

Artificial Intelligence Beating Human Opponents in Poker

Stephen Bozak

University of Rochester

Independent Research Project

May 08, 2006

Abstract

In the popular Poker game, Texas Hold'Em, there are never more than three moves an agent can make. However, these three moves are based on hundreds of different decisions opponents make and thousands of different possible outcomes. Using basic game-theory of betting with a probabilistically better hand, a human player or even a computer program can get take advantage of the mistakes an opponent makes in poker.

Motivation

Online poker sites contain thousands of players at any given moment with thousands of dollars being shifted - usually - from the weaker players towards the stronger players. It is also generally true that at the lower stakes, people tend to play more carelessly and in the poker standards “worse” than players at higher stakes. Many books were written about the game of low limit Hold’Em and how it could be beaten by following specific rules that marginally varied from one author to another. The book “Winning Low Limit Hold’em” by Lee Jones documents specific actions to take in almost every situation designed to profit off of other players’ mistakes. These set of rules are extremely precise and they could easily be coded into a poker playing A.I for a systematic method of playing poker.

Methods

Initially, my goal in this project was to create a computer program capable of playing online Poker through an online Poker program such as PartyPoker® or PokerStar® autonomous of human control and capable of making a profit over a long enough period of time. However this goal was unrealistic because I was unable to find an online Poker site that allowed the use of computer programs or “bots” to play the game in place of a human and also because in order to run successful experiments I would have to have the program playing online for hundreds of hours before I would know if it was successful or not. Instead, I decided to simulate a real life low-limit Hold ‘Em table by having a variety of AI agents.

The process of creating the best AI for low-limit Hold ‘Em games started out by creating a program that could play many hands of Texas Hold ‘Em in order to run experiments. A program I had previously made was used as the foundation and adapted in order to meet the goals for this program. The new program is capable of running eight AI programs that each chooses an option based on its current state. For the experiments, I decided to code five separate AI programs to represent five different styles of playing Poker.

-The first AI, which I will call “tight passive,” contains all the starting hand requirements listed in Lee Jone’s book and will bet or raise according to how strong the starting hand is on a scale from one to five with the following rules:

- 1 Fold it
- 2 Call late position only
- 3 Call middle and late position
- 4 Call early, raise middle and late position
- 5 Raise and reraise

where position is relative position to the start of the action. This AI will then only bet after the flop if it has a strong hand and will not hesitate to fold if it did not make a pair after the flop. The AI also considers the possibility of making a straight or a flush and will calculate the “pot odds” to decide if it should stay.

-The second AI, which I will call “loose passive,” is simply an AI that will call every bet on every round regardless of what the hand it is holding or what the bet is.

-The third AI, which I will call “loose aggressive,” is simply an AI that will bet and raise every single round regardless of the situation, up to four bets, which is the maximum amount of bets per round.

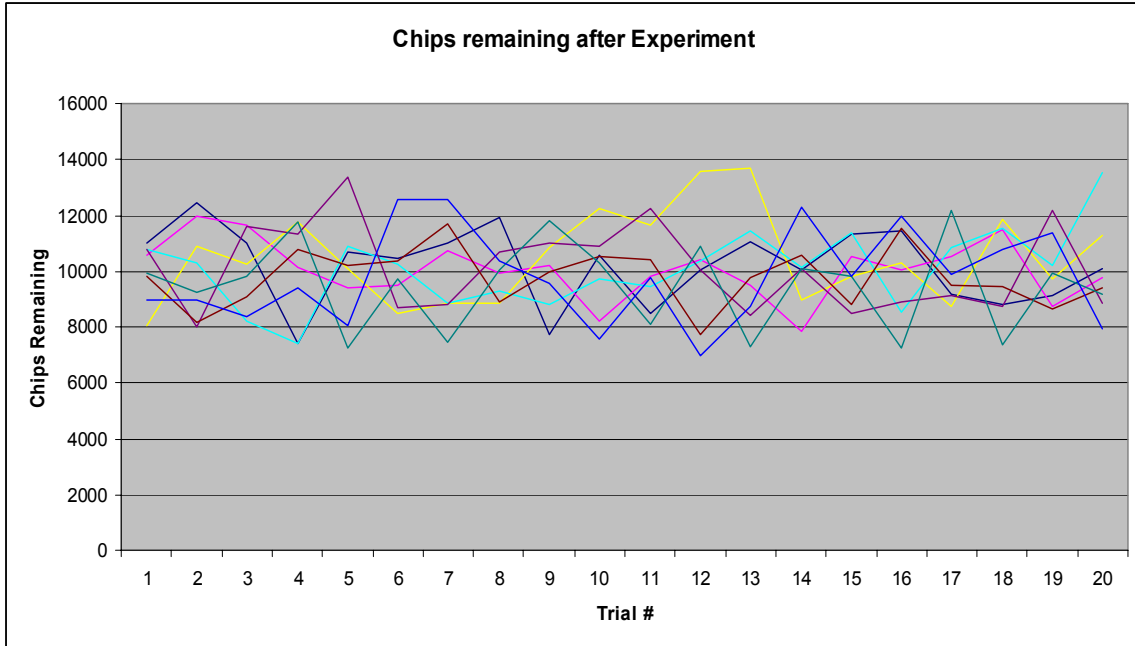
-The fourth AI, which I will call “random” will choose a random move, (check, bet) or (call, raise, fold), at any turn taking no information into consideration.

-The last AI I will call “tight, aggressive” which is very similar to the first AI except it is less hesitant to fold and more willing to bluff, which means bet when you don’t necessarily have a hand. This last AI will have certain values such as “aggressiveness” and “randomfactor” that will be adjusted to find the best possible agent.

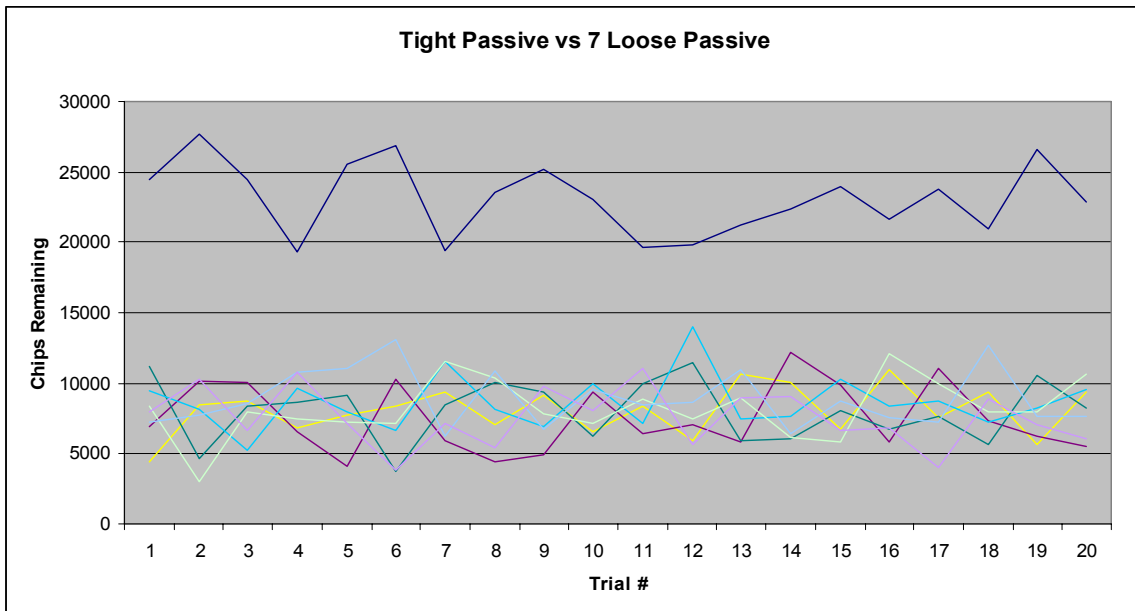
For consistency in all my experiments, I decided on using 1000 sequential hands using the same deck (shuffled) over an iteration of 20 independent games. I used these numbers because they are large enough to eliminate some of the “randomness” in Poker and yet small enough to run the experiments in a manageable amount of time.

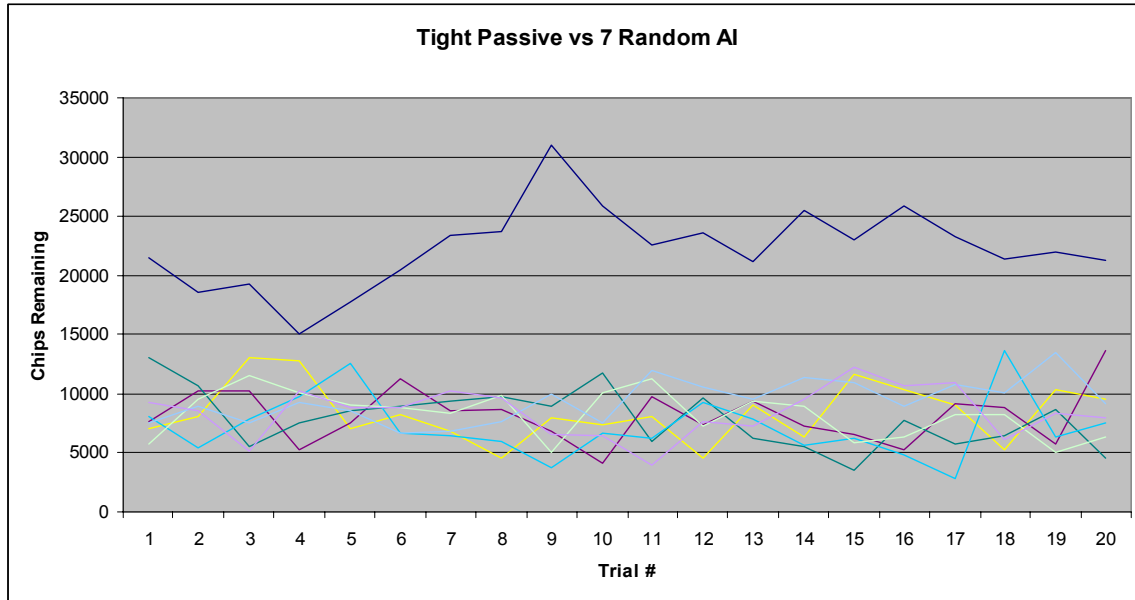
Results

I predict that some AI programs will do better versus other AI programs and perform better in different tables. As a basis to compare the results against, I used a table with eight “tight passive” bots. This resulted in a standard deviation of 1409 which is equal to 35 big bets which is very marginal.

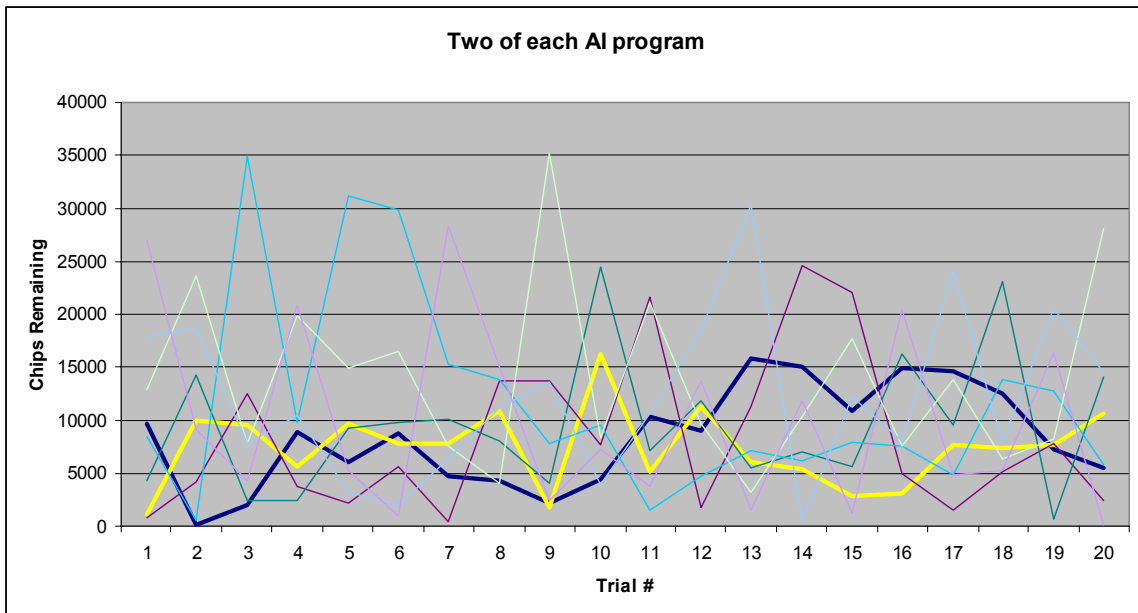
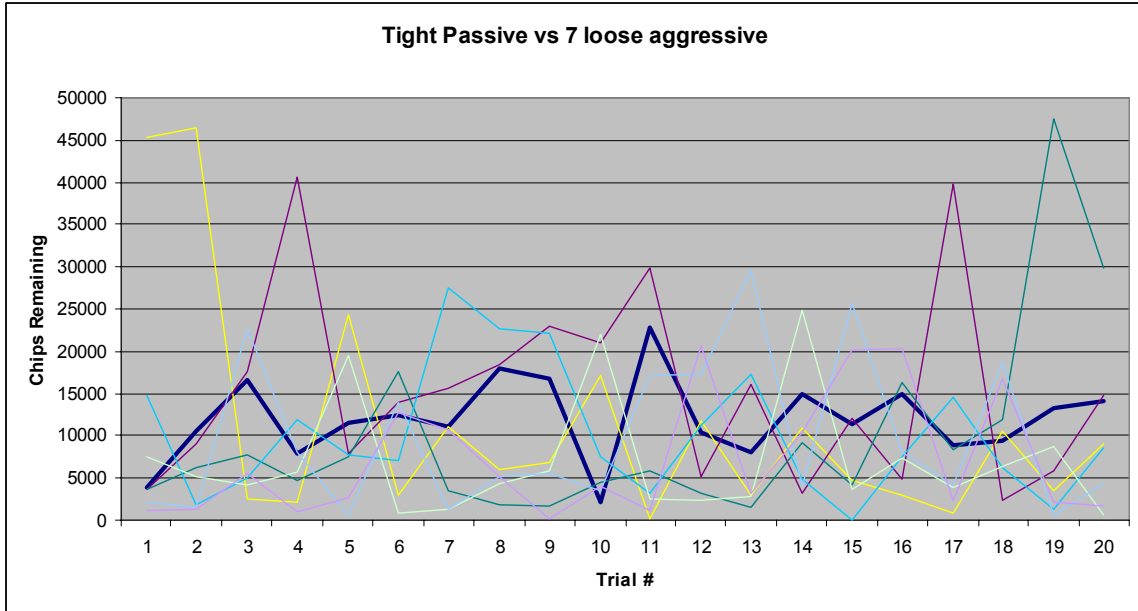


After programming in the basic rules of starting hand requirements, I decided to see how well it would work versus a “dumb” poker bots. The dark blue line in each graph represents the tight-passive AI with the coded starting hand rules.





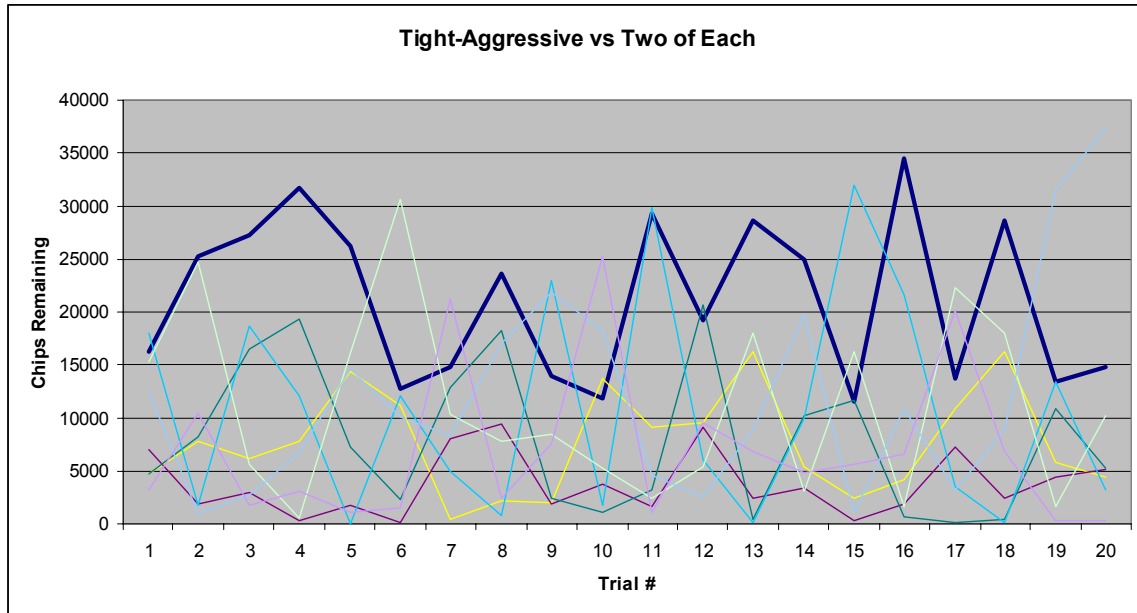
The program does very well in the previous two experiments which is proof that a program can beat opponents if it could find such a table. The “7 loose passive” table is described in Lee Jones’ book as the ideal table to play at following these rules. However, such a table is very unlikely, at least a table with the extreme amount of loose-passive players. The following experiment shows where this AI fails to perform well.



Bold lines are the Tight-Passive program

The rule set is good versus opponents with completely random actions or opponents who only call. At a table with opponents who only bet and raise, this AI shows no clear advantage. Also, at a table with two of each of the aforementioned programs, there is no

clear advantage of which AI is superior. Therefore, it is necessary to develop an improved AI that we will call “tight-aggressive” in the next experiments.



In this experiment, the tight-aggressive AI goes up against the tight-passive AI (yellow) and 2 of each of the other AI programs. The thick blue line represents the new and improved algorithm which has a random factor that causes it to sometimes call a bet. This new AI has very similar performance on all the one tight-passive did well on and also performs well on the ones tight-passive did poorly on. Clearly, this AI could be capable of making money at a real table.

Discussion

The loose-aggressive AI program really sets the pace for the entire table. In a real life situation, a player who plays this way will most certainly bust out of the game fairly

quickly and the only reason the AI did not bust out was because of the large amount of starting chips. Therefore, it is better, for this set of experiments, to not include this anomaly if we wish to determine how the agent would perform at a real life table. However, because it is difficult to simulate the randomness and variety of players that a program could encounter at a real table, I chose to leave it in. However, it is important to keep in mind how drastically these agents change the way the game is played. I do believe however, that at a real table, this AI program would perform, at worst, average at a low-limit Hold 'Em table. There are many more aspects that could be coded into the AI program that I did not include, such as remembering how opponents play and adapt based on that. Perhaps someday in the future, AI programs will play with the best of human players on online poker sites without players even realizing that the skilled player behind the screen is really a sequence of code written by a programmer.

References

PartyPoker <http://www.partypoker.com/>

“Winning Low Limit Hold’em” Lee Jones © 2000

“A Poker Player” article written by Chris Brown 2006