

Strategien bei der Entwicklung von Poker-Agenten

Andreas Eismann

ANDREAS.EISMANN@GMX.NET

Abstract

Die Masterarbeit von Michael Johanson war Grundlage dieser Zusammenfassung verschiedener Strategien bei der Implementierung von Pokerprogrammen. Die fundamentalen statistischen und spieltheoretischen Hintergründe waren ebenso Bestandteil der Thesis, wie auch die tatsächliche, praktische Anwendung des zusammengetragenen Wissens bei der Erstellung eines Poker-Bots sowie der Teilnahme mit diesem an der AAI Computer Poker Challenge 2007.

1. Einleitung

Poker ist aus spieltheoretischer Sicht ein Spiel mit unvollständiger Information. Kennt ein Spieler selbst nur seinen eigenen Typ, während andere Spieler dessen Situation lediglich wahrscheinlichkeits-theoretisch schätzen können, so spricht man von unvollständiger, speziell asymmetrischer Information. In diesem Zusammenhang können Reputationseffekte auftreten, kurz zusammengefasst bedeutet dies, dass es sich für einen Spieler lohnen kann, einige Spielrunden mit Absicht zu seinem eigenen Nachteil zu spielen, um aus seinem daraus entstandenen Ruf bei seinen Mitspielern in den folgenden Spielrunden mehr als diesen entstandenen Nachteil zu erwirtschaften.

Pokerterminologie+ Spielregeln Poker besteht aus vier Spielrunden (Preflop, Flop, Turn, River); nach Austeilen der Karten erfolgt in jeder Spielrunde eine Setzrunde mit den unten genannten drei Aktionsmöglichkeiten für jeden Spieler. Preflop: Austeilen von zwei (privaten) Karten pro Spieler, danach Setzrunde mit den Aktionsmöglichkeiten Flop: Drei Karten werden offen in der Mitte ausgeteilt. Alle Spieler können Ihre privaten Karten mit den offenen kombinieren um eine der Kombinationen zu erhalten. Turn: Eine weitere Karte wird offen aufgedeckt. River: Die letzte der insgesamt nun fünf Karten wird offen aufgedeckt.

Nach der Spielrunde River kommt es zum sogenannten Showdown. Alle noch im Spiel befindlichen Spieler zeigen offen ihre zwei privaten Karten. Derjenige Spieler mit der höchsten Karten-Kombination gewinnt den gesamten Pot. Jeder Spieler hat die Auswahl aus drei Aktionsmöglichkeiten in jeder Spielrunde (Fold, Call, Raise) bedacht werden sollten etwas deutlicher.

Fold: Spielt ein Spieler fold, so verlässt er die aktuelle Spielrunde und verliert damit gleichzeitig die Chance den Pot dieser Runde zu gewinnen. Im Gegenzug verliert er kein (weiteres) Geld.

Call: Beim call erhöht der Spieler seinen eigenen Einsatz bis zu der Gesamtsumme, die derjenige Spieler mit dem aktuell höchsten Einsatz in den Pot eingezahlt hat. Hat in der aktuellen Spielrunde noch niemand einen Betrag gesetzt, so wird diese Aktion check genannt. **Bet/Raise:** Ein Spieler erhöht den aktuellen Einsatz über das Ausmaß des bis

dahin höchsten Einzeleinsatzes hinaus. Hat in der aktuellen Spielrunde schon ein weiterer Spieler den Einsatz erhöht, so wird diese Aktion raise genannt.

Varianten von Texas Holdem Limit/No-Limit In der Limit-Variante ist der Betrag der Wettgröße vorher fix festgelegt. Bei jeder bet/raise-Aktion kann jeder Spieler immer nur den gleichen, festgelegten Betrag setzen. In Turn und River wird dabei der Betrag verdoppelt. Der Betrag aus Preflop und Flop nennt sich small bet, derjenige in Turn und River big bet. Small Blind und Big Blind, den die beiden Spieler links vom Kartengeber zu setzen haben sind hierbei 0,5 small bet bzw. 1 small bet. In der No-Limit-Variante werden lediglich small blind und big blind festgelegt, anschließend ist jeder Spieler frei, Beträge zu setzen wie es für seine Strategie am günstigsten ist.

Heads-Up / Ring In der Heads-Up-Variante spielen lediglich zwei Spieler bzw. Poker-Agenten gegeneinander. Jedes Spiel mit mehr als zwei Mitspielern wird als Ring-Game bezeichnet.

Weitere Terminologie: Bluff Bei einem Bluff hat man selbst ein schwaches Blatt und versucht dem Gegner durch geschicktes Setzen zu suggerieren, man hätte ein starkes Blatt.

Semi-Bluff - Im Unterschied zum Bluff hat man beim Semi-Bluff mit dem eigenen Blatt noch eine Chance, ein gutes Blatt durch die öffentlichen Karten zu bekommen.

Trapping Hierbei versucht ein Spieler mit einem starken Blatt durch Auslassen einer Möglichkeit des bet/raise den Mitspielern zu suggerieren er habe ein schwaches Blatt, wodurch im Optimalfall für diesen Spieler die übrigen Gegenspieler motiviert werden trotz schwacher Blätter weiter zu setzen und somit mehr Geld in den Pot einbringen.

Value Bet Ein Spieler erhöht den Einsatz per bet und damit die potentielle Pot-Größe, die er mit seinem starken Blatt höchstwahrscheinlich gewinnt.

2. Hintergründe

Grundlegend handelt es sich bei der Entwicklung eines guten Poker-Agenten um die Lösung bzw. Annäherung der Lösung der Spieltheorie. Es wird dabei versucht, ein Nash-Gleichgewicht zu seinem Gegner zu finden, also eine Situation, in welcher keiner der beiden Spieler einen Vorteil daraus ziehen kann, wenn er von seiner Strategie abweicht.

2.1 Typen von Pokerstrategien

Bei einer gegenspielerabhängigen Spielstrategie ist eine Abwägung zu treffen zwischen dem Durchschauen des Gegners und Ausnutzen dieses Wissens und der eigenen Durchschaubarkeit. Gegen schwächere Gegner wird hierbei von einer anderen, mitunter offensiveren Spielstrategie ausgegangen, gegen einen stärkeren Gegner oder Gegner mit unbekannter Stärke wird eher auf Sicherheit gespielt, um nicht zu verlieren oder den Versuch zu unternehmen zunächst den Gegner zu durchschauen. Dabei gibt es zwei verschiedene, strategische Ansätze. Im ersten Ansatz werden verschiedene Strategien mit Wahrscheinlichkeiten gewichtet und dann zufällig (aber gewichtet) ausgewählt. Beim zweiten Ansatz wird mit einer (Nash-)Gleichgewichtsstrategie begonnen und fortgefahren, bis die Schwächen des Gegners entdeckt sind und darauf angemessen reagiert werden kann.

2.2 Pokerprogramme

Poker-Agenten spielen einige tausend Spiele pro Sekunde, so dass die statistische Auswertung mit hoher Genauigkeit erfolgen kann. In der vorliegenden Arbeit kommen zwei Techniken zur Anwendung, um die Varianz der auszuwertenden Strategie / Situation zu reduzieren und die Ergebnisse damit signifikanter zu machen: duplicate games und DIVAT Analysis. Die Performanz eines Poker-Agenten wird in small-bets/game (sb/g) gemessen. Diese Maßeinheit vergleicht den Einsatz eines Poker Agents mit dem Gewinn in der Serie von Spielen. Bei der Duplicate-games-Technik werden Spiele mit gleichen Kartenkonstellationen wiederholt um die Varianz zu reduzieren. Die DIVAT-Technik vergleicht bei jeder Kartenkonstellation die Stärken der Blätter der Spieler und entscheidet, wie eine normale (Basis-)Strategie das Spiel für jeden Spieler durchführen würde. Beide Techniken können kombiniert werden, um noch signifikantere Ergebnisse zu erzielen.

2.3 Erfassung von Spielsituationen

Man kann sich die Serien von Entscheidungen, die ein Pokerspieler während eines Spiels treffen muss als Baum vorstellen. Ein solcher Poker-Baum ist ein gerichteter Graph mit dem Startzustand des Spiels als Wurzel. Jeder Knoten im Baum stellt eine Entscheidung des Spielers dar. Es gibt dabei zwei verschiedene Strategieausprägungen, die sich auch im Detaillierungsgrad des Baumes niederschlagen. Bei der einfachen Strategie wird bei einer bestimmten Kartenkonstellation immer dieselbe Aktion vollzogen (fold, call oder raise). Beispielsweise wird bei 2 Assen im Preflop immer die Aktion raise gewählt. Im Rahmen der verhaltensabhängigen Strategie wird mit einer festgelegten Wahrscheinlichkeit eine der drei Aktionen bei gleicher Kartenkonstellation ausgewählt. Beispielsweise wird mit (70% raise; 20 % call; 10 % fold) bei 2 Assen im Preflop vollzogen. Zur Herleitung der perfekten Pokerstrategie für den Poker-Agenten ist es notwendig, alle möglichen Spielsituationen sowie alle möglichen Spielzüge eines Spielers zu erfassen. Beim Texas Holdem (Heads up / Limit) ergeben sich sehr große Poker-Bäume, die sehr unübersichtlich erscheinen und dementsprechend schwer auszuwerten sind, um den besten Poker-Agenten zu entwickeln. Die Anzahl der Spielzustände in einem Pokerbaum hat eine Dimension von ca. $3 \cdot 10^{17}$, die der möglichen Spielzüge ca. $3 \cdot 10^{14}$.

2.4 Abstraktionsebenen

Die Größe des Pokerbaums muss also verringert werden, so dass sich handhabbare Größen ergeben. Da sich durch Ausnutzen des Kartenisomorphismus der Baum nicht hinreichend verkleinern lässt, verwendet man bei der Erfassung von Pokerspielsituationen die Bucketing- (Eimer-) Technik. Diese überführt die Spielgröße effizient in besser auszuwertende Dimensionen. Es wird eine definierte Anzahl von Eimern bereitgehalten, in welche diejenigen Spielkartenkonstellationen eingefüllt werden, die ähnliche oder gleiche Eigenschaften besitzen. Alle Konstellationen in einem Eimer werden anschließend mit der gleichen Strategie behandelt. Die Anzahl der bereitgehaltenen Eimer ist variabel. Werden beispielsweise 5-10 Eimer für die Kartenkonstellationen in der Preflop-Phase verwendet, so entsteht eine handhabbare Größe von Spielsituationen und Spielstrategien. Die Grundform Bucketing-Technik wird in weiteren Varianten verfeinert. Von der Computer Poker Research Group (CPRG) wurden abstrakte Spiele entwickelt, die unter Einsatz von 5-10 Eimern Strategien für Poker-Agenten entwickelt haben. Beim Vergleich dieser Strategien sowie der daraus resultierenden

Poker-Agenten ergab sich ein abnehmender Grenznutzen hinsichtlich einer größer werdenden Anzahl an Eimern und der Güte der Poker-Agenten. Die Poker-Agenten basierend auf sehr vielen Eimern werden also ab einer gewissen Anzahl nicht mehr signifikant besser beim Hinzufügen von Eimern. Es ist demzufolge hinreichend eine kleine Anzahl an Eimern zu wählen, da hieraus schon ein starker Poker-Agent entwickelt werden kann. Die Anzahl der Eimer steht in diesem Zusammenhang für die Abstraktionsebene.

2.5 Gegenzugstrategie

Generell ist es sehr schwierig und in Rechenschritten gesehen teuer auf die Strategie eines Gegners den bestmöglichen Gegenzug bzw. eine Gegenzugstrategie zu entwickeln. Eine Annäherung dieses Ansatzes liefert die Gegenstrategie best response. In jeder Entscheidungssituation (der Spieler befindet sich gerade am Zug) wählt best response diejenige Aktion, die den Nutzen maximiert. Da dieser Algorithmus die Strategie bei jedem Spielzug sowie ebenso Informationen über die des Gegners als Eingabevariablen besitzt ist er bei der Entwicklung von Poker-Agenten nur insofern nützlich, als entwickelte Poker-Agenten mit ihrer Gegenstrategie hinsichtlich ihrer Stärke getestet werden können.

2.6 Auswahl aus mehreren Agenten

Ein weiterer Ansatz bei der Auswahl bzw. Entwicklung eines guten Poker-Agenten ist es, mehrere Poker-Agenten mit unterschiedlichen Strategien zu entwickeln und anhand der Performanz der einzelnen Agenten in vergangenen Spielsituationen denjenigen in einer ähnlichen oder gleichen Spielsituation erneut auszuwählen, der am besten in der damaligen Situation abgeschnitten hat. Doch auch hier ergeben sich Schwierigkeiten, da das Auswahlverfahren unglückliche Umstände bei der Entscheidung einzelner Agenten in der Vergangenheit nicht bzw. kaum berücksichtigen kann und es damit zur Situation kommen kann, dass der beste Agent zu Beginn der Spielserie einige unglückliche Situationen und Entscheidungen getroffen hat und somit unter Umständen nie wieder ausgewählt wird, obwohl dessen Strategie mitunter die beste wäre.

3. Modellierungsstrategien: Nicht-Verlieren vs. Gewinnen

Bei einer kleinen Anzahl von Eimern in der Bucketing-Technik wird im Allgemeinen derjenige Agent, der auf der kleineren Anzahl von Eimern basiert gegen denjenigen mit einer größeren Anzahl von Eimern als Basis verlieren. Demnach wird versucht, die größtmögliche Anzahl von Eimern in der Abstraktion zu verwenden und mathematisch zu lösen. Beschränkende Faktoren sind dabei u.a. die Größe des Speichers des verwendeten Rechners sowie die für einen Zug zur Verfügung stehende Zeit.

3.1 Nicht-Verlieren

Zunächst wird versucht, lediglich nicht zu verlieren. Der dabei behandelte Ansatz Counterfactual Regret Minimization (CFR) benötigt dabei Speicher in linearer Abhängigkeit zu den möglichen Spielzügen (nicht zu den Spielständen). Durch diesen Zusammenhang ist es möglich, bessere Nash-Gleichgewichtsstrategien zu entwickeln als bei den vorigen Ansätzen. Die Regret-Regel ist eine der strategischen Grundlagen der Entscheidungstheorie aus dem Bereich Operations Research . Es werden Opportunitätskosten für vergangene

Entscheidungen ermittelt. Beim CFR wurden zwei Spieler (dealer, opponent) konstruiert, die immer wieder (wiederholte) Spiele gegeneinander spielen und dabei ihre Strategie nach jedem Spiel so anpassen, dass die Opportunitätskosten (regret) minimiert werden. Kennt der Algorithmus in diesem Kontext beide Strategien, so ist er, ähnlich wie best response lediglich zur Untersuchung der angewandten Strategien zu verwenden. Die Errechnung des Regrets wird in folgendem Beispiel verdeutlicht: Gegeben sind die Wahrscheinlichkeiten für die drei Aktionen fold (0.5), call (0.25) und raise (0.25). Die Veränderung des Vermögens des Spielers sei bei fold (-5), call (+10) und raise (+20). Der Erwartungswert errechnet sich in diesem Beispiel zu $E(x) = 0,5 \cdot (-5) + 0,25 \cdot 10 + 0,25 \cdot 20 = 5$. Der Regret ist nun die Differenz aus der Veränderung des Vermögens jeder einzelnen Aktion und dem Erwartungswert. Für fold ergibt sich $-5 - E(x) = -10$. Für die anderen beiden Aktionen ergibt sich $\text{Regret}(\text{call}) = 5$ sowie $\text{Regret}(\text{raise}) = 15$. Anschließend wird jeder einzelne regret-Wert mit der Wahrscheinlichkeit des Gegners für die jeweilige Aktion gewichtet. Hat dieser für (fold/call/raise) jeweils $1/3$ als Wahrscheinlichkeit, so ergeben sich als akkumulierte CFR nun $(-1,33, 1,33, 5)$. Anschließend werden die eigenen Wahrscheinlichkeiten neu gewichtet. Der Aktion fold wird die Wahrscheinlichkeit 0 zugewiesen, da sie einen negativen Regret-Wert besitzt. Der Aktion call wird $1,33 / 5 = 0,2666$ und der Aktion raise $1 - 0,2666 = 0,7333$ als neue Wahrscheinlichkeit zugewiesen. So verfährt die Regret-Strategie weiter, bis ein Grenzwert des Regrets, das Nash-Gleichgewicht erreicht ist. Um den Algorithmus so effizient wie möglich zum Nash-Gleichgewicht zu bringen, wird in Implementierungen versucht, so viele Zweige abzuschneiden, wie es die Situation zulässt. In unserem Beispiel wäre dies der Fall bei der Aktion fold, da hier die Wahrscheinlichkeit aus Sicht beider Gegner auf 0 gesetzt wird. So kann die aufwendige Neuberechnung des Regrets für diese Aktion eingespart werden, was wiederum wichtig für die Einhaltung des Zeitlimits ist bzw. zur Ausnutzung besserer Berechnungen des Nash-Gleichgewichts innerhalb des Zeitlimits dient. Durch das exakte und intelligente Lösen großer Nash-Gleichgewichtsprobleme war es nun möglich neue Gleichgewichtsstrategien zu entwickeln, die in der Lage waren, alle bisher bekannten Poker-Agenten in dieser Sektion zu besiegen. Allerdings befasste sich dieser Abschnitt mit der Strategie bzw. Absicht, nicht zu verlieren (not-to-lose), was in der Realität nicht ausreichend ist. In der Realität möchte man beim Pokern Geld gewinnen und nicht nur keines bzw. möglichst wenig verlieren. Denn es könnte einen Poker-Agenten geben, der allen anderen Spielern wesentlich mehr Geld als der eigene Poker-Agent entzieht und so in der Endabrechnung, trotz keinem einzigen verlorenen Spiel vor dem eigenen Agenten platziert ist. Um einen Wettbewerb zu gewinnen muss der eigene Poker-Agent also auch Spielen um zu gewinnen (playing-to-win), womit sich der folgende Abschnitt befasst.

3.2 Gewinn-Strategie

Die Strategie best response nutzt die Strategie des Gegners am besten aus und maximiert den eigenen Gewinn. Wie schon im vorigen Abschnitt beschrieben, ist es sehr rechenaufwendig und schwierig diese Strategie exakt zu implementieren. In der vorliegenden Master-Thesis wurde deshalb nach einer good response bzw. abstract game best response-Strategie (AGBR) verfahren. Das Ergebnis dieser Strategie ist die Untere Schranke für die wirkliche best response-Strategie, welche ein mindestens genauso gutes Ergebnis beim Pokern erzielen kann. Allerdings beinhaltet die AGBR drei entscheidende Nachteile: - Die Abstraktion des Gegners muss bekannt sein - Die Strategie des Gegners muss ebenfalls

eingetragen werden - Gegenstrategien reagieren ebenfalls auf eigene Anpassungen. Daher kommt diese Strategie erst in einem zweiten Schritt zur Anwendung, nämlich dann, wenn genügend Informationen über den Gegner gesammelt sind. Im ersten Schritt wird über diese AGBR-Strategie die Frequentist Best Response-Strategie (FBR) aufgesetzt. Diese beobachtet den Gegner in einigen (Trainings-)Spielrunden, anschließend kommt AGBR zur Anwendung, um eine good response auf die Strategie des Gegners zu erwirken. Auch bei FBR gibt es zu Beginn eine Einschränkung: Zu dessen Anwendung müssen zu Beginn tausende Spiele unter vollständiger Information gespielt werden. Es müssen also alle Karten, sowohl die öffentlichen, als auch die privaten bekannt sein, und zwar auch dann wenn der Gegner die Karten per fold verwirft. Die Größenordnung der Anzahl der Spiele wird in folgendem Beispiel deutlich: Für die 5-Eimer-Variante müssen zwischen 100.000 und 1 Million Spiele gespielt werden, bis genügend Informationen gesammelt sind. Erst dann kann die AGBR angewendet werden.

3.2.1 AGBR UND FBR IN DER PRAXIS

Je nachdem, wie durchschaubar die Strategie des Gegners ist, benötigt FBR weniger bzw. mehr Trainingsspiele um auf die Gegnerstrategie zu reagieren. Gegen verschiedene Poker-Agenten als Gegner mussten zwischen 10.000 und 600.000 Trainingsspiele absolviert werden um den Break-Even-Punkt zu überschreiten.

3.2.2 WAHL DER ABSTRAKTIONSEBENE

Die Wahl der Abstraktionsebene, in diesem Falle der Anzahl der Eimer hat großen Einfluss auf das Ergebnis (millibets/game). Gegenüber einer 5-Eimer-Abstraktion mit 125 millibets/game hat die 8-Eimer-Abstraktion mit ca. 175 millibets/game einen klaren Vorteil. Des Weiteren wurde dabei deutlich, dass sich in der Größenordnung von 10 Millionen Trainingsspielen bei keiner der Abstraktionen eine Veränderung dieser Zahlen mehr ergibt. Es scheint demnach einen Grenzwert zu geben, gegen den die Poker-Agenten bei der Eruiierung der FBR konvergieren.

3.2.3 ZUSAMMENFASSUNG DER MERKMALE VON FBR

FBR erlaubt es uns, die eigene Strategie zu entwickeln und gleichzeitig die des Gegners auszuspienieren. Die Restriktionen, die FBR mit sich bringt sind bei weitem nicht so gravierend, wie die der ursprünglichen best response-Strategie. Dennoch gibt es zwei Situationen, in denen FBR keine optimalen Ergebnisse liefert. Ist der Gegner falsch modelliert, so zieht FBR mitunter die falschen Schlüsse, was im Wettbewerb eventuell mit großen Verlusten bezahlt werden muss. Außerdem setzt FBR voraus, dass der Gegner seine Strategie nicht ändert, während wir mit diesem trainieren. Ändert der Gegner auch seine Strategie während der Trainingsspiele, so zieht FBR ebenfalls die falschen Schlüsse hinsichtlich dessen Strategie, was zum selben Ergebnis führt, wie ein falsch modellierter Gegner-Agent.

4. Modellierungsstrategie: Restricted Nash Response

Bislang wurden Strategien von Poker-Agenten betrachtet, die gegen eine bekannte Gegnerstrategie gute Ergebnisse erzielten, gegen unbekannte Gegner jedoch teilweise sehr schwache Performanz boten. Nun wird mit der Restricted Nash Response (RNR) eine Möglichkeit

betrachtet, auch gegen unbekannte Gegner und bekannte Gegner mit teilweise dynamisch veränderbarer Strategie gute Gewinnergebnisse zu erzielen (stabile Gegenstrategie). Die Aufgabe von RNR ist es, die (mitunter neue bzw. geänderte) Strategie des Gegners herauszufinden und eine stabile Gegenstrategie in der Art zu bilden, dass der Gegner diese Strategie nicht gegen den eigenen Agenten nutzen kann bzw. nicht durchschauen kann. Im gebildeten Modell wird davon ausgegangen dass der Poker-Agent des Gegners mit Wahrscheinlichkeit p die Strategie unverändert verfolgt und mit Wahrscheinlichkeit $(1-p)$ die Strategie in den Grenzen der üblichen Spieltheorie dahingehend verändert, dass für ihn das Optimum erreicht wird, bzw. der Versuch dessen gestartet wird. Es muß dabei ein Kompromiss eingegangen werden zwischen der eigenen Durchschaubarkeit (für den Gegner) und dem Durchschauen der Strategie des Gegners und deren (teilweise für den Gegner offensichtlichen) Ausnutzung. Für den Startzustand bedient sich RNR ebenfalls der FBR-Strategie. Diese wird anschließend anhand der Wahrscheinlichkeiten mit dem RNR-Algorithmus zu einer stabilen Gegenstrategie weiterentwickelt. Die Anwendung von RNR ist sinnvoll, wenn wie bei der AAAIPoker Challenge 2008 der Kontostand jedes Spielers nach Ende des Turniers eine Rolle spielt. Einfache best response-Strategien sind hierfür nicht ausreichend, da sie sehr leicht vom Gegner umgangen werden können. Zur weiteren Verbesserung von RNR ist es, wie in Abschnitt 2.6 erwähnt nun sinnvoll anhand eines Teams von Agenten mit einem intelligenten Auswahlverfahren den für die Spielsituation besten Agenten zu wählen und nach dessen Strategie die Pokerspielsituation für sich zu entscheiden. Auch an dieser Stelle muss erneut der Kompromiss zwischen Durchschaubarkeit und Durchschauen+Ausnutzen eingegangen werden. Das Team kann aus FBR-Agenten oder RNR-Agenten bestehen. Auch das Hinzufügen eines Agenten mit reiner Nash-Gleichgewichtsstrategie ist möglich. Zur Auswahl des Agenten mit der besten Chance das einzelne Pokerspiel zu gewinnen wird ein eigener Algorithmus (UCB1) eingesetzt. Bei Tests im Rahmen der behandelten Master-Thesis hatten die Teams aus RNR-Agenten eine weitaus bessere Performanz, so dass diese in Wettbewerben mit oftmals neuen Gegnern besser abschneiden werden, als FBR-Agenten. Teilweise waren diese auch besser als die Nash-Gleichgewichtsstrategie, dies macht sich in Wettbewerben, in denen die Gewinner anhand der Gesamtgewinne (Kontostände) ausgewählt werden. An dieser Stelle ist die Nash-Gleichgewichtsstrategie die sicherere Strategie. Unter einigen RNR-Agenten im Team kann aber unter Umständen ein Teammitglied gefunden werden, welches gegen einen neuen Gegner die Nash-Gleichgewichtsstrategie übertreffen kann.

5. Fazit und Ausblick

Aus Sicht von Michael Johansson gibt es einige Arbeitsfelder, die in Zukunft, auch in Bezug auf weitere AAAI Poker Challenges bearbeitet werden sollten. Zum einen ist dies die Ausnutzung von paralleler Bearbeitung der Probleme (CRM, RNR) auf mehreren Rechnern. Dadurch verspricht sich der Autor, größere Abstraktionslevel effizient errechnen zu können und somit stärkere Poker-Agenten zu implementieren. Ebenso sieht er Möglichkeiten die Modellierung von Poker-Agenten-Teams weiter zu verbessern und so die Strategiewahl des Gegners vorherzusehen und darauf zu reagieren. Er nennt des Weiteren eine Möglichkeit die Anzahl der Abstraktionsebenen von bisher 12 Eimern im Preflop auf einige tausend zu erhöhen, indem einige Aktionen der Vergangenheit verworfen werden und somit mehr Spe-

icher im System für die weiteren Abstraktionsebenen frei wird. In der Master-Thesis wurden verschiedene Ansätze der Modellierung von Poker-Agenten aufgezeigt, sowie Anstöße für weitere Forschungen auf dem Gebiet der Computer Poker Challenges gegeben. Ebenso wurde das Ausmaß der Rechenoperationen und Ausprägungen die im Rahmen der Entwicklung eines Poker-Agenten bedacht werden sollten etwas deutlicher. Auch für die Arbeit an den Poker-Agenten im Rahmen dieses Seminars/Praktikums wurden einige Ideen dargelegt, deren Befolgen ein besseres Abschneiden bei der AAAI Poker Challenge 2008 versprechen kann.