Cooperative Agents in Euchre

Taras Mychaskiw

April 16th, 2015

Abstract

Typically, agents use a set of rules or follow a specific strategy when given options in order to maximize their expected utility. This project explores such a setting in the card game euchre. Euchre gives an interesting setting where two teams of two agents compete in an incomplete information game. A suite of euchre playing agents are given which each follow a specific and quick strategy in order to maximize not only their personal utility, but their team utility.

Contents

1 Introduction

A good heuristic or static evaluation function in games helps an agent make decisions which maximize their utility. This idea works well, however in team settings, maximizing a personal score may not be a good team strategy. In this project, we investigate this setting in the card game euchre, where two teams of two complete to maximize their team score. Some agents show, among those studied, that cooperation is a much more powerful strategy. This idea was defined in the prisoner's dilemma, where two rational agents that maximize their personal score end up both losing. If these agents cooperated instead, their personal scores and "team" scores would be optimal.

Card games remain popular to study, as their offer interesting settings which lead to good understandings and findings. For example, a common thread in many card games is that they are incomplete information games – other players' hands or the deck is not known to the agent. Perhaps the most interesting about card games is that humans are (generally) much better than the best known artificial agent. Famously, poker was solved in [1], however they study a very small version of poker compared to what is normal in tournament play. Not to discredit their results, but just to say that much work needs still be done in order for artificial agents to consistently beat the best human players in many card game settings. Euchre, compared to other card games, is less conventional in that it is also a cooperative game. This project details various artificial agents for euchre in order to discover a strategy that is superior.

1.1 Euchre

Simply put, euchre is a trick taking card game. It plays very similarly to other trick taking games, such as hearts or bridge. The euchre deck only consists of 24 cards, 9 through ace of each suit. Each of the four players is dealt five cards which makes up their hand. Each players' hand is only known to them. The 21st card of the deck is turned face up to be offered as trump. The remaining three cards remain hidden and are not used in the hand (known as the "kitty").

Players then take turns deciding on if they wish to "order up" the offered trump card. If so, the dealer must pick up the offered card, place it in his hand and secretly discard a card. If the card is not ordered up, players take turns having the option to call a trump suit or pass – but players cannot call the same trump suit as what was offered prior. Players can also choose to "go alone" when they call a trump suit, which forbids their partner from playing the hand in the hopes of scoring more points.

Once trump is called, the main portion of the game takes place. The player left of the dealer leads the first trick. Each player adds a card to the trick in turn. The winner of the trick leads the next trick. This play continues for five rounds, until all players are out of cards. Afterwards, teams count how many trick their won, and the team that took at least 3 of the tricks scores points as:

1 point : if called trump and took 3-4 tricks

2 points:	if called trump and took all 5 tricks
2 points:	if didn't call trump and took 3-5 tricks
4 points:	if went alone and took all 5 tricks

This play continues, shifting who deals every hand, until a team scores 10 points. That is euchre in a nutshell – though trump hasn't been explained. The general play follows the above guidelines. There are additional rules with trump and which cards you can play into a trick, which are explained below.

When a player leads a trick, the card they play determines the lead suit of the trick. Other players must play a card of the same suit if possible, otherwise they can play any card from their hand. The player with the highest ranked card in the trick wins it, however those cards which are not the lead suit or trump cannot ever win the trick. For example, if I lead the 9♡ and you followed with A♢, my lowly 9 is still winning the trick because you did not follow suit (assuming ♢ is not trump).

Trump in euchre is peculiar, and leads to much confusion for beginners. For non-trump suits, the rankings of cards is as you would expect, ace > king > queen . . . . The lowest trump cards beat the highest non-trump cards (easy enough), however the rankings in among trump is not the same as non-trump. Let ♠ be trump for a concrete example. The ordering of trump then goes J♠ > J♣ > A♠ > K♠ > Q♠ > 10♠ > 9♠. The jack is the highest card in the game, followed by the other jack of the same colour – in our example J♠ is the best and J♣ is the second best (called the "right" and "left" bowers respectfully). The left bower is the confusing part. Though it is printed J♣, it becomes a ♠ as long as ♠ is trump. So when ♣ is led and your only ♣ is J♣, you are not forced to play it as it is not a ♣, but a ♠. Likewise, when ♠ is led, you must play J♣ since it is a ♠. This means there are 7 ♠ in the game, 6 of each ♢ and ♡ and only 5 ♣. Similarly when a red suit is called trump, the other red jack becomes the left bower and it becomes on the trump suit rather than it's printed suit.

1.2 Agent Considerations

With the rules of euchre in mind, we can consider what information is known and important to an artificial agent. When it comes times to play a card into a trick, the following information is known:

the rules of euchre
their own hand
the card offered as trump
all of the cards played so far, and by whom
the trump suit
the lead suit of the trick, is any
the overall rankings of each card

This is all the information an agent has access to. Additional information can be inferred, such as when a player doesn't follow the lead suit, it is known that the player cannot possibly have any cards of that suit in their hand. Making the best use of all the information available will lead to a powerful euchre agent. In the next section, we introduce several euchre agents which make different uses of all this available information. Afterwards, we compare these strategies to each other to determine which is best among them. Lastly, we will discuss the strengths and weaknesses of particular strategies and conclude with ideas for future work.

2 Euchre Agents

The overall objective in euchre is to take as many tricks as possible. This ensures the other team does not benefit from them. Generally, this strategy works well, but it ignores the team aspect of the game. In this section, we give several euchre strategies making various use of the information available in the game. All agents know the basic rules of the game, as a useful agent could not be constructed without this knowledge. Additionally, no agent can cheat. When we say an agent chooses to play a card with some property, it is assumed that the card is chosen from the set of legally playable cards. These strategies range from simple to selfish to complex. These strategies aimed to produce a very quick but good decision when playing a card from their hand into a trick.

2.1 Simple Strategies

The first strategies discussed will be very simple, but serve as a very good base for more complex strategies. These agents only make use of the information given directly to them, their hand. The first agent plays a random card from their hand. This agent exists to serve as a basis for comparison. The next two agents are opposites of each other, one always plays the lowest card in their hand while the other always plays the highest card. Since these agents do not use any additional information, they all play selfishly.

The agents High and Low serve as an excellent foundation for more complex agents. In euchre, the overall goal is to maximize the number of tricks your team earns. Playing High will give you the best possible chance of winning a trick, while Low allows you to throw away your worst card if you resign a trick. Playing a "Middle" strategy would not help compared to High and Low as it seems to give you the worst traits of both the other strategies. With Middle, you could have an increased chance of taking a trick, but if you lost the trick, you've lost a decent card which could have been used later to win a different trick. Do to this, no Middle agent was explored.

2.2 More Complex Strategies

The next agents become slightly more complex. These agents make use of the information given to them in the trick. The first one, called HighLow, behaves as follows:

```
if I have a card that can win the trick, play High
otherwise, there's no hope, play Low
```

This HighLow strategy proves to perform quite well, and makes a lot of sense. If it is possible to take a trick, playing the best card gives the highest chance of actually winning the trick. If there's no hope of winning a trick, playing your worst card saves your better cards in the hopes that they can win later tricks.

A cooperative version of HighLow was created as well, called CoopHighLow. The logic of this agent is as follows:

```
if partner has played in the trick and is winning, play Low
otherwise, play HighLow
```

This strategy aims to not take tricks from your partner if it can be avoided. If your partner is winning the trick, saving your good cards for later and allowing them to win will generally be a very powerful strategy. However, if your partner is losing the trick, or hasn't played, the semi-aggressive HighLow strategy makes sense to play to try to win tricks or avoid losing better cards.

2.3 Markov Decision Strategy

The next agent uses a simplified Markov decision process to determine it's hand strength compared to the strength of remaining cards. The agent uses the ratio of these values to guess at whether a card in his hand is strong and has a good chance of winning the trick. At a high level, the Markov agent keeps track of all cards that have been seen so it knows what cards have not been seen. A value is assigned to each card in the deck which represents the strength of that card. When it comes time to make a decision, the Markov agent performs:

$$\text{action}(H, D, \tau) = \begin{cases} \text{play High} & : \frac{Value(H)}{Value(U \setminus D)} \geq \tau \\ \text{play Low} & : \frac{Value(H)}{Value(U \setminus D)} < \tau \end{cases}$$

where H is the set of cards in the agent's hand, D is the set of cards that have been seen and U is the set of all cards. The function $Value(\cdot)$ gives the sum of the values of all the cards in the given set. The threshold parameter τ determines the operating point of the agent, as τ grows, the Markov agent becomes more and more aggressive. Essentially, when the Markov agent sees a high ratio value, it seems it's hand as good, so a High card is played. Otherwise, the weak is viewed as weak and a Low card is played.

The specific values for τ and the cards used in our experiments are discussed in the next section.

2.4 Card Counting Strategy

The CardCounting agent tries to use all available information to it's advantage. This agent keeps track of whether or not each player can possibly have a card or not. When a card is seen, it is known that no player can possibly hold that card in their hand, so this information is tracked. For additional information, whenever a player does not follow the lead suit, it is known that the player cannot have any of the lead suit in their hand. This information is also recorded.

In addition to using all available information, this agent employs a different strategy depending on how many cards are in the trick. It tries to use players being out of a suit (or strong in a suit) to the team advantage. At a high level, the CardCounting agent performs the following thought process before making a decision:

Leading the trick

- if partner is out of a suit and has trump, play Low in that suit hoping the partner can play trump
- if partner is “strong” in a suit compared to both other players, play Low in that suit
- otherwise, play High

Second to play

- if partner can possibly win and is “stronger” than the other remaining player, play Low
- if partner doesn’t have the lead suit:

  if trump was led, play HighLow

  otherwise, if partner has trump, play Low
- otherwise, play HighLow

Third to play

- if partner is winning so far but the last player can win the trick, play HighLow
- otherwise, play CoopHighLow

Last to play

- play CoopHighLow

This method introduces complexity in the hopes of intelligent performance. This agent aims to maximize the number of tricks the team wins through trying to let the partner take as many tricks as possible; except when the partner cannot help, where this agent tries to play selfishly for the benefit of the team.

The word “strong” comes up in the above explanation. The strength of a player in a suit is defined as the sum of the ranks of each card that player can possible have of that suit, where a high ranking means a stronger card. Then, $S_a > S_b$ implies that player a has a stronger hand than player b in that suit.

2.5 Hybrid Strategy

The Hybrid strategy combines the logics of the CoopHighLow, Markov and CardCounting agents. Essentially, this agent using the same Markov decision process as the Markov agent, except when the agent's hand is good, the CoopHighLow strategy is played. Other times, the CardCounting strategy is played. This strategy was created to hopefully compensate for the passive play of the CardCounting agent with the aggressive behaviour of the CoopHighLow agent.

2.6 Monte Carlo Agent

An overly simple strategy in euchre (or any card game) would be to simulate all possible games which follow from the current game state. In incomplete information settings such as euchre, this is simply not computationally feasible. Instead of traversing the entire game tree, the MonteCarlo agent randomly traverses the game tree for a set amount of time, keeping track of how many tricks were won in each path. This is similar to the strategy laid out in [2].

To help reduce the search space further, the MonteCarlo agent retains the idea from the CardCounting agent to keep track of which cards every other player can possibly have. When it comes time to make a decision, a random possible hand is assigned to each other player and games are simulated until finished. This process is repeated until some amount of time has passed, then the card that led to the most number of trick wins is played. At the start of a game, this agent will likely perform poorly as the game tree is still enormous, while nearing the end of the game, the game tree shrinks to a point where it is possible to simulate all possible games which will greatly strengthen play.

3 Agent Comparisons

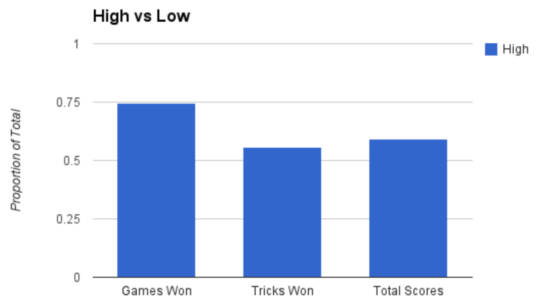
For each table in this section, we compare two of the strategies discussed in the previous section. At first, we look at teams where the two agents follow the same strategy. Mixed agent teams were also studied, however these results are lacklustre. Unless otherwise noted, all the statistics were taken from a simulation of 10,001 games. In these games, it was forced that the dealer call trump to be ♠ for simplicity, as the agents do not have a strategy for calling trump. This setup might be considered unfair to the dealing team, but all players share the disadvantage and aim to make the best out of the situation they find themselves in. Given enough games, the disadvantages will even out. Each game was played until a team had 10 points (teams can get 11 points if they score 2 points when they are at 9 points).

Each agent was also timed when making it's decisions. However, each of these made over 500,000 decisions and the slowest agent took about 3 seconds to make all these decisions. Due to this, the timings won't be included as they are indistinguishable during play against a human. The exception, of course, is the MonteCarlo agent, who was given a certain amount of time to make each decision. These decisions were slower, and the times will be discussed later.

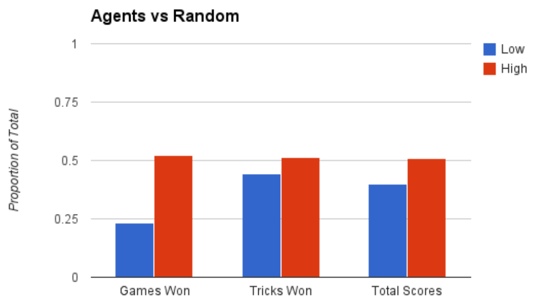
A small note about the following charts. Each show the performance of other agents versus the target agent. A value of 0.25 in Games Won for example means the other agent won 25% of the games against the target agent, so the target agent performed much better than the other one.

3.1 Simple Strategies

The simple strategies were compared to each other. Figures 1a and 1b show the results of their competition. As to be expected, Low did poorly overall. Low managed to still win some games however, since Low essentially keeps it's highest cards for the end of the game. When they get a lucky deal, they can manage to scrape together enough tricks to win a hand. Generally though, playing Low is worse than playing Random. On the other hand, playing High proved to be slightly better than playing Random, though not by very much. Also not surprising is that High beats Low, likely due to winning early tricks. Each of the simple agents perform very quickly, each performing their decisions in very little time.



(a) Results of the Low agent against the High agent.



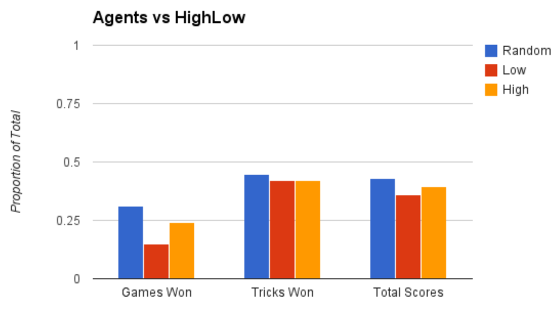
(b) Results of the Random agent against other agents.

Figure 1: Results for the simple agents against each other.

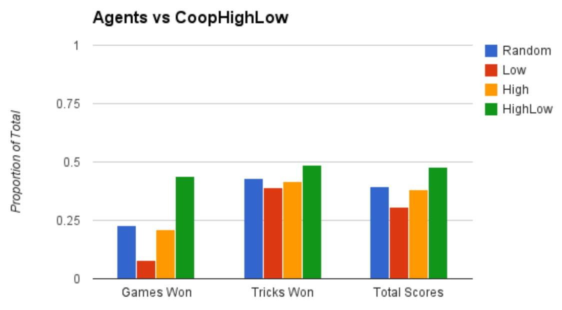
3.2 More Complex Strategies

Figure 2a show the results for the HighLow agent against the simple strategies. The results show a staggering favour for the HighLow agent, able to win handily against any of the simple strategies. This is not surprising, as the HighLow strategy is generally decent enough play in human play. It tries to maximize the number of tricks each card can win, which leads it to be a much stronger strategy than the simple three.

Figure 2b show the results for the CoopHighLow agent against the other strategies. Following the trend of the HighLow agent, the CoopHighLow proves to be very powerful compared to the simple strategies. This also shows the results of an important competition. The CoopHighLow and HighLow agents both prove strong agents, but only one of the tries to cooperate with their team mate. The results favour the CoopHighLow agent, who took on average about a point more per game.



(a) Results of the HighLow agent against other agents.



(b) Results of the CoopHighLow agent against other agents.

Figure 2: Results for the two HighLow family strategies.

3.3 Markov Decision Strategy

Here, we show the results of two different Markov agents. They follow the same strategy, however the values they assign to each card is different. The first agent uses the number of tricks a card wins out of all possible tricks given the current trump suit as the value for each card. This proved to perform strangely, so an additional value base was formed. Markov2 uses a very similar value base; the value of a card is the number of tricks the card can win which derive from the current trick and given the trump suit. This different value base lends itself more closely to the actual values of the cards. For example, an ace that isn't the lead suit or trump cannot win any tricks, so it should be considered a powerful card. When leading a trick, Markov2 behaves identically to Markov. Both Markov agents used a threshold $\tau = 0.25$ due to there being 4 players in the game.

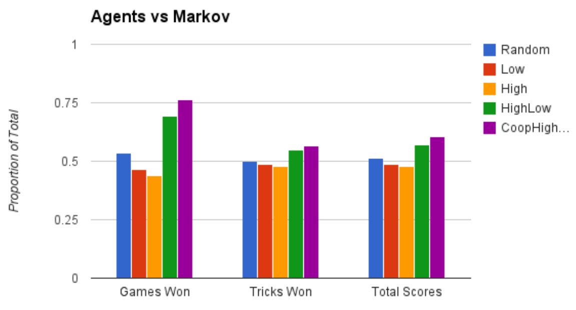
Something worth mentioning, these Markov strategies precomputed all the values of the cards and stored them for quick lookup when needed during the games. They perform much faster this way, but used quite a bit of memory (around 1GB). However, the time taken to precompute the values was only on the order of a second or two, so in real games they can be calculated on the fly quickly enough, human players would not notice a considerable delay.

As mentioned, Markov performs strangely. Markov proves worse than Random, but slightly better than both Low and High. This is strange due to the results shown in Figure 1b, showing that High is better than Random. When played against HighLow and

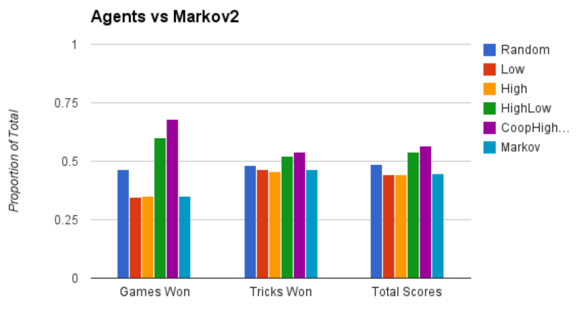
CoopHighLow, Markov did not do well at all, losing most of the games and averaging relatively low scores.

The Markov2 agent shows more promise than the Markov agent, showing it is actually slightly better than Random and still better than both High and Low. Markov2 also is quite a bit better than the Markov agent.

This high that Markov2 experienced is quickly brought down by the soul crushing HighLow and CoopHighLow agents. Although Markov2 performed much better than Markov against these two agents, Markov2 was still trounced. Rather unfortunate. These results solidify the Markov agents on par with the simple strategies.



(a) Results of the Markov agent against other agents.



(b) Results of the Markov2 agent against other agents.

Figure 3: Results for the two Markov family strategies.

3.4 Card Counting Strategy

The CardCounting was modelling off of how I personally play euchre, for the most part. Human players tend to keep track of what suit players hold and play accordingly. The CardCounting, similar to the HighLow agent, performs very well against the simple strategies. The results are promising, as the agent achieves very high average scores in each game.

Surprisingly, the more complex strategy modelled after more human play performs not as well as expected against the relatively simple CoopHighLow agent. CardCounting was able to beat the HighLow agent. However, against the CoopHighLow agent, the CardCounting essentially tied, performing slightly worse but taking more tricks. It is due to this disappointing result that the Hybrid strategy was created, in hopes to beat the CoopHighLow agent.

As shown in the previous subsection, the Markov agents are not very powerful. They are swept aside by the CardCounting agent.

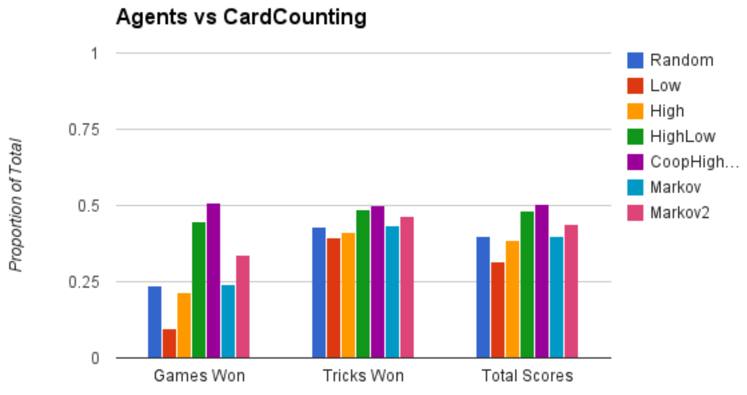


Figure 4: Results of the CardCounting agent against other agents.

3.5 Hybrid Strategy

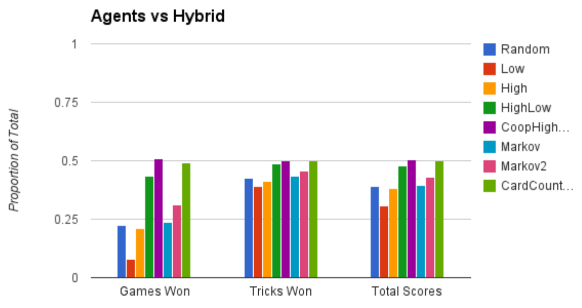
The Hybrid strategy was born out of the disappointing results of the CardCounting agent. A weakness of the CardCounting agent is that it tends to play relatively passively, leading with low cards or generally playing Low too often. The Hybrid strategy used the Markov decision process in order to determine which of the CardCounting and CoopHighLow strategies to play. When the hand was strong, a CoopHighLow strategy was played, and CardCounting otherwise. The Hybrid agent uses the same card values as the Markov agent. After seeing the improvement of Markov2 over Markov, a Hybrid2 agent was created as well in hopes of seeing the same type of improvement. Both Hybrid agents used a threshold $\tau = 0.25$ again, same as with the Markov agents.

Figure 5a shows the results of the Hybrid agent against the other agents. Again, the wrath is the CoopHighLow agent is seen, the Hybrid agent essentially equivalent to the CoopHighLow agent. Hybrid, however, does prove a slight improvement over the CardCounting agent.

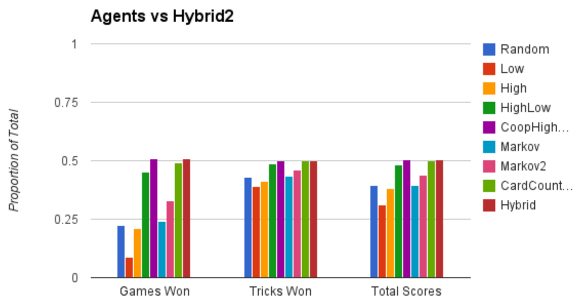
Figure 5a shows the results of the Hybrid2 agent against the other agents. After seeing the improvement of the Markov2 agent over the Markov agent, it was hoped that Hybrid2 would see a similar improvement. Strangely, Hybrid2 tended to perform slightly worse than Hybrid overall, losing to Hybrid in addition to losing to the CoopHighLow agent.

3.6 Monte Carlo Agent

The MonteCarlo agent seems to be the only agent capable of toppling the CoopHighLow agent. Two different time limits were given to the MonteCarlo agent, MonteCarlo10 was given 10 seconds for each decision and MonteCarlo60 was given 60 seconds. Due to the amount of time the MonteCarlo agents took, fewer games were played and only against the strongest other agent, CoopHighLow. MonteCarlo10 played 31 games against CoopHigh-



(a) Results of the Hybrid agent against other agents.



(b) Results of the Hybrid2 agent against other agents.

Figure 5: Results for the two Hybrid family strategies.

Low, while MonteCarlo60 only played 5. More games would be required for a stronger conclusion, however due to time constraints, this was not possible.

Figure 6 shows the results of the two MonteCarlo agents against CoopHighLow. MonteCarlo10 notably does poorly. This is likely due to not being able to search enough of the game tree before being forced to make a decision. At the beginning of the game, only a few hundred runs were able to complete. Not enough information could be gathered in order to make a good decision, as there are $\binom{18}{5}\binom{13}{5}\binom{8}{5} \approx 600$million possible branches to traverse at the start of the game – not including playing out each branch.

Given much more time, the MonteCarlo60 agent was able to beat the CoopHighLow agent. However, given a minute per decision makes for a very slow game of euchre, so this is not a great outcome. Additionally, the results are still not wholly conclusive as not enough trials could be performed.

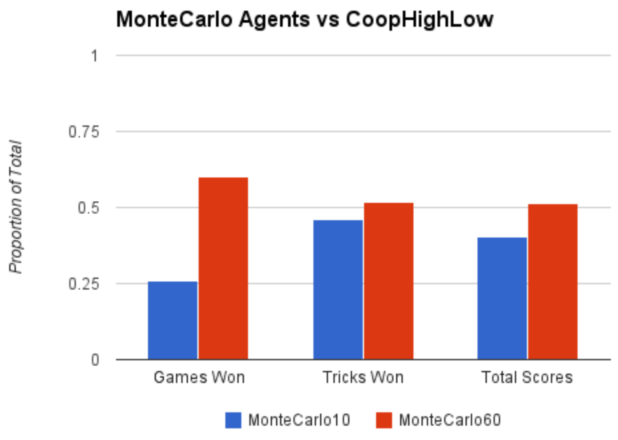


Figure 6: Results of the two MonteCarlo agents against CoopHighLow.

3.7 Mixed Agent Teams

In addition to the above tests, mixed agent teams were also studied. These tests were ideally performed to view which agents were powerful cooperators. Unfortunately, all the results were, simply put, rather boring. No mixed agent team was able to beat a team of CoopHighLow agents consistently. Other games ended up as expected. So, to save the reader from many more charts, these results are omitted.

4 Discussion

Euchre is a game where the goal is to maximize the number of tricks your team takes. With this in mind, it is not totally surprising that such a simple strategy such as CoopHighLow performs so well. This strategy aims, in a very simple way, to maximize the number of tricks the team takes. When it is possible to win a trick, try as much as possible to win, otherwise playing the lowest card saves your better cards in hopes they can win later tricks. The CardCounting and Hybrid strategies also perform very well in this regard, but do not prove to be any better than the simplistic CoopHighLow. This section discusses strengths and weaknesses of particular agents, and what could be done better in the future.

4.1 Strengths of Particular Agents

First of all, praise needs to be given to an underrated agent: Low. Low by itself performed worse than every other agent, however this agent is a key ingredient when creating powerful agents. The Low strategy can be summed up as "I'll save this for later!" This isn't necessarily a strength of the Low agent, but it's value should not be taken for granted.

It's the passive value of Low combined with the aggressive value of High, along with some team play, that makes CoopHighLow such a good strategy. As mentioned, CoopHighLow strives to win tricks were it can as long as it's not hindering the team. This idea can prove a great foundation for any euchre playing agent.

The CardCounting agent shows more promise compared to other agents – it is a strategy that actually uses all available information to it's advantage. The particular rules followed could be improved upon to create a more useful and intelligent agent.

MonteCarlo obviously has the most potential for perfect play. In fact, nearing the end of the game, MonteCarlo10 should be close to perfect play, if not perfect already. This is because as the game is played, more and more information becomes known. As mentioned, there are about 600million branches in the game tree at the beginning of the game. When leading a trick with only three cards left, the tree size is reduced drastically to $\binom{12}{3}\binom{9}{3}\binom{6}{3} \approx 370,000$. This tree size becomes easily traversable, all games can be played out before making a decision.

4.2 Weaknesses of Particular Agents

For the praise of Low, there's also an inherent flaw. When playing Low, you save good cards for later but you are unlikely to lead a trick. If you do not lead a trick, high non-trump suit cards can potentially be worthless. This is why Low by itself performs so poorly. Pairing Low with High removes the flaws of Low.

CoopHighLow, even being the best agent overall, plays very poorly sometimes. Particularly, it is very when leading a trick. Blindly following the rules, CoopHighLow will lead with it's highest card. This means leading trump frequently, which most human players will tell you is a bad play. For example, CoopHighLow will happily lead it's 9♠ as it's highest card, but this is the lowest trump – this type of play is essentially asking other plays to freely take his trump away.

It's for this reason CardCounting as a high potential. It can avoid those type of mistakes using it's more complex rule set. The proposed rule set however, seems to not be perfect. Additional rules may need to be added, or some rules may require changes. Perhaps the largest flaw with the CardCounting agent is that it fails to ever check if opponents are missing a suit. For example, when leading CardCounting checks if their partner is out of a suit and tries to lead that suit hoping the partner can trump it – but perhaps this rule should not be followed if the opponents are also out of the suit and are stronger in trump than my partner. A more complex rule set may lead to a more well rounded agent.

The Markov agents, and by extension the Hybrid agents suffer from one major flaw: they are parametrized. The values of each card and for τ need to be assigned. In our experiments, somewhat natural values were assigned, however a natural value may not lead to the best performing agent.

The MonteCarlo agents, as discussed previously, cannot search enough of the game tree in the early game. This may lead to poor decisions early which snowball into an unwinnable late game.

5 Conclusions and Future Work

This project aimed to create an intelligent euchre playing agent, but found that relatively simple strategies performed just as well if not better than more complex ones. This leads to the conclusion that euchre is mostly not about trick taking, but about trump calling. Unfortunately, none of the strategies given are able to call a meaningful trump suit. This is not to say that CoopHighLow, the best performing agent overall, is an optimal trick taking euchre strategy. On the contrary, we have given some weaknesses that the CoopHighLow strategy has and ways to improve upon them.

Some future work would be to create a more complete rule set for the CardCoutning agent to improve it. This would in turn improve the Hybrid agent. Further improvements can be made to Hybrid, the parameters can be learned instead of arbitrarily assigned, perhaps leading to a much better agent overall. Perhaps the learned values can be used to make an intelligent decision when calling trump. Additionally, a more complex Markov decision process can be used instead of a simple threshold.

In this project, several trick taking euchre strategies are given. A simple rule base consisting of Random, High and Low build up into more complex and more powerful agents. Counter intuitively, a relatively simple strategy proves to be the best trick taker out of those studied. Though it is not a perfect strategy, it can serve as an excellent foundation for future trick taking strategies. This result leads to the conclusion that trick taking is euchre is not as difficult as the decision to call trump, when very little about the state of the game is known.

All of the source code for this project is freely available at https://github.com/twentylemon/euchre

References

[1] Michael Bowling, Neil Burch, Michael Johanson, and Oskari Tammelin. Heads-up limit holdem poker is solved. *Science*, 347(6218):145–149, 2015.

[2] Furtak. *Symmetries and Search in Trick-Taking Card Games*. PhD thesis, University of Alberta, 2013.