	159/28	116/21	451/84	73/14	425/84	103/21	19/4	193/42	373/84	30/7

Figure A.22: Continuation of Figure A.21.

A/D	1	2	3	4	5	6	7	8	9	10	11	12
1	8/9	7/9	2/3	5/9	4/9	1/3	2/9	1/9	0	0	0	0
2	16/9	14/9	4/3	10/9	26/27	8/9	22/27	20/27	2/3	16/27	14/27	4/9
3	8/3	7/3	2	5/3	13/9	4/3	11/9	10/9	1	17/18	8/9	5/6
4	32/9	28/9	8/3	20/9	52/27	16/9	44/27	40/27	4/3	34/27	32/27	10/9
5	40/9	35/9	10/3	25/9	65/27	20/9	55/27	50/27	5/3	85/54	40/27	25/18
6	16/3	14/3	4	10/3	26/9	8/3	22/9	20/9	2	17/9	16/9	5/3
7	56/9	49/9	14/3	35/9	91/27	28/9	77/27	70/27	7/3	119/54	56/27	35/18
8	64/9	56/9	16/3	40/9	104/27	32/9	88/27	80/27	8/3	68/27	64/27	20/9
9	8	7	6	5	13/3	4	11/3	10/3	3	17/6	8/3	5/2
10	73/9	65/9	19/3	49/9	130/27	40/9	110/27	100/27	10/3	85/27	80/27	25/9
11	74/9	67/9	20/3	53/9	143/27	44/9	121/27	110/27	11/3	187/54	88/27	55/18
12	25/3	23/3	7	19/3	52/9	16/3	44/9	40/9	4	34/9	32/9	10/3
13	76/9	71/9	22/3	61/9	169/27	52/9	143/27	130/27	13/3	221/54	104/27	65/18
14	77/9	217/27	68/9	191/27	178/27	55/9	152/27	139/27	14/3	119/27	112/27	35/9
15	26/3	73/9	23/3	65/9	61/9	19/3	53/9	49/9	5	85/18	40/9	25/6
16	79/9	221/27	70/9	199/27	188/27	59/9	166/27	155/27	16/3	136/27	128/27	40/9
17	80/9	223/27	71/9	203/27	193/27	61/9	173/27	163/27	17/3	289/54	136/27	85/18
18	9	25/3	8	23/3	22/3	7	20/3	19/3	6	17/3	16/3	5
19	9	227/27	145/18	209/27	401/54	64/9	367/54	175/27	37/6	158/27	299/54	47/9
20	9	229/27	73/9	211/27	203/27	65/9	187/27	179/27	19/3	163/27	155/27	49/9
21	9	77/9	49/6	71/9	137/18	22/3	127/18	61/9	13/2	56/9	107/18	17/3
22	9	233/27	74/9	215/27	208/27	67/9	194/27	187/27	20/3	173/27	166/27	53/9
23	9	235/27	149/18	361/45	70/9	113/15	328/45	317/45	34/5	59/9	284/45	91/15
24	9	79/9	25/3	121/15	47/6	38/5	221/30	107/15	69/10	20/3	193/30	31/5
25	9	239/27	151/18	73/9	71/9	23/3	67/9	65/9	7	61/9	59/9	19/3
26	9	80/9	76/9	367/45	143/18	116/15	677/90	329/45	71/10	62/9	601/90	97/15
27	9	9	17/2	41/5	8	39/5	38/5	37/5	36/5	7	34/5	33/5
28	9	9	77/9	371/45	217/27	353/45	1033/135	1007/135	109/15	191/27	929/135	301/45
29	9	9	155/18	373/45	218/27	71/9	208/27	203/27	22/3	193/27	188/27	61/9
30	9	9	26/3	25/3	73/9	119/15	349/45	341/45	37/5	65/9	317/45	103/15

Figure A.23: Table of values for $B = 9$, given as rational numbers.

A/D	13	14	15	16	17	18	19	20	21	22	23	24
1	0	0	0	0	0	0	0	0	0	0	0	0
2	10/27	8/27	2/9	4/27	1/9	0	0	0	0	0	0	0
3	7/9	13/18	2/3	11/18	5/9	1/2	4/9	7/18	1/3	5/18	2/9	1/6
4	28/27	44/45	14/15	8/9	38/45	4/5	34/45	32/45	2/3	28/45	26/45	8/15
5	35/27	11/9	7/6	10/9	19/18	1	26/27	25/27	8/9	23/27	22/27	7/9
6	14/9	22/15	7/5	4/3	19/15	6/5	52/45	10/9	16/15	46/45	62/63	20/21
7	49/27	77/45	49/30	14/9	133/90	7/5	182/135	35/27	56/45	161/135	31/27	10/9
8	56/27	88/45	28/15	16/9	76/45	8/5	208/135	40/27	64/45	184/135	248/189	80/63
9	7/3	11/5	21/10	2	19/10	9/5	26/15	5/3	8/5	23/15	31/21	10/7
10	70/27	22/9	7/3	20/9	19/9	2	52/27	50/27	16/9	46/27	310/189	100/63
11	77/27	121/45	77/30	22/9	209/90	11/5	286/135	55/27	88/45	253/135	341/189	110/63
12	28/9	44/15	14/5	8/3	38/15	12/5	104/45	20/9	32/15	92/45	124/63	40/21
13	91/27	143/45	91/30	26/9	247/90	13/5	338/135	65/27	104/45	299/135	403/189	130/63
14	98/27	154/45	49/15	28/9	133/45	14/5	364/135	70/27	112/45	322/135	62/27	20/9
15	35/9	11/3	7/2	10/3	19/6	3	26/9	25/9	8/3	23/9	155/63	50/21
16	112/27	176/45	56/15	32/9	152/45	16/5	416/135	80/27	128/45	368/135	496/189	160/63
17	119/27	187/45	119/30	34/9	323/90	17/5	442/135	85/27	136/45	391/135	527/189	170/63
18	14/3	22/5	21/5	4	19/5	18/5	52/15	10/3	16/5	46/15	62/21	20/7
19	265/54	209/45	133/30	38/9	361/90	19/5	494/135	95/27	152/45	437/135	589/189	190/63
20	139/27	44/9	14/3	40/9	38/9	4	104/27	100/27	32/9	92/27	620/189	200/63
21	97/18	77/15	49/10	14/3	133/30	21/5	182/45	35/9	56/15	161/45	31/9	10/3
22	152/27	242/45	77/15	44/9	209/45	22/5	572/135	110/27	176/45	506/135	682/189	220/63
23	262/45	251/45	16/3	229/45	218/45	23/5	598/135	115/27	184/45	529/135	713/189	230/63
24	179/30	86/15	11/2	79/15	151/30	24/5	208/45	40/9	64/15	184/45	248/63	80/21
25	55/9	53/9	17/3	49/9	47/9	5	130/27	125/27	40/9	115/27	775/189	250/63
26	563/90	272/45	35/6	253/45	487/90	26/5	676/135	130/27	208/45	598/135	806/189	260/63
27	32/5	31/5	6	29/5	28/5	27/5	26/5	5	24/5	23/5	31/7	30/7
28	877/135	851/135	55/9	799/135	773/135	83/15	721/135	139/27	223/45	643/135	124/27	40/9
29	178/27	173/27	56/9	163/27	158/27	17/3	148/27	143/27	46/9	133/27	899/189	290/63
30	301/45	293/45	19/3	277/45	269/45	29/5	253/45	49/9	79/15	229/45	310/63	100/21

Figure A.24: Continuation of Figure A.23.

B Plots

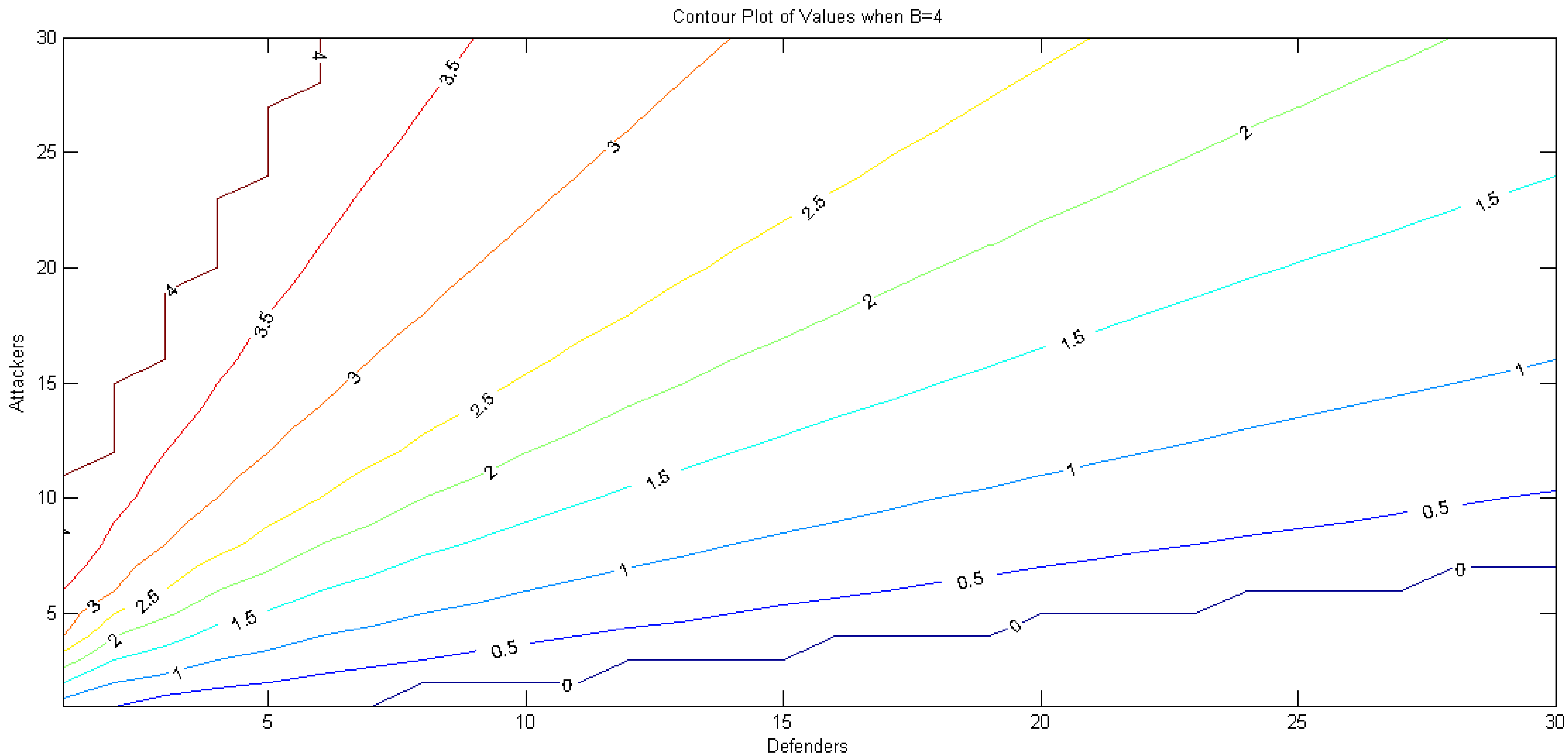

Figure B.1: Contour plot for $B = 4$. Notice the symmetry about $V = 2$.

C MATLAB Code

C.1 ListStrats

```
function Strategies = ListStrats( NumArmies, NumBases )
%ListStrats Creates a matrix with all the strategies for a number
    of armies
%   Lists all the strategies that a player has when the number of
    troops
%   they have is NumArmies and the number of bases to partition
    among is
%   given by NumBases. The strategies are given as a matrix where
%   each row represents a strategy.

if NumBases == 2
    i = NumArmies;
    j = 0;
    Strategies = [i j];

    while i > j+1
        i = i - 1;
        j = j + 1;
        Strategies = [Strategies; i j];
    end
else
    i = NumArmies;
    j = 0;
    Strategies = [i j zeros(1,NumBases-2)];
```

```
    while i > ceil(NumArmies/NumBases)
        i = i - 1;
        j = j + 1;
        ExtraStrats = ListStrats(j, NumBases - 1);
        NumRows = size(ExtraStrats,1);
        LeadingNumbers = zeros(NumRows, 1);

        for r = 1:NumRows
            LeadingNumbers(r) = i;
        end

        Strategies = sort([Strategies; LeadingNumbers ExtraStrats], 2, 'descend');
        Strategies = unique(Strategies, 'rows', 'stable');
    end
end
end
```

C.2 CreateGameMatrix

```
function [ Game ] = CreateGameMatrix( A,D,B )
%CreateGameMatrix Creates a the matrix for the blotto game with A attackers
%                  D defenders, and B bases

    %First, obtain the Strategy matrices
    AStrats = ListStrats(A,B);
    numAstrats = size(AStrats,1);
```

```
    DStrats = ListStrats(D,B);
    numDstrats = size(DStrats,1);

    %Next, initialize the game matrix with placeholder values
    Game = zeros(numAstrats,numDstrats);

    %Go through each element, compare the A & D strategies and
    %populate the Game matrix with outcomes

    for i = 1:numAstrats
        for j = 1:numDstrats
            BaseCapture = 0;%reset ticker each time you move to a new entry in the Game matrix

            %Count the number of bases captured given each strategy
            for a = 1:B
                for d = 1:B
                    if AStrats(i,a) > DStrats(j,d)
                        BaseCapture = BaseCapture + 1;
                    end
                end
            end
            Game(i,j) = BaseCapture / B;%Enter the expected number of bases captured
        end
    end
```

```
end
```

C.3 GameSolve

```
function [ Rstrat, Cstrat, GameValue ] = GameSolve( GameMatrix, options )
%GameSolve Uses linprog to solve Matrix games
%   When a Game Matrix is passed as an argument, this function will solve
%   the game using linprog, finding the optimal strategies of both players
%   and the associated value of the game.
%   The options argument passes options to linprog.

%% Determine the size of the incoming matrix and setup linprog options%%
[m,n] = size(GameMatrix);
if nargin ~= 2
    options = optimset('LargeScale','off','Algorithm','simplex','Simplex','on');
end

%% Column Player %%
ColGame = horzcat(GameMatrix, -ones(m,1));
Cf = [zeros(1,n), 1];
Cb = zeros(m,1);
Clb = [zeros(1,n),-inf];
CGeq = [ones(1,n), 0];
```

```
Cbeq = 1;

Csolution = linprog(Cf,ColGame,Cb,CGeq,Cbeq,Clb,[],[],options);
Cstrat = Csolution(1:n);
Cvalue = Csolution(n+1);

%% Row Player %%
RowGame = horzcat(-GameMatrix.', ones(n,1));
Rf = [zeros(1,m), -1];
Rb = zeros(n,1);
Rlb = [zeros(1,m), -inf];
RGeq = [ones(1,m), 0];
Rbeq = 1;

Rsolution = linprog(Rf,RowGame,Rb,RGeq,Rbeq,Rlb,[],[],options);
Rstrat = Rsolution(1:m);
Rvalue = Rsolution(m+1);

GameValue = [Cvalue; Rvalue];

end
```

C.4 BlottoTableCreator

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%       BlottoTableCreator.m
```

```
% %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% This should create a 4-dimensional array, storing all the solution
% information for Colonel Blotto Games
% For any parameters     A := number of Attacking Armies
%                        D := number of Defending Armies
%                        B := number of Bases
%
% The information stored in the BlottoTable is as follows:
% BlottoTable{A,D,B,1} := The optimal Attacking strategy
% BlottoTable{A,D,B,2} := The optimal Defending strategy
% BlottoTable{A,D,B,3} := The value of the game

%% Parameters - adjust here! %%
SmallestAttacker = 1;
SmallestDefender = 1;
SmallestBases = 2;
LargestAttacker = 30;
LargestDefender = 30;
LargestBases = 5;
%% Create and solve the Blotto Games in the ranges above. %%
BlottoTable = cell(LargestAttacker,LargestDefender,LargestBases,3);
tic
    for k = SmallestBases:LargestBases
```

```
        for i = SmallestAttacker:LargestAttacker
            for j = SmallestDefender:LargestDefender
                Game = CreateGameMatrix(i,j,k);
                [A,B,C] = GameSolve(Game);
                BlottoTable{i,j,k,1} = A;
                BlottoTable{i,j,k,2} = B;
                BlottoTable{i,j,k,3} = C;
            end
        end
        fprintf('BlottoTable populated for %d bases, %d attackers, and %d defenders.\n', k, LargestAttacker, LargestDefender);
    end
toc
disp('BlottoTable created.');
%% Check to make sure the dual programs found the same values. %%
for k = SmallestBases:LargestBases
    for i = SmallestAttacker:LargestAttacker
        for j = SmallestDefender:LargestDefender
            if size(BlottoTable{i,j,k,3},1) > 1
                A = BlottoTable{i,j,k,3};
                if A(1) == A(2)
                    BlottoTable{i,j,k,3} = A(1);
                else
                    fprintf('At (%d,%d,%d) the values are %d and %d\n', i, j, k, A(1),A(2));
                end
```

```
            end
        end
    end
end
disp('Game values checked. If displayed values are equal, amend them as well.');
%% Uncomment to add to an already-created Blotto Table
% SmallestAttacker = 30;
% LargestAttacker = 30;
% SmallestDefender = 19;
% LargestDefender = 30;
%
% outputMessage = '';
% for k = 9:9
%     for i = SmallestAttacker:LargestAttacker
%         for j = SmallestDefender:LargestDefender
%             Game = CreateGameMatrix(i,j,k);
%             [A,B,C] = GameSolve(Game);
%             BlottoTable{i,j,k,1} = A;
%             BlottoTable{i,j,k,2} = B;
%             BlottoTable{i,j,k,3} = C;
%
%             if abs(C(1)-C(2)) < max(C(1),C(2))*1e-12
%                 BlottoTable{i,j,k,3} = C(1);
%             else
%                 outputMessage = strvcat(outputMessage,sprintf('At (%d,%d,%d) the values are %d and %d', i, j, k, C(1),C(2)));
```

```
%                   end
%                   fprintf('Finished solving game at (%d,%d,%d).\n',i,j,k)
%               end
%           end
%           save('Blotto Data/BlottoTable (in progress).mat','BlottoTable')
% end
% %Print values that don't match for error checking
% disp(outputMessage);
```

References

Offline

[1] Friedman, Erich. *Introduction to Game Theory*. 2010. Print.

[2] Singleton, Robert R., and William F. Tyndall. *Games and Programs: Mathematics for Modeling*. San Francisco: W.H. Freeman, 1974. Print.

[3] Williams, J. D. *The Compleat Strategyst: Being a Primer on the Theory of Games of Strategy*. New York: Dover, 1986. Print.

Online

[4] Adamo, Tim, and Alexander Matros. "A Blotto Game With Incomplete Information." Economics Letters 105.1 (2009): 100-102. Business Source Premier. Web. 1 Dec. 2014.

[5] Barron, E.N. Wiley Series in Operations Research and Management Science : Game Theory : An Introduction (2nd Edition). New York, NY, USA: John Wiley & Sons, 2013. ProQuest ebrary. Web. 1 December 2014.

[6] Bellman, Richard. "Short Notes: On "Colonel Blotto" and Analogous Games." *SIAM Review* 11.1 (1969): 66-68. *JSTOR Journals*. Society for Industrial and Applied Mathematics. Web. 27 Oct. 2014. <http://www.jstor.org/stable/2028140>.

[7] Borel, Émile. "The Theory of Play and Integral Equations with Skew Symmetric Kernels." *Econometrica* 21.1 (1953): 97-100. *JSTOR Journals*. The Econometric Society. Web. <http://www.jstor.org/stable/1906946>.

[8] Doyle, Peter. *Linear Programming and the Theory of Games*. Web. 1 Dec. 2014. <https://math.dartmouth.edu/~doyle/docs/finite/fm3/scan/7.pdf>.

[9] Ferguson, Thomas S. *Game Theory*. Web. 1 Dec. 2014. <http://www.math.ucla.edu/~tom/Game_Theory/mat.pdf>.

[10] Golman, Russell, and Scott E. Page. "General Blotto: Games of Allocative Strategic Mismatch." *Public Choice* 2009: 279. *JSTOR Journals*. Web. 3 Oct. 2014.

[11] Gross, Oliver, and Robert Wagner. *A continuous Colonel Blotto game*. No. RAND-RM-408. RAND PROJECT AIR FORCE SANTA MONICA CA, 1950.

[12] Kovenock, Dan, and Brian Roberson. *Coalitional Colonel Blotto games with application to the economics of alliances*. No. SP II 2008-02. Discussion papers//WZB, Wissenschaftszentrum Berlin für Sozialforschung, Schwerpunkt Märkte und Politik, Abteilung Marktprozesse und Steuerung, 2008.

[13] Roberson, Brian, and Dmitriy Kvasov. "The Non-Constant-Sum Colonel Blotto Game." Economic Theory 51.2 (2012): 397-433.Business Source Premier. Web. 1 Dec. 2014.

[14] Straffin, Philip D. Anneli Lax New Mathematical Library, Volume 36 : Game Theory and Strategy. Washington, DC, USA: Mathematical Association of America, 1993. ProQuest ebrary. Web. 1 December 2014.