Searching for Optimal Strategies in *Knock 'm Down*

Matthew T. Fluet

Advisor: Prof. Arthur T. Benjamin

Harvey Mudd College

Claremont, CA 91711

May 1999

Abstract

Knock 'm Down is a game of dice that is so easy to learn that it is being played in classrooms around the world as a way to develop students' intuition about probability. However, as analysis has shown, lurking underneath this deceptively simple game are many surprising and highly *un*intuitive results. In the original description of the game, two players are each given one six sided die, 12 tokens and a board labeled with the values 2, 3, ..., 12. Each player distributes his/her tokens among the values on his/her board. Now, the players roll their dice together and each removes a token from his/her board on the value equal to the sum of the dice (if he/she has one there). Turns continue in this fashion. The winner is the first player to remove all twelve tokens. The problem posed by this game is to determine which allocation of tokens will maximize a player's chances of winning. Results will demonstrate that the answer to this question depends on many factors, and small variations in the rules of the game can lead to markedly different answers. In addition to the major theoretical results, the principal computational challenges of this problem will be discussed.

Contents

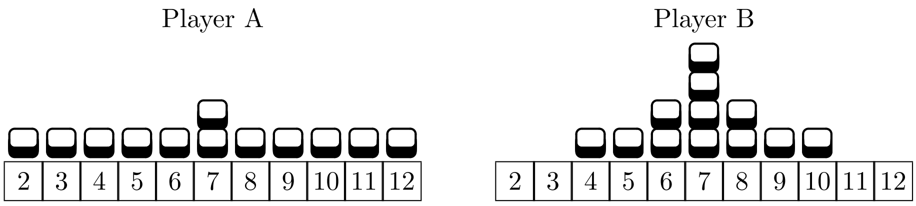

Figure 1: Allocations for Player A and Player B

1 Introduction

Knock 'm Down is a game of dice that is so easy to learn that it is being played in classrooms around the world. Although this game has been effective at developing students' intuition about probability (see [3] and [4]), our analysis will shown that lurking underneath this deceptively simple game are many surprising and highly *un*intuitive results.

The game is played by two players, each of whom is given a 6-sided die, 12 tokens, and a board with the numbers 2 through 12. The players allocate their tokens among the eleven numbers on the board however they want. Let's suppose players A and B allocate their tokens as shown in Figure 1. Next the players roll their dice together and each removes a token from his or her board on the value equal to the sum of the dice. For instance, if a 6 and a 2 are rolled, then both players remove a token from the 8 spot. Whereas if a 6 and a 5 are rolled, then player A removes a token from the 11 spot but – since player B has no tokens on the 11 spot – B's allocation is unchanged. The first player to remove all tokens is the winner. (If both players remove their last token on the same roll, then the game is a draw.)

In this paper, we investigate the question: what is the optimal strategy for playing a game of *Knock 'm Down*? That is, what allocation of tokens should one choose to maximize the probability that one wins the game?

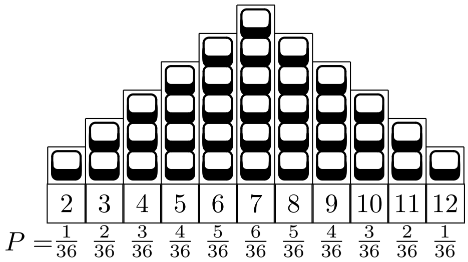

Figure 2: Histogram of Probabilities and an Allocation of 36 Tokens

2 A Variety of Examples

Before presenting a number of theoretical results, we will examine some interesting instances of *Knock 'm Down* games in order to develop a better intuitive understanding of the game. Additionally, these examples will allow us to become better acquainted with the types of theoretical results we wish to prove about the game.

2.1 Players A and B Revisited

Recall the example allocations shown in Figure 1 for the 12-token game with P corresponding to the roll of two six-sided dice. Instinctively, the allocation of player B seems superior since it more closely resembles the shape of the histogram of probabilities in Figure 2. In fact, player B wins against player A in 75% of the games, draws in 9% of the games, and loses in only 16% of the games. We say that the allocation of player B is *favored* over the allocation of player A, since it wins at least as often as it loses.

Our instincts told us that the optimal allocation should resemble the histogram as much as possible. If we were given 36 tokens to allocate, then we could allocate them exactly proportional to the probabilities in Figure 2. By all that is sensible, we felt this should be the optimal allocation. But as we soon learned, in this innocent little dice game, all is not sensible!

Before revealing our solution to the original 12-token game, let's find the best allocations for some simpler games. (Here's a hint: The best 12-token allocation can be obtained by moving just one token in player B's allocation in Figure 1.)

2.2 Surprises in a 4-Valued Game

Consider the 4-valued game consisting of outcomes α, β, γ, and δ with respective probabilities .4, .3, .2, and .1. How should you allocate 10 tokens? Can you predict which of the two allocations in Figure 3 is better? Notice that the first allocation has exactly the same triangular shape as the histogram of probabilities.

Surprisingly, the answer depends on what you mean by "better." It seems reasonable that we should want the allocation that requires, on average, the fewest number of turns to remove all tokens. Let $A = [4, 3, 2, 1]$,

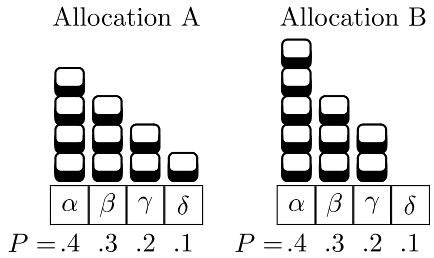

Figure 3: Which allocation is better?

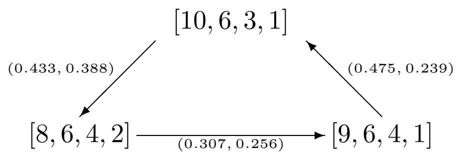

Figure 4: Subset of the Emperor Cycle in the 20-token Game with $P = (.4, .3, .2, .1)$

$B = [5, 3, 2, 0]$, and let $E(X)$ denote the average number (i.e., the *expected value*) of the number of turns needed to clear all the tokens from the allocation X.[1] In fact, using the calculations described in Section 6, we can show that B has the smallest expectation among *all* allocations of 10 tokens. We call such an allocation a *minimal allocation*. Armed with this information, it appears that B is the superior position. Or is it?

When we play the two positions against each other, we find that B loses to A, more than one and a half times as often as it beats it! Why does this happen? Essentially, it is due to the following fact: if a 1 is rolled anytime before B is finished, then allocation A becomes a *sub-allocation* of B, so B can not possibly win (it must lose or draw). However, *five* 4s must be rolled before B can achieve that status against A. In fact, allocation A is favored over *all* other allocations of 10 tokens. Using terminology from [5], we call such an allocation an *emperor*.

Does this same phenomenon occur when we increase the number of tokens? Alas no. When we play the same 4-valued game with 20 tokens, allocation $[10, 6, 3, 1]$ has the lowest expected value *and* it beats the triangular allocation $[8, 6, 4, 2]$ in head-to-head competition. (We note that 10, 6, 3, and 1 are *triangular* numbers, but that's just a coincidence!) But here's the strange part: allocation $[8, 6, 4, 2]$ beats $[9, 6, 4, 1]$ which in turn beats $[10, 6, 3, 1]$! In other words, we have a situation with *non-transitive probabilities*, as illustrated in Figure 4. The arc from $[10, 6, 3, 1]$ to $[8, 6, 4, 2]$ indicates that $[10, 6, 3, 1]$ beats $[8, 6, 4, 2]$ with probability 0.433 and loses to $[8, 6, 4, 2]$ with probability 0.388, (and therefore draws with probability 0.179). Although this game has no emperor, the three allocations above plus a few others form an *emperor cycle*:

[1] This informal notation is actually an abuse of the expected value notation, since X is not a random variable. See Section 3 for the formal notation that will be used in later sections.

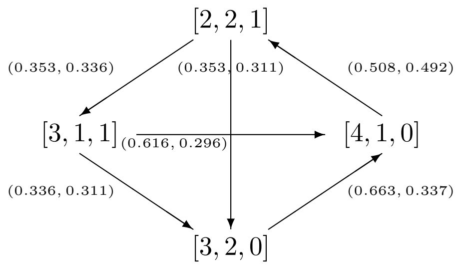

Figure 5: Emperor Cycle in the 5-Token Game with $P = (\frac{3}{6}, \frac{2}{6}, \frac{1}{6})$

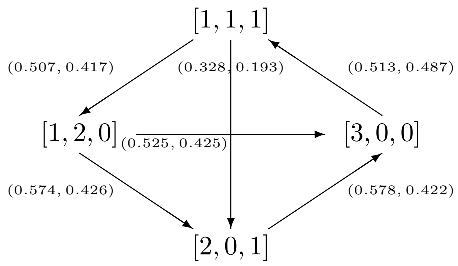

Figure 6: Cycles in the 3-Token Game with $P = (\frac{3}{6}, \frac{2}{6}, \frac{1}{6})$

the smallest set of allocations such that each allocation in the cycle is favored over all allocations not in the cycle.

2.3 A Scam with a 3-Valued Game

An emperor cycle also exists in the 5-token game with probability vector $P = (3/6, 2/6, 1/6)$. Here, allocation (3,2,0) defeats (4,1,0), which defeats (2,2,1), which defeats (3,1,1), which defeats (3,2,0) as illustrated in Figure 5.

When the same game is played with 3 tokens, then allocation $(2, 1, 0)$ is both the minimal allocation and the emperor. However, several non-emperor cycles exist in this game, as illustrated in Figure 6. One can easily imagine lucrative scams based on these non-transitive properties, easily played with a single six-sided die.

2.4 A Simple 2-Valued Game

To get a feel for how some of these results could be calculated, we will look at a very small example. (See Section 6 for a formal treatment of these calculations in the general case.) Consider the 2-valued game consisting of the outcomes H (heads) and T (tails) with respective probabilities of $\frac{2}{3}$ and $\frac{1}{3}$. Then there are only three allocations of two tokens, as depicted in Figure 7. How might we calculate the expected clearing time of allocation A? Well, clearly the first turn of the game reduces A to having only one token on one

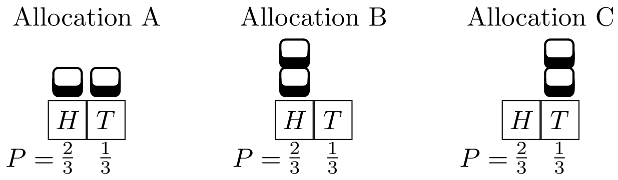

Figure 7: The Allocations for the 2-Token Game with $P = (\frac{2}{3}, \frac{1}{3})$.

of the two values. With probability $\frac{2}{3}$ heads comes up first and A must wait for a tails to appear. Since tails appear with probability $\frac{1}{3}$, then the expected time to clear one T token is ${\frac{1}{3}}^{-1} = 3$. Likewise, with probability $\frac{1}{3}$, A must wait an expected ${\frac{2}{3}}^{-1} = \frac{3}{2}$ turns until a heads appears. The analysis is even easier for allocations B and C. B must clear two tokens on H, waiting an expected $\frac{3}{2}$ turns for each, and C must clear two tokens on T, waiting an expected 3 turns for each. Thus, we have

$$\begin{aligned} E([1,1]) &= 1 + \frac{1}{3}\left(\frac{3}{2}\right) + \frac{2}{3}(3) = 3.5 \\ E([2,0]) &= \frac{3}{2} + \frac{3}{2} = 3 \\ E([0,2]) &= 3 + 3 = 6. \end{aligned}$$

Therefore, allocation B is the minimal allocation.

However, allocation A wins against allocation B in $\frac{5}{9}$ of the games and loses the other $\frac{4}{9}$ of the games. Hence, allocation A beats allocation B in the majority of games. Why? Well, note that allocation A wins all sequences of turns that begin like $(H, T, \ldots)$, $(T, H, \ldots)$ or $(T, T, \ldots)$, which are $\frac{5}{9}$ of the sequences. On the other hand, allocation A only loses sequences of turns that begin like $(H, H, \ldots)$, which are $\frac{4}{9}$ of the sequences. Similarly, we can show that allocation A is favored over allocation C in $\frac{8}{9}$ of the games and loses the other $\frac{1}{9}$ of the games. Therefore, allocation A is the emperor.

2.5 Minimal Allocations

The examples above give a flavor of some of the unexpected results that appear in an analysis of *Knock 'm Down*. Here, we wish to examine minimal allocations in more detail. Recall that a minimal allocation for a fixed probability distribution is an allocation X whose expected clearing time is less than the expected clearing time of all other allocations with the same number of tokens. If X is a minimal allocation, we will denote it as X^*.

There are some properties which we would expect a minimal allocation of a fixed number of tokens to possess. Clearly, the most tokens should appear on the most probable value, while the least number of tokens should appear on the least probable value. Further, values with equal probability should have approximately equal numbers of tokens. One token allocation which satisfies these properties is the distribution of tokens

	X^*											
t	2	3	4	5	6	7	8	9	10	11	12	$E(X^*)$
1	0	0	0	0	0	1	0	0	0	0	0	6.000
2	0	0	0	0	1	1	0	0	0	0	0	9.927
3	0	0	0	0	1	1	1	0	0	0	0	12.505
4	0	0	0	1	1	1	1	0	0	0	0	15.476
5	0	0	0	1	1	1	1	1	0	0	0	17.768
6	0	0	0	1	1	2	1	1	0	0	0	19.762
7	0	0	0	1	2	2	1	1	0	0	0	22.279
8	0	0	0	1	2	2	2	1	0	0	0	24.306
9	0	0	1	1	2	2	2	1	0	0	0	26.430
10	0	0	1	1	2	3	2	1	0	0	0	28.267
11	0	0	1	1	2	3	2	1	1	0	0	29.865
12	0	0	1	2	2	3	2	1	1	0	0	31.922
13	0	0	1	2	2	3	2	2	1	0	0	33.700
18	0	0	1	2	4	5	3	2	1	0	0	42.665
24	0	1	2	3	4	5	4	3	2	0	0	52.139
30	0	1	2	4	5	7	5	3	2	1	0	60.772
36	0	1	3	4	6	8	6	4	3	1	0	69.569

Figure 8: Minimal Allocations for t-token Games with the Original *Knock 'm Down* Probabilities of $P = (\frac{1}{36}, \frac{2}{36}, \frac{3}{36}, \frac{4}{36}, \frac{5}{36}, \frac{6}{36}, \frac{5}{36}, \frac{4}{36}, \frac{3}{36}, \frac{2}{36}, \frac{1}{36})$

in the same proportions as the probability vector. However, even when possible, we have already seen that this may not be the minimal allocation. For example, if $P = (\frac{1}{2}, \frac{1}{3}, \frac{1}{6})$ and $t = 6$, we might well expect the minimal allocation optimal of tokens to be $[3, 2, 1]$, but $[4, 2, 0]$ is the minimal allocation.

Figures 8, 9, 10, and 11 show the minimal allocations for a variety of probability distributions and tokens.

We reach a number of interesting conclusions from studying these results. First, we note that these calculations support the conjecture that every minimal allocation of t tokens contains a minimal allocation of $t-1$ tokens as a sub-allocation. However, these calculations show that it is rarely the case that the minimal allocation is exactly proportional to the relative probability of the values. In fact, the case of $P = (\frac{2}{3}, \frac{1}{3})$ seems to suggest that the deviation of the minimal allocation of tokens from the distribution by relative probabilities increases with the number of tokens.

One other expectation that we might have is the following. Suppose we wish to find the minimal allocation X^* for $P = (p_1, p_2, \ldots, p_N)$ with t tokens. Further, suppose we know x_1^*, but not $x_2^*, \ldots, x_N^*$. Then it may

	X^*		
t	$\frac{2}{3}$	$\frac{1}{3}$	$E(X^*)$
1	1	0	1.500
2	2	0	3.000
3	2	1	4.333
4	3	1	5.389
5	4	1	6.593
6	5	1	7.895
7	5	2	8.949
8	6	2	10.053
9	7	2	11.261
300	204	96	313.325
600	405	195	618.844
900	606	294	923.093
1200	807	393	1226.673

Figure 9: Minimal Allocations for t tokens with $P = (\frac{2}{3}, \frac{1}{3})$

be the case that we can let $X' = [x'_2, \dots, x'_N]$ be the minimal allocation for $P' = (\frac{p_2}{p_2+\cdots+p_N}, \dots, \frac{p_N}{p_2+\cdots+p_N})$ with $t - x_1^*$ tokens, with the hope that $X^* = [x_1^*, X']$. In some cases, this appears to be true. Take $P = (\frac{3}{6}, \frac{2}{6}, \frac{1}{6})$ with $t = 10$. We have $X^* = [6, 3, 1]$. Further, $P' = (\frac{2}{3}, \frac{1}{3})$ and $t - x_1^* = 4$. Thus $X' = [3, 1]$ and $X^* = [x_1^*, X']$. However, this is not always the case. Take $P = (\frac{4}{10}, \frac{3}{10}, \frac{2}{10}, \frac{1}{10})$ with $t = 14$. We have $X^* = [7, 5, 2, 0]$. Further, $P' = (\frac{3}{6}, \frac{2}{6}, \frac{1}{6})$ and $t - x_1^* = 7$. However, $X' = [4, 2, 1]$ and $X \neq (x_1^*, X')$.

Finally, we might also expect the minimal allocation to be unique for a fixed probability distribution and a fixed number of tokens. However, this is not the case. Consider Figure 8. The minimal allocation for 7 tokens which appears on the table is $[0, 0, 0, 1, 2, 2, 1, 1, 0, 0, 0]$, but it should be clear that the allocation $[0, 0, 0, 1, 1, 2, 2, 1, 0, 0, 0]$ has the same expected clearing time, since the values 6 and 8 have the same probability of being rolled on a turn. So, we can revise our expectation and hope that minimal allocations are unique up to symmetries. Unfortunately, this is not the case either. Using the methods described in Section 6, we can show that for $P = (\frac{\sqrt{5}-1}{2}, \frac{3-\sqrt{5}}{2})$, the allocations $[2, 0]$ and $[1, 1]$ have the same expected clearing time, and both are less than the expected clearing time of $[0, 2]$. Despite these setbacks, we will show in Section 4 that there are some necessary conditions which all minimal allocations must satisfy.

t	X^* $\frac{3}{6}$	$\frac{2}{6}$	$\frac{1}{6}$	$E(X^*)$
1	1	0	0	2.000
2	1	1	0	3.800
3	2	1	0	5.080
4	3	1	0	6.648
5	3	2	0	8.074
6	4	2	0	9.400
7	4	2	1	10.923
8	5	2	1	12.149
9	5	3	1	13.207
10	6	3	1	14.387
11	6	4	1	15.684
12	7	4	1	16.802
13	8	4	1	18.104
14	8	5	1	19.342
15	9	5	1	20.568

Figure 10: Minimal Allocations for t tokens with $P = (\frac{3}{6}, \frac{2}{6}, \frac{1}{6})$

	X^*				
t	.4	.3	.2	.1	$E(X^*)$
01	1	0	0	0	2.500
02	1	1	0	0	4.405
03	2	1	0	0	6.088
04	2	1	1	0	7.916
05	3	1	1	0	9.428
06	3	2	1	0	10.569
07	4	2	1	0	12.079
08	4	3	1	0	13.519
09	5	3	1	0	14.979
10	5	3	2	0	16.327
11	5	4	2	0	17.724
12	6	4	2	0	18.923
13	7	4	2	0	20.395
14	7	5	2	0	21.739
15	7	5	3	0	23.057

Figure 11: Minimal Allocations for t tokens with $P = (.4, .3, .2, .1)$

t	Emperor
01	[1,0,0]
02	[1,1,0]
03	[2,1,0]
04	[2,1,1]
05	—
06	[3,2,1]
10	[5,3,2]
12	[6,4,2]
18	—

Figure 12: Emperors for t tokens with $P = (\frac{3}{6}, \frac{2}{6}, \frac{1}{6})$

2.6 Tournaments, Emperors and Emperor Cycles

Having considered the minimal allocations for a variety of games, we now turn our attention to the emperors and emperor cycles. Recall that an emperor is an allocation that is favored over all other allocations. If X is an emperor, we will denote it as $\hat{X}$. Similarly, an emperor cycle is the smallest set of allocations such that every allocation in the cycle is favored over every allocation not in the cycle. Note that an emperor, if it exists, is a member of the emperor cycle.

Intuitively, we might expect that the minimal allocation, having the smallest expected clearing time, would be likely to win when competing against any other allocation – but, we have already seen that this is not the case. We might also have intuitively thought that favoring would be a transitive property; i.e., if X is favored over Y and Y is favored over Z, then X is favored over Z. The existence of emperor cycles shows that this conjecture is false.

Recall the 3-valued game with $P = (\frac{3}{6}, \frac{2}{6}, \frac{1}{6})$. The emperors for different numbers of tokens are shown in Figure 12. Even with only these eight examples, we begin to see curious phenomenon. First, we note that for tournaments with 1, 2 and 3 tokens, the emperor is the minimal allocation for the same number of tokens. However, beginning with tournaments of 4 tokens, the emperor is no longer the allocation with minimum expected clearing time. In fact, tournaments with 5 and 18 tokens have emperor cycles instead of emperors, while tournaments with 6 and 12 tokens have emperors which are distributions by relative probability.

The existence of emperor cycles seriously compromises our ability to conclusively answer the question of which allocation we should choose to maximize our chances of winning the game. If we know our opponent's allocation, then we can always choose to play an allocation which is favored over it (unless we are both playing emperors). On the other hand, if our opponent's allocation is unknown and the tournament has an emperor

t	Emperor
10	[5,3,2,0]
20	[9,6,4,1]

Figure 13: Emperors for t tokens with $P = (.4, .3, .2, .1)$ for *Independent Knock 'm Down*

cycle, it becomes difficult to answer the question of which allocation is "best" to play, especially taking into account the fact that our opponent may be pursuing a similar analysis on which allocation we intend to play. Further, emperor cycles may not accurately capture our idea of the "best" allocations. In Figure 5, we saw that the emperor cycle in the 5-token game was comprised of 4 allocations, all of which seemed reasonable from an intuitive point of view. However, the emperor cycle in the 18-token game is comprised of 70 allocations, and includes allocations such as $[3, 10, 5]$ and $[17, 0, 1]$, which do not seem reasonable from an intuitive point of view.

Surprisingly, cycles occur quite often in *Knock 'm Down* tournaments, although non-emperor cycles are less interesting in terms of choosing a best allocation. In Figure 6, three distinct cycles occur among the allocations $[3, 0, 0]$, $[1, 1, 1]$, $[1, 2, 0]$, and $[2, 0, 1]$ in the tournament of 3 tokens, even though $[2, 1, 0]$ is an emperor allocation for this game. In the tournament of 10 tokens with the same probability vector, there are over $42,000$ distinct cycles occurring among the 66 allocations, with the longest cycle passing through 13 different allocations.

In hindsight, we might suspect that the cycles appearing in *Knock 'm Down* tournaments occur because the players are using the same value to remove tokens. We briefly consider a variation of *Knock 'm Down* called *Independent Knock 'm Down.* Each player distributes tokens as in the original *Knock 'm Down*, but on each turn, a random value is produced for each player and they remove tokens accordingly. Hence, we could turn the original description of *Knock 'm Down* into an instance of *Independent Knock 'm Down* by equipping each player with a pair of dice that they roll on each turn. Notice that in *Independent Knock 'm Down*, players could simply compare the number of turns each required to clear his board to determine the winner of the game. Despite the strange behavior of the original *Knock 'm Down*, we might reasonably believe that an allocation's expected clearing time is an accurate measure of its ranking in the associated tournament in *Independent Knock 'm Down.*

Unfortunately, *Independent Knock 'm Down* neither equates minimal allocations with emperors nor eliminates cycles. We return to the game with $P = (.4, .3, .2, .1)$. Figure 13 shows the emperors in the *Independent* version of this game for 10 and 20 tokens. In the case of 10 tokens, we find that the emperor does indeed correspond to the minimal allocation. However, in the case of 20 tokens, the emperor corresponds to an allocation that is neither $[10, 6, 3, 1]$, the minimal allocation, nor $[8, 6, 4, 2]$, the distribution by relative probabilities, although this allocation did appear in the emperor cycle for 20 tokens in the original *Knock 'm*

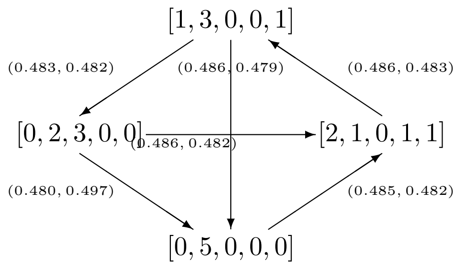

Figure 14: Cycles in the 5-Token Game with $P = (\frac{7}{18}, \frac{5}{18}, \frac{3}{18}, \frac{2}{18}, \frac{1}{18})$ for *Independent Knock 'm Down*

Down game. In fact, $[9, 6, 3, 1]$ wins against $[10, 6, 3, 1]$ in 48.18% of the games and loses against $[10, 6, 3, 1]$ in 46.20% of the games, while $[9, 6, 3, 1]$ wins against $[8, 6, 4, 2]$ in 51.83% of the games and loses against $[8, 6, 4, 2]$ in 43.13% of the games.

Independent Knock 'm Down tournaments also suffer from non-transitive favorability, although cycles in these tournaments are much less common than they are in the original *Knock 'm Down* tournaments. One example which has been found occurs in the case of $P = (\frac{7}{18}, \frac{5}{18}, \frac{3}{18}, \frac{2}{18}, \frac{1}{18})$. In that tournament we have $[1, 3, 0, 0, 1]$ favored over $[0, 2, 3, 0, 0]$ favored over $[0, 5, 0, 0, 0]$ favored over $[2, 1, 0, 1, 1]$ favored over $[1, 3, 0, 0, 1]$, as shown in Figure 14. On the other hand, no emperor cycle has been found in an *Independent Knock 'm Down* tournament; rather, every tournament we examined has had an emperor.

Finally, we wish to note that the tournaments described are not true tournaments. To expound on this fact, we need to clarify the definition that an allocation X is favored over an allocation Y. Generally, we have interpreted this to mean that the probability that X wins against Y is greater than the probability that X loses against Y. However, when an allocation X plays itself, it's probability of winning and of losing are both zero. Similarly, if X and Y are allocations that have equal numbers of tokens on values with equal probability, then the probability that X wins against Y is equal to the probability that X loses against Y. For example, if $P = (\frac{2}{4}, \frac{1}{4}, \frac{1}{4})$ then we have that $[1, 1, 0]$ wins against $[1, 0, 1]$ in 41.67% of the games (42.12% of the games in *Independent Knock 'm Down*) and loses with the same probability. These two cases can be considered trivial, since we can clearly recognize them *a priori*.

However, there are examples of non-trivial allocations which have equal probability of winning the game. With $P = (\frac{\sqrt{2}}{2}, 1 - \frac{\sqrt{2}}{2})$, we have that $[2, 0]$ wins against $[1, 1]$ with probability $\frac{1}{2}$ and $[2, 0]$ loses against $[1, 1]$ with probability $\frac{1}{2}$. A similar pair of probabilities can be found such that $[2, 0]$ and $[1, 1]$ have equal probability of winning in *Independent Knock 'm Down*, although the probabilities are solutions to a fifth degree polynomial and are not expressible in radicals. These examples show that a *Knock 'm Down* tournament in which a win is awarded to the allocation which has the greater probability of winning is not always a true tournament; i.e., we must decide whether there are some pairs of allocations which have two edges in the tournament graph (introducing trivial two-cycles) or have no edges in the tournament graph (complicating

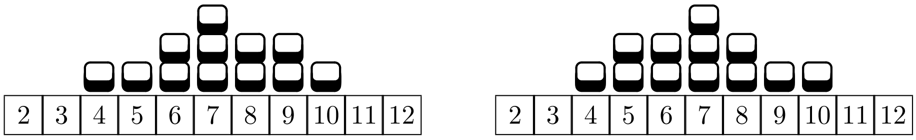

Figure 15: Minimal Allocations and Emperors for the 12-Token Game with the Original *Knock 'm Down* Probabilities

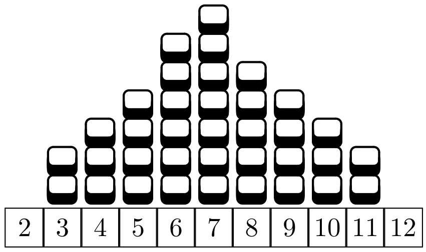

Figure 16: A Local Emperor for the 36-Token Game with the Original *Knock 'm Down* Probabilities

the definition of emperor and emperor cycles). We elect to take the first alternative, since our tournaments will then be super-graphs of true tournament graphs, potentially allowing us to make use of results that hold on tournament graphs. Therefore, we will adopt the convention that an allocation is favored over another allocation if the first allocation is at least as likely to win as to lose; formally, we say that allocation X is favored over allocation Y if $Pr(X \text{ wins against } Y) \geq Pr(Y \text{ wins against } X)$. However, this will force us to define an emperor cycle to be the smallest set of allocations such that every allocation in the set is favored over every allocation not in the cycle, and no allocation not in the cycle is favored over any allocation in the cycle. This new definition does not invalidate any of the examples described above, but it will have bearing on some of the subsequent results.

2.7 Solution to the Original *Knock 'm Down*

After considering all of these results, we might wonder if there is any hope of choosing a "best" allocation in the original *Knock 'm Down* game, with 12 tokens played on a board with the values corresponding to the roll of a pair of dice. Surprisingly, this instance of the game has a rather tidy solution. We have verified that the allocations $[0, 0, 1, 1, 2, 3, 2, 2, 1, 0, 0]$ and $[0, 0, 1, 2, 2, 3, 2, 1, 1, 0, 0]$ are both minimal allocations and emperors (see Figure 15).

This should not imply that the original *Knock 'm Down* game is free of surprises. Consider the case with 36 tokens. In that game, $[0, 1, 3, 4, 6, 8, 6, 4, 3, 1, 0]$ is the minimal allocation. This is consistent with other examples above, where the allocation according to the probability distribution failed to be minimal. Further, allocation $[0,1,3,4,6,8,6,4,3,1,0]$ wins against $[1,2,3,4,5,6,5,4,3,2,1]$ in 45.14% of the games and loses against $[1,2,3,4,5,6,5,4,3,2,1]$ in 30.12% of the games, a decisive "victory" for the minimal allocation. On the other hand, the allocation $[0,2,3,4,6,7,6,4,3,1,0]$ is favored over $[0,1,3,4,6,8,6,4,3,1,0]$, winning 19.40% of the games and losing 12.93% of the games. Hence, $[0,1,3,4,6,8,6,4,3,1,0]$ is not an emperor. We conjecture that the allocations $[0,2,3,4,6,7,5,4,3,2,0]$ $[0,2,3,4,5,7,6,4,3,2,0]$ are emperors in this game (see Figure 16), although this has not been verified. We have been able to verify that these allocations are at least *local emperors* in that each is favored over all *neighboring* allocations, i.e., all allocations reachable by moving a single token. In addition, these allocations are favored over the allocation $[1,2,3,4,5,6,5,4,3,2,1]$.

3 Notation and Generalization

Having considered a variety of results that occur when playing particular instances of *Knock 'm Down*, we now wish to turn our attention to some general results that hold for all instances of the game. In order to consider the largest set of games similar to the original description of *Knock 'm Down*, we will adopt the following conventions and notations to describe a generalized version of *Knock 'm Down.* Let N be an integer representing the number of values labeled on a player's board. For the original version of the game described in Section 1, N is 11. Let $P = (p_1, p_2, \ldots, p_N)$ be a probability vector; i.e., $0 < p_i < 1$ and $\sum_{i=1}^{N} p_i = 1$. In most cases, we will write this probability vector as a non-increasing sequence ($p_1 \geq p_2 \geq \cdots \geq p_N$) and in Section 6 we will occasionally require the probability vector to be a non-increasing sequence. However, the results in other sections are not affected by using an arbitrary probability vector. On each turn of the game, a value on the board is produced by some random process. The i^{th} value is produced with probability p_i. Finally, let t be the number of tokens to be allocated on the board by a player. For notational convenience, we will call this the *t-token game with* $P = (p_1, p_2, \ldots, p_N)$. Hence, the original *Knock 'm Down* game in Section 1 describes a 12-token game with $P = (\frac{1}{36}, \frac{1}{18}, \frac{1}{12}, \frac{1}{9}, \frac{5}{36}, \frac{1}{6}, \frac{5}{36}, \frac{1}{9}, \frac{1}{12}, \frac{1}{28}, \frac{1}{36})$.

Let $X = [x_1, x_2, \ldots, x_N]$ be a vector of non-negative integers, representing the token allocation with x_i tokens on the i^{th} value. Let e_i be the i^{th} unit vector of N elements, so $X + e_i$ represents the token allocation identical to X except incremented by one token on the i^{th} value. Since a player has t tokens to allocate, an initial token allocation X satisfies $\sum_{i=1}^{N} x_i = t$. The allocations of players A and B in Figure 1 would be represented as $A = [1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 1]$ and $B = [0, 0, 1, 1, 2, 4, 2, 1, 1, 0, 0]$ respectively. We define the *neighbors* of an allocation X to be all valid allocations of the form $X + e_i - e_j$; i.e., those allocations which can be reached from X by the transfer of a single token.

Next we consider three versions of this generalized *Knock 'm Down. Solitaire Knock 'm Down* is for a single player. The player arranges his t tokens among the N values. Each turn a random value is produced and the player may remove a token from that value if he has one or more tokens there. The player attempts to arrange his tokens so as to minimize the number of turns required to clear his initial allocation. *Original Knock 'm Down* and *Independent Knock 'm Down* are for two players racing to be the first to remove all of his tokens. Each player distributes his tokens among the N values. For *Original Knock 'm Down*, a single random value is produced each turn and each player may remove a token from that value if he has one or more tokens there. For *Independent Knock 'm Down*, two random values are produced each turn, one for each player, who may only remove a token from his generated value if he has one or more tokens there. Naturally, the first player to remove all tokens is the winner and if both players remove their last token on the same turn, then the game is a draw. Hence, the description of *Knock 'm Down* in Section 1 is played under *Original* rules, but may easily be adapted to *Independent* rules by equipping each player with two dice. Notice that while draws are possible under both sets of two player rules, not all pairs of initial allocations in

Original games can result in draws, while all pairs of initial allocations in *Independent* games can result in draws. We will concentrate our study on *Original Knock 'm Down* and will frequently refer to it as simply *Knock 'm Down* when it is clear from context that a two-player game is being discussed.

Finally, in order to answer the question of which allocation will maximize a player's chances of winning in a particular version of *Knock 'm Down*, we consider the methods by which allocations can be compared. In *Solitaire Knock 'm Down*, the player wishes to minimize the number of turns required to clear his board. If T_X is the random value representing the number of turns required to remove all tokens from the allocation X, then an optimal allocation in *Solitaire Knock 'm Down* is an allocation which minimizes $E(T_X)$; that is, an allocation which minimizes the expected number of turns required to clear the allocation. In general, $E(T_X)$ will rank all of the allocations in a t-token game with fixed P in an order such that the allocations which minimize $E(T_X)$ are the most desirable and the allocations which maximize $E(T_X)$ are the least desirable. If X is an allocation in a t-token game with fixed P such that $E(T_X) \leq E(T_Y)$ for all other allocations in the game, then we call X a *minimal allocation* for the game. A minimal allocation will typically be denoted as X^*. Recall from Section 2 that minimal allocations are not necessarily unique.

In *Original Knock 'm Down* and *Independent Knock 'm Down*, allocations can be compared either by direct competition or by tournament play. To compare two allocations by direct competition, we introduce the notion of a WDL (pronounced "widdle") function. If X and Y are two allocations which share the same probability vector (although not necessarily with the same number of tokens), we say $WDL(X,Y) = (w,d,l)$, where $w = Pr(X \text{ wins against } Y)$, $d = Pr(X \text{ draws against } Y)$, and $l = Pr(X \text{ loses against } Y)$. Naturally, it must always be true that $w + d + l = 1$. We let WDL_O be the WDL function which corresponds to a game played under *Original* rules and let WDL_I be the WDL function which corresponds to a game played under *Independent* rules. However, since we will concentrate our study on *Original Knock 'm Down*, we will frequently write WDL instead of WDL_O.

Let X and Y be allocations in the same game and $WDL(X,Y) = (w,d,l)$. If $w \geq l$, then we say that allocation X is *favored* over allocation Y. Hence, a player given a choice to play one allocation against another would choose the favored allocation to maximize his chances of winning. However, in general, a player can choose any allocation from the t-token game with fixed P. Thus, we can construct a tournament in which every allocation is compared by direct competition with every other allocation by means of the appropriate WDL function. This tournament can be considered as a graph, with vertices drawn from the set of allocations and and edge from X to Y if X is favored over Y. In the two-player versions of *Knock 'm Down*, an optimal allocation is one that is favored over the most other allocations. In general, this tournament will rank all of the allocations in a t-token game with fixed P in an order such that the allocations which are favored over the most other allocations are the most desirable and the allocations which are favored over the least other allocations are the least desirable.

Suppose there is an allocation X in a t-token game with fixed P which is favored over all other allocations. Then, using terminology from [5], we call such an allocation an *emperor* of the game. If X is an emperor, we will denote it as $\hat{X}$ (since the emperor traditionally wears a hat!). Recall from Section 2 that an emperor need not exist nor be unique for a particular instance. Hence, we define an *emperor cycle* to be the smallest set of allocations such that every allocation in the cycle is favored over all allocations not in the cycle, and no allocation not in the cycle is favored over any allocation in the cycle. The existence of emperor cycles demonstrates that the allocations which are favored over the most other allocations may not always be "best" in competition. In fact, emperor cycles demonstrate that in some instances, no strategy (in the sense of choosing a single allocation for all competitions) is "best." An intriguing possibility, but one will will not be investigated here, is to consider a "mixed strategy," where allocations in some subset of the emperor cycle are assigned probabilities of being played in an arbitrary competition to maximize our chances of winning the game. A study of these mixed strategies from a game theoretic point of view could be very fruitful.

4 Theoretical Results

In Section 2, we saw that many of our conjectured relationships between the distribution of tokens by relative probabilities, the minimal allocations, and the emperors or emperor cycles failed to hold in general. Ideally, we would like to have available theorems which could directly construct the minimal allocation (for an optimal solution to *Solitaire Knock 'm Down*) and the emperor or emperor-cycle (for an optimal solution to *Knock 'm Down* and *Independent Knock 'm Down*) from the probability distribution and the number of tokens. Unfortunately, the unintuitive nature of the game suggests that such a theorem would be very complicated for the N-valued game. In Section 5, we will prove two such theorems in the special case of a 2-valued game. In this section, we will demonstrate that significant progress towards a characterization of the allocations which are minimal has been made.

4.1 The Minimal Allocation Theorems

Recall that in Section 2.5, we discussed some of the properties we would expect a minimal allocation to posses. Here, we wish to formalize those intuitive ideas and to prove that they are indeed true. In this discussion, we are interested in the t-token game with $P = (p_1, p_2, \dots, p_N)$ and an allocation $X^* = [x_1^*, x_2^*, \dots, x_N^*]$ will always denote a minimal allocation for the game.

Our first intuitive notion is that values with greater probability should have a greater number of tokens. Formally, we write

$$\text{if } p_a < p_b, \text{ then } x_a^* \leq x_b^*. \tag{1}$$

We note that when $p_a < p_b$ it is certainly possible that $x_a^* = x_b^*$. For example, in the 2-token game with $P = (\frac{3}{6}, \frac{2}{6}, \frac{1}{6})$, the optimal allocation is $X^* = [1, 1, 0]$. Another result that we might expect is the following:

$$\text{if } p_a = p_b, \text{ then } |x_a^* - x_b^*| \leq 1. \tag{2}$$

This result coincides with our intuition that values having equal probability should have equal numbers of tokens. Alternatively, we can assert that any allocation that does not satisfy this result can be improved by "evening out" the distribution of tokens.

We will shortly show that (2) is true for all minimal allocations X^*. We have often made reference to the fact that our intuition for this game suggests that the minimal allocation of tokens should resemble the histogram of probabilities – but we have seen numerous examples that this is not the case. What is not apparent from the examples is the following result that we will prove, which implies that the optimal allocation must at least "respect" proportions, in that $\frac{x_a^*}{x_b^*}$ cannot exceed $\frac{p_a}{p_b}$ by much. Specifically,

$$\text{if } p_a < p_b, \text{ then } p_b(x_a^* - 1) < p_a x_b^*. \tag{3}$$

We note that the 2 token example above illustrates that the stronger conclusion $p_b x_a^* < p_a x_b^*$ is not attainable. Notice that (3) implies and is strictly stronger than (1).

We proceed in showing these results in three steps. First, we will prove the *Token Adding Theorem* which will allow us to consider what happens to an allocation's expected clearing time when a new token is added. The insight gained in proving this theorem will allow us to prove the *Second Token Adding Theorem* which makes use of a different set of hypotheses to reach the same conclusion. Finally, these two theorems will allow us to prove two *Minimal Allocation Theorems* which correspond to (3) and (2).

4.2 The *Token Adding Theorem*

The starting point for proving the *Minimal Allocation Theorems* is the *Token Adding Theorem.* Surprisingly, the *Token Adding Theorem* does not deal with minimal allocations at all. Instead, we consider situations in which we are able to determine when it is wrong (in the sense of minimizing the expected clearing time) to add a token to the less probable of two values. Recall that T_X is the random variable denoting the number of turns required to remove all tokens from the allocation X and $X + e_i$ is the token allocation identical to X except incremented by one token on the i^{th} value. Then the theorem is stated as follows:

Token Adding Theorem. *Let X be a token allocation with t tokens such that $p_b x_a \geq p_a x_b$, where $p_a < p_b$. Then $E[T_{X+e_a}] > E[T_{X+e_b}]$.*

Hence, if $\frac{x_a}{x_b} \geq \frac{p_a}{p_b}$, we are guaranteed that adding the token to the more probable value is the better choice. However, if the conditions of the theorem are not satisfied, then it is unclear whether adding the token to the more probable or less probable value is the better choice.

We will prove the *Token Adding Theorem* by showing that "the probability that the last token to remove in $X + e_a$ is the added a token" is greater than the probability that "the last token to remove in $X + e_b$ is the added b token." This, coupled with the fact that the expected time to clear one a token is longer than the expected time to clear one b token, will allow us to prove the theorem with ease.

4.2.1 The *Token Adding Theorem*

Token Adding Theorem. *Let X be a token allocation with t tokens such that $p_b x_a \geq p_a x_b$, where $p_a < p_b$. Then $E[T_{X+e_a}] > E[T_{X+e_b}]$.*

Proof. Let X be a token allocation with t tokens such that $p_b x_a \geq p_a x_b$, where $p_a < p_b$.

Let $X_a = X + e_a$ and $X_b = X + e_b$.

We wish to show $E[T_{X_a}] > E[T_{X_b}]$. Equivalently, we show that $E[T_{X_a}] - E[T_{X_b}] > 0$.

Note $T_{X_a} = T_X + R_a$, where R_a is the number of turns needed to clear X_a after clearing X; i.e., it is the "rest" of the turns needed to remove the added a if it remains after clearing X. By the linearity of expectation, $E[T_{X_a}] = E[T_X] + E[R_a]$. After X is cleared, we either have 0 tokens or 1 token remaining on a. Thus, $E[R_a] = \frac{1}{p_a} Pr(R_a > 0)$. Note that $Pr(R_a > 0)$ is the probability X is cleared with exactly x_a a's. Similarly, we note that $E[T_{X_b}] = E[T_X] + E[R_b]$, where $E[R_b] = \frac{1}{p_b} Pr(R_b > 0)$ and $Pr(R_b > 0)$ is the probability X is cleared with exactly x_b b's. Hence,

$$\begin{aligned} E[T_{X_a}] - E[T_{X_b}] &= \left(E[T_X] + \frac{1}{p_a} Pr(R_a > 0) \right) - \left(E[T_X] + \frac{1}{p_b} Pr(R_b > 0) \right) \\ &= \frac{1}{p_a} Pr(R_a > 0) - \frac{1}{p_b} Pr(R_b > 0) \end{aligned}$$

and it suffices to show that $\frac{1}{p_a} Pr(R_a > 0) > \frac{1}{p_a} Pr(R_b > 0)$.

For reasons which will become apparent shortly, let $A_{n,m}$ be the set of sequences of values of length n such that X is cleared on the n^{th} turn with exactly x_a a's and $x_b + m$ b's. Note that $A_{n,m} = \emptyset$ for some values of N, t, X, n, and m. Define $Pr(A_{n,m}) = \sum_{\alpha \in A_{n,m}} Pr(\alpha)$, where $Pr(\alpha)$ is simply the product of the probabilities of the values in the sequence. Similarly, let $B_{n,m}$ be the set of sequences of values of length n such that X is cleared on the n^{th} turn with exactly x_b b's and $x_a + m$ a's and use an analogous definition for $Pr(B_{n,m})$.

Thus, we have the following:

$$Pr(R_a > 0) = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} Pr(A_{n,m})$$

and

$$Pr(R_b > 0) = \sum_{n=0}^{\infty} \sum_{m=0}^{\infty} Pr(B_{n,m}).$$

As we noted earlier, $A_{n,m}$ and $B_{n,m}$ equal $\emptyset$ for some values of N, t, X, n, and m. In particular, $A_{n,m} = B_{n,m} = \emptyset$ for $n < t$ and for $m > n - t$, regardless of the values of N and X. Therefore, we can "simplify" the equations to:

$$Pr(R_a > 0) = \sum_{n=t}^{\infty} \sum_{m=0}^{n-t} Pr(A_{n,m})$$

and

$$Pr(R_b > 0) = \sum_{n=t}^{\infty} \sum_{m=0}^{n-t} Pr(B_{n,m}).$$

Recall that we wish to show $\frac{1}{p_a} Pr(R_a > 0) > \frac{1}{p_a} Pr(R_b > 0)$. Based on the definitions above, it suffices to show that

$$\frac{1}{p_a} Pr(A_{n,m}) \geq \frac{1}{p_b} Pr(B_{n,m})$$

for all n and m and that the inequality is strict in at least one case. Taking this fact (which will be demonstrated in a lemma shortly) on faith for the time being, we note that we can now claim $\frac{1}{p_a} Pr(R_a > 0) > \frac{1}{p_b} Pr(R_b > 0)$, and conclude that $E[T_{X+e_a}] > E[T_{X+e_b}]$. $\square$

4.2.2 Proof of the *Token Adding Lemma*

Here we prove the lemma which we required in the proof of the *Token Adding Theorem.*

Token Adding Lemma. *Let X be a token allocation with t tokens and N values such that $p_b x_a \geq p_a x_b$, where $p_a < p_b$. Then for $n > t$ and $0 \leq m \leq n - t$, $\frac{1}{p_a} Pr(A_{n,m}) \geq \frac{1}{p_b} Pr(B_{n,m})$. Further, the inequality is strict in the case $n = t$ and $m = 0$.*

Proof. Let X be a token allocation with t tokens and N values such that $p_b x_a \geq p_a x_b$, where $p_a < p_b$.

We begin with the case $n = t$ and $m = 0$. Recalling the definition of $A_{n,m}$ and $B_{n,m}$, we note that both $A_{t,0}$ and $B_{t,0}$ are the set of sequences of values of length n such that X is cleared on the t^{th} turn with exactly x_a a's and x_b b's. Hence $A_{t,0} = B_{t,0}$ and $Pr(A_{t,0}) = Pr(B_{t,0})$. Further, since $n = t$, neither $A_{t,0}$ nor $B_{t,0}$ is empty (with the possibility that $A_{t,0}$ and $B_{t,0}$ contain only the empty sequence, which has probability 1, in the case $t = 0$). Since, $p_a < p_b$, then $\frac{1}{p_a} Pr(A_{t,0}) > \frac{1}{p_b} Pr(B_{t,0})$.

Next, consider the case $n > t$ and $m = 0$. Again, we note that $A_{n,0} = B_{n,0}$ and $Pr(A_{n,0}) = Pr(B_{n,0})$. Thus, $\frac{1}{p_a} Pr(A_{t,0}) \geq \frac{1}{p_b} Pr(B_{t,0})$. The inequality is strict except when $N = 2$ and $m > t$, where $Pr(A_{n,0}) = Pr(B_{n,0}) = 0$.

Finally, consider the case $n > t$ and $0 < m \leq n - t$.

Let X' be the allocation of $t - x_a - x_b$ tokens with the x_a a's and x_b b's removed from X. That is, let $X' = X - x_a e_a - x_b e_b$.

We express $Pr(A_{n,m})$ as the sum of two probabilities: $Pr(A_{n,0}) = A_1 + A_2$, where

$$A_1 = Pr(X \text{ is cleared on the } n^{th} \text{ turn with exactly } x_a \ a\text{'s and } x_b + m \ b\text{'s,}$$
$$\text{and the last turn is an } a)$$
$$A_2 = Pr(X \text{ is cleared on the } n^{th} \text{ turn with exactly } x_a \ a\text{'s and } x_b + m \ b\text{'s,}$$
$$\text{and the last turn is neither an } a \text{ nor a } b).$$

Note that either or both of the terms may be zero for some values of N, t, X, n, and m.

Likewise, we express $P[B_{n,m}]$ as the sum of two probabilities: $Pr(B_{n,0}) = B_1 + B_2$, where

$$B_1 = Pr(X \text{ is cleared on the } n^{th} \text{ turn with exactly } x_b \ b\text{'s and } x_a + m \ a\text{'s,}$$
$$\text{and the last turn is a } b)$$
$$B_2 = Pr(X \text{ is cleared on the } n^{th} \text{ turn with exactly } x_b \ b\text{'s and } x_a + m \ a\text{'s,}$$
$$\text{and the last turn is neither a } b \text{ nor an } a).$$

Note that one or more of the terms may be zero for some values of N, t, X, n, and m.

In order to show that $\frac{1}{p_a} Pr(A_{n,m}) \geq \frac{1}{p_b} Pr(B_{n,m})$, we show that $\frac{1}{p_a} A_1 \geq \frac{1}{p_b} B_1$ and $\frac{1}{p_a} A_2 \geq \frac{1}{p_b} B_2$.

First, show $\frac{1}{p_a}A_1 \geq \frac{1}{p_b}B_1$.

If $B_1 = 0$, then clearly $\frac{1}{p_a}A_1 \geq \frac{1}{p_b}B_1$.

On the other hand, if $B_1 \neq 0$, then it suffices to show that $\frac{\frac{1}{p_a}A_1}{\frac{1}{p_b}B_1} \geq 1$.

Note, for $B_1 \neq 0$,

$$A_1 = \binom{n-1}{x_a-1}\binom{n-x_a}{x_b+m} p_a^{x_a} p_b^{x_b+m} Pr(X' \text{ is cleared within } n - x_a - x_b - m \text{ relevant turns})$$

$$B_1 = \binom{n-1}{x_b-1}\binom{n-x_b}{x_a+m} p_b^{x_b} p_a^{x_a+m} Pr(X' \text{ is cleared within } n - x_a - x_b - m \text{ relevant turns}),$$

where relevant turns for X' are non-a and non-b randomly generated values. Also note that X' is cleared *within* $n - x_a - x_b - m$ turns.

Then,

$$\begin{aligned}
\frac{\frac{1}{p_a}A_1}{\frac{1}{p_b}B_1} &= \frac{\frac{1}{p_a}\binom{n-1}{x_a-1}\binom{n-x_a}{x_b+m}p_a^{x_a}p_b^{x_b+m}Pr(X' \ldots)}{\frac{1}{p_b}\binom{n-1}{x_b-1}\binom{n-x_b}{x_a+m}p_b^{x_b}p_a^{x_a+m}Pr(X' \ldots)} \\
&= \frac{\frac{(n-1)!}{(x_a-1)!(x_b+m)!(n-x_a-x_b-m)!}p_b^{m+1}}{\frac{(n-1)!}{(x_b-1)!(x_a+m)!(n-x_a-x_b-m)!}p_a^{m+1}} \\
&= \left(\frac{p_b}{p_a}\right)^{m+1} \frac{(x_b-1)!(x_a+m)!}{(x_a-1)!(x_b+m)!} \\
&= \left(\frac{p_b}{p_a}\right)^{m+1} \frac{x_a(x_a+1)\cdots(x_a+m)}{x_b(x_b+1)\cdots(x_b+m)} \\
&= \prod_{k=0}^{m} \frac{p_b(x_a+k)}{p_a(x_b+k)}.
\end{aligned}$$

(Note, if $x_b = 0$, then B_1, the probability of clearing X with b on the last turn, would be zero. Hence, the product above is well defined.)

Note, $\frac{p_b(x_a+k)}{p_a(x_b+k)} \geq 1$ iff $p_b(x_a+k) - p_a(x_b+k) \geq 0$.

By assumption, $p_b x_a \geq p_a x_b$ and $p_a < p_b$.

Hence,

$$p_b(x_a+k) - p_a(x_b+k) = p_b x_a - p_a x_b + k(p_b - p_a) \geq 0.$$

Thus, $\frac{p_b(x_a+k)}{p_a(x_b+k)} \geq 1$ and therefore

$$\frac{\frac{1}{p_a}A_1}{\frac{1}{p_b}B_1} = \prod_{k=0}^{m} \frac{p_b(x_a+k)}{p_a(x_b+k)} \geq 1.$$

So, we have $\frac{1}{p_a}A_1 \geq \frac{1}{p_b}B_1$.

Next, we show $\frac{1}{p_a}A_2 \geq \frac{1}{p_b}B_2$.

If $B_2 = 0$, then clearly $\frac{1}{p_a}A_2 \geq \frac{1}{p_b}B_2$.

On the other hand, if $B_2 \neq 0$, then it suffices to show that $\frac{\frac{1}{p_a}A_2}{\frac{1}{p_b}B_2} \geq 1$.

Note, for $B_2 \neq 0$,

$$A_2 = \binom{n-1}{x_a}\binom{n-x_a-1}{x_b+m} p_a^{x_a} p_b^{x_b+m} Pr(X' \text{ is cleared in exactly } n-x_a-x_b-m \text{ relevant turns})$$
$$B_2 = \binom{n-1}{x_b}\binom{n-x_b-1}{x_a+m} p_b^{x_b} p_a^{x_a+m} Pr(X' \text{ is cleared in exactly } n-x_a-x_b-m \text{ relevant turns}),$$

where relevant turns for X' are non-a and non-b randomly generated values. Also note that X' is cleared *in* $n - x_a - x_b - m$, which means that the last token of X' must be removed on the $(n - x_a - x_b - m)^{th}$ turn.

Then,

$$\begin{aligned}
\frac{\frac{1}{p_a}A_2}{\frac{1}{p_b}B_2} &= \frac{\frac{1}{p_a}\binom{n-1}{x_a}\binom{n-x_a-1}{x_b+m} p_a^{x_a} p_b^{x_b+m} Pr(X' \ldots)}{\frac{1}{p_b}\binom{n-1}{x_b}\binom{n-x_b-1}{x_a+m} p_b^{x_b} p_a^{x_a+m} Pr(X' \ldots)} \\
&= \frac{\frac{(n-1)!}{x_a!(x_b+m)!(n-1-x_a-x_b-m)!} p_b^{m+1}}{\frac{(n-1)!}{x_b!(x_a+m)!(n-1-x_a-x_b-m)!} p_a^{m+1}} \\
&= \left(\frac{p_b}{p_a}\right)^{m+1} \frac{x_b!(x_a+m)!}{x_a!(x_b+m)!} \\
&= \left(\frac{p_b}{p_a}\right)^{m+1} \frac{(x_a+1)(x_a+2)\cdots(x_a+m)}{(x_b+1)(x+b+2)\cdots(x_b+m)} \\
&= \frac{p_b}{p_a}\prod_{k=1}^{m} \frac{p_b(x_a+k)}{p_a(x_b+k)}.
\end{aligned}$$

It has already been established that $\frac{p_b(x_a+k)}{p_a(x_b+k)} \geq 1$ and $p_a < p_b$ by assumption, so

$$\frac{\frac{1}{p_a}A_2}{\frac{1}{p_b}B_2} = \frac{p_b}{p_a}\prod_{k=1}^{m} \frac{p_b(x_a+k)}{p_a(x_b+k)} \geq 1.$$

So, we have $\frac{1}{p_a}A_2 \geq \frac{1}{p_b}B_2$.

Hence, we conclude that for $n > t$ and $0 < m \leq n - t$, $\frac{1}{p_a}Pr(A_{n,m}) \geq \frac{1}{p_b}Pr(B_{n,m})$.

Therefore, for $n > t$ and $0 \leq m \leq n - t$, $\frac{1}{p_a}Pr(A_{n,m}) \geq \frac{1}{p_b}Pr(B_{n,m})$. Further, the inequality is strict in the case $n = t$ and $m = 0$. $\square$

4.2.3 The *Second Token Adding Theorem*

Before proving the two *Minimal Allocation Theorems*, we first prove a slightly different version of the *Token Adding Theorem*, called the *Second Token Adding Theorem*.

Second Token Adding Theorem. *Let X be a token allocation with t tokens such that $x_a > x_b$, where $p_a = p_b$. Then $E[T_{X+e_a}] > E[T_{X+e_b}]$.*

Proof. The method of proof used in the *Token Adding Theorem* will work in this case, provided we can demonstrate that the conclusion of the *Token Adding Lemma* holds under the new hypotheses. Hence, we wish to show that if $x_a > x_b$ and $p_a = p_b$, then for $n > t$ and $0 \leq m \leq n-t$, $\frac{1}{p_a}Pr(A_{n,m}) \geq \frac{1}{p_b}Pr(B_{n,m})$ and the inequality is strict in at least once case.

Unfortunately, because $p_a = p_b$, it is no longer the case that $\frac{1}{p_a}Pr(A_{t,0}) > \frac{1}{p_b}Pr(B_{t,0})$. In fact, $\frac{1}{p_a}Pr(A_{n,0}) = \frac{1}{p_b}Pr(B_{n,0})$ for all $n \geq t$.

On the other hand, showing that $\frac{1}{p_a}Pr(A_{n,m}) \geq \frac{1}{p_b}Pr(B_{n,m})$ required only that

$$\prod_{k=0}^{m} \frac{p_b(x_a+k)}{p_a(x_b+k)} \geq 1 \qquad \text{and} \qquad \frac{p_b}{p_a}\prod_{k=1}^{m} \frac{p_b(x_a+k)}{p_a(x_b+k)} \geq 1$$

when $B_1 \neq 0$ and $B_2 \neq 0$. Since $p_a = p_b$, these products simplify to

$$\prod_{k=0}^{m} \frac{x_a+k}{x_b+k} \qquad \text{and} \qquad \prod_{k=1}^{m} \frac{x_a+k}{x_b+k}.$$

Further, because $x_a > x_b$, then clearly each product is strictly greater than 1. However, with this information, we cannot conclude that $\frac{1}{p_a}Pr(A_{n,m}) > \frac{1}{p_b}Pr(B_{n,m})$ unless we can demonstrate that there is a case when $B_1 \neq 0$ or $B_2 \neq 0$, since we could only conclude that $\frac{1}{p_a}Pr(A_{n,m}) \geq \frac{1}{p_b}Pr(B_{n,m})$ when $B_1 = 0$ and $B_2 = 0$.

Alternatively, we can show that there is an m such that $\frac{1}{p_a}Pr(A_{n,m}) > \frac{1}{p_b}Pr(B_{n,m})$ in all cases. In particular, we examine $m = n-t$. Recall that

$$\begin{aligned} A_1 = Pr(&X \text{ is cleared on the } n^{th} \text{ turn with exactly } x_a\ a\text{'s and } x_b+m\ b\text{'s},\\ &\text{and the last turn is an } a). \end{aligned}$$

Since $x_a > x_b \geq 0$, then $x_a \geq 1$. Therefore, $A_1 > 0$, because $x_a \geq 1$ and all of the "extraneous" rolls are filled with b's. If $B_1 = 0$, then $\frac{1}{p_a}Pr(A_{n,n-t}) > \frac{1}{p_b}Pr(B_{n,n-t})$. On the other hand, if $B_1 \neq 0$, then by the arguments made above, $\frac{1}{p_a}Pr(A_{n,n-t}) > \frac{1}{p_b}Pr(B_{n,n-t})$.

Hence, we have shown that for $n > t$ and $0 \leq m \leq n-t$, $\frac{1}{p_a}Pr(A_{n,m}) \geq \frac{1}{p_b}Pr(B_{n,m})$ and the inequality is strict for all $m = n-t$. Thus, by a proof analogous to the one used to prove the *Token Adding Theorem*, we conclude that $E[T_{X+e_a}] > E[T_{X+e_b}]$. □

4.2.4 The *Minimal Allocation Theorems*

The *Minimal Allocation Theorems* demonstrate necessary (although not sufficient) conditions for an allocation X^* to be minimal for the t-token game with fixed P. Actually, the theorems seem to be more general than that – notice that these theorems are independent of t. We will prove these two theorems here, but will delay exploring their applications until Section 6.

First Minimal Allocation Theorem. *Let X^* be a minimal token allocation with t tokens. If $p_a < p_b$, then $p_b(x_a^* - 1) < p_a x_b^*$.*

Proof. Either $x_a^* = 0$ or $x_a^* > 0$. First, suppose $x_a^* = 0$. Then $p_b(x_a^* - 1) < 0 \leq p_a x_b^*$, since $x_b^* \geq 0$. Hence, the theorem holds for $x_a^* = 0$.

Now suppose, by way of contradiction, that X is a minimal token allocation with t tokens and $x_a > 0$ such that $p_b(x_a - 1) \geq p_a x_b$, where $p_a < p_b$.

Let $X' = X - e_a$. Since $x_a > 0$, then X' is a valid allocation of $t - 1$ tokens.

Then $x'_a = x_a - 1$ and $x'_b = x_b$.

Hence, $p_b x'_a = p_b(x_a - 1) \geq p_a x_b = p_a x'_b$. Then by the *Token Adding Theorem*,

$$E[T_X] = E[T_{X'+e_a}] > E[T_{X'+e_b}] = E[T_{X-e_a+e_b}].$$

Thus, $X - e_a + e_b$ is an allocation with t tokens, whose expected clearing time is less than the expected clearing time of X, contradicting the minimality of X.

Hence, $p_b(x_a^* - 1) < p_a x_b^*$ for a minimal allocation X^*. □

Second Minimal Allocation Theorem. *Let X^* be a minimal token allocation with t tokens. If $p_a = p_b$, then $|x_a^* - x_b^*| \leq 1$.*

Proof. Suppose, by way of contradiction, that X is a minimal token allocation with t tokens such that $|x_a - x_b| > 1$, where $p_a = p_b$.

Either $x_a - x_b > 1$ or $x_b - x_a > 1$. Without loss of generality, suppose $x_a - x_b > 1$. Then $x_a - 1 > x_b$.

Let $X' = X - e_a$. Since $x_a > 1$, then X' is a valid allocation with t tokens.

Then $x'_a = x_a - 1$ and $x'_b = x_b$.

Hence $x'_a = x_a - 1 > x_b = x'_b$.

Then, by the *Second Token Adding Theorem*,

$$E[T_X] = E[T_{X'+e_a}] > E[T_{X'+e_b}] = E[T_{X-e_a+e_b}].$$

Thus, $X - e_a + e_b$ is an allocation with t tokens, whose expected clearing time is less than the expected clearing time of X, contradicting the minimality of X.

Similarly, if $x_b - x_a > 1$, then $X - e_b + e_a$ is an allocation with t tokens, whose expected clearing time is less than the expected clearing time of X, contradicting the minimality of X.

Thus, $|x_a^* - x_b^*| \leq 1$ for a minimal allocation X^*. □

5 *Knock 'm Down* in the Case of Two Values ($N = 2$)

Section 2 demonstrated that a theorem or algorithm for constructing the minimal allocations and the emperors or emperor cycle for the t-token game with fixed P would appear to be fairly complicated. However, there is one exceptional case, namely the game *Knock 'm Down* played with exactly two values. Th game with two values has some particularly nice properties. First, we note that $P = (p_1, p_2) = (p_1, 1 - p_1)$, so the probabilities of the values can be easily expressed in terms of a single variable. Similarly, for t tokens, any allocation X must be of the form $[x_1, x_2] = [x_1, t - x_1]$, so the token allocation can also be easily expressed in terms of a single variable. However, we will often use p_2 and x_2 to ease notation when it is convenient to do so. In Section 6, we will demonstrate that $E(T_X)$ can be calculated for a general *Knock 'm Down* game by means of a recursive calculation. However, in the 2-valued game, we will derive two non-recursive formulae for $E(T_X)$. Using one of these formulae, we will show that it is possible to determine the minimal allocations for the game using a simple calculation. Next, we will show that an even simpler calculation yields the emperors for the game. Finally, we will investigate the asymptotic shapes of the minimal allocation and emperor as we increase the number of tokens.

5.1 Formulae for $E(T_X)$

In order to derive formulae for $E(T_X)$, we will appeal to the definition of expected value. That is, if Z is a random integer variable, then

$$E(Z) = \sum_k kPr(Z = k).$$

In the case of $E(T_X)$, the random variable T_X is never less than the total number of tokens, so

$$Pr(T_X = k) = 0 \qquad\qquad \text{if } k < x_1 + x_2.$$

Now, we wish to express $Pr(T_X = k)$ for $k \geq x_1 + x_2$. If $T_X = k$, then we cleared the last token from X on the k^{th} turn. Hence, the k^{th} turn produced either the x_1^{th} 1 value, with $k - x_1$ 2 values produced on earlier turns, or the x_2^{th} 2 value, with $k - x_2$ 1 values produced on earlier turns. Therefore, we have

$$\begin{aligned} Pr(T_X = k) &= \binom{k-1}{x_1-1} p_1^{x_1} p_2^{k-x_1} + \binom{k-1}{x_2-1} p_1^{k-x_2} p_2^{x_1} \qquad\qquad \text{if } k \geq x_1 + x_2 \\ &= \left(\frac{p_1}{p_2}\right)^{x_1} \binom{k-1}{x_1-1} p_2^k + \left(\frac{p_2}{p_1}\right)^{x_2} \binom{k-1}{x_2-1} p_1^k. \end{aligned}$$

Hence, we have that:

$$\begin{aligned} E(T_X) &= \sum_{k=0}^{\infty} kPr(T_X = k) \\ &= \sum_{k=x_1+x_2}^{\infty} k \left[\left(\frac{p_1}{p_2}\right)^{x_1} \binom{k-1}{x_1-1} p_2^k + \left(\frac{p_2}{p_1}\right)^{x_2} \binom{k-1}{x_2-1} p_1^k \right]. \end{aligned}$$

To simplify this infinite sum, we will make use of the identity

$$\sum_{k=0}^{\infty} \binom{k}{x} p^k = \frac{p^x}{(1-p)^{x+1}} \qquad \text{for } 0 < p < 1 \text{ and integral } x \geq 0,$$

which can be demonstrated using a simple proof by induction on x and the identity $\binom{k}{x+1} = \binom{k-1}{x+1} + \binom{k-1}{x}$. Returning to the sum at hand, we have that

$$\begin{aligned}
E(T_X) &= \left(\frac{p_1}{p_2}\right)^{x_1} \sum_{k=x_1+x_2}^{\infty} k\binom{k-1}{x_1-1} p_2^k + \left(\frac{p_2}{p_1}\right)^{x_2} \sum_{k=x_1+x_2}^{\infty} k\binom{k-1}{x_2-1} p_1^k \\
&= \left(\frac{p_1}{p_2}\right)^{x_1} \sum_{k=x_1+x_2}^{\infty} x_1\binom{k}{x_1} p_2^k + \left(\frac{p_2}{p_1}\right)^{x_2} \sum_{k=x_1+x_2}^{\infty} x_2\binom{k}{x_2} p_1^k \\
&= x_1\left(\frac{p_1}{p_2}\right)^{x_1} \sum_{k=x_1+x_2}^{\infty} \binom{k}{x_1} p_2^k + x_2\left(\frac{p_2}{p_1}\right)^{x_2} \sum_{k=x_1+x_2}^{\infty} \binom{k}{x_2} p_1^k \\
&= x_1\left(\frac{p_1}{p_2}\right)^{x_1} \left[\sum_{k=0}^{\infty} \binom{k}{x_1} p_2^k - \sum_{k=0}^{x_1+x_2-1} \binom{k}{x_1} p_2^k\right] + x_2\left(\frac{p_2}{p_1}\right)^{x_2} \left[\sum_{k=0}^{\infty} \binom{k}{x_2} p_1^k - \sum_{k=0}^{x_1+x_2-1} \binom{k}{x_2} p_1^k\right] \\
&= x_1\left(\frac{p_1}{p_2}\right)^{x_1} \left[\frac{p_2^{x_1}}{p_1^{x_1+1}} - \sum_{k=0}^{x_1+x_2-1} \binom{k}{x_1} p_2^k\right] + x_2\left(\frac{p_2}{p_1}\right)^{x_2} \left[\frac{p_1^{x_2}}{p_2^{x_2-1}} - \sum_{k=0}^{x_1+x_2-1} \binom{k}{x_2} p_1^k\right] \\
&= \frac{x_1}{p_1} - x_1\left(\frac{p_1}{p_2}\right)^{x_1} \sum_{k=x_1}^{x_1+x_2-1} \binom{k}{x_1} p_2^k + \frac{x_2}{p_2} - x_2\left(\frac{p_2}{p_1}\right)^{x_2} \sum_{k=x_2}^{x_1+x_2-1} \binom{k}{x_2} p_1^k,
\end{aligned}$$

which is a finite formula for $E(T_X)$.

Using a slightly less intuitive expression for $Pr(T_X = k)$, we can derive another formula for $E(T_X)$. As we saw above

$$Pr(T_X < x_1 + x_2) = 0$$

and

$$Pr(T_X = x_1 + x_2) = \binom{x_1+x_2}{x_1} p_1^{x_1} p_2^{x_2}.$$

We wish to express $Pr(T_X = x_1 + x_2 + k)$ for $k > 0$. We will again consider whether the last turn cleared the last 1 value or the last 2 value, but will consider what values appear in the first t turns separately from the remainder of the turns. For example, suppose the last turn is a 1. Then the probability that the first t turns produced exactly i 1 values is given by

$$\binom{x_1+x_2}{i} p_1^i p_2^{x_1+x_2-i}.$$

Likewise, the probability that a sequence of k turns produced exactly $x_1 - i$ 1 values with a 1 value on the k^{th} turn is given by

$$\binom{k-1}{x_1-1-i} p_1^{x_1-i} p_2^{k-x_1-i}.$$

Finally, we note that if the last turn cleared the last 1 value, then i can range from $\max\{0, x_1 - k\}$. The same analysis works if the last turn cleared the last 2 value. Hence, the probability that the allocation was cleared in exactly $x_1 + x_2 + k$ turns for $k > 0$ is given by

$$\begin{aligned}
Pr(T_X = x_1 + x_2 + k) = &\sum_{i=\max\{0,x_1-k\}}^{x_1-1} \binom{x_1+x_2}{i} p_1^i p_2^{x_1+x_2-i} \binom{k-1}{x_1-1-i} p_1^{x_1-i} p_2^{k-x_1-i} \\
&+ \sum_{i=\max\{0,x_2-k\}}^{x_2-1} \binom{x_1+x_2}{i} p_1^{x_1+x_2-i} p_2^i \binom{k-1}{x_2-1-i} p_1^{k-x_2-i} p_2^{x_2-i} \\
&= p_1^{x_1} p_2^{x_2} \sum_{i=\max\{0,x_1-k\}}^{x_1-1} \binom{x_1+x_2}{i} \binom{k-1}{x_1-1-i} p_2^k \\
&+ p_1^{x_1} p_2^{x_2} \sum_{i=\max\{0,x_2-k\}}^{x_2-1} \binom{x_1+x_2}{i} \binom{k-1}{x_2-1-i} p_1^k.
\end{aligned}$$

Hence, we have

$$\begin{aligned}
E(T_X) &= \sum_{k=0}^{\infty} kPr(T_X = k) \\
&= \sum_{k=x_1+x_2}^{\infty} kPr(T_X = k) \\
&= \sum_{k=0}^{\infty} (x_1 + x_2 + k) Pr(T_X = x_1 + x_2 + k) \\
&= (x_1 + x_2) \sum_{k=0}^{\infty} Pr(T_X = x_1 + x_2 + k) + \sum_{k=1}^{\infty} kPr(T_X = x_1 + x_2 + k) \\
&= x_1 + x_2 + \sum_{k=1}^{\infty} kPr(T_X = x_1 + x_2 + k),
\end{aligned}$$

since the probability that the allocation in cleared in $x_1 + x_2$ or more turns is 1. We now concentrate on simplifying the sum.

$$\begin{aligned}
(x_1 + x_2) \sum_{k=1}^{\infty} kPr(T_X = x_1 + x_2 + k) = \sum_{k=1}^{\infty} k \Bigg[&p_1^{x_1} p_2^{x_2} \sum_{i=\max\{0,x_1-k\}}^{x_1-1} \binom{x_1+x_2}{i} \binom{k-1}{x_1-1-i} p_2^k \\
&+ p_1^{x_1} p_2^{x_2} \sum_{i=\max\{0,x_2-k\}}^{x_2-1} \binom{x_1+x_2}{i} \binom{k-1}{x_2-1-i} p_1^k \Bigg].
\end{aligned}$$

We will concentrate on the first sum, since the second sum is similar. In this case, we have

$$\sum_{k=1}^{\infty} k \left[p_1^{x_1} p_2^{x_2} \sum_{i=\max\{0,x_1-k\}}^{x_1-1} \binom{x_1+x_2}{i} \binom{k-1}{x_1-1-i} p_2^k \right]$$

$$
\begin{aligned}
&= p_1^{x_1} p_2^{x_2} \sum_{k=1}^{\infty} \sum_{i=\max\{0,x_1-k\}}^{x_1-1} k \binom{x_1+x_2}{i} \binom{k-1}{x_1-1-i} p_2^k \\
&= p_1^{x_1} p_2^{x_2} \sum_{i=0}^{x_1-1} \sum_{k=x_1-i}^{\infty} \binom{x_1+x_2}{i} \binom{k-1}{x_1-1-i} p_2^k \\
&= p_1^{x_1} p_2^{x_2} \sum_{i=0}^{x_1-1} \binom{x_1+x_2}{i} \left[\sum_{k=x_1-i}^{\infty} \binom{k-1}{x_1-1-i} p_2^k \right] \\
&= p_1^{x_1} p_2^{x_2} \sum_{i=0}^{x_1-1} \binom{x_1+x_2}{i} (x_1-i) \frac{p_2^{x_1-i}}{p_1^{x_1-i+1}} \\
&= \frac{p_2^{x_1+x_2}}{p_1} \sum_{i=0}^{x_1-1} (x_1-i) \binom{x_1+x_2}{i} \left(\frac{p_1}{p_2}\right)^i .
\end{aligned}
$$

Combining this with the second sum, we have

$$
\begin{aligned}
E(T_X) &= x_1 + x_2 + \frac{p_2^{x_1+x_2}}{p_1} \sum_{i=0}^{x_1-1} (x_1-i) \binom{x_1+x_2}{i} \left(\frac{p_1}{p_2}\right)^i + \frac{p_1^{x_1+x_2}}{p_2} \sum_{i=0}^{x_2-1} (x_2-i) \binom{x_1+x_2}{i} \left(\frac{p_2}{p_1}\right)^i \\
&= t + \sum_{i=0}^{x_1-1} (x_1-i) \binom{t}{i} p_1^{i-1} p_2^{t-i} + \sum_{i=0}^{x_2-1} (x_2-1) \binom{t}{i} p_1^{t-i} p_2^{i-1} \\
&= t + \sum_{i=0}^{x_1-1} (x_1-i) \binom{t}{i} p_1^{i-1} p_2^{t-i} + \sum_{i=x_1+1}^{t} (i-x_1) \binom{t}{i} p_1^{t-i} p_2^{i-1}
\end{aligned}
$$

We will apply this formula to the problem of finding the minimal allocations for the 2-valued game.

5.2 Minimal Allocations

Suppose that $X = [x_1, x_2]$ is an allocation of t tokens such that $x_2 > 0$. Then we note that

$$
\begin{aligned}
E(T_{X+e_1-e_2}) - E(T_X) &= \left(t + \sum_{i=0}^{x_1} (x_1+1-i) \binom{t}{i} p_1^{i-1} p_2^{t-i} + \sum_{i=x_1+2}^{t} (i-x_1-1) \binom{t}{i} p_1^{t-i} p_2^{i-1} \right) \\
&\quad - \left(t + \sum_{i=0}^{x_1-1} (x_1-i) \binom{t}{i} p_1^{i-1} p_2^{t-i} + \sum_{i=x_1+1}^{t} (i-x_1) \binom{t}{i} p_1^{t-i} p_2^{i-1} \right) \\
&= \sum_{i=0}^{x_1-1} \binom{t}{i} p_1^{i-1} p_2^{t-i} + \binom{t}{x_1} p_1^{x_1-1} p_2^{t-x_1} \\
&\quad - \sum_{i=x_1+2}^{t} \binom{t}{i} p_2^{t-i} p_2^{i-1} - \binom{t}{x_1+1} p_1^{t-x_1-1} p_2^{x_1+1-1} \\
&= \sum_{i=0}^{x_1} \binom{t}{i} p_1^{i-1} p_2^{t-i} - \sum_{i=x_1+1}^{t} \binom{t}{i} p_2^{t-i} p_2^{i-1}.
\end{aligned}
$$

Thus, moving one token from the 2 value to the 1 value changes the expected clearing time by $E(T_{X+e_1-e_2}) - E(T_X)$. If $E(T_{X+e_1-e_2}) - E(T_X) < 0$, then it is to our advantage to make the move. On the other hand, if $E(T_{X+e_1-e_2}) - E(T_X) > 0$, then it is not to our advantage to make the move. Hence, if we begin with all of our tokens on the 2 value, we simply move tokens one-by-one until $E(T_{X+e_1-e_2}) - E(T_X) \geq 0$. In fact, recalling the Binomial Theorem, we note that

$$\sum_{i=0}^{t} \binom{t}{i} p_1^i p_2^{t-i} = (p_1 + p_2)^t = 1.$$

Thus, we have

$$\sum_{i=0}^{x_1} \binom{t}{i} p_1^i p_2^{t-i} \geq p_1 \qquad \text{iff} \qquad \sum_{i=x_1+1}^{t} \binom{t}{i} p_1^i p_2^{t-i} \leq p_2,$$

so

$$\sum_{i=0}^{x_1} \binom{t}{i} p_1^{i-1} p_2^{t-i} \geq 1 \qquad \text{iff} \qquad \sum_{i=x_1+1}^{t} \binom{t}{i} p_1^i p_2^{t-i-1} \leq 1.$$

Finally, we have that

$$E(T_{X+e_1-e_2}) - E(T_X) \geq 0 \qquad \text{iff} \qquad \sum_{i=0}^{x_1} \binom{t}{i} p_1^i p_2^{t-i} \geq p_1.$$

Thus, to minimize $E(T_X)$ for a fixed t, let $X = (x_1, t-x_1)$ where $x_1 = \min\{0 \leq x_1 \leq t : \sum_{i=0}^{x_1} \binom{t}{i} p_1^i p_2^{t-i} \geq p_1\}$. But this is simply the definition of the p_1^{th} percentile of the $Binomial(t, p_1)$ distribution! Hence, we have proved the following.

Minimal Allocation Theorem for Two Values. *For the 2-valued, t-token game with $P = (p, 1-p)$, if $X^* = [x^*, t-x^*]$ where x^* is the p^{th} percentile of the $Binomial(t, p)$ distribution, then X^* is a minimal allocation.*

For a shorter proof of the same result, we simply compare the allocation $[x, t-x]$ with the allocation $[x+1, t-x-1]$ and condition on z, the number of 1 values that occur in the first t turns. If x or less 1 values occur in the first t turns, then both allocations will have cleared all tokens off the second value and the allocation $[x+1, t-x-1]$ will require an expected $\frac{1}{p}$ more turns than the allocation $[x, t-x]$ to clear the remaining tokens off of the first value. Likewise, if more than x 1 values occur in the first t turns, then both allocations will have cleared all tokens off the first value and the allocation $[x, t-x]$ will require an expected $\frac{1}{1-p}$ more turns than the allocation $[x+1, t-x-1]$ to clear the remaining tokens off the second value. Thus, we have that

$$E(T_{[x+1,t-x-1]}) - E(T_{[x,t-x]}) = \frac{1}{p} Pr(z \leq x) - \frac{1}{1-p} Pr(z > x).$$

Next, we note that

$$\begin{aligned}
E(T_{[x+1,t-x-1]}) - E(T_{[x,t-x]}) \geq 0 &\Leftrightarrow \frac{1}{p} Pr(z \leq x) - \frac{1}{1-p} Pr(z > x) \geq 0 \\
&\Leftrightarrow (1-p) Pr(z \leq x) \geq p Pr(z > x) \\
&\Leftrightarrow (1-p) Pr(z \leq x) \geq p(1 - Pr(z \leq x)) \\
&\Leftrightarrow Pr(z \leq x) \geq p.
\end{aligned}$$

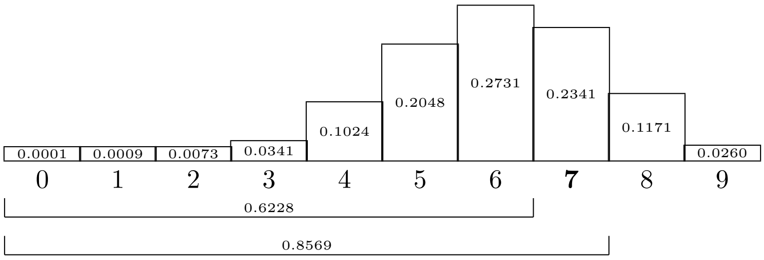

Figure 17: The $\frac{2}{3}$ Percentile of the $Binomial(9, \frac{2}{3})$ Distribution

Since z is a $Binomial(t, p)$ random variable, then we have again proven the *Minimal Allocation Theorem for Two Values.*

As an application of this theorem, we consider the 9-token game with $P = (\frac{2}{3}, \frac{1}{3})$. The $Binomial(9, \frac{2}{3})$ distribution is shown in Figure 17 with the $\frac{2}{3}$ percentile marked. Hence, the minimal allocation for the game is $[7, 2]$.

5.3 Emperors

Having found a characterization of the minimal allocations in the 2-valued game, we turn our attention to the emperors. Surprisingly, the emperor has a very similar characterization.

Emperor Theorem for Two Values. *For the* 2*-valued,* t*-token game with* $P = (p, 1-p)$*, if* $\hat{X} = [\hat{x}, t-\hat{x}]$ *where* $\hat{x}$ *is the median of the* $Binomial(t, p)$ *distribution, then* $\hat{X}$ *is an emperor.*

Proof. Let $\hat{X} = [\hat{x}, t-\hat{x}]$ be an allocation of t tokens where $\hat{x}$ is the median of the $Binomial(t, p)$ distribution.

First, we show that $\hat{X}$ is favored over any allocation $Y = [y_1, y_2]$ of t tokens such that $y_1 < \hat{x}_1$. Consider the first t turns of the game. If $\hat{x}_1$ 1 values occur in the first t turns, then $\hat{X}$ wins. Suppose more than $\hat{x}_1$ 1 values occur in the first t turns, then both $\hat{X}$ and Y have cleared all their tokens off the 1 value. However, both $\hat{X}$ and Y have tokens remaining on the 2 value. Further, $\hat{x}_2 < y_2$, so regardless of the remaining rolls, $\hat{X}$ will be cleared before Y. Hence $\hat{X}$ wins any game such that at least $\hat{x}_1$ 1 values occur in the first t turns. But,

$$\sum_{i=\hat{x}_1}^{t} \binom{t}{i} p^t (1-p)^{t-1} \geq \frac{1}{2}$$

by definition of the median of the $Binomial(t, p)$ distribution. Hence, $\hat{X}$ wins at least half of the games against Y, and $\hat{X}$ is favored over Y.

Similarly, $\hat{X}$ is favored over any allocation $Y = [y_1, y_2]$ of t tokens such that $y_1 > \hat{x}_1$. Again, if $\hat{x}_1$ 1 values occur in the first t turns, then $\hat{X}$ wins. If less than $\hat{x}_1$ 1 values occur in the first t turns, then both $\hat{X}$ and Y have cleared all of their tokens off the 2 value. However, both $\hat{X}$ and Y have tokens remaining on

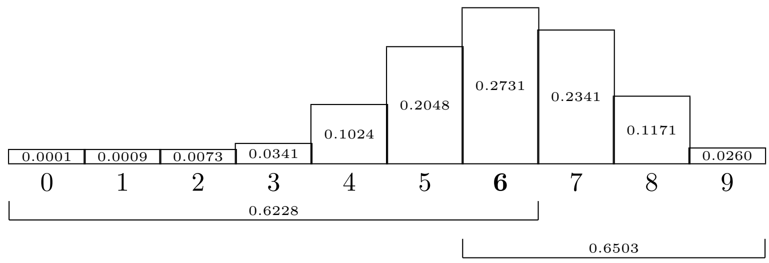

Figure 18: The Median of the $Binomial(9, \frac{2}{3})$ Distribution

the 1 value. Again, regardless of the remaining rolls, $\hat{X}$ will be cleared before Y, since $\hat{x}_1 < y_1$. Thus, $\hat{X}$ wins any game such that at most $\hat{x}_1$ 1 values occur in the first t turns. But,

$$\sum_{i=0}^{\hat{x}_1} \binom{t}{i} p^t (1-p)^{t-1} \geq \frac{1}{2}$$

by definition of the median of the $Binomial(t, p)$ distribution. Hence, $\hat{X}$ wins at least half of the games against Y, and $\hat{X}$ is favored over Y.

Thus, $\hat{X}$ is favored over all other allocations and is an emperor. □

For ease of calculation, we note that the median of the $Binomial(t, p)$ distribution is an integer whose distance from tp is less than one (see [2]).

As an application of this theorem, we consider the 9-token game with $P = (\frac{2}{3}, \frac{1}{3})$. The $Binomial(9, \frac{2}{3})$ distribution is shown in Figure 18 with the median marked. Hence, the minimal allocation for the game is $[6, 3]$.

5.4 Asymptotic Shape of X^* and $\hat{X}$

We can use the two theorems above to investigate the asymptotic shape of X^* and $\hat{X}$. Recall that a minimal allocation $X^* = [x^*, t - x^*]$ is given by

$$x^* = \min\{0 \leq x^* \leq t : \sum_{i=0}^{x^*} \binom{t}{i} p^i (1-p)^{t-i} \geq p\}.$$

In general, when t is large, x^* can be estimated quite accurately using a *normal* approximation. Specifically,

$$x^* \approx tp + z_p \sqrt{tp(1-p)}$$

and

$$t - x^* \approx t(1-p) + z_{1-p} \sqrt{tp(1-p)},$$

where z_p is the p^{th} percentile of the standard normal distribution. For example, with $P = (\frac{2}{3}, \frac{1}{3})$ and $t = 1200$, then normal approximation yields $x^* \approx 800 + 0.4307\sqrt{\frac{800}{3}} \approx 807.0337$, and similarly, $t - x^* \approx 400 - 0.4307\sqrt{\frac{800}{3}} \approx 392.9662$, in accordance with the minimal allocation listed in Figure 9.

Likewise, since an emperor $\hat{X} = [\hat{x}, t - \hat{t}]$ is given by

$$\hat{x} = \min\{0 \leq \hat{x} \leq t : \sum_{i=0}^{\hat{x}} \binom{t}{i} p_1^i (1-p)^{t-i} \geq \frac{1}{2}\},$$

then we can use the normal approximations

$$x^* \approx tp + z_{\frac{1}{2}}\sqrt{tp(1-p)} = tp$$

and

$$t - x^* \approx t(1-p) + z_{\frac{1}{2}}\sqrt{tp(1-p)} = t(1-p)$$

for large values of t. Hence, with $P = (\frac{2}{3}, \frac{1}{3})$ and $t = 1200$, we find that $[800, 400]$ is an emperor.

A surprising result of these approximations is that as the total number of tokens is increased, the minimal allocations and emperors all approach the distribution according to relative probabilities. In fact, we have the following.

Asymptotic Shape Theorem for Two Values. *For the 2-valued with $P = (p, 1-p)$, let $X^*(t) = [x^*(t), t - x^*(t)]$ and $\hat{X}(t) = [\hat{x}(t), t - \hat{x}(t)]$ be functions representing a minimal allocation and an emperor allocation for the games with t tokens. Then*

$$\lim_{t\to\infty} \frac{X^*(t)}{t} = P = \lim_{t\to\infty} \frac{\hat{X}(t)}{t}.$$

Proof. We simply observe that

$$\begin{aligned}
\lim_{t\to\infty} \frac{X^*(t)}{t} &= \lim_{t\to\infty} \left[\frac{x^*(t)}{t}, \frac{t - x^*(t)}{t}\right] \\
&= \lim_{t\to\infty} \left[\frac{tp + z_p\sqrt{tp(1-p)}}{t}, \frac{t(1-p) - z_{1-p}\sqrt{tp(1-p)}}{t}\right] \\
&= \lim_{t\to\infty} \left[p + z_p\sqrt{\frac{p(1-p)}{t}}, (1-p) - z_{1-p}\sqrt{\frac{p(1-p)}{t}}\right] \\
&= [p, 1-p] \\
&= P.
\end{aligned}$$

Likewise,

$$\begin{aligned}
\lim_{t\to\infty} \frac{\hat{X}(t)}{t} &= \lim_{t\to\infty} \left[\frac{\hat{x}(t)}{t}, \frac{t-\hat{x}(t)}{t}\right] \\
&= \lim_{t\to\infty} \left[\frac{tp}{t}, \frac{t(1-p)}{t}\right] \\
&= [p, 1-p] \\
&= P.
\end{aligned}$$

$\square$

So, despite our misgivings after the evidence in Section 2, we can finally justify our intuitive notion that the "best" allocation is the distribution according relative probabilities – at least in the 2-valued case. Actually, we conjecture that this result hold for the N-valued case (see Section 7 for evidence that this may indeed be true).

6 Computational Details

In Section 2, we examined a number of interesting instances of *Knock 'm Down* in order to develop a better intuitive understanding of the game. In this section, we wish to investigate some of the details associated with computing the results cited earlier. As we shall see, calculating the expected clearing time of an allocation, the winning and losing probabilities of pairs of allocations, and determining the emperor or emperor cycle from the tournament graph are computational challenges in their own right. In Section 6.1, we will examine the principal equations and techniques used for calculating $E(T_X)$ and $WDL(X,Y)$. In Section 6.2, we will examine some of the advanced methods which allowed us to investigate games which were too "large" to be handled by the primary methods.[2]

6.1 Primary Methods

In this section, we discuss the primary methods used to develop the results in Section 2. Two basic principles underly these methods. First, the laws of conditional expectation and conditional probability are exploited to derive recursive formulae for the desired quantities. Second, a modification of the standard dynamic-programming technique is used to increase the time and space efficiency of the calculations. We will use the notation developed in Section 3, letting N be an integer representing the number of values labeled on a player's board, $P = (p_1, p_2, \ldots, p_N)$ be a probability vector, $X = [x_1, x_2, \ldots, x_N]$ be a vector of non-negative integers (representing a token allocation), and e_i be the i^{th} unit vector of N elements.

6.1.1 Calculating $E(T_X)$

To calculate $E(T_X)$, we exploit the law of conditional expectation. Recall that if Z is a random variable, whose value is conditional on the random integer variable V, then

$$E(Z) = \sum_i Pr(V = i) E(Z | V = i).$$

We let V be a random integer variable representing the value "rolled" on the first turn of an N-valued game. Thus, we can express

$$E(T_X) = \sum_i Pr(V = i) E(T_X | V = i)$$

where $P(V = i) = p_i$ for $i \in \{1, 2, \ldots, N\}$ and $Pr(V = i) = 0$ otherwise. To determine $E(T_X | V = i)$, we reason as follows. If the first turn produced a value that we could not use, i.e., we had no tokens on the i^{th} value, then we "wasted" a roll and have not changed our situation. On the other hand, if the first turn

[2] For the interested reader, source code implementing these methods is available by request from **Matthew_Fluet@hmc.edu**. The code is written in Standard ML utilizing the SML '97 Basis Library. The computational results cited in this section and Section 2 were computed using the SML/NJ compiler and development environment version 110 on an UltraSparc Enterprise 3000 running Solaris 2.6.

produced a value that we could use, then we still "used" a roll, but have improved our situation by reducing the number of tokens we need to remove subsequently. Formally, we have

$$E(T_X|V=v) = \begin{cases} 1+E(T_X) & \text{if } x_v = 0 \\ 1+E(T_{X-e_v}) & \text{if } x_v \neq 0. \end{cases}$$

For instance, in the game with $P = (.4, .3, .2, .1)$, we have

$$E(T_{[4,3,2,1]}) = 1 + .4E(T_{[3,3,2,1]}) + .3E(T_{[4,2,2,1]}) + .3E(T_{[4,3,1,1]}) + .1E(T_{[4,3,2,0]})$$

and

$$\begin{aligned} E(T_{[5,3,2,0]}) &= 1 + .4E(T_{[4,3,2,0]}) + .3E(T_{[3,2,2,0]}) + .2E(T_{[5,3,1,0]}) + .1E(T_{[5,3,2,0]}) \\ &= \frac{10}{9}(1 + .4E(T_{[4,3,2,0]}) + .3E(T_{[3,2,2,0]}) + .2E(T_{[5,3,1,0]})) \end{aligned}$$

Solving for $E(T_X)$ yields the following recursive formula:

$$E(T_X) = \frac{1 + \sum_{i:x_i \neq 0} p_i E(T_{X-e_i})}{1 - \sum_{v:x_i=0} p_i}. \tag{4}$$

with the base case $E(T_{[0,\dots,0]}) = 0$.

At this point it seems as though a straightforward implementation of (4) into any programming language would allow us to calculate $E(T_X)$ with relative ease. However, (4) is deceptively simple. Notice that it makes use of recursion in the summation of the numerator. It is fairly straightforward to calculate that the number of recursions required to calculate $E(T_X)$ is $\prod_i (1+x_i)$. In addition to the large number of recursive calls necessary to calculate $E(T_X)$, we note that the recursions do not form a "tree" structure; rather, many of the recursions make use of identical sub-values. Presented with a recursive formula with these characteristics, one would naturally attempt to implement (4) using a dynamic-programming technique.

Unfortunately, making use of dynamic-programming in this situation is difficult. Recall that in order to make use of dynamic-programming, it is necessary to save the results of earlier calculations for use in future calculations. While this seems natural enough to implement in this situation, consider the following. Suppose we wish to calculate the expected time required to clear allocations with up to twelve tokens on any of eleven values, as in the original description of *Knock 'm Down*. Then we would need to allocate space for 13^{11} intermediate values. Stored as two-byte floating point numbers, this is over 3 terabytes of data, too much memory to allocate on a single machine. The alternative is to forgo dynamic-programming and to simply use recursion, recalculating values as needed. But this can become time consuming, vastly inflating the time necessary to calculate $E(T_X)$.

As a compromise solution, we have implemented the calculation of $E(T_X)$ in the following manner. We note that in a"top-level" request for $E(T_X)$, the recursion will only ever calculate $E(T_{X'})$, where X' is a sub-allocation of X. Therefore, at the time a top-level request is made, we allocate space for $\prod_i (1+x_i)$ intermediate values. Then we can proceed with calculating (4), storing intermediate values as they are calculated and reusing these values when they are needed. When we have finished evaluating the top-level request, we dispose of the intermediate values. Using this method, allocations with a total of twelve tokens on any of eleven values require at most 2 kilobytes of memory. This method has the advantage of dynamic-programming without the memory overhead. The drawback is that the evaluation of $E(T_X)$ for one allocation cannot make use of the intermediate values used in evaluating $E(T_Y)$, where Y is an allocation that shares sub-allocations with X.

6.1.2 Finding Minimal Allocations

Now that we have an efficient method for evaluating $E(T_X)$, it remains to calculate minimal allocations with t tokens. The allocations of t tokens among N values are equivalent to the the $\binom{t+N-1}{t}$ compositions of t into N parts. This can be difficult, but many combinatorial algorithms books describe a method for generating the next composition from the previous composition (see [7]). We can now proceed with a brute force search of all $\binom{t+N-1}{t}$ allocations and choose the set of allocations which have the minimal expected value. However, for large numbers of tokens and values, this method quickly becomes computationally infeasible. As will be described in Section 6.2, we can find the minimal allocations with much less computation.

6.1.3 Calculating $VAR(T_X)$

We briefly examine two formulae for $VAR(T_X)$. Although we have not investigated the properties of $VAR(T_X)$, it is possible that by comparing both the expected clearing time and the variance of the clearing time for a pair of allocations, we could determine if one allocation were favored over the other.

For the first formula, we use the definition of variance:

$$VAR(T_X) = E(T_X^2) - E(T_X)^2. \tag{5}$$

We calculate $E(T_X^2)$ using conditional expectation. Therefore, we have

$$E(T_X^2) = \sum_i Pr(V=i)E(T_X^2|V=i),$$

where

$$\begin{aligned} E[T_X^2|V=i] &= \begin{cases} 1+2E(T_{X-e_i})+E(T_{X-e_i}^2) & \text{if } x_i \neq 0 \\ 1+2E(T_X)+E(T_X^2) & \text{if } x_i = 0 \end{cases} \\ &= \begin{cases} 2(1+2E(T_{X-e_i}))-1+E(T_{X-e_y}^2) & \text{if } x_i \neq 0 \\ 2(1+2E(T_X))-1+E(T_X) & \text{if } x_i = 0 \end{cases}. \end{aligned}$$

Using the identity

$$E(T_X) = \sum_{i:x_i \neq 0} p_i(1 + E(T_{X-e_i})) + \sum_{i:x_i = 0} p_i(1 + E(T_X)),$$

we can solve for $E(T_X^2)$ to yield the following recursive formula:

$$E(T_X^2) = \frac{2E(T_X) - 1 + \sum_{i:x_i \neq 0} p_i E(T_{X-e_y}^2)}{1 - \sum_{i:x_i=0} p_i},$$

with the base case $E(T_{[0,\ldots,0]}^2) = 0$. We can now calculate $VAR(T_X)$ using the definition of variance above.

Alternatively, we can use conditional variance. Most formally, we have that

$$VAR(T_X) = E(VAR(T_X|V)) + VAR(E(T_X|V)).$$

Using the definition of variance and conditional expectation, we can derive that

$$VAR(T_X) = \sum_i p_i VAR(T_X|V = i) + \left(\sum_i p_i E(T_X|V = i) - \left(\sum_i p_i E(T_X|V = i) \right)^2 \right).$$

We note that

$$\begin{aligned} VAR(T_X|V = i) &= E\left((T_X - E(T_X|V = i))^2 | V = i\right) \\ &= \begin{cases} VAR(T_{X-e_i}) & \text{if } x_i \neq 0 \\ VAR(T_X) & \text{if } x_i = 0. \end{cases} \end{aligned}$$

Also, we note that

$$\sum_i p_i E(T_X|V = i)^2 - \left(\sum_i p_i E(T_X|V = i) \right)^2 = 2E(T_X) - 1 + \sum_i p_i \left(E(T_{X-e_i})^2 - E(T_X)^2 \right).$$

Solving for $VAR(T_X)$ yields the following recursive formula:

$$VAR(T_X) = \frac{2E(T_X) - 1 + \sum_{i:x_i \neq 0} p_i(VAR(T_{X-e_i}) + E(T_{X-e_i})^2 - E(T_X)^2)}{1 - \sum_{i:x_i=0} p_i}. \quad (6)$$

While both (5) and (6) are valid formulae for calculating $VAR(T_X)$, they each have advantages and disadvantages. A disadvantage shared by both methods is a dependence on $E(T_X)$. In practice this is not a serious disadvantage, since most likely one is looking for both $E(T_X)$ and $VAR(T_X)$. However, in light of the recursive nature of these formulae it makes sense to calculate $E(T_X)$ and $VAR(T_X)$ at the same time, taking advantage of the dynamic-programming techniques described previously to avoid redundant calculations. An advantage of (5) is that it appears to be a simpler sequence of calculations to carry out. On the other hand, the (6) second method has the advantage of being an explicit expression for $VAR(T_X)$ and may be more useful in theoretical considerations of variance.

6.1.4 Calculating $WDL_O(X, Y)$ and $WDL_I(X, Y)$

Recall that $WDL_O(X, Y)$ is the WDL function which corresponds to a game played under *Original Knock 'm Down* rules, in which players remove tokens from the same value on a turn. To calculate $WDL_O(X, Y)$, we exploit the law of conditional probability. Again, we let V be the random variable representing the value "rolled" on the first turn of an N-valued game. Thus, we can express

$$WDL_O(X, Y) = \sum_i Pr(V = i) WDL_O(X, Y|V = i),$$

where

$$WDL_O(X, Y|V = i) = \begin{cases} WDL_O(X, Y) & \text{if } x_i = 0, y_i = 0 \\ WDL_O(X, Y - e_i) & \text{if } x_i = 0, y_i \neq 0 \\ WDL_O(X - e_i, Y) & \text{if } x_i \neq 0, y_i = 0 \\ WDL_O(X - e_i, Y - e_i) & \text{if } x_i \neq 0, y_i \neq 0, \end{cases}$$

since producing a value on the first turn for which there are no tokens in either allocation does not change the situation, while producing a value on the first turn for which either or both allocations have tokens does change the situation. Solving for $WDL_O(X, Y)$ yields the following recursive formula:

$$\begin{aligned} WDL_O(X, Y) = \Bigg(& \sum_{i: \substack{x_i \neq 0 \\ y_i \neq 0}} p_i WDL_O(X - e_i, Y - e_i) + \sum_{i: \substack{x_i \neq 0 \\ y_i = 0}} p_i WDL_O(X - e_i, Y) \\ & + \sum_{i: \substack{x_i = 0 \\ y_i \neq 0}} p_i WDL_O(X, Y - e_i) \Bigg) \Big/ \Bigg(1 - \sum_{i: \substack{x_i = 0 \\ y_i = o}} p_i \Bigg), \end{aligned} \tag{7}$$

with the important base cases

$$\begin{aligned} WDL_O([0, \dots, 0], [0, \dots, 0) &= (0, 1, 0) \\ WDL_O([0, \dots, 0], Y) &= (1, 0, 0) \\ WDL_O(X, [0, \dots, 0]) &= (0, 0, 1). \end{aligned}$$

In *Independent Knock 'm Down*, players remove tokens from independently random values on a turn. We can use the same conditional probability technique to find a formula for WDL_I, but we must condition on the values produced on the first turn for both players. When calculating $WDL_I(X, Y)$, we let V be the random variable representing the value on the first turn for the player with allocation X and W be the random variable representing the value on the first turn for the player with allocation Y. We can express

$$WDL_I(X, Y) = \sum_{i,j} Pr(V = i) Pr(W = j) WDL_I(X, Y|V = i, W = j).$$

As with the case of WDL_O, we have four cases to consider:

$$WDL_O(X,Y|V=i,W=j) = \begin{cases} WDL_I(X,Y) & \text{if } x_i = 0, y_j = 0 \\ WDL_I(X,Y-e_j) & \text{if } x_i = 0, y_j \neq 0 \\ WDL_I(X-e_i,Y) & \text{if } x_i \neq 0, y_j = 0 \\ WDL_I(X-e_i,Y-e_j) & \text{if } x_i \neq 0, y_j \neq 0. \end{cases}$$

Solving for WDL_I yields the following recursive formula:

$$\begin{aligned} WDL_I(X,Y) = \Bigg(&\sum_{i:x_i\neq 0}\sum_{j:y_j\neq 0} p_i p_j WDL_I(X-e_i,Y-e_j) + \sum_{i:x_i\neq 0}\sum_{j:y_j=0} p_i p_j WDL_I(X-e_i,Y) \\ &+ \sum_{i:x_i=0}\sum_{j:y_j\neq 0} p_i p_j WDL_I(X,Y-e_j)\Bigg) \Bigg/ \left(1 - \sum_{i:x_i=0}\sum_{j:y_j=} p_i p_j\right), \end{aligned} \tag{8}$$

with the important base cases

$$\begin{aligned} WDL_I([0,\ldots,0],[0,\ldots,0) &= (0,1,0) \\ WDL_I([0,\ldots,0],Y) &= (1,0,0) \\ WDL_I(X,[0,\ldots,0]) &= (0,0,1). \end{aligned}$$

Notice that the formulae for $WDL_O(X,Y)$ and $WDL_I(X,Y)$ have the same recursive properties as the formula for $E(T_X)$, so the same dynamic-programming technique can be used. However, note that they are recursive formula in $2N$ variables, with a dynamic-programming storage requirement of $3\prod_i(1+x_i)(1+y_i)$ intermediate values (one for each of the three probabilities being calculated). In Section 6.2, we will show that there is a more efficient and potentially faster method for calculating $WDL_O(X,Y)$.

6.2 Optimized Methods

In this section, we discuss a number of techniques which lead to more efficient implementations of two of the methods discussed in the last section.

6.2.1 Finding Minimal Allocations

In the last section, we developed an efficient means of calculating $E(T_X)$, but finding the minimal allocations for the t-game with fixed P was limited to a brute force search of all $\binom{t+N-1}{t}$ allocations. Throughout this discussion, we will return to the problem of finding the minimal allocations for twelve tokens in the original description of *Knock 'm Down* where P corresponds to the roll of two of six-sided dice. However, we will write this probability vector as a non-increasing sequence: $P = (\frac{6}{36}, \frac{5}{36}, \frac{5}{36}, \frac{4}{36}, \frac{4}{36}, \frac{3}{36}, \frac{3}{36}, \frac{2}{36}, \frac{2}{36}, \frac{1}{36}\frac{1}{36})$. We understand that many of these techniques cannot be analyzed for a quantitative measure of improvement, but we hope that the concrete example of the original *Knock 'm Down* will provide evidence for a substantial qualitative improvement. To begin, we note that a brute force search of all allocations of twelve tokens on eleven values would calculate $E(T_X)$ for $\binom{12+11-1}{12} = 646,646$ allocations.

The first improvement that can be made when searching for the minimal allocations is to note the following: if $p_i = p_{i+1}$ and $X = [x_1, \ldots, x_i, x_{i+1}, \ldots, x_N]$ and $X' = [x_1, \ldots, x_{i+1}, x_i, \ldots, x_N]$, then $E(T_X) = E(T_{X'})$. Hence, we define an allocation X to be in *normal form* if for all $p_i = p_{i+1}$ then $x_i \geq x_{i+1}$. For example, the allocation $[4, 2, 3, 1, 1, 0, 0, 0, 0, 0, 0]$ is not in normal form, but the allocation $[4, 3, 2, 1, 1, 0, 0, 0, 0, 0, 0]$ is in normal form. Further, an allocation X which is not in normal form can be mapped to an allocation X' which is in normal form such that $E(T_X) = E(T_{X'})$ and X' can be constructed from X by reordering the tokens on values with equal probability. Thus, it suffices to calculate $E(T_X)$ only for those allocations which are in normal form. Once a minimal allocation is found, other minimal allocations which are not in normal form can be easily constructed. Since the original description of *Knock 'm Down* has five pairs of values with equal probability, it benefits greatly from this improvement. In fact, of the $646, 646$ allocations, only $68, 354$ allocations are in normal form. Unfortunately, there is one drawback to this technique; namely, it requires some values to have equal probability. Therefore, more powerful techniques are required.

Recall that in Section 4, we proved:

First Minimal Allocation Theorem. *Let X^* be a minimal token allocation with t tokens. Then $p_b(x_a^* - 1) < p_a x_b^*$, where $p_a < p_b$.*

and

Second Minimal Allocation Theorem. *Let X^* be a minimal token allocation with t tokens. Then $|x_a^* - x_b^*| \leq 1$, where $p_a = p_b$.*

These two theorems give necessary (although not sufficient) conditions for a token allocation to be minimal. Hence, when searching for the minimal allocations, we need only calculate $E(T_X)$ for those allocations which satisfy the conclusions of the *Minimal Allocation Theorems*. To apply the *First MAT* to a particular allocation, we must examine every pair of values which have unequal probabilities. Likewise, to apply the *Second MAT*, we must examine every pair of values which have equal probabilities. Applying these results to the 12-token game, we find that there are only 175 allocations which satisfy the *First MAT* and there are $35, 014$ allocations which satisfy the *Second MAT*. Intersecting these two sets of allocations, we find that there are only 109 allocations which satisfy both the *First* and *Second MATs*.

We now prove a *Third Minimal Allocation Theorem*. It should be clear from the statement of the theorem that the *Third MAT* is an implicit result of the *First* and *Second MATs*, but we make it explicit here for its computational implications.

Third Minimal Allocation Theorem. *Let X^* be a minimal token allocation in normal form with t tokens for $P = (p_1, \ldots, p_N)$, such that $p_1 \geq \cdots \geq p_n$. Then $x_1^* \geq \cdots \geq x_N^*$.*

Proof. Suppose, by way of contradiction, that X is a minimal token allocation in normal form with t tokens such that $x_i < x_{i+1}$ for $p_i \geq p_{i+1}$. Since X is in normal form and $x_i < x_{i+1}$, then $p_i > p_{i+1}$. Further, since $x_i < x_{i+1}$, then $x_{i+1} > 0$ and $x_i \leq x_{i+1} - 1$. Let $X' = X - e_{i+1}$. Since $x_{i+1} > 0$, then X' is a valid token allocation. Then $x'_{i+1} = x_{i+1} - 1$ and $x'_i = x_i$. Hence, $p_{i+1}x'_i = p_{i+1}x_i \leq p_i(x_{i+1} - 1) = p_{i+1}x'_i$. Then, by the *Token Adding Theorem*,

$$E(T_X) = E(T_{X'+e_{i+1}}) > E(T_{X'+e_i}) = E(T_{X-e_{i+1}+e_i}).$$

Thus, $X - e_{i+1} + e_i$ is an allocation with t tokens, whose expected clearing time is less than the expected clearing time of X, contradicting the minimality of X.

Thus, $x_i \geq x_{i+1}$ for $p_i \geq p_{i+1}$. Hence, $x_1^* \geq \cdots x_N^*$ for a minimal allocation in normal form with $P = (p_1, \ldots, p_N)$, such that $p_1 \geq \cdots \geq p_n$. □

Notice that the *Third MAT* implies that every minimal allocation of t tokens in normal form can be written as a non-increasing list of tokens. If we search for the minimal allocations with t tokens using only the *First* and *Second MATs*, then we must generate every allocation X which corresponds to a *composition* of t into N parts. If X satisfies the conclusions of the *First* and *Second MATs*, then we calculate $E(T_X)$, compare it with the running minimum, and move onto the next allocation. Although we calculate the expected clearing time of only some of the generated allocations, we briefly consider all $\binom{t+N-1}{t}$ allocations. (We remarked in Section 6.1 that there are efficient combinatorial algorithms for generating the combinations of t into N parts, but we would still like to eliminate extraneous calculations.) On the other hand, if we search for the minimal allocations with t tokens using the *First*, *Second*, and *Third MATs*, then we are guaranteed that the minimal allocations in normal form can be expressed as a non-increasing list of tokens. Hence, we need only generate every allocation X which corresponds to a *partition* of t into at most N parts, expressed as a non-increasing list of parts. We can then continue as above: if X satisfies the conditions of the *First* and *Second MATs*, then we calculate $E(T_X)$, compare it with the running minimum, and move onto the next allocation. Although we calculate the expected clearing time of the same number of normal form allocations, we generate fewer allocations. Since there is no closed form for the expression $\pi(t, N) = \sum_{k=1}^{N} p(t, k)$, we cannot argue quantitatively about the relative sizes of the number of allocations considered using partitions versus the number of allocations considered using combinations, but we can be sure that the partition method considers (significantly) fewer allocations (since the partitions of t into at most N parts can be expressed as a subset of the compositions of t into N parts).

Although there are combinatorial algorithms for generating all partitions of t (see [7]), we would like an algorithm for generating all partitions of t into N parts, with "unused" parts filled in with 0s. Obviously, we could generate such a list from all partitions of t, but when t is large relative to N, we will again be "wasting" time generating partitions corresponding to allocations that we will never even consider as being a potentially minimal allocation. We developed an algorithm for generating all partitions of t into N parts which avoids generating extraneous partitions.[3]

By taking into account the *Third MAT*, we find that there are only 76 partitions of twelve into at most eleven parts. Hence, there are only 76 allocations in normal form which satisfy the *Third MAT*. Combining all three *MATs*, we find that there are only 49 normal form allocations which are potentially minimal allocations and hence we need to calculate $E(T_X)$ for only these 49 allocations.

However, we can do even better. Note that the allocation $[12, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]$ is in normal form and satisfies all three *MATs*. Likewise, $[11, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0]$ is also in normal form and satisfies all three *MATs*. However, these allocations seem to be clearly far from minimal. In general, there are often a number of allocations which satisfy all three *MATs*, but are clearly not minimal. Therefore, we modify our searching algorithm in the following manner. First, we note that for any allocation X, $E(T_X) \geq \frac{x_1}{p_1}$ since in order to clear any allocation, we must clear all of the tokens on the most probable value. Next, we note that among those allocations which have many tokens on the most probable values and satisfy all three *MATs*, then $\frac{x_1}{p_1}$ becomes a better approximation for $E(T_X)$. Suppose we have a allocation X^* which has the minimum expected clearing time of all allocations considered thus far and X is the next potentially minimal allocation to check. If $E(T_{X^*}) < \frac{x_1}{p_1}$, then clearly $E(T_{X^*}) < E(T_X)$, so we are not required to perform the more expensive calculation of $E(T_X)$. By generating our allocations in an order which places the least number of tokens on the most probable value first and places the greatest number of tokens on the most probable value last, then using the running minimum and the approximation by tokens on the most probable value, we are able to eliminate even more allocations from consideration. A general quantitative measurement of this type of improvement is unknown, but experience has shown it to be significant in some cases. Returning to the case of twelve tokens in the original *Knock 'm Down*, we have found that of the 49 allocations which satisfy all three *MATs*, 17 allocations are eliminated by the method described. Hence, of the original $646,646$ allocations, in order to find the allocations with the minimum expected clearing time, we need only calculate $E(T_X)$ of 32 normal form allocations.

6.2.2 Calculating $WDL_O(X, Y)$

In the last section, we developed a recursive formula for the calculation of $WDL_O(X, Y)$. We noted that the formula was recursive in $2N$ variables, for which a naive implementation would use a dynamic-programming storage requirement of $3\prod_i (x_i + 1)(y_i + 1)$ intermediate values. However, with a little thought it becomes clear that not all of those intermediate values are filled, since the two allocations are influenced by the

[3]See source code available by request from **Matthew_Fluet@hmc.edu** for the implementation details.

same value. For example, consider calculating $WDL_O([4,3,2,1],[5,3,2,0])$. Among the intermediate values calculated are $WDL_O([3,3,2,1],[4,3,2,0])$, $WDL_O([4,2,2,1],[5,2,2,1])$ and $WDL_O([3,2,2,1],[4,2,2,1])$. However, $WDL_O([3,3,2,1],[5,2,2,1])$ is never calculated, since the first allocation cannot remove a token on the first value without the second allocation also removing a token from the first value, and conversely, the second allocation cannot remove a token from the second value without the first allocation removing a token from the second value. Hence, $3\prod_i (x_i+1)(y_i+1)$ is overcounting the number of intermediate values we will need.

A more efficient means of calculating $WDL_O(X,Y)$ can be found by noting that it is sometimes possible to decide the winner of a game without that player removing all of tokens from his allocation. For example, suppose the first player chooses the allocation $[4,3,2,1]$ and the second player chooses the allocation $[5,3,2,0]$. If the first 7 values generated for the game are $\alpha,\alpha,\gamma,\alpha,\alpha,\delta,\gamma$, then the first player will be left with the allocation $[0,3,1,0]$ and the second player will be left with the allocation $[0,3,1,0]$. Clearly, any sequence of values that clears the first player's allocation will also clear the second player's allocation, so this game can only end in a draw. On the other hand, if the first 7 values generated for the game are $\beta,\gamma,\beta,\alpha,\beta,\gamma,\delta$, then the first player will be left with the allocation $[3,0,0,0]$ and the second player will be left with the allocation $[4,0,0,0]$. Clearly, any sequence of values that clears the second player's allocation will have first cleared the first player's allocation, so this game can only end in a win for the first player. This motivates us to define the following. Let $Max(X,Y) = [\max\{x_1,y_1\}),\max\{x_2,y_2\},\ldots,\max\{x_N,y_N\}]$. Let $F=\{f : x_f < y_f\}$, for the values on which the *f*irst player has fewer tokens, $B=\{b : x_b = y_b\}$, for the values on which *b*oth players have equal tokens, and $S=\{s : x_s > y_s\}$, for the values which the *s*econd player has fewer tokens. Finally, define

$$WDL'_O(Z;F,B,S) = \begin{cases} (1,0,0) & \text{if } \forall f\in F, z_f = 0 \text{ and } \forall b\in B, z_b = 0 \\ (0,1,0) & \text{if } \forall f\in F, z_f = 0 \text{ and } \forall s\in S, z_s = 0 \\ (0,0,1) & \text{if } \forall b\in B, z_b = 0 \text{ and } \forall s\in S, z_s = 0 \\ \frac{\sum_{i:z_i\neq 0} p_v WDL'_O(Z-e_i;L,B,R)}{1-\sum_{i:x_i=0} p_i} & \text{otherwise.} \end{cases}$$

since having tokens only on values where the second player has more tokens is a guaranteed win for the first player, having tokens only on values where both players have equal tokens is a guaranteed draw for the first player, having tokens only on values where the first player has more tokens is a guaranteed loss for the first player, and anything else requires a conditional expression. Hence, we note that $WDL_O(X,Y) = WDL'_O(Max(X,Y);F,B,S)$. Further, $WDL'_O(Z;F,B,S)$ is a recursive formula in only N variables with a dynamic-programming storage requirement of $3\prod_i(1+z_i) = 3\prod_i(1+\max\{x_i,y_i\})$ intermediate values.

6.3 Finding Emperors and Emperor Cycles

Lastly, we consider the methods by which we find emperors and emperor cycles in a tournament. We construct a tournament using an adjacency matrix representation for the directed graph. The i, j element of the matrix is $WDL(X_i, X_j)$, where X_i and X_j are, respectively, the i^{th} and j^{th} allocations in the tournament. If $(w_{i,j}, d_{i,j}, l_{i,j})$ is the i, j element of the matrix and $w_{i,j} \geq l_{i,j}$, then we consider a directed edge to exist from X_i to X_j. We can easily check for the existence of emperors by examining the adjacency matrix for any rows i where $w_{i,j} \geq l_{i,j}$ for all j.

In order to detect emperor cycles and to examine the non-transitivity of the whole tournament, we developed an algorithm to find all cycles of a directed graph given its adjacency matrix. Although the problem is exponential in general (since the completely connected directed graph of n vertices has $\sum_{k=2}^{n} \frac{n!}{(n-k)!k}$ cycles), for reasonable sized tournaments, the algorithm runs reasonably fast. On the other hand, if we only wish to determine the allocations that are in the emperor cycle, we can first group allocations of the tournament into strongly connected components.[4] Since we are considering a tournament, if allocation X and Y are members of separate strongly connected components and X is favored over Y, then every member of X's component is favored over every member of Y's component. Further, each component of the tournament contains a cycle through all of the allocations of the component. Since we can easily group allocations of a tournament into strongly connected components, if we knew one member of the emperor cycle, we could determine the strongly connected component that contains all members of the emperor cycle. Unfortunately, we do not have a means of determining candidate members of the emperor cycle. In Section 7, we will examine some conjectures for determining candidate members of the the emperor cycle.

[4] Recall that a set of vertices in a directed graph is a strongly connected component if there exists a directed path from every vertex in the set to every other vertex in the set.

7 Conjectures, Open Questions, and Future Directions

Despite the number of interesting results cited earlier, there are many interesting questions which remain unanswered. In this section, we wish to describe some of the conjectures that we are most interested in seeing proved or disproved. Also, we will examine some of the directions in which future research could progress.

In Section 5, we developed as simple description of the minimal allocations and emperors for the 2-valued game. A closed form solution to the n-valued game remains elusive for $n \geq 3$. We saw in Section 2 that non-trivial emperor cycles occur in games with three values, so a simple description may be difficult to develop. Failing that, we would like to find more ways to characterize the minimal allocations and emperors. We are particularly interested in variations of the *First Minimal Allocation Theorem* which might provide interesting *lower* bounds on x_a/x_b (or perhaps $(x_a + 1)/x_b$) for $p_a < p_b$. Shortly, we will consider one conjecture that might provide such a theorem. We are also interested in favorability results which avoid the calculation of $WDL(X, Y)$. This could greatly increase the efficiency with which the tournament graph for a game is constructed. We will explore the possibility of using majorization to achieve such a result in *Independent Knock 'm Down.*

7.1 The Minimal Sub-allocation Conjecture

In Section 2, we remarked that we might expect that the minimal allocation of t tokens for a fixed P would contain the minimal allocation of $(t - 1)$ tokens as a sub-allocation. We formalize this conjecture in a more useful form as follows:

Minimal Sub-allocation Conjecture. *If X^* is a minimal allocation for the t-token game with fixed P, then there exists a minimal allocation for the $(t+1)$-token game that properly contains X^* as a sub-allocation.*

If this conjecture were true, we could add use it to greatly improve the efficiency with which we could search for minimal allocations. In particular, it would mean that we would only need to consider N potentially minimal allocations when moving from the t-token game to the $(t + 1)$-token game. Taking into account the *Minimal Allocation Theorems*, we would probably be required to calculate $E(T_X)$ for even fewer allocations. Better still, if this conjecture were true, we would further like to develop a simple rule to determine which value deserves the next token. One major difficulty in proving the *Minimal Allocation Conjecture* is the fact that we do not have a simple formula for $E(T_{X+e_i})$. Such a formula would greatly aid in considering what happens to allocations when small changes are made.

7.2 The Splitting Conjecture

We have recently considered the following conjecture:

Splitting Conjecture. *Let $P = (p_1, \ldots, p_{N-1}, p_N)$. Let $Y^* = [y_1^*, \ldots, y_{N-1}^*]$ be a minimal allocation for t-tokens with $P' = (p_1, \ldots, p_{N-1} + p_N)$. Then there exists a minimal allocation $X^* = [x_1^*, \ldots, x_{N-1}^*, x_N^*]$ for the t-token game with P, such that $y_1^* \leq x_1^*$, $y_2^* \leq x_2^*$, $\ldots$, $y_{N-2} \leq x_{N-2}$ and $y_{N-1}^* \geq x_{N-1}^* + x_N^*$.*

This conjecture makes sense from an intuitive point of view. When we split $p_{N-1} + p_N$, we decreased the probabilities of the last two values. Therefore, all things being equal among the other values, we would want no more than y_{N-1} tokens split between those values. In fact, if we split a high probability value into two small probability values, then we would want those less probable values to have even less tokens; since t is fixed, these "extra" tokens must be distributed among the other values.

A rigorous proof of this conjecture would yield both theoretical and computational advantages. We will briefly explore two of the consequences of this theorem, because they demonstrate the usefulness *Splitting Conjecture*

First, we note what can be accomplished through repeated applications of the *Splitting Conjecture*.

Conjecture. *Let $P = (p_1, p_2, \ldots, p_N)$. Let $Y^* = [y_1^*, y_2^*]$ be a minimal allocation for the t-token game with $P' = (p_1, p_2 + \cdots + p_N)$. Then there exists a minimal allocation $X^* = [x_1^*, x_2^*, \ldots, x_N^*]$ for the t-token game with P, such that $y_1^* \leq x_1^*$.*

Proof assuming the Splitting Conjecture. We prove the conjecture by induction on N.

Basis ($N = 2$): Let $P = (p_1, p_2)$. Let $Y^* = [y_1^*, y_2^*]$ be a minimal allocation for the t-token game with $P' = (p_1, p_2)$. Then there clearly exists a minimal allocation $X^* = [x_1^*, x_2^*]$ for the t-token game with P, such that $y_1^* \leq x_1^*$ – namely, $X^* = Y^*$.

Induction Step: Let $P = (p_1, p_2, \ldots, p_N, p_{N+1})$. Let $Y^* = [y_1^*, y_2^*]$ be a minimal allocation for the t-token game with $P' = (p_1, p_2 + \cdots + p_N + p_{N+1})$. Then, by the Induction Hypothesis, there exists a minimal allocation $Z^* = [z_1^*, \ldots, z_N]$ for the t-token game with $P'' = (p_1, p_2, \ldots, p_N + p_{N+1})$, such that $y_1^* \leq z_1^*$. By the *Splitting Conjecture*, there exists a minimal allocation $X^* = [x_1^*, \ldots, x_N, x_{N+1}]$ for the t-token game with P, such that $z_1^* \leq x_1^*$, $z_2^* \leq x_2^*$, $\ldots$, $z_{N-1} \leq x_{N-1}$ and $z_N^* \geq x_N^* + x_{N+1}^*$. Hence, $y_1^* \leq x_1^*$. □

Using the results of Section 5, we can calculate Y^* fairly easily. Further, we would hope that we could calculate a Y_i^* corresponding to each value in such a way that $y_{i,1} \leq x_i$ for each value. Then we could prove that:

Conjecture. *Let $P = (p_1, \ldots, p_{N-1}, p_N)$. Let $Y_i^* = [y_{i,1}^*, y_{i,2}^*]$ be a minimal allocation for the t-token game with $P' = (p_i, 1 - p_i)$. Then there exists a minimal allocation $X = [x_1^*, \ldots, x_N^*]$ for the t-token game with P, such that $y_{i,1}^* \leq x_i^*$ for all i.*

Hence, we would have a lower bound on the number of tokens that must appear on each value. This could be of great computational benefit, since if $y_{i,1}^* > 0$ for $i > 1$, then we need not consider the allocation

$[t, 0, \ldots, 0]$ as a potentially minimal allocation. In fact, we can often eliminate a fair number of the allocations that are potentially minimal according to the *Minimal Allocation Theorems*. For example, when finding the minimal allocation for the 30-token game with $P = (.4, .3, .2, .1)$, the *MATs* admit 132 potentially minimal allocations – but using the conjecture above, only 22 are potentially minimal.

The last conjecture also has some nice theoretical implications. Recall that in Section 5, we showed that

$$\lim_{t \to \infty} \frac{X^*(t)}{t} = P.$$

If $Y_i^*(t) = [y_{i,1}^*(t), y_{i,2}^*(t)]$ is a minimal allocation for the t-token game with $P' = (p_i, 1 - p_i)$, then we have that

$$\lim_{t \to \infty} \frac{y_{i,1}^*(t)}{t} = p_i.$$

Hence, by the last conjecture, have that

$$\lim_{t \to \infty} \frac{x_i^*(t)}{t} \geq \lim_{t \to \infty} \frac{y_{i,1}^*(t)}{t} = p_i.$$

Since this holds for all i and $\sum_{i=1}^N p_i = 1$, we must therefore have that

$$\lim_{t \to \infty} \frac{x_i^*(t)}{t} = p_i,$$

which shows that

$$\lim_{t \to \infty} \frac{X^*(t)}{t} = P$$

is true for all N-valued games.

7.3 Majorization and *Independent Knock 'm Down*

The game of *Independent Knock 'm Down* may be easier to analyze than *Original Knock 'm Down*. Although we saw one example of non-transitivity in Section 2, it was found only after searching through many *Independent* tournaments on a wide variety of probability distributions. We suspect that *Independent Knock 'm Down* may be provably transitive in some cases. In particular, it may be possible to show that certain sets of allocations must have transitive favorability.

One method for showing such a result might be to use the concept of majorization (see [1]). We say that a (finite or infinite) sequence $A = [a_0, a_1, \ldots]$ *majorizes* a sequence $B = [b_0, b_1, \ldots]$ if

$$\sum_{i=0}^{k} a_i \geq \sum_{i=0}^{k} b_i \text{ for all } k.$$

The property of majorization is transitive.

We can use majorization to analyze *Independent* games in the following way. For allocations X and Y, let $\langle X \rangle = [Pr(T_X = 0), Pr(T_X = 1), \ldots, Pr(T_X = i), \ldots]$ and $\langle Y \rangle = [Pr(T_Y = 0), Pr(T_Y = 1), \ldots, Pr(T_Y =$

$i), \ldots]$. We note that allocation X majorizes allocation Y if and only if

$$\langle X\rangle^T A \langle Y\rangle \geq 0,$$

where A is the infinite matrix

$$\begin{bmatrix} 0 & 1 & 1 & \\ -1 & 0 & 1 & \cdots \\ -1 & -1 & 0 & \\ & \vdots & & \ddots \end{bmatrix} = [sgn(j-i)]_{ij}.$$

The expression $\langle X\rangle^T A\langle Y\rangle$ is the difference between the probability that X wins against Y and the probability that Y wins against X. From another point of view, it is possible to consider A as the payoff matrix of a game, where $\langle X\rangle$ and $\langle Y\rangle$ are competing strategies. Considered in this light, it is clear that the optimal strategy for the payoff matrix A is the strategy with the probability vector $[1, 0, 0, \ldots]$, which will win against any strategy other than itself, with which it will draw. However, in *Knock 'm Down* it is impossible to construct an allocation of tokens with such a probability vector when there are multiple values on which to put a token.

We conjecture that if $\langle X\rangle$ majorizes $\langle Y\rangle$, then $\langle X\rangle^T A\langle Y\rangle \geq 0$ and the allocation X is favored over the allocation Y. Computationally, this is an infeasible method for determining favorability, since it requires the construction of infinite vectors. On the other hand, if we were able to prove that $\langle X\rangle$ majorizes $\langle Y\rangle$ using only the structure of X and Y, then we could determine the favored allocation without needing to compute $WDL_I(X,Y)$.

7.4 Other Directions

Finally, we wish to conclude by remarking on some of the directions of research that we have considered, but have not had time to pursue. In Section 6, we developed formulae for $VAR(T_X)$. However, we have not seriously considered the variance of the clearing time of an allocation as a potential indicator of the allocation's performance in a game of *Knock 'm Down*. One reason that the minimal allocation is not always the emperor might be the following: a minimal allocation with a large variance may be beaten by an allocation with a slightly higher expected value, but smaller variance. Intuitively, the minimal allocation might have the smallest expected clearing time, but if this is "spread out" over too wide a range of values, another allocation with a smaller variance might have the advantage in competition.

We conjectured earlier that

$$\lim_{t\to\infty} \frac{X^*(t)}{t} = P.$$

We might also consider the expressions

$$\lim_{t\to\infty} \frac{E(T_{X^*(t)})}{t} \qquad \text{and} \qquad \lim_{t\to\infty} \frac{VAR(T_{X^*(t)})}{t^2}$$

to see what asymptotic properties are held by minimal allocations.

Previously, we saw that it might be possible to use the 2-valued game results from Section 5 to increase the efficiency of our searches for minimal allocations in the N-valued game. We briefly conjectured that the 2-valued result could be used to directly construct the minimal allocation in the N-valued game. Suppose that we considered the N-valued game, but knew, through some means, that there was a minimal allocation which used exactly $t_{i,j}$ tokens on the i^{th} and j^{th} values together. Then, it seems reasonable that the $t_{i,j}$ tokens should be allocated between the i^{th} and j^{th} values in the same manner that t_0 tokens are allocated in the t_0-token game with $P = (\frac{p_i}{p_i+p_j}, \frac{p_j}{p_i+p_j})$. For example, in the 10-token game with $P = (.4, .3, .2, .1)$, the minimal allocation is $[5, 3, 2, 0]$. We note that the minimal allocation to the 7-token game with $P = (\frac{2}{3}, \frac{1}{3})$ game is $[5, 2]$, corresponding to the tokens on the first and third values in the minimal allocation to the 4-valued game. However, in the 20-token game, the minimal allocation is $[10, 6, 3, 1]$, but the minimal allocation to the 13-token game with $P = (\frac{2}{3}, \frac{1}{3})$ is not $[10, 3]$ but $[9, 4]$. One interesting direction of research would be to discover when the method described above works and why it does not in certain cases.

We remarked in Section 2 that emperor cycles in tournaments may not accurately capture our idea of the "best" allocations in a particular game. We might investigate more sophisticated rankings in tournaments. Vol'skii (see [8]) considers a number of different methods for choosing the best alternatives on directed graphs and tournaments, including "winning cycles" that are equivalent to our emperor cycles. However, all of these methods fail to take into consideration information other than the direction of the edges in the tournament graph. In the case of a *Knock 'm Down* tournament, each edge can have an associated weight – either the probability that the favored allocation will win the match or the difference between the favored allocation's winning and losing probabilities. Moon (see [6]) considers generalized tournaments where there is a directed edge between every pair of vertices and each edge is weighted such that $w_{ij} + w_{ji} = 1$ for all i and j. The tournaments considered by Moon do not admit draws between elements, but we might seek after such results. Since we have calculated an allocation's winning probability, it does not seem appropriate to discard this information when choosing the "best" allocation.

The question of local versus global results has not been adequately answered either. It is certainly not the case that a local emperor is necessarily a global emperor; see, for example, Figure 5, where $[4, 1, 0]$ is the only allocation favored over $[2, 2, 1]$, but $[4, 1, 0]$ is not a neighbor of $[2, 2, 1]$. However, we conjecture that a local emperor is always a member of the emperor cycle of the tournament. On the other hand, we have not found any counter-examples to the conjecture that a local minimal allocation is a global minimal allocation. If this were true, it might suggest that a "hill-climbing" strategy might be incorporated into a search for the minimal allocations. We have experimented with a hill-climbing method using WDL functions. Surprisingly, all tournaments seem to have a local emperor towards which a hill-climbing algorithm converges. This, along with the method for detecting cycles using strongly connected components might be a useful method for determining the emperor cycles of a tournament.

Finally, we could consider truels and other multi-player games. The rules of *Knock 'm Down* would appear to easily extend to multiple players. We suspect that different emperors and emperor cycles (appropriately formulated) would appear in the tournament graphs induced by these games.

8 Acknowledgments

Thanks to Steve Kremm and Professor Hank Krieger for bringing this game to our attention. Grateful thanks also to Professor Janet Myhre and the Institute for Decision Sciences for supporting this research.

Thanks to Professor Arthur T. Benjamin for support, guidance and encouragement during this research.

References

[1] M. Lawrence Clevenson and William Watkins. Majorization and the Birthday Inequality. *Mathematics Magazine*, 64:183–188, 1991.

[2] Persi Diaconis and Sandy Zabell. Closed Form Summation for Classical Distributions: Variations on a Theme of De Moivre. *Statistical Science*, 6:284–302, 1991.

[3] Dan Fendel and Diane Resek, with Lynne Alper and Sherry Fraser. *Interactive Mathematics Program*, Year 1. Key Curriculum Press, 1997.

[4] Gordon Hunt. Knock 'm Down. *Teaching Statistics*, 20:59–62, 1998.

[5] Stephen B. Maurer. The King Chicken Theorems. *Mathematics Magazine*, 53:67–80, 1980.

[6] John W. Moon. *Topics on Tournaments.* New York: Holt, Rinehart and Winston, Inc., 1968.

[7] Albert Nijenhuis and Herbert S. Wilf. *Combinatorial Algorithms.* New York: Academic Press, 1975.

[8] V. I. Vol'skii. Choice of Best Alternatives on Directed Graphs and Tournaments. *Automation and Remote Control*, 49:267–278, 1988.