

Opponent Modelling in RoShamBo

Timo Bozsolik

FB Informatik, TU Darmstadt

TIMO.BOZ@GMX.DE

Abstract

Diese Arbeit befasst sich mit dem Einsatz von computerisierten Spielern in RO SHAMBO, besser bekannt als “Stein, Schere, Papier”. Dieses einfache Spiel eignet sich besonders gut, um die generellen Techniken in Bezug auf die Modellierung und Analyse von Gegenspielern in einer Spielsituation aufzuzeigen und zu untersuchen. Der erste internationale RO SHAMBO-Programmierwettbewerb hat verschiedene Strategien für das *Opponent Modelling* hervorgebracht, von denen einige hier erläutert werden sollen. Die meisten dieser Vorgehensweisen lassen sich auch auf andere Spiele wie zum Beispiel Poker übertragen und sind daher von generellem Interesse für die Entwicklung computerisierter Spieler.

1. Einleitung

RO SHAMBO oder “Stein, Schere, Papier” ist ein weit verbreitetes Knobelspiel, bei dem zwei Spieler gegeneinander antreten und mit Handzeichen jeweils gleichzeitig eines der Symbole Stein, Schere oder Papier formen. Je nachdem, welches Symbol der eine und welches der andere Spieler wählt, gewinnt einer der beiden die aktuelle Runde; das gesamte Spiel gewinnt dann derjenige Spieler, der die meisten Runden für sich entscheiden konnte. Die genauen Regeln können Tabelle 1 entnommen werden. Wählen beide Spieler das gleiche Symbol, ergibt sich ein Unentschieden; wählt etwa der erste Spieler Schere und der zweite Papier, so gewinnt ersterer (“Schere schneidet Papier”).

	Stein	Schere	Papier
Stein	—	Stein	Papier
Schere	Stein	—	Schere
Papier	Papier	Schere	—

Tabelle 1: Regeln bei RO SHAMBO

Da jedes Symbol gegen je ein anderes gewinnt und gegen eines verliert (beispielweise gewinnt Stein gegen Schere, verliert jedoch gegen Papier), sind alle Symbole spieltechnisch gleichwertig und führen in jeder Runde zunächst zu den gleichen Gewinnchancen. Ist jedoch das Verhalten des Gegners beim RO SHAMBO bekannt oder vorhersehbar (Gegner spielt immer Stein), so kann immer und eindeutig die optimale Gegenmaßnahme ergriffen werden (spiele Papier), welche garantiert zu Erfolg führt. Auf diese Weise entwickelt das auf den ersten Blick trivial und rein zufällig wirkende Spiel eine unerwartete strategische Komplexität und stellt so ein interessantes Forschungsfeld für die Techniken des *Opponent Modelling* dar.

Im Folgenden werden in Kapitel 2 zuerst einige theoretische Überlegungen zur optimalen Strategie in RO SHAMBO vorgenommen und darauf in Kapitel 3 einige einfache,

aber sub-optimale Vorgehensweisen in Form von Standard-Bots vorgestellt und entsprechende Gegenmaßnahmen analysiert. Hierauf wird im Abschnitt 4 mit *Iocaine Powder* ein komplexer Computerspieler und der Gewinner des ersten internationalen RO SHAMBO-Programmierwettbewerbs genauer untersucht. Zum Schluss wird in Kapitel 5 auf Grundlage der gewonnenen Informationen die Frage geklärt, inwieweit sich die Ergebnisse des Opponent Modelling in RO SHAMBO sich auf das Pokerspiel beziehen lassen.

2. Theoretische Betrachtung

Aufgrund der theoretischen Gleichwertigkeit aller Symbole ist die optimale Strategie beim RO SHAMBO die, schlicht in jeder Runde eine zufällige Entscheidung zu treffen. Gegen diese Strategie gibt es keinerlei erfolgsversprechende Gegenmaßnahmen, da das Verhalten bei einem guten Zufallsgenerator komplett indeterministisch ist und vom Gegner nicht eingeschätzt werden kann. Damit unterliegt der Ausgang einer Runde ebenfalls dem Zufall. Eine genügend große Anzahl von Runden vorausgesetzt, verliert und gewinnt ein Spieler, der diese Strategie anwendet, in etwa gleicher Anzahl, egal welches Spiel und welche Analysen der Gegner verfolgt.

Weiß man jedoch genau, nach welcher Taktik der Gegenüber handelt, so ist es leicht, entsprechend darauf zu reagieren und man wird in jeder Runde gewinnen. Damit ist das Endergebnis in diesem Fall also entscheidend besser als das der optimalen Strategie. Dies zeigt, dass die zufällige Auswahl zwar global gesehen nicht geschlagen werden kann. Jedoch kann diese wiederum selbst auch einfache Strategien nicht mit einer signifikanten Differenz besiegen. Betrachtet man beispielsweise einen Computerspieler, der diese zufällige Strategie verfolgt, und lässt diesen in 1000 Runden gegen einen Bot antreten, welcher stets Stein spielt, so wird ersterer mit hoher Wahrscheinlichkeit in etwa 333 Runden gewinnen, in 333 Runden verlieren und 333 Runden werden unentschieden ausgehen. Die Differenz wird also mit einer großen Wahrscheinlichkeit betragsmäßig nahe bei 0 liegen. Ein Spieler, welcher die Taktik des Stein-Bots von vorneherein erkennt, wird alle 1000 Runden gewinnen und so im Vergleich besser abschneiden.

Analog zu den obigen Ausführungen hat der erste internationale RO SHAMBO - Programmierwettbewerb gezeigt, dass es beim Entwurf eines Computerspielers nicht darauf ankommt, dass dieser gegen alle Spieler die gleichen Gewinnchancen hat. Vielmehr gilt es, die anderen sub-optimalen, weniger guten Computerspieler möglichst hoch zu schlagen (siehe [1]). Idealerweise muss das System des Gegenspielers also erkannt oder zumindest angenähert werden, wobei das eigene Verhalten möglichst nicht preisgegeben werden darf, da der Gegner das gleiche Ziel verfolgt. Um einen Eindruck in die Vielfalt der Gegnerstrategien zu geben und damit deren Erkennung zu ermöglichen, soll im Folgenden ein Überblick über verschiedene Basisstrategien gegeben und mögliche Gegenmaßnahmen diskutiert werden.

3. Die Dummy-Bots

Um mögliche Gegner zu simulieren und den Teilnehmern eine grundsätzliche Vielfalt an zu erkennenden Strategien vorzugeben, wurden beim ersten internationalen RO SHAMBO-Programmierwettbewerb 11 sogenannte *Dummy-Bots*, also Computerspieler mit relativ einfachen Strategien, eingesetzt (siehe [1]). Diese waren von Anfang an nicht darauf ausgelegt,

die anderen Teilnehmer zu schlagen und das Turnier zu gewinnen, sondern sollten vielmehr grundsätzliche Verhaltensweisen simulieren, die von den anderen Bots erkannt und deren Schwachstellen ausgenutzt werden sollten. Daher agieren diese Bots meist unabhängig von ihrem aktuellen Gegner nach mehr oder weniger vorbestimmten Mustern, betreiben also selbst keinerlei *Opponent Modelling*. Trotzdem ist deren Analyse unabdinglich, da diese verschiedene Spielweisen erkannt werden und entsprechende Gegenstrategien entwickelt werden müssen. Aus diesem Grund sollen hier einige dieser *Dummy-Bots* auf deren grundsätzliche Strategie hin analysiert und auf deren Erkennung eingegangen werden.

Die optimale Strategie in Form eines **Random-Bots**, der jede Aktion völlig zufällig wählt, wurde schon im letzten Abschnitt diskutiert. Eine Vorhersage des nächsten Spielzuges ist hier - ein guter Zufallsgenerator vorausgesetzt - grundsätzlich selbst mit maximaler Informationsmenge nicht möglich, weshalb jede mögliche Gegenstrategie im Durchschnitt ein ähnliches Ergebnis (ein Unentschieden) produziert. Wie erwartet konnte dieser Bot im Rahmen des ausgerichteten Turniers einen mittleren Platz erreichen, da er gegen keinen der anderen Teilnehmer weder mit einer größeren Differenz gewann noch verlor. Eine Alternative zu diesem Bot sind pseudo-zufällige Strategien, welche zwar nach einem festen und deterministischen Muster handeln, welches jedoch nicht offensichtlich ist. Beispiele dafür sind der **PI-Bot**, welcher seine Aktionen anhand der Ziffern der Zahl PI (modulo 3) bestimmt, oder der **Text-Bot**, welcher dafür die Buchstaben eines beliebigen Textes benutzt. Letzterer konnte aufgrund der in der menschlichen Sprache auftretenden Muster jedoch leicht von besseren Pattern-Detektoren geschlagen werden. Diese beiden Bots sind in ihren Aktionen vollständig vorhersehbar und damit leicht zu schlagen, jedoch nur, wenn man die zugrunde liegende Zahl bzw. den Text kennt. Daher können effektive Gegenstrategien auch nur auf diese Daten ausgelegt sein und sind im Endeffekt nicht tragbar.

Auf den ersten Blick genauso zufällig, jedoch völlig deterministisch und nach einem erkennbaren Modell arbeitend ist der **De Bruijn-Bot**. Dieser wählt seine Aktionen nach einer Zeichenkette bestimmter Länge, welche nach einem festen System generiert wird. In einer solchen De Bruijn-Sequenz kommen alle Kombinationen einer bestimmten Länge aus einer bestimmten Anzahl von Zeichen genau einmal vor. Nimmt man beispielsweise die Zeichen "r" (für Stein), "s" (für Schere) und "p" (für Papier) als Grundlage, so enthält die Sequenz "rrpspprrs" jede Sub-Sequenz der Länge 2 über diesem Alphabet genau einmal (also "rr", "rp" etc.). Dieser Bot schnitt erstaunlicherweise als einer der besten im Feld ab, was daran liegen könnte, dass die Aktionen hier gleichmäßiger verteilt sind als beim **Random-Bot** und dass kein Muster zweimal vorkommt, was jede Erkennung von Patterns von vorneherein unnütz und sogar kontraproduktiv werden lässt.

Während die bisher vorgestellten Bots bei der Wahl ihres nächsten Spielzuges keinerlei Informationen über den bisherigen Spielverlauf zugrunde legen, richten sich die folgenden Bots nach den letzten selbst getroffenen Entscheidungen. Der **Switch-a-lot-Bot** etwa wählt mit einer hohen Wahrscheinlichkeit immer ein Symbol, welches unterschiedlich zum letzten Zug ist. Ist diese Strategie ersteinmal erkannt, gestalten sich die Abwehrmaßnahmen denkbar einfach. War der letzte Spielzug dieses Bots etwa **Schere**, so wird der nächste entweder **Stein** oder **Papier** sein. Die Reaktion mit **Papier** wird in der Hälfte der Fälle ein Unentschieden ergeben und in der anderen Hälfte zum Gewinn der Runde führen, jedoch nie mit einem Verlust. Ein ähnlich vorhersehbares Verhalten liegt dem **Flat-Bot** zugrunde, welcher seine Aktionen zwar auch zufällig wählt, jedoch durch Analyse seiner bisherigen Spielzüge

versucht, die Anzahl der jeweils gespielten Symbole, in etwa gleich zu halten. So wählt dieser Bot in 80% der Fälle immer die bisher seltenste Aktion. Ist ein Gegner in der Lage, dieses Verhalten zu durchschauen, so kann die optimale Reaktion leicht berechnet werden. Durch ein solches Verhalten werden wie auch beim **Switch-a-lot-Bot** in gewisser Weise menschliche (meist unbewusste) Strategien simuliert, da auch der Mensch lange Sequenzen des gleichen Symbols eher vermeidet und so die aktuelle Entscheidung von den vorherigen abhängig macht.

Der komplexeste und einzige auf Erfolg ausgelegte Dummy-Bot im ausgerichteten Turnier, war der **Anti-rotn-Bot**. Dieser analysiert die gegnerischen Spielzüge in Form von Rotationen (aufeinanderfolgende gleiche Symbole werden als 0, verschiedene als +1 bzw. -1 gewertet) und trifft seine Entscheidungen auf Basis dieser Historie. Ähnlich wie bei der optimalen Gegenstrategie beim **Switch-a-lot-Bot** richtet sich dann der nächste Spielzug nicht nach dem wahrscheinlichsten nächsten Spielzug des Gegners, sondern nach dem mit der geringsten Wahrscheinlichkeit. Wird beispielsweise Stein als unwahrscheinlichste nächste Aktion des Kontrahenten angenommen, so wird die Reaktion mit Schere gegen die beiden vermutlich häufiger auftretenden Symbole Schere und Papier nie verlieren. Zusätzlich werden hier die eigenen Verluste mitgezählt und ab einem Gesamtverlust von 4% nur noch nach zufälliger Strategie gespielt, um hohe Niederlagen zu vermeiden.

Die in diesem Kapitel besprochenen *Dummy-Bots* verfolgten alle **eine** einfache, relativ leicht zu erkennende Spielweise. Ein großer Teil der eigentlich am Turnier teilnehmenden Computerspieler bediente sich dem Konzept der *gemischten Strategien*. In diesem Fall werden verschiedene einfache Spielweisen parallel implementiert, in einer Vorentscheidung die nächste Aktion jeder einzelnen berechnet und dann diejenige mit dem größten Erfolg in der Vergangenheit endgültig verwendet. Zwar kann sich dieser Ansatz leicht an eine Vielzahl an Gegnern anpassen, jedoch richtet sich die gesamte Qualität nach der der besten Teil-Strategie, weshalb sich im Turnier allgemeinere Verfahren (siehe z.B. *Iocaine Powder* in Kapitel 4) eher durchgesetzt haben. Eine weitere Unterteilung der Bots kann man in Bezug auf die Analyse der bisherigen Spielrunden treffen (siehe [4]). Hier existieren verschiedene statistische Methoden (wie etwa beim **Flat-Bot**) sowie Formen der direkten Analyse der Historie, wie etwa die Erkennung von Mustern. Damit die eigene Strategie außerdem verschleiert und vor hohen Verlusten gegen starke Gegner bewahrt wird, wurde in vielen Ansätzen ein gewisses Rauschen in Form von zufälligen Aktionen eingeführt. Viele Bots weichen sogar erst von der randomisierten Spielweise ab, wenn diese eine signifikante Schwäche beim Gegner erkennen. Eine kurze Vorstellung der besten Computerspieler im Turnier und die kompletten Ergebnisse finden sich in [3].

4. Iocaine Powder

Mit dem Gewinner des ersten internationalen RO SHAMBO-Programmierwettbewerbs, *Iocaine Powder* (siehe [2]), soll nun ein komplexer, sehr spielstarker Bot detailliert untersucht werden. Dessen Strategie ist sehr generischer Natur und zeigt, dass ein guter Prognose-Algorithmus auch auf viele andere Probleme angewendet werden kann, wie etwa Datenkompression, eMail-Klassifizierung oder Data Mining.

4.1 Metastrategien

Jeder direkte Algorithmus zur Vorhersage der Aktionen eines Spielers hat eine gravierende Schwäche: der Gegner kann die eigene Vorgehensweise erkannt haben und selbst entsprechend handeln bzw. agieren. Um diesem Nachteil entgegenzuwirken, werden in *Iocaine Powder* sogenannte Meta-Strategien eingesetzt. Diese sechs übergeordneten Taktiken erweitern dann ein gegebenes Prognose-Modul, so dass auf Basis von dessen Entscheidung eine neue getroffen wird, die wieder weitere Faktoren berücksichtigt. Der entscheidende Aspekt ist hier die Frage, wie man dagegen vorgehen kann, dass ein Gegner die eigene Spielweise durchblickt. Dazu wird im Folgenden von einem Vorhersage-Modul \mathbf{P} ausgegangen, dass die nächste Aktion des Gegners unabhängig voraussagt.

Die einfachste mögliche Meta-Strategie $\mathbf{P.0}$ verkörpert die unmittelbare Anwendung des Vorhersagers und versucht direkt, das von \mathbf{P} bestimmte Symbol zu schlagen. Beispielsweise würde $\mathbf{P.0}$ also Papier liefern, um einen von \mathbf{P} prognostizierten Stein des Gegners zu besiegen. Einen Schritt weiter geht $\mathbf{P.1}$, die Abwehr des sogenannten *second-guessing*, bei der davon ausgegangen wird, dass der Gegner die eigene Vorhersage kennt und entsprechend handelt. Würde \mathbf{P} also Stein liefern und unser Gegner die Funktionsweise von \mathbf{P} kennen, würde er damit rechnen, dass wir Papier spielen. Daher wird der Gegner in dieser Situation selbst Schere spielen, um dieses zu schlagen. $\mathbf{P.1}$ liefert nun Stein, um wiederum die Schere des Gegners zu egalisieren. Wieder eine Stufe weiter geht $\mathbf{P.2}$, bei der davon ausgegangen wird, dass der Gegner das eigene Verfahren von $\mathbf{P.1}$ kennt, wiederum einen Schritt weiter denkt und dieses versucht abzuwehren. Da $\mathbf{P.1}$ als nächste Aktion Stein empfohlen hat und man annimmt, dass der Gegner auch $\mathbf{P.1}$ durchblickt hat und als Reaktion entsprechend Papier spielen würde, begegnet man diesem mittels $\mathbf{P.2}$ mit Schere. Diese Kette könnte nun unendlich fortgesetzt werden, in dem man sich weiter die Frage stellt, was ist, wenn der Gegner die jeweils vorherige Strategie durchschaut hat und dies wiederum abwehren will. Jedoch empfiehlt $\mathbf{P.0}$ bereits das Spiel von Papier, $\mathbf{P.1}$ liefert Stein und nach $\mathbf{P.2}$ sollte Schere gespielt werden, so dass alle möglichen Aktionen bereits abgedeckt sind und weitere Schritte wieder nur die gleichen Spielzüge liefern würden (es gilt $\mathbf{P.3}=\mathbf{P.0}$, $\mathbf{P.4}=\mathbf{P.1}$ usw.).

Umgekehrt zur bisherigen Annahme, nämlich dass die Aktionen des Gegners von \mathbf{P} bestimmt werden, gehen die folgenden drei Strategien davon aus, dass der Gegner selbst \mathbf{P} benutzt. Daher wird hier auch nicht \mathbf{P} zur Vorhersage

herangezogen, sondern \mathbf{P}' , bei welchem die Position des eigenen Bots und des Gegners vertauscht ist. Damit liefert \mathbf{P}' quasi eine Vorhersage des eigenen Spiels. Lautet diese beispielsweise Stein, so wird angenommen, dass der Gegner Papier spielt, weswegen $\mathbf{P'.0}$ Schere suggeriert. $\mathbf{P'.1}$ und $\mathbf{P'.2}$ sind dann analog zu $\mathbf{P.1}$ und $\mathbf{P.2}$ wieder genau die Taktiken, die davon ausgehen, dass der Gegner das eigene Vorgehen von $\mathbf{P'.0}$ und $\mathbf{P'.1}$ kennt, entsprechend einen Schritt weiter gehen und darauf reagieren.

Die diskutierten Ansätze stellen eine wirkungsvolle Abwehr gegen intelligente Gegenspieler dar, jedoch kennt man die Taktik des Gegners und vor allem dessen Level des "Weiterdenkens" nicht. Damit kann jede dieser Vorgehensweisen zu einem gegebenen Vorhersage-Modul die beste sein, jedoch ist a priori nicht bekannt, welche. Und da es sechs verschiedene Strategien gibt, wovon jeweils zwei Stein, zwei Schere und zwei Papier empfehlen, kann die Wahl der nächsten Aktion auch nicht in irgendeiner Weise eingeschränkt werden. Es wird jedoch angenommen, dass die Aktionen des Gegners konsistent sind und dieser sein grundsätzliches

Verhalten nicht (oder nur sehr langsam) verändert. Daher kann man die zurückliegenden Spielzüge benutzen, um zu entscheiden, welche dieser sechs Strategien bisher die beste war und diese auch im nächsten Zug einsetzen. Aufgrund der im Nachhinein ja bekannten Erfolgsquote jeder Strategie wird also diejenige mit der besten Performance ausgewählt und angewandt.

4.2 Algorithmen zur Vorhersage

Jede noch so durchdachte Meta-Strategie ist nutzlos ohne einen Basis-Algorithmus zur Vorhersage des nächsten Zuges des Gegners, auf deren Grundlage dann die weitere Entscheidung abgeleitet werden kann. Hier benutzt *Iocaine Powder* drei verschiedene Algorithmen, wobei die aktuell anzuwendende Variante wieder durch den Vergleich der Güte in den vorangegangenen Spielzügen bestimmt wird.

Eine einfache statistische Methode, die nächste Aktion des Gegners vorauszusehen, stellt die **Frequenzanalyse** dar. Diese findet schlicht das in der Vergangenheit am häufigsten eingesetzte Symbol des Gegners und sagt dieses auch für den nächsten Zug voraus. Damit kann zum einen das Verhalten extrem einfacher Bots vorhergesagt werden, auf der anderen Seite werden in Kombination mit den bereits diskutierten Meta-Strategien Bots, die selbst eine Frequenzanalyse einsetzen, leicht geschlagen.

Im Gegensatz dazu analysiert die Methode des **History-Matchings** direkt die bereits gespielten Runden und sucht dort nach auftretenden Mustern. Bestanden die letzten Runden beispielweise aus **Stein gegen Schere**, **Stein gegen Papier** und **Papier gegen Schere**, so sucht dieser Algorithmus nach weiteren Vorkommen dieser Sequenz in der Vergangenheit und sagt das Symbol voraus, welches dem am häufigsten folgenden nächsten Zug des Gegners entspricht. Dabei werden Vorkommen als aussagekräftiger bewertet, je länger die übereinstimmende Abfolge und je öfter diese bereits aufgetreten ist.

Eine Absicherung gegen die erfolgreiche Analyse und Vorhersage des eigenen Spiels durch einen stärkeren Gegner bildet das **zufällige Raten** des nächsten Zuges. Falls ein Gegenspieler in der Lage ist, die oberen beiden Vorhersager und alle Variationen durch die Meta-Strategien abzuwehren, erreichen diese für die vergangenen Runden eine schlechte Erfolgsquote. Als Konsequenz ist die Quote dieses Algorithmus die beste und gegen besonders gute Gegner wird einfach nach dem Zufallsprinzip gespielt. Damit wird *Iocaine Powder* auch gegen wesentlich stärkere Gegner nicht besonders hoch verlieren oder sogar gewinnen, da es gegen diese Taktik keine wirksame Gegenstrategie gibt.

4.3 Zusätzliche Erweiterungen

Die Kombinationen aus drei von Grund auf verschiedenen Algorithmen zur Vorhersage des Gegnerverhaltens in Kombination mit den sechs Meta-Strategien, die die Verwendung von ähnlichen Möglichkeiten der Vorhersage bei Gegnern kompensieren sollen, hat sich insgesamt als gute Gesamt-Strategie herausgestellt. Die Annahme, dass der Gegenspieler sein Verhalten im Zeitverlauf nicht ändert, kann jedoch offensichtlich nicht immer getroffen werden. Daher wurde eine weitere Ebene von Meta-Meta-Strategien eingeführt, welche den Bewertungshorizont zur Auswahl der eigentlichen Meta-Strategie (z.B. **P.0** oder **P.1** etc.) ändert. Dazu wird wiederum die Erfolgsquote jeder Entscheidung basierend auf den letzten 1000, 100, 10 etc. Zügen verglichen und der Zeitraum, der im gesamten Spiel am

besten war, ausgewählt. Damit können auch Gegner, die ihr Spiel in einer bestimmten Regelmäßigkeit ändern, besser eingeschätzt werden. Weiterhin werden auch die Zeithorizonte der Vorhersage-Algorithmen variiert, verglichen und die beste Variante zur Bestimmung des nächsten Zuges gewählt.

Insgesamt hat sich herausgestellt, dass diese generische Vorgehensweise aus den drei Ebenen der Vorhersage-Algorithmen, der Meta-Strategien und der Variation der Bewertungszeiträume als sehr generische Variante den hoch spezialisierten Algorithmen überlegen ist. Dabei wurden in *Iocaine Powder* keinerlei Mechanismen zur Verschleierung der eigenen Spielweise eingebaut, so dass davon auszugehen ist, dass ein Bot, der speziell darauf programmiert ist, *Iocaine Powder* zu schlagen, darin auch erfolgreich sein wird. Im Hinblick auf die Vielzahl der möglichen Strategien, die ein Gegner anwenden kann, kann dieser Ansatz jedoch als sehr robust und effektiv bewertet werden.

5. Schlussfolgerungen

Zusammenfassend kann man sagen, dass sich hinter dem zunächst trivial erscheinenden Spiel RO SHAMBO ein auf den zweiten Blick komplexes und vielschichtiges Szenario verbirgt. Dieses Spiel kann als reine Anwendung des Opponent-Modelling angesehen werden, weshalb es für dessen Erforschung besonders nützlich sein kann. Die hier entwickelten Techniken zur Vorhersage des Gegnerverhaltens können dann direkt oder in Abwandlungen auch in anderen Spielen mit unvollkommener Information wie Poker eingesetzt werden. Natürlich kommen hier weitere Spielfaktoren hinzu und es müssen mehr Aspekte als nur der wohl wahrscheinlichste nächste Zug des Gegners beachtet werden. Beispielsweise ist dieser unwichtig, wenn man beim Poker das definitiv beste Blatt am Tisch besitzt. Da diese Situation jedoch sehr selten vorkommt, ist es von Vorteil zu wissen, was der Gegner vermutlich als nächstes tun wird, um so zum Beispiel unnötige Bluffs zu vermeiden. Außerdem weiß jeder Pokerspieler, dass man auch mit zwei Assen auf der Hand verlieren kann und daher jede verfügbare Information bei seinem Spiel beachten sollte. Grundsätzlich ist es also ein wichtiger Faktor zu wissen, wie sich ein Spieler in einer bestimmten Situation verhält, um sein eigenes Spiel darauf ausrichten zu können. Gerade in einem Kontext wie dem Heads-Up-Game spielt es auch eine große Rolle, zu ahnen was der Gegner denkt und was er denkt, was man selbst denkt etc. In diesem Fall könnten prinzipiell die Überlegungen zu den Meta-Strategien von *Iocaine Powder* nützlich sein.

Literatur

- [1] Billings, D. Thoughts on RO SHAMBO, 2000. ICGA Journal, Vol. 23, No. 1, pp. 3-8.
- [2] Egnor, D. Iocaine Powder, 2000. ICGA Journal, Vol. 23, No. 1, pp. 33-35.
- [3] Billings, D. The First International RO SHAMBO Programming Competition, 2000. ICGA Journal, Vol. 23, No. 1, pp. 42-50.
- [4] Billings, D. The Second International RO SHAMBO Programming Competition, 2001. <http://www.cs.ualberta.ca/~darse/rsbpc.html>.