Optimal Strategy for Casino Blackjack: A Markov Chain Approach

Kevin Coltin*

August 7, 2012

Abstract

By modeling the game of casino blackjack as a Markov chain, we derive the optimal strategy for the game without using simulations. We present a replicable algorithm for calculating this optimal strategy that can be easily modified to account for variations in casino rules. A player using this strategy will reduce the casino's advantage to 0.498%, meaning that the player will lose an average of 50 cents for every $100 bet.

1 Introduction

The game of casino blackjack lends itself to mathematical analysis using probabilistic models. Each player plays only against the dealer, who is required to play a known, deterministic strategy. As a result, an optimal strategy for the game can be computed exactly without using simulation and without requiring knowledge or assumptions of other players' strategies. In this paper we present a method using Markov chains for deriving this optimal strategy and for determining the player's probability of winning a game given the set of cards he has. Computer simulations of several million hands of blackjack are used to confirm the correctness of this method.

This paper is about playing strategy—the choices that a player makes when playing a given hand, as opposed to betting strategy—the strategy of how much a player should bet before the cards are dealt. Countless books and websites purport to reveal the optimal playing strategy, but the advice given by these sources differs. This is both because strategy guides are constructed based different variations of blackjack rules, of which there are many [6], and because they use different methods to determine the best strategy. For example, the strategies advocated by two canonical sources, Thorp [3] and the Hoyle games company [1], differ slightly from each other and from the strategy derived in this paper. This paper may resolve disputes among these sources by presenting

*Arizona State University. Email: kcoltin@asu.edu. Thanks to Eric Kostelich and Courtney Bruce for their help.

a verifiable algorithm used to compute the optimal playing strategy that can be easily modified to account for differences in rules.

The main results—the optimal strategy and player's probabilities of winning with each possible hand—are presented in Table 6.3. Although it is impossible to beat the casino on average, a player playing with this strategy will have expected losses of only 0.498% of the amount bet.

2 Rules and Assumptions

2.1 The rules of blackjack

The basic rules of casino blackjack are as follows: The player plays only against the dealer. The player and the dealer are dealt two cards from a standard deck of 52. Each card is given its face value, with face cards worth ten points. As such, the terms ten and face card are used interchangeably. Aces are worth either one or 11 points, at the discretion of the player holding them. A hand is called "soft" if it contains an ace being counted as 11, and "hard" otherwise. For example, a hand consisting of an ace and a six is a soft 17, and a hand with ace, ten, and six or of a ten and a seven is a hard 17. The word "hand" refers both to a player's current set of cards and to each round of the game, ending when both player and dealer either stand or bust; the meaning is clear in context. The goal is to finish with more points than the dealer, without exceeding 21 points. Both of the player's cards are dealt face up, while one of the dealer's cards is dealt face up (the "up card") and the other face down (the "down card"). The player may then choose to "hit," draw another card, or "stand," stop hitting and stay with the cards he has. Hitting allows the player to increase his points but also carries the risk that his total will exceed 21 and he will "bust." After the player has either stood or busted, the dealer takes her turn. She is required to hit until she has a total of 17 or greater and then stand. At this point the player wins the hand if he finishes a higher total than the dealer or if the dealer busts, and loses if dealer has a higher total or if the player busts. If both bust, the dealer wins. If both end up with an equal total, it is a "push" and the player keeps his bet.

2.2 Assumptions used

We assume the following standard rules: the dealer is required to hit when her total is 16 or less and stand with 17 or more, with the exception that she hits with a soft 17 (a 17 including an ace being used as 11). Splitting and doubling are allowed with any hands; these strategies are described in Sec. 6. Surrender is not allowed. A blackjack (being dealt an ace and a ten) pays three to two on the player's bet. The algorithm is flexible enough that it could be easily modified to account for changes in these rules.

A few other assumptions should be noted. First and most importantly, it is assumed that the probability of drawing any particular card from the deck is always 1/13, regardless of the cards that have already been drawn. When playing with six or more decks as is standard in most casinos, the impact of this assumption is negligible [4]. For example, if there is a seven in play, the actual probability of drawing another seven from a stack of six decks is actually 23/312, so the difference in approximating it as $1/13 = 24/312$ is negligible. The results from simulations, described in Sec. 8, confirm that this assumption is justified. This also means that we are ignoring card counting strategies, which are beyond the scope of this paper.

Next, it is assumed that the desire of the player is to maximize the expected value of his winnings—or, more accurately, to minimize his expected losses, since it is not possible to beat the casino on average. This means that in some circumstances playing the optimal strategy derived here will mean increasing one's bet by doubling down or splitting, which increases the expected value of the player's winnings at the cost of also increasing the variance.

We ignore the option of taking insurance, a side bet that allows the player to hedge against the dealer having blackjack, because it is widely agreed to be unprofitable for the player (see for example [7]).

Finally, in calculating the player's strategy we ignore the possibility of either the player or the dealer having "blackjack," meaning being dealt an ace and a ten or face card in the initial hand. If this happens the game ends and the party with the blackjack wins automatically before the player having the chance to take any action. Therefore, given that the player is even in a position to make a strategy choice, it must be that neither party has blackjack. The effect of this is to change the probabilities of what the dealer's down card may be. E.g., if the dealer's up card is an ace, we may assume that her down card is not a ten, so by Bayes' rule the probability of the down card being any card in two through nine or an ace is now 1/9 rather than 1/13.

3 Blackjack as a Markov Chain

The game of blackjack can be formulated elegantly as a Markov chain. A discrete-time stochastic process is a family of random variables (X_i), $i \in \mathbb{N}$, and a Markov chain is a stochastic process that has the Markov property:

$$\mathbb{P}(X_i = x_i | X_{i-1} = x_{i-1}, \ldots, X_1 = x_1) = \mathbb{P}(X_i = x_i | X_{i-1} = x_{i-1}) \qquad (1)$$

for all $i \in \mathbb{N}$ and where the x_i's are possible states of the system. That is, the probability distribution of each state depends only on the value of the state immediately preceding it, and is memoryless with respect to how the system arrived at the preceding state. In blackjack, the states are the possible hands that a player may have: hard 4 through 21, soft 12 through 21, and bust (22 or more). (Because aces take the value 11 when it is possible to do so without busting, it is impossible to have two or more cards totaling less than four, or totaling less than 12 if one of them is an ace.) Each time indexed by i represents the player drawing another card: X_1 is the two-card hand the player is originally dealt, X_2 is his hand after one hit, and so on. The chain stops at X_N for some finite N when the player either chooses to stand or goes bust. For $m > N$, define X_m to equal X_N.

It is clear from the rules of the game that it is irrelevant how a player arrived at his current hand. For example, a hand of two face cards is no different than one of two sixes and an eight. This means the game obeys (1). The only factor distinguishing two hands with the same total value is whether they are hard or soft (whether they contain an ace being used as 11). Thus at any point in the game the only factors relevant to the player are his current hand and the dealer's up card. The up card can be considered as a parameter of the system that is constant throughout each hand of play.

4 The Markov Chain for the Dealer's Hand

To optimize the player's strategy for beating the dealer's point total, the dealer's possible outcomes must be modeled. Since the only available information about the dealer's hand is her up card, and since the dealer's play is independent of the player's, the only question that must be answered about the dealer's hand is: given the dealer's up card, what is the probability that she will finish with each possible point total?

As in Sec. 3, the dealer's hand can be described as a Markov chain (Y_i) taking values in a state space $\mathcal{H}$, the set of all possible hands. A Markov chain may be completely defined by a transition matrix $Q = (q_{ij})$, $i, j \in \mathcal{H}$, where $q_{ij} = \mathbb{P}(Y_n = j | Y_{n-1} = i)$ at every time n. That is, the entry q_{ij} is the probability that the dealer will next have hand j given that she currently has hand i. By raising Q to the power n we obtain the n-state transition matrix. Defining $q_{ij}^{(n)}$ to be the i, j entry of Q^n, we have that $q_{ij}^{(n)}$ is the probability that the dealer will be holding hand j after n steps given that her current hand is i.

A state j is accessible from a state i if for some $n > 0$, $p_{ij}^{(n)} > 0$. A state i in a Markov chain is an essential state if for all j such that j is accessible from i, i is also accessible from j. It is an absorbing state if it is impossible to move from that state to any other state. Define $A \subset \mathcal{H}$ as $A := \{\text{hard 17–21, soft 18–21, bust}\}$. Recall that the dealer is required to hit until her hand is at least 17 (at least 18 if it is a soft hand) and then must stand. Thus, A is the set of all absorbing states in $\mathcal{H}$ and is also precisely the set of essential states.

The dealer's transition matrix Q is computed using the following pseudocode algorithm. Note that the cards in the deck are indexed by $k \in \{1, 2, \ldots, 10\}$ with one standing for ace and face cards counted as ten. The probability of drawing card k is $4/13$ if $k = 10$ and $1/13$ otherwise.

Program 1.

```
for i = 1 to number of possible hands:
   if i is in A:
      P(i,i) = 1
      P(i,j) = 0 for all j /= i
```

```
else:
   for k = 1 to 10:
      j = calculateNewHand (currentHand=i, cardDrawn=k)
      p = probability of drawing card k
      P(i,j) = P(i,j) + p
```

The function `calculateNewHand(i,k)` should return the hand j that results when a player hits while holding hand i and draws a new card k. Pseudocode for this function follows. In it, a hand is implemented as a data structure with attributes value, the point value of the hand, and isSoft, a boolean value indicating whether it is a soft hand.

Program 2.

```
function calculateNewHand (currentHand, cardDrawn) {
newHand.value = currentHand.value + cardDrawn

// Count aces as 11 initially
if cardDrawn = 1:
   newHand.value = currentHand.value + 10
   newHand.isSoft = true
// If the hand is soft and exceeds 21, count the ace as 1
if newHand.isSoft = true and newHand.value > 21:
   newHand.value = newHand.value - 10
   if currentHand was soft and cardDrawn is an ace:
      newHand.isSoft = true
   else:
      newHand.isSoft = false
if newHand.value > 21:
   newHand = bust

return newHand
}
```

Because the dealer continues to hit until she reaches an absorbing state, taking Q^∞ will give us the transition matrix showing the probabilities that the dealer will finish with each possible hand given her starting hand. (In practice, it is sufficient to simply take Q^{20} since that is already more hits than the dealer can possibly take before she either stands or busts.) Let $\pi^0 = (\pi_i^0)$ be a row vector of length $|\mathcal{H}|$ where π_i^0 is the probability that the dealer is initially holding hand i. Then by right-multiplying by the transition matrix we get

$$\pi^0 Q^\infty = \pi$$

where $\pi = (\pi_i)$ and π_i is the probability that the dealer will finish with hand i.

Given the dealer's up card, we can compute the vector π^0 which gives the probability distribution of the starting hands that the dealer could potentially be holding. For example, if the up card is a two, then there is a $4/13$ chance that the dealer's starting hand is twelve (i.e. the down card is a ten), a 1/13 chance that it is 11 (the down card is a nine), and so on. Following this reasoning, the following algorithm computes the vector π^0:

Program 3.

```
function calculateDealersStartingHand (upCard) {
for i = 1 to 10: // i ranges among all possible "down" cards
   hand = getHand (upCard, i) // returns the hand consisting of the
                              // up card and card i
   pi0(hand) = pi0(hand) + probability of drawing card i[1]

return pi0
}
```

Finally, we multiply to get $\pi = \pi^0 Q^\infty$ where π is the vector of probabilities that the dealer will end up with each possible hand. Clearly $\pi_i = 0$ for all i not in A, the set of absorbing states. The up card gives the probabilities that the dealer will finish with each value between 17 and 21 inclusive or that she will bust.

5 The Player's Optimal Strategy

The problem of determining the player's optimal strategy consists of determining whether he should hit or stand given each possible combination of his hand and the dealer's up card. We will ignore the more advanced options of doubling down and splitting hands until Sec. 6. Since the choice of whether to hit or stand depends solely on the player's current hand and the up card, with the series of steps by which the player arrived at his current hand being irrelevant, the player's strategy is a Markov chain (X_i) taking values in $\mathcal{H}$, the set of possible hands. The up card is a parameter of the chain that does not change within each chain. This means we can write $X_i = X_i(k)$ where k, the up card, is constant over all i.

The player's "strategy chart," $C = (c_{ik})$, can be implemented as a $|\mathcal{H}| \times 10$ matrix of data structures called "Strategies." A Strategy has the following data members: a parameter taking values "hit" or "stand" and values *winpct* and *losspct* indicating the player's probabilities of winning and losing a hand. Note that it is necessary to include both *winpct* and *losspct* as they may not add up to 100% because a hand may end in a push, or tie. Therefore, the entry c_{ik} is a Strategy corresponding to the scenario where the player is holding hand i and the dealer's up card is k: it contains the action that the player should take in this situation along with his probabilities of winning or losing the hand.

The Markov transition matrix used to obtain the player's optimal strategy is the "hit transition matrix" $P = (p_{ij})$, $i, j \in \{1, \dots, |\mathcal{H}|\}$. This is the one-step transition matrix of a player whose strategy is to hit on every hand until he

[1]As described in Sec. 2.2, this is actually the probability of drawing card i given that the dealer's hand is not a blackjack.

busts. The entry p_{ij} gives the probability that a player holding hand i who hits will next have hand j. For example, if i is a hard ten and j is a hard 18, then $p_{ij} = 1/13$, the probability that the card drawn will be an eight. The matrix P is created by iterating over all combinations of hands and cards, and for each hand i and card k, adding the probability of drawing k (4/13 for a ten, 1/13 otherwise) to entry p_{ij}, where j is the hand resulting from adding k to hand i. Pseudocode follows:

Program 4.

```
function makeHitTransitionMatrix () {
N = number of possible hands
P = N x N zeros matrix

for i = 1 to N:
   if i = bust:
      P(i,i) = 1 // Bust is an absorbing state - player can't leave.
   else:
      for k = 1 to 10:
         j = calculateNewHand (i, k) // see Program 2
         P(i,j) = P(i,j) + probability of drawing card k

return P
}
```

We now have the tools necessary to compute the strategy chart C. For each combination of player's hand i and dealer's up card k, we can calculate the player's probabilities of winning and of losing the hand both if he stands and if he hits. The probabilities of winning and losing if the player stands are simple to calculate from the matrix Q^∞ which gives the probability that the dealer will end up with each possible total 17–21 or bust given her up card. If the player's hand is worth n points and the up card is k, then his probability of winning the hand is the sum of the probability that the dealer busts and the probabilities that the dealer finishes with p points for $p \in \{17, \ldots, n-1\}$ (recalling that the dealer must finish with at least 17 points). Conversely, the probability of losing the hand is the sum of the probabilities that the dealer finishes with p points for $p \in \{n+1, \ldots, 21\}$.

By the law of total probability, the probability that the player wins the hand given that he hits is the sum of the probabilities that his next hand after hitting is each possible hand multiplied by the probabilities that he will win given each possible hand:

$$\mathbb{P}(\text{win}|\text{hand } i) = \sum_{j \in \mathcal{H}} \mathbb{P}(\text{next hand is } j|\text{current hand is } i)\, \mathbb{P}(\text{win}|\text{hand } j) \quad (2)$$

where all probabilities are implicitly conditioned on the event that the up card is k. If we define C' to be the matrix (c'_{ij}) whose i, j entry is the *winpct* data member of the corresponding entry of C—i.e., c'_{ij} is the probability the player will win if he is holding hand i against up card j—then (2) becomes, in vector notation:

$$\mathbb{P}(\text{win}|\text{hand } i) = \mathbf{p}_i \mathbf{c}'_k \tag{3}$$

where $\mathbf{p}_i$ is the ith row vector of P and $\mathbf{c}'_k$ is the kth column vector of C'. Having calculated the probabilities of winning and losing whether the player stands or hits, and letting β be the amount of the player's bet, the player's expected value for the hand in either case is:

$$\begin{aligned} \mathbb{E}(\text{winnings}|\text{stand}) &= \beta\,\mathbb{P}(\text{win}|\text{stand}) - \beta\,\mathbb{P}(\text{loss}|\text{stand}) \\ \mathbb{E}(\text{winnings}|\text{hit}) &= \beta\,\mathbb{P}(\text{win}|\text{hit}) - \beta\,\mathbb{P}(\text{loss}|\text{hit}). \end{aligned} \tag{4}$$

All probabilities are of course conditioned on the events that the current hand is i and the up card is k. Then the player's action, stored in c_{ik}, should be to stand if $\mathbb{E}(\text{winnings}|\text{stand}) > \mathbb{E}(\text{winnings}|\text{hit})$ and hit otherwise. This choice is independent of the bet β.

There is one obstacle to this approach: computing each entry c_{ik} requires knowing the vector $\mathbf{c}'_k$, which may not be fully known yet. As such it is necessary to compute the c_{ik}'s in the correct order so as not to create a circular definition. If the player hits on a hard 21 then he is guaranteed to bust, so hard 21 is computable. Hitting on hard 20 can only lead to a hand of hard 21 or a bust, so hard 20 is computable, and so on. Soft hands complicate this somewhat, but the process still works without becoming circular. In practice it is unnecessary to figure out the exact order in which the hands must be solved, because a computer can simply loop repeatedly over the entries of C until all are solved. The following program computes the entire chart C:

Program 5.

```
function calculateStrategyChart () {
P = makeHitTransitionMatrix ()
do
   for i = 1 to number of possible hands:
      for k = 1 to 10:
         stand_winpct = probability that dealer finishes with fewer
                        points than hand i is worth, given up card k
         loss_winpct = probability that dealer finishes with more
                       points than hand i is worth, given up card k
         hit_winpct = p(i) * c_win(k) // row i of P, column k of C'
         hit_losspct = p(i) * c_loss(k)

         if hit_winpct and hit_losspct cannot be calculated:
            continue

         if stand_winpct - stand_losspct > hit_winpct - hit_losspct:
            C(i,k).action = stand
         else:
            C(i,k).action = hit
```

```
until every entry of C is solved

return C
}
```

6 Advanced Strategies: Doubling Down and Splitting Hands

In addition to hitting and standing, there are two more actions the player may take. These may only be done before the player has made any other move, when he is holding only the two cards he is originally dealt. This is why we can compute the original strategy chart with Program 5 before we have to consider these strategies: the player can never transition to a hand in which they are permitted, so they do not affect the choice of whether to hit or stand on any other hand.

When a player doubles down, he doubles his bet on the hand and draws exactly one more card, after which he must stand. This allows him to double his potential winnings on hands that are favorable. For example, players often double down on hands of 11 because of the high chance they will be dealt a face card and get 21. The downside, besides the possibility of losing twice as much, is that the player can only draw one more card, even if that card is low. For instance, a player who doubles down on nine and gets a two is then unable to hit with a hand of 11.

Splitting may only be done when the player has two identical cards, such as a pair of eights. In this case he may choose to split the cards into two separate hands and play each hand separately. Each new hand is played as though it were an ordinary hand, and the player may even double down on a new hand or, if dealt a third identical card, split the hand again. He must put down a second bet equal to his original bet so that the original bet amount is being wagered on each of the two new hands. This can transform a weak hand into two better hands and, like doubling down, allows the player to double hit bet in a favorable situation, but it also carries the risk of losing twice as much.

Throughout this section, all probabilities are implicitly conditioned on the event that the dealer's up card is k.

6.1 Doubling down

To determine whether doubling down is advisable for a given combination of player's initial hand i and dealer's up card k, we must determine the probabilities of winning and losing the hand if the player doubles down, and then determine if the expected value of the hand with doubling

$$\mathbb{E}(\text{winnings}|\text{double down}) = 2\beta\,\mathbb{P}(\text{win}|\text{d.d.}) - 2\beta\,\mathbb{P}(\text{loss}|\text{d.d.}) \tag{5}$$

is greater than the expected value that was previously calculated ignoring the possibility of doubling—the greater of the two formulas in (4). The size of the bet β is doubled.

To calculate the probabilities of winning and losing if the player doubles down, note that the distribution vector of the current hand i is $\mathbf{e}_i$, the $1 \times |\mathcal{H}|$ vector whose ith entry is one and the rest zeros. Multiply $\mathbf{e}_i P$, where P is the hit transition matrix from Sec. 5, and let the result be π. This is now the distribution vector of the hand that the player will end up with. Lastly, the expected value when doubling down is $\pi\left(\mathbf{p}_{\text{win}} - \mathbf{p}_{\text{loss}}\right)$ with $\mathbf{p}_{\text{win}}$ being the $|\mathcal{H}| \times 1$ vector whose ith entry is the probabiility of winning if the player stands on hand i, and $\mathbf{p}_{\text{loss}}$ defined analogously. The entries of these two vectors are calculated in the same was as the player's probabilities of winning and losing if he stands, in Sec. 5.

6.2 Splitting hands

Determining the player's expected value if he splits his cards is slightly more complicated because of the possibility of repeated splitting: after splitting a hand, say a pair of eights, the player may be dealt another eight, making one of his two new hands the same as his original hand. Let the "new hand" mean the hand resulting from one of the two split hands plus the next card dealt to it. (After splitting, the player is automatically dealt one new card to each the cards he split, forming two new hands of two cards each.) If we let w_h equal the player's winnings from each one of his new hands, we have:

$$\mathbb{E}(w_h) = \sum_{i \in \mathcal{H},\, i \neq h} \left(p_i \mathbb{E}(w_i)\right) + p_h \cdot 2\mathbb{E}(w_h) \tag{6}$$

where p_i is the probability that the new hand will be hand i, and $\mathbb{E}(w_i)$ is the expectation of the player's winnings if he is holding hand i. By the additivity of expected values, the player's expected value from the entire hand is $\mathbb{E}(w_h) + \mathbb{E}(w_h)$, the sum of the expected values of each of the new hands. This explains the $2\mathbb{E}(w_h)$ term in (6), since it is the expected value of the new hand if another identical card is dealt, making it the same as the original hand.

Rearranging (6), the expected value if the player splits is given by

$$\mathbb{E}(\text{winnings}|\text{split}) = 2\mathbb{E}(w_h) = \left(2 \sum_{i \in \mathcal{H},\, i \neq h} p_i \mathbb{E}(w_i)\right) / (1 - 2p_h). \tag{7}$$

Each value p_i can be computed easily; it is the probability of drawing the card, which is unique if it exists, that will transform one of the two split cards into hand i. For example, if the player splits a pair of eights, then the probability that each one of the new hands will be a soft 19 is 1/13, the probability of drawing an ace. The values of $\mathbb{E}(w_i)$ are computed from the strategy chart C, by subtracting the relevant probability of losing *losspct* from the probability of winning *winpct* and multiplying by the size of the bet β, or 2β if the player would double down with the given hand. When comparing the expected value of the player's winnings if he splits to his expected winnings otherwise, β again cancels out, making the decision independent of the size of the bet.

6.3 Strategy chart

The completed chart showing the player's optimal strategy for each combination of his hand and the dealer's up card is shown as Table 6.3. In the table, each cell indicates whether to Hit, Stand, SPLit, or Double Down. The numbers X/Y means the player has an $X\%$ chance of winning the hand and a $Y\%$ chance of losing. Totals do not add to 100% because of ties. For split hands, this is the percent chance of winning each of the two new hands. Some hands for which the strategy is considered obvious are omitted to save space. On hands where the chart says to double down, when doubling is not possible, the player should stand on soft 18 or 19 and hit otherwise.

7 Expected Value of this Strategy

To compute the expected value of a player's winnings when playing with the strategy advocated here, the following formula was used:

$$\mathbb{E}(\text{winnings}) = \pi_0^\top \mu. \tag{8}$$

The vector π_0 is the probabilities that the player starts with each possible hand. This is simple to calculate: for example the probability of starting with a hand of seven is $2(1/13)^2 + 2(1/13)^2$, the sum of the probabilities of being dealt a two and a five and of being dealt a three and a four. Then μ is the vector of expected values of the player's winnings when starting with each hand, which is calculated as described in Sections 5 and 6.

The player's expected value when playing with this strategy is -0.00498β. A player betting \$10 per hand and playing at a typical rate of 100 hands per hour [2] will lose an average of \$4.98 per hour. The casino's "house edge" is 0.498%. This compares favorably with other casino games.

In practice, the player's expected losses could be greater or less than this. It is well documented that the player's odds are better if a smaller number of decks is used [5]. By assuming a constant probability of drawing each card as explained in Sec. 2.2, we have effectively been assuming that the number of decks is infinite. Although the effect of this assumption on the choice of optimal strategy is negligible, it does have the effect of slightly worsening the player's overall odds as compared to what they would be with a finite number of decks. On the other hand, a real player's odds could be worse due to less than perfect adherence to optimal strategy, either because of the difficulty of memorizing the strategy for every situation or because he may decide not to split or double when strategy calls for it, particularly in situations where optimal strategy dictates resplitting or doubling down on a split hand. The player would be deciding to prioritize a lower variance of returns over maximizing his expected value.

	2	3	4	5	6	7	8	9	10	A
9	H 50/42	DD 50/44	DD 51/42	DD 53/41	DD 55/39	H 53/36	H 49/39	H 39/44	H 36/52	H 37/49
10	DD 55/37	DD 56/36	DD 58/35	DD 59/34	DD 61/32	DD 56/36	DD 53/39	DD 49/42	H 43/40	H 45/42
11	DD 58/34	DD 59/33	DD 61/32	DD 62/31	DD 63/30	DD 58/35	DD 55/38	DD 52/41	DD 51/42	DD 48/43
12	H 35/60	H 36/59	S 40/60	S 42/58	S 44/56	H 36/57	H 33/60	H 29/63	H 27/65	H 27/65
13	S 36/64	S 38/62	S 40/60	S 42/58	S 44/56	H 33/60	H 30/63	H 27/66	H 25/68	H 25/68
14	S 36/64	S 38/62	S 40/60	S 42/58	S 44/56	H 31/63	H 28/65	H 25/68	H 23/70	H 23/70
15	S 36/64	S 38/62	S 40/60	S 42/58	S 44/56	H 28/65	H 26/68	H 23/70	H 22/72	H 21/72
16	S 36/64	S 38/62	S 40/60	S 42/58	S 44/56	H 26/68	H 24/70	H 22/72	H 20/74	H 20/74
A,2	H 49/45	H 51/43	H 52/42	H 54/41	DD 53/43	H 52/39	H 47/42	H 43/46	H 39/50	H 39/49
A,3	H 48/46	H 50/45	H 52/43	DD 51/45	DD 53/43	H 50/42	H 45/44	H 41/48	H 38/52	H 38/51
A,4	H 47/47	H 49/46	DD 49/46	DD 51/45	DD 53/43	H 48/44	H 44/46	H 39/50	H 36/54	H 36/53
A,5	H 47/49	H 48/47	DD 49/46	DD 51/45	DD 53/43	H 46/46	H 42/49	H 38/52	H 35/55	H 35/55
A,6	H 46/46	DD 47/45	DD 49/43	DD 51/42	DD 53/40	H 44/39	H 41/48	H 37/52	H 34/54	H 34/56
A,7	DD 49/43	DD 50/42	DD 52/40	DD 54/39	DD 55/38	S 63/23	S 37/27	H 39/49	H 37/52	H 35/51
A,8	S 62/25	S 63/24	S 65/23	S 66/22	DD 58/35	S 77/15	S 73/14	S 47/18	S 47/41	S 49/30
A,A	SPL 65/26	SPL 66/25	SPL 67/24	SPL 68/23	SPL 69/22	SPL 70/21	SPL 66/23	SPL 62/27	SPL 59/30	SPL 58/30
2,2	H 42/54	H 44/52	SPL 47/49	SPL 49/47	SPL 51/46	SPL 44/46	H 37/53	H 33/57	H 31/60	H 31/60
3,3	SPL 43/53	SPL 45/51	SPL 47/50	SPL 48/48	SPL 50/47	SPL 43/49	H 35/57	H 31/60	H 29/63	H 28/63
4,4	H 45/47	H 46/46	H 48/45	SPL 47/50	SPL 49/48	H 48/40	H 38/44	H 34/55	H 32/57	H 30/57
5,5	DD 55/37	DD 56/36	DD 58/35	DD 59/34	DD 61/32	DD 56/36	DD 53/39	DD 49/42	H 43/40	H 45/42
6,6	SPL 42/55	SPL 44/53	SPL 45/52	SPL 47/50	SPL 49/48	H 36/57	H 33/60	H 29/63	H 27/65	H 27/65
7,7	SPL 41/51	SPL 43/50	SPL 45/49	SPL 47/47	SPL 49/45	SPL 38/44	H 28/65	H 25/68	H 23/70	H 23/70
8,8	SPL 46/46	SPL 47/45	SPL 49/43	SPL 51/42	SPL 52/42	SPL 50/38	SPL 39/43	SPL 35/54	SPL 33/56	SPL 31/55
9,9	SPL 50/43	SPL 51/42	SPL 53/40	SPL 54/39	SPL 56/38	S 63/23	SPL 50/41	SPL 39/44	S 35/53	S 28/51
10,10	S 75/12	S 76/12	S 77/11	S 78/11	S 78/11	S 85/7	S 86/7	S 82/6	S 59/4	S 70/10

Table 1: The player's optimal strategy as a function of the player's current hand (vertical axis) and the dealer's up card (horizontal axis). See details in Sec. 6.3.

8 Simulations

Although no simulations were needed to calculate the player's optimal strategy, Monte Carlo simulation may be used to verify its correctness. We simulated 10,000 hands of blackjack played for each possible combination of the player's starting hand and the dealer's up card. (That is, 10,000 hands were simulated where the player started with a pair of twos and the dealer's up card was a two; 10,000 where the player started with a pair of twos and the up card was a three; etc.) The computer "player" was programmed to play using this optimal strategy. For every possible combination of hand and up card, the percentage of times that the computer won and the percentage of times that it lost were both within 2% of the computed probabilities shown in Table 6.3. This confirms that the algorithm used for calculating the strategy is correct. This result held even when drawing from as few as two decks of cards: this indicates that the assumption that the probability of drawing each card is always 1/13, as though they were drawn from an infinite number of decks, is justified.

9 Conclusion

This paper has presented a replicable, verifiable algorithm for calculating the optimal strategy for the game of casino blackjack. By adhering to this strategy a player can reduce the casino's "house edge" to a mere 0.498%.

The strategy could be further tested by using Monte Carlo methods to simulate thousands of hands played and confirm that the expected value of -0.498% cannot be improved on by making any changes in the strategy for each hand. Although the simulations described in Sec. 8 verified that the percentages shown in Table 6.3 and the expected value of the strategy are correct, they do not provide additional proof for the claim that this strategy maximizes the expected value as compared to other possible strategies.

Future work should also focus on relaxing the strongest assumption that this paper makes: the assumption that the probability of drawing each card in the deck is always 1/13. It should be noted that this assumption was made even in the simulations described in Sec. 8, as the probability of drawing each card in the simulations was set to 1/13. As argued in Sec. 2.2, the effect of relaxing this assumption would likely be negligible as long as the number of decks is still large, such as six. However, changes could be made to re-optimize the strategy for single-deck blackjack, for which the differences could be substantial. It would also be valuable to estimate the variance of the player's winnings and determine what changes should be made for a player who values minimizing the variance of his losses in addition to maximizing his expected value. This might mean hitting or standing instead of doubling down or splitting in some cases in which doubling or splitting increases the player's expected value only slightly, in order to avoid the possibility of losing twice his bet.

References

[1] "Game Rules: Blackjack." *Hoyle*. United States Playing Card Company, n.d. Web. 21 May 2012.

[2] Kilby, Jim, et al. *Casino Operations Management*. New York: John Wiley & Sons, 1998.

[3] Thorp, Edward O. *Beat the Dealer: A Winning Strategy for the Game of Twenty-One*. New York: Random House, 1966.

[4] Wakin, Michael B., and Christopher J. Rozell. "A Markov Chain Analysis of Blackjack Strategy." 2004. Web.

[5] Werthamer, N. Richard. *The Science of Casino Blackjack*. New York: Springer, 2009.

[6] Wolfe, David. "Distinguishing Gamblers from Investors at the Blackjack Table." *Lecture Notes in Computer Science* 2883 (2003): 1-10.

[7] Yu, Kam, and Huan Wang. "Economic Analysis of Blackjack: An Application of Prospect Theory." Canadian Economic Association 43rd Annual Conference (2009).