1991

The Probabilities Associated with the Game Jai-alai

Robert Hesse
College of Saint Benedict/Saint John's University

The Probabilities Associated with the Game Jai-alai

A THESIS
The Honors Program
College of St. Benedict/St. John's University

In Partial Fulfillment
of the Requirements for the Distinction "All College Honors"
and the Degree Bachelor of Arts
In the Department of Mathematics

by
Robert Hesse
April 1991

The Probabilities Associated with the Game Jai-alai

Approved by

Philip Byrne

Associate Professor of Mathematics

Michael Tangredi

Associate Professor of Mathematics

Robert H. Dumonceaux

Professor of Mathematics

Michael Tangredi

Chair, Department of Mathematics

Mark C. Thamert, OSB.

Director, Honors Program

1 Table of Contents

2 Acknowledgements

I would like to thank the following people for their time and effort involved in this work. First and foremost I thank my advisor Philip Byrne for giving me the problem, the main approach used towards finding the solution to the problem, and several hours of patient listening and commenting. I would also like to thank my readers, Dr. Robert Dumonceaux, and Dr. Michael Tangredi, for their patience and willingness to read through my written work and mold it into something coherent. Lastly I would like to thank Dr. Lynn Ziegler and Dr. Tom Sibley for allowing me to come in and bounce ideas off of them when that was needed.

3 Introduction

This paper studies the probabilities within the game Jai-alai. Before explaining the probabilities, one first needs to understand the game itself. Jai-alai is a racquetball-like game where eight ordered players compete against one another in the following manner: Player 1 plays Player 2. The winner of this game plays Player 3, while the loser goes to the end of the line behind Player 8. In the first seven games, 1 versus 2, ? versus 3, ..., ? versus 8, the winner is awarded one point, and in every game after that two points. A player wins the match when he is the first to acquire seven or more points. The question involving probabilities is the following: Suppose all players are of equal ability. Is there any advantage to being in the front of the line (like players 1 and 2) versus being in the end of the line? Another way of restating the problem is calculate the probabilities each player has of winning the match given that they all are of equal ability.

Two approaches are used towards solving this problem. The first method involves running simulations on a computer and tabulating these results. Then using statistics a range can be set in which the probabilities are bounded. The second approach involves modeling the game by a Markov Chain. The Markov Chain can be written in matrix form, which in turn can be manipulated so that the probabilities of the players winning the game can be calculated exactly. Also in this second approach, it was necessary to look at smaller player number versions of the Jai-alai game to detect patterns that would be in the regular Eight Player Game. Each of these methods has advantages and drawbacks. The advantage of the simulation is that it can give a fair estimate of the probabilities in a short amount of time, while the advantage of the Markov Chain is that it can give the exact probabilities.

4 First Approach: The Simulation

4.1 Basic Idea of Computer Simulation

The first approach towards solving the problem involves the use of simulations. A program was written in the Pascal computer language that operates in the following manner. It takes the player labeled "defender" and has him compete against the player in the front of the line (In the initial case, Player 1 is the "defender", Player 2 is first in line). Next, the computer randomly picks a number between zero and one. If the number is greater then 0.5 (so the probability of each player winning is one-half) the "defender" wins and has either one or two points added to his score depending on how many total games have been played. If the "challenger" wins (if number ≤ 0.5), he is awarded the points, and becomes the new "defender". In either case, the loser is put at the end of the line, and the match continues until one player has acquired the seven or more points needed to win. In each batch the computer simulated 1 million of these matches. The following is the average probability of winning from ten of these batches.

4.2 The Results

Player	*Probability*
1	0.1632634
2	0.1632991
3	0.1383588
4	0.1240282
5	0.1020501
6	0.1025616
7	0.0887368
8	0.1177020

Notice Player 1 and Player 2 have approximately the same probability of winning. This result was expected since they held the same position at the beginning of the game, which means they share the same opportunities of victory. Another point of interest is the steady decrease in probability of victory as the player number increases, except for Player 8 who has better chances of winning then Players 5, 6, and 7. One surprising aspect is Players 5 and 6's approximate equivalence.

4.3 Statistical Analysis of Results

There are several methods of studying the above results using statistics. One method that can be applied is constructing a confidence interval on each players' matches won. In each of these tests, a ninety-five percent confidence interval was used so $z_{0.025} = 1.96$.

Player	$C.I. for Player$
1	(0.163034315, 0.163492484)
2	(0.163069995, 0.163528204)
3	(0.138144795, 0.138572804)
4	(0.123823903, 0.124232496)
5	(0.101862475, 0.102237724)
6	(0.102373559, 0.102749640)
7	(0.088560549, 0.088913050)
8	(0.117502264, 0.117901735)

The above table implies that for a given player ninety-five percent of the time the interval contains his actual probability of victory. Another point that can be studied is if Player 1 and 2 have the same probability of winning the match. This is worthwhile to do since if the possibility that their probabilities cannot be equal occurs, it means something is wrong with the simulation. Again it is possible to calculate whether the two are equal using confidence intervals between two independent sets of simulations. In every case the test was run, the possibility that they equaled occurred so the simulation appears to work. One could also run confidence intervals on players 5 and 6 to see if they too could be equal.

5 Second Approach: Markov Chains

5.1 Theory of Markov Chains

Before delving into the second approach towards solving the problem, it is necessary to have a quick overview of Markov Chains. Thus, here are a few necessary definitions:

(see for example MARKOV CHAINS: THEORY AND APPLICATIONS by Isaacson and Madsen.)

Definition 1 *The set of all possible outcomes of an experiment is called the sample space of the experiment. The sample space will be denoted by the symbol, Ω, and an arbitrary element of Ω will be denoted by ω.*

Definition 2 *A function that maps a sample space into the real numbers is called a random variable.*

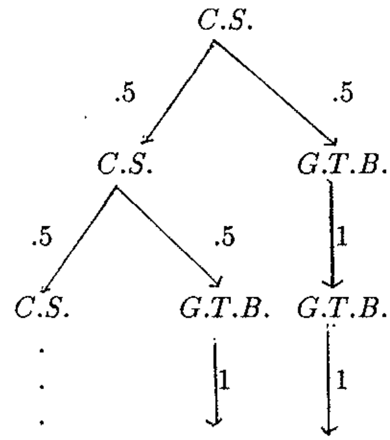
Definition 3 *A stochastic process is a family of random variables defined on some sample space, Ω.*

Definition 4 *The set of distinct values assumed by the stochastic process is called the state space. If the state space of a stochastic process is countable, or finite, the process will be called a chain.*

Definition 5 *A stochastic process X_k, $k = 1, 2, ...$ with state space $S =$ 1,2,3,... is said to satisfy the Markov property if for every n and all states $i_1, i_2, ..., i_n$ it is true that*

$$P[X_n = i_n | X_{n-1} = i_{n-1}, X_{n-2} = i_{n-2}, ..., X_1 = i_1] = P[X_n = i_n | X_{n-1} = i_{n-1}].$$

To clarify, a Markov Chain involves a process that moves from one state to another state with a certain probability. Take for instance the following example: Suppose a student is doing homework and decides that after twenty minutes she will flip a coin to determine whether she will continue studying. If the coin comes up heads, she will study twenty more minutes and then repeat the coin toss. If the coin comes up tails, she will go to bed. The possibilities *continue studying* and *go to bed* are states in the Markov chain each with transition probability 0.5 (Assuming of course she flipped a fair coin). Notice that a process in *continue studying* in turn has 0.5 probability of entering into *continue studying* or *go to bed*. On the other hand a process in *go to bed* can only move into *go to bed* and thus has probability one. (A state that enters only into itself is called an ***absorbing state***). Here is the general picture of this Markov Chain.



5.2 Application to Jai-alai

Looking at Jai-alai, notice the following similarities to the Markov Process. In Jai-alai, the probability of the players having a particular score and place in line is dependent only upon what occurred in the previous game. It also can be shown that the Jai-alai game must have a finite number of states. Thus it is possible to model Jai-alai as a Markov Chain where each game is a state and the probability of moving into a state (game) is only dependent on the outcome of the game before it. But how are these states in the game defined? Consider the following notation: Each player is represented by a set of ordered pairs, and where this set of ordered pairs lies determines their place in line. For example the initial state would look like the this: (1,0)(2,0)(3,0)(4,0)(5,0)(6,0)(7,0)(8,0) Since this is the beginning of the game, each player has zero points and their order is 1-8. Now the two states that a process in the initial state can go into look like:

$Player1wins = (1,1)(3,0)(4,0)(5,0)(6,0)(7,0)(8,0)(2,0)$
$Player2wins = (2,1)(3,0)(4,0)(5,0)(6,0)(7,0)(8,0)(1,0)$
transition probabilities of a process entering each equal $\frac{1}{2}$

5.3 Markov and Matrices

The next set of definitions explain the relation between Markov Chains and matrices.

A Markov Chain can be interpreted as a matrix in the following manner. Each row corresponds to a state in the chain. If a state x can enter state y with probability p, then matrix row x, column y, will have p in it. For instance in the *Study / Go To Bed* example the matrix would look like:

$$\begin{array}{cc} S. & G.T.B \end{array}$$

$$\begin{pmatrix} 0.5 & 0.5 \\ 0 & 1 \end{pmatrix}$$

Notice that every entry is non-negative and less than or equal to one. Also note that the sum of each row equals one. These properties as well as those mentioned in the above paragraph define a *Transition Matrix for a Markov Chain* . Thus a general form form for a transition matrix P of a Markov Chain would look like:

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ . & . & \cdots & . \\ . & . & \cdots & . \\ . & . & \cdots & . \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{pmatrix}$$

Where the p_{ij} are the probabilities of going from state i to j in the Markov Chain in one step.

In the Markov Chain for Jai-alai, all the states where a player wins the match are absorbing states. So for each row i in the transition matrix that corresponds to an absorbing state, column i has a value of 1. Thus if the transition matrix P is reconfigured such that the absorbing states were in the first rows and columns, our new transition matrix would look like:

$$P^* = \begin{pmatrix} 1 & 0 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ 0 & 1 & \ldots & 0 & 0 & 0 & \ldots & 0 \\ . & . & \ldots & . & . & . & \ldots & . \\ . & . & \ldots & . & . & . & \ldots & . \\ . & . & \ldots & . & . & . & \ldots & . \\ 0 & 0 & \ldots & 1 & 0 & 0 & \ldots & 0 \\ r_{1,1} & r_{1,2} & \cdots & r_{1,k} & q_{1,1} & q_{1,2} & \cdots & q_{1,(n-k)} \\ r_{2,1} & r_{2,2} & \cdots & r_{2,k} & q_{2,1} & q_{2,2} & \cdots & q_{2,(n-k)} \\ . & . & . & . & \ldots & . & \ldots & . \\ . & . & . & . & \ldots & . & \ldots & . \\ . & . & . & . & \ldots & . & \ldots & . \\ r_{n,1} & r_{n,2} & \cdots & r_{n,k} & q_{n,1} & q_{n,2} & \cdots & q_{n,(n-k)} \end{pmatrix}$$

Where the $r_{i,j}$ are the probabilities of entering an absorbing state from a non-absorbing state, $q_{i,j}$ are the probabilities of entering into a non-absorbing state from a non-absorbing state. Notice that the submatrix of absorbing states going into absorbing states is the identity matrix, and the submatrix of absorbing states going into non-absorbing states is the zero matrix. Thus matrix P^* could be rewritten as:

$$P^* = \begin{pmatrix} I & 0 \\ R & Q \end{pmatrix}$$

If P^* is an n by n matrix and the number of absorbing states is k, then I is k by k, 0 is k by $(n-k)$, R is $(n-k)$ by k, and Q is $(n-k)$ by $(n-k)$. Finally (as can be found in Isaacson) to calculate the probability of eventually entering an absorbing state from Q, one needs to find $N = (I - Q)^{-1}$, and then calculate NR. Another piece of information that can be calculated is the average number of steps it takes for a non-absorbing state to enter an absorbing state. This is found by taking the desired non-absorbing state's row, and summing up the entries of this row. For the Jai-alai problem this is equivalent to the average number of games played before the match ends.

5.4 Application of Markov Matrices

To create the Transition Matrix, it first is necessary not only to calculate all the possible states in the game, but also to describe all the possible ways a process can move into the various states. A Pascal program, Combo.pas, generated as output all the states, and how a process in one state entered into other states. This output is in the same form as described earlier for the states in the Jai-alai game, except that the parenthesis and commas have been removed. Remember that to find some patterns in the Jai-alai game it is good to look at some smaller player number versions of the game, and thereby detect possible patterns from those versions. Here is the output for the three-player game (first player to acquire two points wins match, first two games are worth one point, every game after is worth two points) and how the computer assigned the states to a matrix.

Combo.out	*continued*	*State*	*StateRepresented*
102030	102030	1	102030
113020	213010	2	113020
122030	221030	3	122030
122030	221030	4	312011
9	9	5	331120
113020	213010	6	221131
312011	311021	7	213010
331120	332110	8	221030
331120	332110	9	311021
9	9	10	322110
312011	311021	11	122131
221131	122131		221131
122131			
9	9		
9	9		
9	9		
	9		

The first string is the initial state for the three-player game. The next string is a state that a process from intial state might enter. This state contains the information that player 1 is the defender, has one point and is about to compete against Player 3 who has zero points, and the next player waiting in line is Player 2 who has zero points. Once again the following string is a state that the previous one could move into, and again contains certain information. This procedure continues on until there exists a state where a the following one is equal to it. This means the state is an absorbing state, and it has probability of one that it will enter into itself. In Jai-alai this state is interpreted as someone winning the game. The lines that contain only the digit 9 exist for the program interpreting the data, Matrix.pas. Matrix.pas has a procedure that converts the above data into a transition matrix P. Then another procedure rearranges the matrix into the P^* mentioned earlier. Now that P^* is created, all that is necessary is to find $N = (I - Q)^{-1}$. However for very large matrices, calculating N can become quite arduous. Thus it becomes necessary to take advantage of any special properties that the matrix $(I - Q)$ may have. Since the matrices for smaller player number versions of the Jai-alai game are similar to what the eight player game matrix looks like, it is useful to observe the Transition matrix for the three-player game and make some conclusions from that.

$$P^* = \begin{array}{cc} \begin{matrix} 3 & 5 & 6 & 8 & 10 & 11 & 1 & 2 & 4 & 7 & 9 \end{matrix} & \\ \begin{pmatrix} 1 & & & & & & & & & & \\ & 1 & & & & & & & & & \\ & & 1 & & & & & & & & \\ & & & 1 & & & & & & & \\ & & & & 1 & & & & & & \\ & & & & & 1 & & & & & \\ & & & & & & 0 & 0.5 & & 0.5 & \\ 0.5 & & & & & & & 0 & 0.5 & & \\ & 0.5 & 0.5 & & & & & & 0 & & \\ & & & 0.5 & & & & & & 0 & 0.5 \\ & & & & 0.5 & 0.5 & & & & & 0 \end{pmatrix} & \begin{matrix} 3 \\ 5 \\ 6 \\ 8 \\ 10 \\ 11 \\ 1 \\ 2 \\ 4 \\ 7 \\ 9 \end{matrix} \end{array}$$

Consequently,

$$(I - Q) = \begin{array}{cc} \begin{matrix} 1 & 2 & 4 & 7 & 9 \end{matrix} & \\ \begin{pmatrix} 1 & -0.5 & & -0.5 & \\ & 1 & -0.5 & & \\ & & 1 & & \\ & & & 1 & -0.5 \\ & & & & 1 \end{pmatrix} & \begin{matrix} 1 \\ 2 \\ 4 \\ 7 \\ 9 \end{matrix} \end{array}$$

Note two interesting observations of $(I - Q)$: first, it is upper triangular; and second, there are at most three entries per row, so for a large matrix $(I - Q)$ is sparse. These observations led towards the first approach of calculating $(I-Q)^{-1}$. Although for larger player number games, 5-Player Game and up, the matrices do not fit the pattern exactly, it is possible to alter them into the same form as for the 3-Player Game. The following is the actual $N=(I-Q)^{-1}$, R, and NR matrices for the three player game.

$$N=(I-Q)^{-1}=\begin{array}{cc} \begin{array}{ccccc} 1 & 2 & 4 & 7 & 9 \end{array} & \\ \begin{pmatrix} 1 & 0.5 & 0.25 & 0.5 & 0.25 \\ & 1 & 0.5 & & \\ & & 1 & & \\ & & & 1 & 0.5 \\ & & & & 1 \end{pmatrix} & \begin{array}{c} 1 \\ 2 \\ 4 \\ 7 \\ 9 \end{array} \end{array}$$

$$R=\begin{array}{cc} \begin{array}{cccccc} 3 & 5 & 6 & 8 & 10 & 11 \end{array} & \\ \begin{pmatrix} & & & & & \\ 0.5 & & & & & \\ & 0.5 & 0.5 & & & \\ & & & 0.5 & & \\ & & & & 0.5 & 0.5 \end{pmatrix} & \begin{array}{c} 1 \\ 2 \\ 4 \\ 7 \\ 9 \end{array} \end{array}$$

$$NR=\begin{array}{cc} \begin{array}{cccccc} 3 & 5 & 6 & 8 & 10 & 11 \end{array} & \\ \begin{pmatrix} 0.25 & 0.125 & 0.125 & 0.25 & 0.125 & 0.125 \\ 0.5 & 0.25 & 0.25 & & & \\ & 0.5 & 0.5 & & & \\ & & & 0.5 & 0.25 & 0.25 \\ & & & & 0.5 & 0.5 \end{pmatrix} & \begin{array}{c} 1 \\ 2 \\ 4 \\ 7 \\ 9 \end{array} \end{array}$$

5.5 First Approach: $(I-Q)^{-1}=I+Q+Q^2+Q^3+...+Q^l+...$

It can be shown that if the sequence of matrices Q^n converges to the zero matrix as n gets large, then the above equation is true (See for instance SPARSE MATRICES by U Schendel). Q does converge to the zero matrix since it is strictly upper triangular and $\mid q_{i,j} \mid < 1$ for all $1 \leq i,j \leq k$ where Q is k by k. But even better is the fact that Q^n converges to the zero matrix for a finite number n by the following: Q is the one-step transition of non-absorbing states into non-absorbing states. This implies Q^n is the n-step transition of non-absorbing states into non-absorbing states. But since the number of states in the Jai-alai game is finite, there is a $\delta > 0$ where δ is equal to the longest possible transition from a non-absorbing state to a non-absorbing state. Thus if $\delta = k$ then $Q^{\delta} = 0$. So by calculating the longest set of steps from the initial state to the last non-absorbing state, the value of δ can be determined. This algorithm gave results for the three, four, five, and six player games which are as follows:

Three-Player Game
The expected length of the match is 2.500000000 games
Player 1 has probability 0.375000000
Player 2 has probability 0.375000000
Player 3 has probability 0.250000000

Four-Player Game
The expected length of the match is 4.187500000 games
Player 1 has probability 0.296875000
Player 2 has probability 0.296875000
Player 3 has probability 0.156250000
Player 4 has probability 0.250000000

Five-Player Game
The expected length of the match is 6.070312500 games
Player 1 has probability 0.251953125
Player 2 has probability 0.251953125
Player 3 has probability 0.183593750
Player 4 has probability 0.175781250
Player 5 has probability 0.136718750

Six-Player Game
The expected length of the match is 8.395874023 games
Player 1 has probability 0.209869385
Player 2 has probability 0.209869385
Player 3 has probability 0.159057617
Player 4 has probability 0.150695801
Player 5 has probability 0.112792969
Player 6 has probability 0.157714844

These results were compared to the corresponding computer simulations, and appear to be valid. However, all attempts to calculate the results for the Seven-Player Game failed due to computer constraints. Thus a different approach was needed to solve this game, and in the end the Eight-Player game.

5.6 Second Approach: Gaussian Backsubstitution

Originally the approach of calculating the inverse using the Gaussian substitution was ruled out since this technique usually takes an excessively large amount of calculations to compute the inverse. However, notice that the elimination half of the Gaussian technique was unnecessary since the matrix $(I - Q)$ was already in upper triangular form. Thus all that needed to be done to compute the inverse was the backsubstitution technique (See for example Burden & Faires). Simply put, the backsubstitution technique does a series of row multiplications and additions to other rows to change $(I - Q)$ to I. At the same time these operations are also performed on an identity matrix, so as $(I - Q)$ is converted to I, I is converted to $(I - Q)^{-1}$. This new technique of calculating $N = (I - Q)^{-1}$ together with the original technique for calculating NR drastically reduced the amount of time necessary to calculate the results for the Five and Six-Player games, and after a long amount of time was able to calculate the results for the Seven-Player game:

Seven-Player Game
The expected length of the match is 10.782016754 games
Player 1 has probability 0.186703682
Player 2 has probability 0.186703682
Player 3 has probability 0.157234192
Player 4 has probability 0.137237549
Player 5 has probability 0.108762741
Player 6 has probability 0.120042801
Player 7 has probability 0.103315353

This approach, although faster then the previous one, was not fast enough to calculate the Eight-Player Game. Once again, a different technique was need to calculate the inverse.

5.7 Third Approach: $\bar{x}(I-Q) = (1\ 0\ 0\ \ldots\ 0)$

In the Jai-alai problem, the only information desired is the probability of each player winning the game given their initial place in line. But the matrices have been defined to have the initial or first game of the matrix as the first row of Q. So instead of calculating the inverse of $(I-Q)$, it is sufficient to calculate the first row of $(I-Q)^{-1}$, which will be called $\bar{x}$. Now since $(I-Q)^{-1}(I-Q) = I$ it follows that $\bar{x}(I-Q) = (1\ 0\ 0\ \ldots\ 0)$. Then $\bar{x}R = \bar{y}$ = set of probabilities of entering absorbing states from initial states. The algorithm calculated $\bar{x} = (x_1, x_2, \ldots, x_k)$ from the following:

$$
\begin{aligned}
x_1(1) &= 1 \\
x_1(q_{1,2}) + x_2(1) &= 0 \\
x_1(q_{1,3}) + x_2(q_{2,3}) + x_3(1) &= 0 \\
\ldots &= 0 \\
\ldots &= 0 \\
\ldots &= 0 \\
x_1(q_{1,k}) + \ldots + x_k(1) &= 0
\end{aligned}
$$

or

$$
\begin{aligned}
x_1 &= 1 \\
x_2 &= -x_1(q_{1,2}) \\
x_3 &= -x_1(q_{1,3}) - x_2(q_{2,3}) \\
. &= \ldots \\
. &= \ldots \\
. &= \ldots \\
x_k &= -x_1(q_{1,k}) - x_2(q_{2,k}) - \ldots - x_{k-1}(q_{k-1,k})
\end{aligned}
$$

Once again this new approach took less computer time then the previous one.

5.8 Calculating the Probabilities for the Eight-Player Game

Before continuing on, it is necessary to notice the growth of the matrix size as the number of players grew.

Game	*States*
Three – Player	11
Four – Player	43
Five – Player	203
Six – Player	1999
Seven – Player	21327
Eight – Player	844767

It would have been impossible to calculate the probabilities for the Eight-Player Game as is with anything less then a Cray SuperComputer. So it became necessary to break the Eight-Player Game down to a size much more manageable. To begin with, since players 1 and 2 share a certain symmetry, it is only necessary to look at half of the matches, say those where player 2 is the winner of the first game. So now the size has been cut down to 422,383. Now suppose this is broken again into two halves, run these two halves separately, and find the average of their respective probabilities. This reduces the matrix size to roughly 211,191. This procedure can be continued until the matrices are at a manageable size for the computer to manipulate. Then each of the results for the matrices (probabilities and expected number of games) can be summed and averaged together to give the average for the half *player2winsfirstgame*. Finally given this result one needs to average Player 1 and 2's probabilities to find the results for the total match. In actuality, the half *player2winsfirstgame* was cut into eighteen smaller matrices (at first sixteen, but two of them were too large and needed to be cut again). Here are the results for the Eight-Player Game:

The expected number of games is 13.777956963
Player 1 has probability 0.163116857
Player 2 has probability 0.163116857
Player 3 has probability 0.138615549
Player 4 has probability 0.124012306
Player 5 has probability 0.102026485
Player 6 has probability 0.102595806
Player 7 has probability 0.088775754
Player 8 has probability 0.117740393

Looking back at the Confidence Intervals in Section 4.3, notice that the Confidence Intervals did contain all of the above actual probabilities except Player 3's chances of victory. This is acceptable since the Confidence Intervals are only ninety-five percent accurate; it is within the realm of possibility to have them fail one of the eight times.

6 Patterns in Jai-alai

While working on different player number Jai-alai games, a few patterns kept appearing. First, in every even player game, the last player in line had better chances of winning then anyone else in the lower half. This could be attributed to the last player having the opportunity of winning the fewest number of games necessary to win the match. For instance in the Four-Player Game, Player 4 needs to win only two games to win the match while every other player needs to win three games. This does not work for the odd player games because the last player does not have the advantage of be able to play fewer games and win the match then everyone else; he shares that advantage with the player in line above him. Another observation involves the way the Transition Matrices were created. To keep the matrices in upper-triangular form, it was necessary to 'pretend' some states were different when in reality they were equal. For example, in the Five-Player game there was the state: 5523334312, which could be generated from 1253233343 or 5312233343. Instead of having a process from these two states go into the same state, the process entered a different state. So in general this technique eliminated the possibility of two process from different states entering into the same state. However this splicing of states did not alter the probabilities. This splicing though drastically increased the sizes of the matrices as made apparent by the following table.

$Game$	$Number of States$	$Number of Unique States$
$Three - Player$	11	11
$Four - Player$	43	43
$Five - Player$	203	199
$Six - Player$	1999	1599
$Seven - Player$	21327	11263
$Eight - Player$	844767	?

So as the number of players increased, the number of spliced states roughly doubled. (The amount of unique states has not been calculated for the eight player game since this would entail merging and sorting data files whose sum is over 30 megabytes). All these extra states make for a strong argument against splicing. However, not splicing creates matrices not in upper triangular form, and also matrices that are no longer sparse. Besides that, these new matrices also could not be broken into smaller pieces as done for the Eight-Player Game, and calculating the inverse (or just the first row of it) would no longer be a simple task.

7 Future Work

The previous paragraph mentions the problems of not splicing the states in the Transition Matrix, but possibly the compact version of the matrix has some 'nice' properties within it that enable it to to solve the problem. This should be looked into, since as the size of the matrices gets larger, any reduction in matrix size reduces memory and thus also time it takes to run the program.

Another aspect that could be looked into is changing the basic assumption. Suppose all players but one are of equal ability, and that one player is only slightly worse then everyone else. Are his chances of winning the game altered drastically from when he was equal to everyone else? Some preliminary work has been done on this involving Player 1 in the Eight-Player game. His chances of defeating each player were forty-nine percent, while everyone else chances remained the same. The first column of victories is with Player One having equal probabilities of winning, while the second one is the output from unequal ability.

Player	*Normal*	*Altered*
1	163564	152877
2	163507	166087
3	138738	140236
4	123593	124577
5	102238	103079
6	102395	103724
7	88165	89638
8	117800	119782

Player One wins 10687 (or one percent) fewer matches in the altered probability simulation then in the second equal probability simulation. So it appears that his chances for victory have changed proportionately with his probability against other players. However, these calculations need to be done to all players in the line, to see if they are affected in any way differently. More importantly however, if they are affected differently, an explanation why they are is in order.

8 Works Consulted

Burden, Richard L and J Douglas Faires. Numerical Analysis Boston: PWS-Kent Publishing Company, 1989.

Isaacson, Dean L, and Richard W Madsen. Markov Chains, Theory and Applications. New York: John Wiley & Sons Inc, 1976.

Kemeny, John G. Finite Mathematics. Englewood Cliffs NJ: Prentice Hall, 1972.

Larsen, Richard J and Morris L Marx. An Introduction to Mathematical Statistics. Englewood Cliffs NJ: Prentice Hall, 1986.

Mizrahi, Abe and Michael Sullivan. Finite Mathematics. New York: John Wiley & Sons Inc., 1976.

Schendel, U.Sparse Matrices. Trans. Brian Conelly. New York: John Wiley & Sons Inc., 1989.