2000

Optimal Expected Values for Cribbage Hands

Philip Martin
Harvey Mudd College

Optimal Expected Values for Cribbage Hands

by

Philip L. Martin

Hank Krieger, Advisor

Advisor: _____

Committee Member: _____

May 2000

Department of Mathematics

Abstract

Optimal Expected Values for Cribbage Hands

by Philip L. Martin

May 2000

The game of Cribbage has a complex way of counting points in the hands that are dealt to each player. Each player has a choice of what cards to keep and what cards to throw into an extra hand, called the crib, that one of the players gets to count towards his score. Ideally, you could try to keep the most points possible in your hand and your crib, or, conversely, the most points in your hand with the fewest points in your opponent's crib. To add to the fun, a final card is randomly chosen that all three hands share. This thesis deals with finding optimal expected values for each player's hand and the crib. Unfortunately, finding the exact optimal values is very difficult. However, I have managed to get bounds on the optimal values.

TABLE OF CONTENTS

ACKNOWLEDGMENTS

I would like to extend a very special thanks to my advisor, Professor Hank Krieger. Without his help, I would have certainly gone in several wrong directions. Also, thanks to my second reader, Professor Arthur Benjamin. Finally, I owe a great deal to the two people who kept all the thesis students on track this year, Professor Lesley Ward, and Professor Andrew Bernoff. Thanks to you all, you made my thesis experience exceptional.

Chapter 1

INTRODUCTION TO THE PROBLEM

1.1 Motivation

The game of Cribbage is not an easy game to pick up quickly. There are many rules that seem to be insignificant and yet are very important. The basic rules are not very difficult to understand, but there are a lot of intricacies that can only be picked up after many hours of play. Even then, there are still questions that have no clear answer. For instance, is it worth trying to get a flush? Should I care more about my hand or the crib? Should I keep the 7, 7, 8, or the 7, 8, 8? It's questions such as these that prompted me to learn more about the mathematics behind this intriguing game.

The game has many distinct phases. There are choices involved in each step that greatly determine how well each hand goes for each player: what cards to keep in your hand, which cards to throw into the crib, and what cards to play during the hand (see Appendix A for the Rules of Cribbage). Ideally, using game theory and a healthy dose of probabilistic modeling, I could determine answers to some of these questions, or at least strategies that increase the points you get out of each phase.

A search for research on Cribbage led to the realization that very little mathematical study has been given to this game. We see all the probabilities associated with other games such as Poker, or Blackjack, but no Cribbage. I suppose that if casinos played Cribbage, more research would go into it, but this lack of former work on the game did not help me. Everything here is directly from my own research.

Of course, it quickly became obvious that there were far too many aspects of the game to take into account at one time, so I was forced to narrow my search down a bit. One question that has always bothered me is how advantageous is it to have the first deal of the game? You get to count one more crib than your opponent, which is certainly an advantage, but how large of an advantage is it? In order to answer this question, the expected value of the crib needs to be determined.

There are two ways to approach finding the expected value of the crib, a Monte Carlo method by dealing out a large set of hands, or a more precise method of determining which cards get thrown into the crib based on the strategies of the players. The Monte Carlo method seems to be far easier, but optimizing this expected value for both players is tricky because one player wants the value to be high, while the other player wants it to be low.

An analysis of strategies is therefore preferable, leading to a more accurate measurement of the expected value of the crib. Unfortunately, there are a lot of strategies, $15\binom{52}{6}$ if you cannot eliminate any of them. Analyzing all of these is a very large problem, far out of the scope of my research, however, because I have some knowledge of the game, I could approach these strategies from a different angle, only examining strategies that seem to work well in the game. Then, however, the question becomes: How good are these strategies?

The obvious answer is: We don't know. Well, we can compare strategies, but we don't know how close they are to the best strategy, and the best strategy might be something unexpected. The natural next step is therefore to determine the optimal expected value and then develop strategies that get close to this expected value. This thesis is devoted to finding this optimal expected value.

1.2 Early Approach to the Problem

The first steps I took were to develop simple strategies for keeping cards in your hand that gave good expected values in your hand and the crib. Using a Monte Carlo method to deal out hundreds of millions of hands, I was able to find expected values of strategies that seemed reasonable to me, but in fact were quite wrong.

Having nothing to compare my numbers to, it was difficult to determine if they were accurate. Fortunately, however, I realized that there were only about 20 million possible hands that could be dealt, and running through those 20 million hands for each strategy was much simpler than dealing out hundreds of million hands and gave an exact expected value, which was better anyway.

After many bug fixes in my code, I was able to get accurate values for the expected values of different strategies. The exact value matched the value I got using the Monte Carlo method, but was very different than the first values I had gotten. However, now I had a method of checking my values and remove unnoticed bugs in my code.

Originally, I intended to come up with 20 or so strategies for both the dealer and the pone (non-dealer), and build a game matrix of the resulting expected payoffs for each strategy against the other strategies. Thus, I could reduce the problem to a simple game theory problem and find an optimal strategy.

At first, this seemed viable, and I was able to determine some strategies that you would never use, and some strategies that seemed quite good. However, this was without paying much attention to the crib. It was easy to find good strategies that helped out the expected value of the cards you kept, but the real question was how valuable were the cards you threw into your or your opponent's crib.

It quickly became obvious that finding the expected value of the crib was heavily dependent on the strategy your opponent was using in throwing his cards. However, in order to develop better strategies for the crib, a method for accurately determining expected values for the crib needed to be developed. This is far more difficult than

just finding the expected value of your hand (see chapter 3). This quickly became the focus of my research.

1.3 Desired results

Ideally, I could have modeled the entire game of Cribbage and worked a thousand strategies against each other to determine the advantage the first dealer has. However, this is impractical and the limited information on the mathematics behind the game of Cribbage made it difficult to make any progress in this direction.

My next idea was to run 10 to 20 strategies against each other and determine what strategies were good and could possibly be developed further for a more optimal strategy. This also became impractical because of the complexity of determining expected values for cribs. Also, I had no idea if the expected values of the strategies I was using were anywhere close to optimal. The optimal value could have been much higher.

Therefore, the focus has shifted to finding optimal expected values for both the hand and the crib. This does not provide a very practical strategy because it involves many probabilistic computations for each hand, but it does give us a value to shoot for in both the hand and the crib.

Chapter 2

EXPECTED VALUE FOR A HAND

2.1 What goes into the Expected Value

When given the five cards that make up a cribbage hand, it is relatively simple to determine the value of that hand. The value is determined by finding all the combinations of cards that give you points from those five cards (see Appendix A for point combinations).

When given just the four cards in your hand, and not knowing what the start card will be, we must do a calculation to find the expected value of those four cards given that one of 46 possible start cards will be chosen to make the five card hand. This also is not very difficult, and can be done simply by finding the mean of the 46 possible hands.

However, we must deal with being dealt a 6 card hand, deciding what 4 cards to keep from those 6, and then determine the expected value of that hand. Fortunately, there are only fifteen choices for the four cards to keep, so the decision is not too difficult. However, all of these steps must go into determining the expected value of any given hand.

2.2 An Example hand

Ok, let's say you are dealt the following six card hand:

$$A\heartsuit\ 4\spadesuit\ 7\heartsuit\ 8\spadesuit\ 9\diamondsuit\ J\diamondsuit$$

Now you are left with the unenviable task of choosing which 4 cards you want to

keep and which you want to throw into the crib. You will certainly want to keep the 7, 8, and 9 because a run of three is good to have, plus the 7 and 8 give you a fifteen. Choosing the fourth is a bit trickier, but for now, lets just say you keep the Ace. Now the start card is randomly chosen, and you are left with this:

$$A\heartsuit \ 7\heartsuit \ 8\spadesuit \ 9\diamondsuit \ \text{Start card: } 7\diamondsuit$$

This is a very nice hand, you have a double run:

$$7\diamondsuit \ 7\heartsuit \ 8\spadesuit \ 9\diamondsuit$$

worth **8** points, and three fifteens:

$$A\heartsuit \ 7\diamondsuit \ 7\heartsuit,$$
$$7\diamondsuit \ 8\spadesuit,$$
$$\text{and } 7\heartsuit \ 8\spadesuit$$

worth a total of **6** points. Therefore, the entire hand is worth **14** points, which is quite good.

2.3 Finding the Expected Value of a Strategy

A Strategy, in the sense that I am using it, is a set of rules such that when given a six card hand, the rules uniquely determine the choice for the 4 cards to keep. For example, one of my strategies was to keep as many points as possible in the four cards you keep. When there is a tie among possible throws, choose the cards with the lowest face value.

Thus, when given a strategy, it is useful to determine the expected value of that strategy given any possible six card hand. There are two ways to do this, the Monte Carlo method, and the heavily computational exact method.

The Monte Carlo method was my first choice. It is simple to code, enables you to get a lot of information quickly, and doesn't involve a lot of computation. You can

find expected values for the Dealer's hand, the Pone's (non-Dealer) hand, and the crib in one pass of the program. Unfortunately, it also has some problems. First of all, it is not exact, and secondly, it takes averaging about a hundred million hands to get a few digits of accuracy in the expected value. Keep in mind that there are only $\binom{52}{6} \approx 20$ million possible hands. There should be a simpler and more accurate method.

The second method I developed is to find the exact expected value for the strategy by finding the expected value of each of the $\binom{52}{6}$ hands and getting the mean. This gives the same expected value as the Monte Carlo method, but it is exact. Unfortunately, it only gives the expected value of the hand, and not the crib (see chapter 3 for crib values). Using this method, I was able to find accurate expected values for several strategies, and it also provided a check on the code I was using for the Monte Carlo method.

The question remains: How good are these strategies? Can I get better? What is the best expected value for a strategy? In other words, What is the least upper bound of expected values for strategies.

2.4   Maximizing the Expected Value

It's fairly easy to see that a strategy to maximize the expected value is not hard to come up with. It is simply: When given a six card hand, go through the 15 possible hands to keep and choose the one with the highest expected value given the 46 possible start cards. It's easy to see that this will produce an achievable upper bound on the expected value, and thus give us our maximum expected value:

$$\frac{7767205480}{46\binom{52}{6}} \approx 8.2939$$

The large number in the numerator comes from adding all the points counted for the $46\binom{52}{6}$ hands that are looked at when the discard is chosen to maximize these points.

The 46 in the denominator is for the 46 possible start cards for each hand, and the $\binom{52}{6}$ is for each of the possible hands that could be dealt.

As you may have noticed, this is all without paying any attention to the value of the crib. Given that the value of the crib is what separates the Dealer from the Pone, it would seem to be important. Unfortunately, finding the expected value of the crib is much more difficult that for just the hand.

Chapter 3

EXPECTED VALUE FOR THE CRIB

3.1   Using the Monte Carlo Method

The Monte Carlo Method provides us with a simple way to find the expected value of
the crib. We give the Dealer a strategy for deciding what cards to throw, and we give
the Pone a similar strategy. We can then deal out a hundred million hands, have the
Dealer and the Pone make their decisions based on their strategies, randomly choose
a start card, and then count up the points in the crib. The average value of the crib
over these hundred million hands is the expected value for the crib.

Unfortunately, this doesn't help us in determining an optimal strategy. The pri-
mary problem with this method is that it is dependent on the two strategies that must
be used, one for the Dealer, and one for the Pone. Thus we come back to the problems
mentioned earlier of playing hundreds of strategies against hundreds of strategies to
determine an optimal strategy for both players. This isn't practical.

Also, it is not useful to use the Monte Carlo method to determine the expected
value of the crib when using the optimal strategy discussed in Chapter 2. There is
nothing that says that the optimal strategy for maximizing points in your hand is the
same one that optimizes the value in the crib. In fact, it would seem counter-intuitive
that these would be similar strategies.

The Dealer would like to maximize the points he gets in his hand and in his
crib. Thus there are many hands that would be better served by throwing a pair or
fifteen into his crib instead of keeping it in his hand, reducing the potential points in
his hand, but increasing the potential points in his crib. Similarly, the Pone might

intentionally reduce the expected value of his hand if it reduced the expected value of his opponents crib by more.

The number for the expected value that the Monte Carlo method spits out is of little use to us because it doesn't help us optimize the strategies of the two players. We need another method to determine optimal strategies.

3.2   Finding An Expected Value

One of the problems with creating a strategy to help decide what two cards to throw (when thinking about the crib) is that there are so many possibilities for your opponents throw into the crib. It's relatively simple to get the expected value of your hand given the four cards you keep because there are only 46 possible start cards to worry about it. As seen in Section 2.4, it is straightforward to develop a strategy that maximizes this expected value.

However, it is far more difficult to guess how valuable are the two cards you throw into your or your opponent's crib. In some cases it's easy to see that you want to throw a pair into your crib, and thus you know you have at least 2 points in your crib. This is a nice situation, and thus comes up rarely. It doesn't help us in finding the optimal expected value, either.

Another way to determine the expected value of the two cards you throw into the crib is to average the resulting possible cribs. There are a lot of them, however, and just finding them is a very slow process, but this is what I chose to do.

To start, I built a very large matrix with all of the possible 4 card cribs and their resulting expected values given the possible start cards. If you think of this as a game matrix with the Dealer's strategies on the left and the Pone's strategies on the top, you can see where I'm going with this. Before seeing any cards, each player has $\binom{52}{2} = 1326$ possible throws into the crib. Each of these is a strategy in my matrix.

This is a very large table $\binom{52}{2}$ by $\binom{52}{2}$, but it is manageable, and more importantly,

	...	♣2♠6	♣2♣7	...
⋮		⋮	⋮	
◇6♠8	...	4.583	9.354	...
◇6♣9	...	7.583	6.042	...
⋮		⋮	⋮	

Figure 3.1: A sample from the Crib Matrix

it can be reduced considerably when you know which six cards you are dealt. When you have six cards dealt to you, you can reduce the number of choices you have to $\binom{6}{2} = 15$. Also, you can reduce the number of choices your opponent might have to $\binom{46}{2} = 1035$.

At this point, we can get an idea of the expected value of the crib for each of these 15 choices. We simply need to average all of the expected values in the row or column of that choice. Thus we have a method of attaching an expected value for the crib to each choice of crib throws. This will help us determine which choice to make for both the Dealer and the Pone.

A minor inaccuracy is introduced in my calculations while finding the averages of these choices. It is easily corrected, but takes a lot of time. See Appendix C for a discussion of this problem.

One obvious problem with this method is that it assumes that each of the choices for your opponent's throw are equally likely. In other words, when finding the expected value, you're assuming that your opponent is just as likely to throw a pair of fives as he is to throw an ace and a king. This is not true, and needs to be addressed.

3.3 How Game Theory Helps

In Game Theory, the concept of the value of the game is the payoff to one of the players when both players use their optimal strategy. For a single hand of cribbage, just counting the points in the two hands and the crib will give us this value. It is the points in the Dealer's hand plus the crib, minus the value of the Pone's hand. This is the value we are looking for in an optimal situation for both players.

Unfortunately, finding this value is outside the scope of my research. There are many options for trying to optimize the values of the crib, and I haven't been able to spend time developing them. We can, however, find bounds for the optimal values of each of these hands.

When we look at the method described in the previous section, this gives us an expected value for the crib that is different for the Dealer and the Pone. Because the averaging assumes all throws are equally likely, from the Dealer's point of view, the expected value will be higher than the actual expected value, and similarly for the Pone, the value will be lower than the actual expected value.

Since we can bound the optimal expected value of the crib, it would be useful if we could get similar bounds on the value of the Dealers' and Pones' hands. As we saw in chapter 2, the upper bound on these two values can be found easily, and the lower bound is given to us by the method in the last section.

3.4 Bounding the Optimal Values

The great thing about the method developed in section 3.2, is that along with the bound found in chapter 2, it gives us all of the boundaries needed.

First of all, for both the Dealer and the Pone, you are trying to maximize the value in your hand plus or minus the value of the crib. This will give a value lower than the bound found in chapter 2 because in using this method you will sometimes choose to keep cards with a lower expected value in order to either raise or lower the

expected value of the crib. Fortunately, this gives us a lower bound on the optimal expected value of the hand for both the Dealer and the Pone.

This is a lower bound because the correct distribution of throws by your opponent will remove some of the throws you thought were really good. This will lead you back to making choices that were better for your hand. In other words, when a more realistic distribution of crib throws is available, it is apparent that you have less control over the crib. Thus, with less control over the crib, you will want to try to increase the value in your hand more and worry about the crib less. Therefore, these expected values are a lower bound.

Lower bounds on the expected value for the hands:

$$
\begin{aligned}
\text{Dealer's hand} &= 8.01249586652 \\
\text{Pone's hand} &= 8.14340876324
\end{aligned}
$$

These values are obtained by averaging the expected value for the hand (not the crib) of each of the $\binom{52}{6}$ hands when you make the correct choice of cards to throw with respect to the crib. For example, when doing this from the Dealer's perspective, in order to get the expected value for the Dealer's crib that is described below, he/she will choose to reduce the expected value in his hand occasionally in order to increase the expected value of the crib. This might be done by throwing a pair into the crib or other similar tactics. These lower bounds come from doing this over all $\binom{52}{6}$ hands.

The method also gives us both a lower and upper bound on the optimal expected value for the crib. The lower bound is the expected value from the Pone's prospective. The reason that this is a lower bound is because it is from the point of view that all throws coming from the dealer are equally likely, which is not true. Actually, some of the nice throws from the Dealer will never occur. This will increase the actual expected value from the value the Pone gets while using this method.

Similarly, the Dealer's expected value is too high using this method. The Dealer

has assumed that it is just as likely that the Pone will throw two good cards into his crib as anything else. This means that the expected value can only go down from what he gets using this method.

The bounds on the expected value for the crib:

$$\text{Lower Bound} = 4.19821737416$$
$$\text{Upper Bound} = 5.22911232834$$

As above, these values are obtained using the crib matrix method described earlier. Both the Dealer and the Pone make their crib throw choices for each of the $\binom{52}{6}$ possible hands. When they make this choice, they get an expected value for the crib from a row (or column) of the crib matrix. This value is artificially raised or lowered because the distribution of opponent's throws is inaccurate. The average of all of these $\binom{52}{6}$ expected values, however, gives us these bounds on the optimal expected value for the crib from both player's perspectives.

A nice thing to note here, is that the range for these possible values is relatively small, and therefore, it would not take an enormous amount of work to tighten these boundaries. Unfortunately, I've run out of time.

Chapter 4

RESULTS

4.1 The Boundaries for Optimal Expected Values

There are three expected values that I have tried to optimize for this thesis: one for
the Dealer's hand, one for the Pone's hand, and one for the crib. Because it was
impossible to compute these optimal values exactly, I have been able to find bounds
for these expected values :

$$8.0125 \; < \; \text{Dealer's hand} \; < \; 8.2939$$
$$4.1982 \; < \; \text{The Crib} \; < \; 5.2291$$
$$8.1434 \; < \; \text{Pone's hand} \; < \; 8.2939$$

Fortunately, these bounds are rather tight and despite being unable to find exact
optimal expected values, we have a very good idea of where they are, and thus I
have accomplished the goal I had: to get an idea of how close the strategies I was
developing were to optimal strategies.

4.2 Why This is Helpful

In my initial ideas for research, I wanted to develop optimal strategies for the Dealer
and the Pone to choose cards to throw into the crib. Using these strategies, I could
further build a model for the game of cribbage and eventually discover how much of
an advantage it was to have the first deal of the game. It is impossible to develop an
accurate model of the game if you have no idea what values are possible for optimal
expected values.

For example, one of my early strategies for the Pone involved always avoiding throwing a pair or fifteen into the crib. It sounds like a good plan because you never give away free points to your opponent. However, it turns out this is not even close to optimal because the expected value for the Pone's hand in this strategy was less than 8. Now this strategy might be useful in certain circumstances, but if your only goal is to negate the Dealer's advantage of counting the crib as much as possible, then this is not a good strategy.

Without any idea of what an optimal strategy was, it was impossible to see that this strategy was hurting the Pone more than helping. With the good idea of where the optimal values are, we can develop strategies that can actually be played by a person with minimal amount of heavy computations, and we can see how close this strategy is to optimal.

4.3   Possible Future Research

The problem with the distribution of throws into the crib is a very interesting one. It is obvious that the uniform distribution that was used here is not correct, but there are many other distributions that might be closer to optimal. I could see a normal distribution being closer to optimal, the really good and really bad throws get thrown much less often, but this seems to have some problems as well. It would appear that some heavy strategical analysis could go into the dealer and the Pone developing a distribution for their opponent. Each distribution is a strategy for the player, and thus you could develop a game matrix and see which strategies brought the bounds on the expected value close together. This seems to be the next step in finding an exact value for the optimal expected value.

Also, the idea of optimal strategies in the pegging round of each hand has obviously not been touched upon. When you want to keep good cards for pegging, as well as close to the end of the game when you know that you won't get to count your hand

and only the pegging points count, certain cards might be much better to keep both to peg points and to prevent pegging. This is not an entirely separate question from the optimal hand to keep. Taking into account good pegging cards will certainly change what is considered optimal and change the values in this thesis. However, it probably won't change them much, because in most cases the deciding factor in what cards to keep is the potential for counting points in the hand.

4.4 Conclusion

How valuable is it to get the first deal in your average Cribbage game? It's not an easy question to answer. When I started this research, I actually thought I might have a chance of answering it. In some sense I have answered it: You get, on average, to start with an extra 4.2 to 5.2 points because you count an extra crib. Unfortunately this isn't all the answer. There are other advantages to having the first deal, and while the few extra points for counting the extra crib are a factor in the answer to this question, they are not the whole picture. The question is far more complex than I would have ever thought, and I'm glad to have figured out just one part of it.

Appendix A

THE RULES OF CRIBBAGE

Taken from the American Cribbage Congress's Web Page.

The object of the game cribbage is to obtain 121 points before your opponent does. This is accomplished by "pegging " or moving your pegs around the board as you score points. It is also a game where etiquette is important. The rituals associated with cutting and dealing, playing and pegging, as well as the terminology, all serve the useful purpose of keeping things in order - and they help to give the game a flavor of its own.

A.1   Terms and Definitions

Card Values or Pip Values Two players use a standard 52 card pack and the Cards rank K (high) Q J 10 9 8 7 6 5 4 3 2 A (low).

Cut for Deal The two players each cut a card from a shuffled deck to determine who starts the game as Dealer and Pone (non Dealer), low card wins. If the cards are equal the deck is shuffled and the players cut again.

Dealer This is the player dealing the cards for a hand (round) and will have the crib that hand.

Pone The player who is dealt to, also the person who is dealt to for the first hand. Dealer and Pone alternate each hand, however, in determining game strategy the first Pone is often referred to as the Pone for the game. In tournament play the Pone cuts the deck before the Dealer deals the hand.

The Deal The Dealer alternately deals 6 cards to both the Pone and himself. The Dealer places the remaining deck of cards face down on her side of the table.

The Discard Both players discard, face down, 2 of the 6 cards to the Dealer's side of the table.

The Starter or Cut Card After the cards have been dealt and the players have contributed to the crib, the Pone cuts the deck and the Dealer turns over the top card of the remaining deck. This is the Starter card. The Starter card is not used during the play of the hands. The Pone must cut at least 4 cards and must leave at least 4 cards during the cut.

Card Combinations Points may be obtained by getting various card combinations such as pairs, runs, flushes etc. They are presented with their point value in the table below.

The Play or "Pegging" The objective here is two fold... Each player, starting with the Pone, takes turns laying one card on the table. As they do, if they achieve any of the card combinations as shown in the later table, they peg the respective points. In addition, if they reach 31 exactly, they peg 2 points. The complete breakdown of card combinations and points follows. Players announce the total pip value of the cards played by both players and any points scored. The player must peg the points announced prior to playing another card. Players peg by moving the rear peg ahead of the front peg the number of points "holes" scored. The front peg is the players most recent score and the rear peg the previous score.

The Crib This refers to four cards that were placed face down on the Dealer's side of the table at the start of a hand of play. They remain face down until the end of the hand at which point they are counted and the points are pegged by

the dealer. Possession alternates as the players each takes his or her turn as the dealer.

The Show or "Counting the Hand" After the Play, each player counts the points in their hand starting with the Pone, then the Dealer and finally the Dealer's crib. The Start card may be used to add to possible point values.

Round A Round is the play of cards during a hand when the players attempt to score points. The round ends when a player scores 31, gets a "Go" for playing the last card of a round without reaching 31, or plays the last card of a hand without reaching 31.

Hand A hand consists of one or more Rounds and the Count. At the end of the hand, the points are counted and pegged, and the Pone becomes the Dealer and deals a fresh hand. The new Dealer also get the crib as well.

The Cribbage Board A cribbage board is a wooden (plastic, ivory, etc.) board with small holes, in groups of five, drilled into it. The players are represented by pegs that fit in the holes. In any case, each player has two pegs: the forward peg shows the player's score to date, and the rear peg shows the previous score. When a player scores points, the rear peg is moved in front to show "Peg" the new score. That way the distance between the pegs shows the amount most recently scored, and the opponent can thereby check it has been scored correctly. Fine Wood tournament boards as used for ACC tournament are available at SCM Tourney Boards.

The Winner The player to peg 121 points or more, first.

The Dead Hole The player to peg 120 points exactly is said to be in the "dead" or "stink" hole. If you lose the game in this position you are "left in the stink

hole."

Skunk A player who loses by more then 30 points is considered to be "skunked", 60 points is double skunked and 90 points is triple skunked.

A.2   The Game Sequence

Starting the game

The game begins with a shuffle of the cards. Each player cuts a card and the lowest card wins, that person gets the first crib and deals the first hand. The cards are then shuffled by the dealer. After the first hand, the opposite player deals the cards and receives the crib. The dealer and possession of the crib alternates each hand until the game ends.

Dealing and the Crib

Each player is dealt six cards. They keep four of their choice and discard two cards into the crib on the Dealer's side of the table. The crib remains face down until the end of each hand when the crib points are tallied and added to the dealer's total. The remaining deck is left face down until each player has made their choices for which cards they wish to keep and which cards they want to discard into the crib.

The Cut

After each player decides which cards they want in their hand and the crib, the Pone cuts the deck and the top card of the remaining deck, the Starter card, is turned over by the Dealer. If the Starter card is a Jack, then the Dealer pegs (receives) two points immediately. (Called "Two for Nibs")

The Play

The object of this part of cribbage is to total the pip values of the cards played to

equal 15 or 31 points without going over. Cards are valued by their number except for the face cards (Jack, Queen and King) which are worth ten points. Aces are always valued as "one". The Pone begins by playing the first card to the table, leading. Each player alternately plays a card to the table, announcing the total pip value of the cards played, until 31 is reached. In addition to going to 31, players may also get points for achieving any of the combinations listed below... such as a pair, run or or three of a kind, etc. If a player only has cards in her hand that will exceed 31, then they say "Go" and the other player may use his/her remaining cards to get to 31 and/or to obtain any of the possible card combinations. If both players only have remaining cards that will exceed 31, then the last player to place a card on the table gets one point for "The Go" and the other player begins the next round. Play continues again until all cards have been used. The person playing the last card not exceeding 31 pegs one point for the "Last Card".

If a player receives a "GO" they must play all cards in their hand that would, in sequence, not exceed 31. This does not mean that if the player has 2 cards and either one could be played under 31, but not both of them, that they are both played, only one or the other, whichever the player chooses to play.

The Show or Count

After the play comes the Show. Starting with the Pone, his/her hand of cards is displayed and the results are pegged. Then the dealer displays his/her hand, pegs the points and then turns over the crib and pegs those points. The objective here is to make any of the mentioned card combinations, except 31. The Start card (the face up card on the deck) is used with each player's hand as well as the crib and the scores are pegged accordingly. After the count the Pone becomes the Dealer and shuffles and deals the cards thus beginning the next round and hand.

A.3 Cribbage Points

Fifteens Any combination of cards totaling 15

> **2** points for each combination

Thirty-one Any combination of cards totaling 31

> **2** points but only during play

Last Card Being the last player to Play a card without exceeding 31

> **1** point but only during play

Pairs Two cards of the same rank

> **2**

Three of a kind Three cards of the same rank (a.k.a Pair Royal)

> **6**

Four of a kind Four cards of the same rank (a.k.a Double Pair Royal)

> **12**

Runs A combination of three or more cards in sequence (they need not be of the same suit)

> **1** point per card in sequence

Double runs Two 3 card runs that containing a pair

> **6** points for two runs + **2** points for the pair = **8**, only counted when the hand is counted
> —Two 4 card runs that contain a pair
> **8** points for two runs + **2** points for the pair = **10**, only counted when the hand is counted

Triple runs Three 3 card runs that share a three of a kind

> **9** points for three runs + **6** points for the three of a kind = **15**, only counted when the hand is counted

Double double runs Four 3 card runs that have two pairs

> **12** points for four runs + **2** points for each pair (total **4**) = **16**, only counted when the hand is counted

Flushes Four cards of the same suit in the hand

> **4**, only counted when the hand is counted
>
> —Five cards of the same suit, four in the hand plus the Start Card
>
> **5**, counted in either the hand or the Crib

His Nobs A jack of the same suit as the Start card

> **1**, only counted when the hand is counted

Nineteen Because it is impossible to obtain this number regardless of the card combinations, it is the term for a useless hand

> **Zero**

Appendix B

JAVA CODE

B.1   Cribbage.class

This class was designed to encompass all of the methods to actually grab the 4 cards you want to keep, count the points in a hand, and keep track of strategies used. It contains all of the utility methods that are useful like seeing if two hands are equal and printing out a hand into readable format. Basically, if it was going to be used in all the other classes, it made it into this class.

```
package philip.cardgames.cribbage;
import java.util.Vector;
public class Cribbage {
    //Keep random cards, pay no attention.  NOT A GOOD STRATEGY !!!!
    public static final int STRATEGY_ZERO = 1000;
    //This strategy is to maximize the points you have in your hand,
    //not caring about the crib at all.  With a tie, keep lowest cards.
    public static final int STRATEGY_ONE = 1001;
    //This strategy is to maximize the points you have in your hand,
    //not caring about the crib at all.  With a tie, keep highest cards.
    public static final int STRATEGY_TWO = 1002;
```

and so on with more strategies and other static variables...

```
    public static void main(String args[]) {
```

While the code is left out here, this method initialized input variables and then created an instance of the main class that could run as many games of cribbage as you wanted to before exiting:

```
    new Cribbage();
}
public void Cribbage() {
```

This method creates all of the variables that I want to keep track of over the life of this program. This method is specifically designed for using the Monte Carlo method and dealing n hands out and averaging all the values. In the end, it prints out a lot of information that is useful about the strategies used and the expected values that were found.

```
}
```

The following two methods are used to count how many points are in the hand that are passed into it. The first is based on a four card hand, whether or not it is the crib, and the start card (upCard) that is used for this hand. The second only counts the points in a four card hand. This is useful for some strategies.

```
public static int countHand(int[] hand, int upCard, boolean isCrib) {
    int retValue = 0;
    retValue += countFifteens(hand, upCard);
    retValue += countPairs(hand, upCard);
    retValue += countRuns(hand, upCard);
    retValue += countFlush(hand, upCard, isCrib);
    retValue += countNobs(hand, upCard);

    return retValue;
```

```
    }
    public static int countHand(int[] hand) {
        int retValue = 0;
        retValue += countFifteens(hand);
        retValue += countPairs(hand);
        retValue += countRuns(hand);
        retValue += countFlush(hand);


        return retValue;
    }
}
```

**The rest of the methods in this class haven't been included because they
are heavily computational and not very interesting. They involve methods
such as countFifteens as seen above that actually count all the fifteens in a
hand. Also there are methods for each strategy to pick the cards that that
strategy would want you to keep. There are also various utility methods
that aren't very interesting.**

B.2 EVofStrategy.class

This class is included almost in its entirety because it utilizes a method to go through
all $\binom{52}{6}$ hands that are possible and averages their expected values. This same method
is used in the CountCrib class (see next section). However, in this class, the crib
doesn't matter, all that matters is the expected value of the hand. This was the
primary class used in finding the maximum expected value in Chapter 2.

```
public class EVofStrategy {
  public static void main(String[] args) {
    //initialize input variables.
```

```
   new EVofStrategy();
}
public EVofStrategy() {
  int counter = 0;
  int card1 = 0;
  int card2 = 1;
  int card3 = 2;
  int card4 = 3;
  int card5 = 4;
  int card6 = 5;
  double numberOfUpCards = 46;
  while(card1 < 47) {
    card2 = card1 + 1;
    while(card2 < 48) {
      card3 = card2 + 1;
      while(card3 < 49) {
        card4 = card3 + 1;
        while(card4 < 50) {
          card5 = card4 + 1;
          while(card5 < 51) {
            card6 = card5 + 1;
            while(card6 < 52) {
              int dealt[] = {card1, card2, card3,
                card4, card5, card6};
              int crib[] = new int[4];
              int hand[] = new int[4];
              Cribbage.choose(dealt, hand, crib,
                true, strategy);
```

```
    double evOfHand = 0;

    int upcard = 0;

    while(upcard < 52) {

      boolean con = false;

      for(int i = 0; i < dealt.length; i++) {

        if(dealt[i] == upcard) {

          con = true;

        }

      }

      if(con) {

      }

      else {

        int counted = Cribbage.countHand(

          hand, upcard, false);

        evOfHand += counted;

      }

      upcard++;

    }

    ev += evOfHand;

    counter++;

    if(counter%10000 == 0) {

      //print out data...

    }

    card6++;

  }

  card5++;

}

card4++;
```

```
            }
            card3++;
        }
        card2++;
    }
    card1++;
}
  }
}
```

B.3 CribCount.class

This class is the primary class to find the boundary values given in Chapter 3. It goes through all $\binom{52}{6}$ hands like EVofStrategy does, but it makes choices based on both the hand and the crib instead of just the hand.

```
public class CribCount {
    public static void main(String args[]) {
        //initialize variables for the class, then the matrix:
        initCribMatrix();
        new CribCount();
    }
    public static void initCribMatrix() {
        //This method creates the large matrix of 1326x1326
        //representing all the possible throws into the crib.
        //Later, this matrix will be manipulated many times to
        //help find expected values of the crib.
    }
    public CribCount() {
```

```
        //This method goes through all approx. 20 million hands
        //like in EVofStrategy.  For each hand, it grabs values
        //for that hand for each of the expected values that we
        //want.  For the Dealer's hand and crib, and the Pone's
        //hand and crib.  All of these values are averaged out and
        //spit out as results.
    }
}
```

B.4   Other classes

There were several other classes that I used over the course of this research. All of these were earlier versions of the three main classes already discussed, or their functions helped at the time, but did not eventually contribute to the final results of this thesis.

Appendix C

A SMALL CRIB MATRIX PROBLEM

In order to reduce the time that it takes to run the program I have written to find the bounds on the expected value of the crib, I start by building a "crib expected value matrix," as described in chapter 3. This matrix is made up of entries that have two cards from the row, and two cards from the column representing the throws into the crib from the Dealer and the Pone. The expected value is then based on the 48 possible start cards that could be turned up given those four cards were in the hand.

The use of 48 start cards is where the inaccuracy occurs. When the start card is turned up, any given player has seen six cards. Therefore, there are 46 possible start cards. Thus, given that you know what six cards are in your hand, each entry in the matrix should have only 44 possible start cards, not the 48 that are used. The 46 you know it might be, minus the two that are thrown into the crib from that specific column of the matrix.

It is fairly simple to write the code that alters each cell in the matrix and removes the possibility of those extra 4 cards. This is what I did at first. The problem with this is that it's changing 3 hundred-million cells over the life of the program, and this slowed the program down by a factor of 20. It would have taken about 100 days to run the program with this level of accuracy.

My solution was to remove this level of accuracy and go with the assumption that those 4 possible start cards in each of the rows that were being averaged would not significantly alter the average over 1035 columns of expected values. Thus the values that are achieved are not exact and there is a simple way to obtain more exact values, it is just very time intensive.