Eine spieltheoretische Betrachtung des Pokerspiels

Stefan Lück STEFAN.LUECK@STUD.TU-DARMSTADT.DE
Fachbereich 01: Wirtschaftsinformatik
Technische Universität Darmstadt

Claudio Weck CLAUDIO.WECK@STUD.TU-DARMSTADT.DE
Fachbereich 20: Informatik
Technische Universität Darmstadt

Abstract

Die Entwicklung eines automatisierten Agenten, der beim Pokerspiel mit menschlichen Gegnern mithalten kann, beschäftigt schon seit längerer Zeit Wissenschaftler im Bereich der Erfoschung von künstlicher Intelligenz. Nachdem das Schachspiel mittlerweile fast optimal durch einen Computer gespielt werden kann, stellt das Pokerspiel eine ganz neue Herausforderung dar. Diese Arbeit beschreibt, wie man mit den Konzepten der Spieltheorie ein Pokerspiel analysieren und auch (theoretisch) lösen kann. Nach einer allgemeinen Einführung in die Spieltheorie und einer Einordnung des Pokerspiels in jener, werden Methoden für die exakte Darstellung und Lösung des Spiels aufgezeigt. Ähnlich wie beim Schachspiel ist allerdings aufgrund der Spielgröße eine vollständige Berechnung nicht möglich. Wir zeigen daher, welche Ansätze für Optimierungen und Approximationen es gibt, und wie man diese auf eine konkrete Problemstellung in Form der AAAI-PokerChallenge anwenden kann.

1. Spieltheoritsche Grundlagen

Die Spieltheorie beschäftigt sich mit der mathematischen Analyse von Entscheidungssituationen mit mehreren Akteuren. Die Akteure (*Spieler*) interagieren dabei miteinander, sodass die Entscheidungen eines Einzelnen von denen der Anderen beeinflußt werden. Erstmals formal beschrieben wurde die Spieltheorie in den 1920er und 1930er Jahren durch John von Neumann[1], der damit als Begründer dieser Forschungsrichtung gilt und zusammen mit Oskar Morgenstern 1944 das erste Standardwerk der Spieltheorie veröffentlichte[2]. Besonders im volkswirtschaftlichen Bereich hat sich die Spieltheorie für die Modellierung komplexer Interaktionen durchgesetzt[3].

Das Ziel der Spieltheorie besteht darin, Methoden für die Analyse von Spielen zur Verfügung zu stellen. Man unterscheidet dabei einerseits zwischen Werkzeugen zur Modellierung und Darstellung von Spielen sowie Verfahren zur Vorhersage von Spielstrategien andererseits. Die Vorhersage von Spielstrategien bezeichnet man auch als das Lösen eines Spiels (siehe dazu Abschnitt 1.3).

1.1 Definition eines Spiels

Formal betrachtet besteht ein Spiel in der Spieltheorie aus einer Menge an *Spielern*, die *Entscheidungen* treffen müssen. Jeder Spieler kann dazu aus einer vorgegebenen Menge

1. Deutsch-ungarischer Mathematiker (*1903 - †1957).
2. Vgl. (von Neumann & Morgenstern, 1944).
3. Z.B bei der Modellierung der Auktion zur Versteigerung von UMTS-Lizenzen.

an *Handlungsalternativen* wählen. Außerdem gibt es für jeden Spieler eine *Nutzenfunktion*, die in Abhängigkeit der getroffenen Entscheidungen *aller* Spieler einen Auszahlungsbetrag, also seinen persönlichen Nutzen, festlegt. Eine *Strategie* für einen Spieler beschreibt nun, für welche der Alternativen sich ein Spieler unter welchen Umständen entscheidet. Beispiel 1 beschreibt den Aufbau eines einfachen Interaktionsspiels.

Definition 1 Mathematische Formulierung eines Spiels

$$\begin{array}{lll} \mathcal{P} & & \textit{Menge an Spielern} \\ \mathcal{A}_p & .\forall p \in \mathcal{P} & \textit{Handlungsalternativen pro Spieler} \\ u_p : \mathcal{A}_1 \times \cdots \times \mathcal{A}_{|\mathcal{P}|} \to \mathbb{R} & .\forall p \in \mathcal{P} & \textit{Auszahlungsfunktion pro Spieler} \end{array}$$

Beispiel 1 *Bei dem Spiel Schere-Stein-Papier gibt es zwei Spieler. Jeder Spieler muss sich entscheiden, welche Handlungsalternative in Form von drei möglichen Symbolen (Schere, Stein oder Papier) er wählt. Dabei gilt, dass Schere gegen Papier gewinnt, Papier gegen Stein und der Stein gegen die Schere. Abhängig von der Entscheidung des Gegners gewinnt also entweder ein Spieler selbst (Auszahlung* $+1$*), sein Gegner (Auszahlung* -1*) oder es gibt ein Unentschieden, wenn beide Spieler das gleiche Symbol wählen (Auszahlung* 0*). Eine mögliche Strategie eines Spielers wäre, immer das Symbol Schere zu wählen. Eine andere Strategie könnte sein, dass er sich zufällig für eines der Symbole entschiedet.*

Für die Darstellung eines Spiel wird häufig die sogenannte *Normalform* eines Spiels gewählt. Diese besteht aus einer $|\mathcal{P}|$-dimensionalen Matrix, wobei jede Dimension der Matrix durch die Handlungsalternativen genau eines Spielers aufgespannt wird. Die Zellen der Matrix geben die Werte der Auszahlungsfunktion für die einzelnen Spieler an. Diese sind voneinander durch Semikola getrennt. Im Fall von zwei Spielern ist die Normalform eine einfache Tabelle. Tabelle 1 zeigt eine Normalformdarstellung des Schere-Stein-Papier-Spiels aus Beispiel 1.

		Spieler 2		
		Schere	Stein	Papier
Spieler 1	Schere	0 ; 0	-1 ; 1	1 ; -1
	Stein	1 ; -1	0 ; 0	-1 ; 1
	Papier	-1 ; 1	1 ; -1	0 ; 0

Tabelle 1: Spiel Schere-Stein-Papier in Normalform

1.2 Klassifikation von Spielen

Spiele können nach ihrer Art und ihren Eigenschaften klassifiziert werden. Dies ist notwendig, um die richtigen Methoden und Instrumente für die Analyse wählen zu können. Eine zentrale Unterscheidung erfolgt in statische und dynamische Spiele. Bei *statischen (oder strategischen) Spielen* erfolgt die Entscheidung der Spieler für eine Handlungsalternative simultan. Kein Spieler kennt die Entscheidungen der anderen zu dem Zeitpunkt, an dem er seine eigene Strategie auswählen muss. Statische Spiele sind gut zur Modellierung von Situationen geeignet, wie sie häufig in der Wirtschaft und Politik auftreten. Dabei müssen die

Handelnden Entscheidungen über die Zukunft treffen, wobei sich die gewählten Strategien der Mitspieler erst im Laufe der Zeit offenbaren. Typische Beispiele für solche Situationen sind Entscheidungen über Markteintritte, einen Kapazitätenausbau oder das Rüstungsverhalten. Auch das Schere-Stein-Papier-Spiel ist ein statisches Spiel. Für die Darstellung von statischen Spielen kann direkt die Normalform verwendet werden.

Für die Modellierung von Gesellschaftsspielen hingegen sind statische Spiele i.S.d. Spieltheorie häufig nicht geeignet, da bei jenen die Spieler zumeist abwechselnd Entscheidungen treffen. Solche rundenbasierten Situationen werden mit *dynamischen Spielen* beschrieben. Die Spieler sind dabei nacheinander am Zug und müssen jeweils eine Entscheidung treffen. Jede dieser Entscheidungen beeinflusst den weiteren Verlauf des Spiels und beschränkt häufig den Alternativenraum der nachfolgenden Spieler (z.B. wenn ein Spieler beim Schachspiel den König seinenes Gegners bedroht). Die Normalform kann daher nur schwer zur intuitiven Darstellung dynamischer Spiele verwendet werden. Ein besser geeignetes Instrument ist die extensive Darstellung, die in Abschnitt 2.1 beschrieben wird.

Häufig wird der Spielverlauf auch durch (bewusst herbeigeführte) zufällige Ereignisse beeinflusst. Dies ist insbesondere bei Würfelspielen (durch Würfeln) und Kartenspielen (durch Mischen) der Fall. Durch den Einfluss des Zufalls kann es passieren, dass eine Strategie bei ansonsten identischen Spielverlauf entweder eine sehr hohe oder sehr niedrige Auszahlung bewirken kann. Dies ist z.B. offensichtlich bei Würfelwettspielen: Man setzt einen bestimmten Betrag auf eine gerade oder ungerage Augenzahl. Abhängig von der gefallenen Zahl bekommt man nun entweder den doppelten Einsatz zurück oder man verliert seinen Einsatz. Die Existenz von zufälligen Momenten in einem Spiel ist also eine zentrale Eigenschaft, die bei einer Analyse des Spiels berücksichtigt werden muss.

Eine weitere wichtige Eigenschaft von Spielen ist die Existenz von privaten Informationen. Unter privaten Informationen sind spielrelevante (also entscheidungsbeeinflussende) Daten zu verstehen, die nicht allen Spielern offen vorliegen. Spieler sind dadurch in der Lage Entscheidungen zu treffen, die andere nicht vorhersagen können. Bei einem Kartenspiel stellen zum Beispiel die Karten, die ein Spieler auf der Hand hat, eine private Information dar. Auch die eigenen Entscheidungen im Spielverlauf stellen häufig eine private Information dar. Beim Spiel Schiffe-Versenken z.B. liegt die erste Entscheidung für jeden Spieler in der Platzierung der Schiffe. Diese Information behält jeder Spieler allerdings für sich und hat dadurch einen Informationsvorsprung. Spiele, in denen keine privaten Informationen vorliegen, bezeichnet man als *vollkommene* oder *perfekte* Spiele. Ein Beispiel für ein Solches Spiel ist das Schachspiel, bei dem alle Informationen über das Spiel (man spricht vom *Spielstatus*) jedem Spieler offen zugänglich sind.

Tabelle 2 gibt eine kurze Übersicht und klassifiziert beispielhaft Spiele nach ihren strukturellen Eigenschaften. Bei den aufgeführten Spielen handelt es sich um dynamische Spiele.

	ohne Zufall	mit Zufall
vollkommen	Schach Go	Backgammon Kniffel
nicht vollkommen	Schiffe versenken	Poker Rommé

Tabelle 2: Klassifikation von Spielen (Beispiele)

Neben der Klassifikation von Spielen nach strukturellen Kriterien kann man eine weitere Untergliederung nach inhaltlichen Aspekten vornehmen. Diese Eigenschaften können dann häufig zur Vereinfachung der Lösungssuche verwendet werden. Eine mögliche Eigenschaft eines Spiels ist z.B. die *Symmetrie*. Ein Spiel ist genau dann symmetrisch, wenn alle Spieler über identische Alternativenräume und Nutzenfunktionen verfügen ($\mathcal{A}_p = \mathcal{A}_q \wedge u_p(a_1, \ldots, a_{|\mathcal{P}|}) = u_q(a_1, \ldots, a_{|\mathcal{P}|})\ .\forall p, q \in \mathcal{P}$). Für einen Spieler ist es also unerheblich, ob er an erster oder zweiter Position spielt. Ein Beispiel für ein symmetrisches Spiel ist das Schere-Stein-Papier-Spiel.

Außerdem kann man zwischen nicht-kompetitiven und kompetitiven Spielen unterscheiden. Bei nicht-kompetitiven Spielen erreichen die Spieler alle einen höheren Nutzen, wenn sie zusammenarbeiten. Ein Beispiel für ein solches Spiel ist die Mammutjagd, wo Jäger in der Gruppe jagen müssen, um ein großes Mammut zu erbeuten und so einen hohen Nutzen zu erzielen. Bei (strikt) kompetitiven Spielen hingegen Verhalten sich die Auszahlungen der Spieler reziprok zueinander, d.h. damit ein Spieler eine höhere Auszahlung bekommen kann, muss sich gleichzeitg die Auszahlung einer seiner Mitspieler (Gegner) verringern. Eine typische Klasse von Spielen, die diese Eigenschaft besitzen, sind die Nullsummenspiele. Dabei ergibt die Summe aller Auszahlungen immer Null (oder allgemein eine Konstante c, was formal äquivalent ist: $\sum_{p \in \mathcal{P}} u_p(a_1, \ldots, a_{|\mathcal{P}|}) = c$), woraus das reziproke Verhalten der Nutzenfunktionen direkt folgt.

Wie bereits aus Tabelle 2 zu entnehmen ist, handelt es sich bei Poker um ein dynamisches Spiel mit unvollkommener Informationen und Zufallsereignissen. Poker ist dabei ein strikt kompetatives Nullsummenspiel (einer gewinnt, alle anderen verlieren) und nicht symmetrisch, da die zu wählenden Handlungsalternativen von der Position am Tisch abhängen. Diese Klassifikation trifft auf die meisten Kartenspiele zu. Das Zufallsmoment resultiert dabei aus dem Mischen der Karten und der anschließenden zufälligen Verteilung. Durch das Halten der Karten auf der Hand schützt jeder Spieler seine privaten Informationen, dadurch ist das Spiel unvollkommen. Ein Beispiel für ein Kartenspiel mit perfekten Informationen hingegen ist z.B. Offizierskat, wo alle Karten offen auf dem Tisch liegen.

1.3 Vorhersage des Spielverlaufs

Im Rahmen eines Spiels trifft jeder Spieler Entscheidungen, durch die die Auszahlungen am Ende des Spiels beeinflusst werden. Die Art und Weise, wie ein Spieler seine Entscheidungen trifft, nennt man seine *Strategie*. Die Gesamtheit der Strategien aller Spieler kann man zu einem Strategienvektor zusammenfassen. Dieser Strategienvektor bestimmt den Verlauf des Spiels eindeutig, da für jede mögliche Situation des Spiels die Handlungsweise des jeweils am Zug befindlichen Spielers definiert wird. Fasst man die Strategien von allen Spielern bis auf einen zusammen, erhält man einen Fremdstrategienvektor, der den Verlauf des Spiels in Abhängigkeit der Entscheidungen eines einzelnen Spielers wiederspiegelt. Eine Betrachtung dieses Vektors ist sinnvol, wenn man ein Spiel aus der Sicht eines einzelnen Spielers betrachtet und die Strategien der Mitspieler als gegeben ansieht.

Definition 2 Strategien
Eine Strategie $s_p \in \mathcal{S}_p$ eines Spielers $p \in \mathcal{P}$ beschreibt für jede denkbare Entscheidungssituation eines Spielers, für welche Handlungsalternative er sich entscheiden wird. Sie definiert also genau das Verhalten eines Spielers während des Spiels.

Ein Strategienvektor S ist eine Sammlung, die jeweils genau eine Strategie für jeden Spieler enthält. Dadurch wird das Verhalten aller Spieler und damit der gesamte Spielablauf definiert:

$$S = \{s_p | \forall p \in \mathcal{P}\}$$

Ein Fremdstrategienvektor S_{-p} enthält eine Sammlung aller Strategien außer der eines einzelnen Spielers $p \in \mathcal{P}$:

$$S_{-p} = \{s_q | \forall q \in \mathcal{P} \land q \neq p\}$$

Jeder Spieler kann seine Strategie für ein Spiel frei wählen. Dabei geht man davon aus, dass sich ein Spieler aber nicht für eine beliebige Strategie entscheiden wird, sondern versucht, durch die Wahl einer geeigneten Strategie seinen eigenen Nutzen am Spielausgang zu maximieren. Dieses grundlegende Verhalten eines Individuums, dass man in der Spieltheorie für alle Spieler annimmt, nennt man das Rationalitätsprinzip.

Definition 3 Rationalitätsprinzip
Das Rationalitätsprinzip besagt, dass sich jeder Spieler für genau die Strategie s_p^* entscheiden wird, die ihm die höchste Auszahlung am Spielende sichert. Nimmt man die Strategien der Mitspieler als gegeben an, ist eine optimale Strategie *der Spieler also:*

$$s_p^* = \arg\max_{s_p \in \mathcal{S}_p} u_p(s_p | S_{-p}) \quad .\forall p \in \mathcal{P}$$

Damit ein Spieler eine optimale Strategie wählen kann, müsste er aber bereits die Strategie des Gegners kennen. Dies ist normalerweise aber nicht der Fall, daher muss ein Spieler Annahmen über die Strategien seiner Mitspieler treffen. Diese Annahmen können aus Erfahrungen von früheren Teilnahmen an einem Spiel resultieren. Es ist dabei davon auszugehen, dass jeder Spieler seine Strategie solange ändert, bis er seinen Nutzen nicht mehr verbessern kann[4]. Wenn jeder Spieler eine solche Strategie gefunden hat, befindet sich das Spiel in einem Gleichgewicht, d.h. keiner der Spieler wird seine Strategie ändern, solange auch alle anderen Spieler ihre Strategie beibehalten. Ein Strategienvektor, der diese Bedingung erfüllt, nannt man *Nash-Gleichgewicht*[5].

Definition 4 Gleichgewichtsstrategien
Ein Strategienvektor stellt genau dann eine Gleichgewichtsstrategie[6] dar, wenn die Strategien der Spieler ihre jeweils gegenseitig besten Antworten *auf die Strategien ihrer Mitspieler darstellen. Es kann also kein Spieler seine Auszahlung durch das Ändern seiner Strategie erhöhen, das Rationalitätsprinzip ist für alle Spieler gleichzeitig erfüllt:*

$$S^* \rightarrow \forall p \in \mathcal{P}. \quad s_p^* = \arg\max_{s_p \in \mathcal{S}_p} u_p(s_p | S_{-p}^*)$$

4. Für weitere Erklärungen über das Zustandekommen eines Gleichgewichts siehe (Osborne, 2004, S. 134ff).
5. Benannt nach dem Mathematiker und Nobelpreisträger John Forbes Nash Jr., der diese Eigenschaft zuerst 1950 beschrieben hat.
6. Vgl. zum Nash-Gleichgewicht insbesondere (Nash, 1951).

In der Spieltheorie bezeichnet man eine Gleichgewichtsstrategie, also die Menge der einzelnen optimalen Spielerstrategien, als die *Lösung* eines Spiels. Ein Spiel kann mehrere Lösungen haben oder auch gar keine. Bei einer Betrachtung des Spiels Schere-Stein-Papier aus Beispiel 1 fällt auf, dass es offensichtlich keine Kombination von Symbolen gibt, die dieses Kriterium erfüllt. Unabhängig von dem Symbol, das ein Spieler wählt, gibt es immer ein Symbol, gegen das er damit verlieren wird. Der Gegner würde also seine Strategie anpassen wollen um ein Symbol zu wählen, mit dem er gewinnt. Es gibt kein Gleichgewicht, und daher auch keine Lösung.

In dem vorangegangenen Beispiel wurde unterstellt, dass sich ein Spieler immer für genau ein Symbol entscheiden muss und seine Strategie daher in der Wahl genau einer einzigen Handlungsalternative besteht. Man spricht in diesem Fall von einer *reinen Strategie*. Eine wichtige Erweiterung der Spieltheorie besteht in dem Konzept, dass man auch gemischte Strategien zulässt. Bei einer *gemischten Strategie* spielt ein Spieler jede Handlungsalternative mit einer bestimmten (positiven) Wahrscheinlichkeit. Die Auszahlung, die ein Spieler mit dieser Strategie erreicht, ist eine nach den Wahrscheinlichkeiten gewichtete Kombination der Auszahlungen, also der Erwartungswert der Auszahlung.

Definition 5 Gemischte Strategie und erwartete Auszahlung
Eine gemischte Strategie ordnet jeder möglichen Handlungsalternative $a_p \in \mathcal{A}_p$ eines Spieler eine bestimmte Wahrscheinlichkeit π zu, mit der er diese Alternative wählen wird.

$$s_p \in \mathcal{S}_p = \left\{ \begin{pmatrix} \pi_1 \\ \vdots \\ \pi_{|\mathcal{A}_p|} \end{pmatrix} \mid \sum_{i=1}^{|\mathcal{A}_p|} \pi_i = 1 \ \wedge \ \forall \pi_i : \pi_i \geq 0, \ i \in \{1, \ldots, |\mathcal{A}_p|\} \right\}$$

Die erwartete Auszahlung ergibt sich aus der Gewichtung der Einzelauszahlungen einer Handlungsalternative mit der Wahrscheinlichkeit, dass diese gespielt wird.

$$u_p(s_p|S_{-p}) = E\left(u_p|S_{-p}\right) = \sum_{i=1}^{|\mathcal{A}_p|} \pi_i \cdot u_p(a_{p,i}|S_{-p})$$

Das Konzept der Gleichgewichtsstrategien kann man nun analog auf gemischte Strategien ausweiten. Lässt man z.B. gemischte Strategien für das Spiel Schere-Stein-Papier zu, exisitert ein Gleichgewicht in der Strategiekombination $s_1 = s_2 = \left(\frac{1}{3} \text{ Schere}, \frac{1}{3} \text{ Stein}, \frac{1}{3} \text{ Papier}\right)$[7]. Allgemein kann man beweisen, dass es für jedes Spiel mit einer endlichen Menge an Handlungsalternativen eine Gleichgewichtsstrategie in gemischten Strategien gibt[8]. Bei strikt kompetitativen Spielen ist dieses Gleichgewicht obendrein eindeutig. Für eine mögliche Berechung des Gleichgewichts siehe Abschnitt 2.2.2.

2. Poker: Ein unvollkommenes dynamisches Spiel mit Zufall

Wie bereits in Abschnitt 1.2 dargestellt, handelt es sich beim Pokerspiel um ein unvollkommenes dynamisches Nullsummenspiel mit Zufallseinfluß. Für diese Klasse von Spielen

7. Die Strategien im Gleichgewicht müssen für alle Spieler gleich sein, da das Spiel symmetrisch ist.
8. Vgl. dazu z.B. (Nash, 1951).

stellt die Spieltheorie spezielle Werkzeug und Methoden zur Verfügung, die für eine exakte Darstellung und genaue Lösung verwendet werden können. Im folgenden Abschnitt 2.1 wird eine Darstellungsform für dynamische Spiele vorgestellt, die eine einfache und intuitive Interpretation ermöglicht, die extensive Darstellung. Im Anschluss daran werden in Abschnitt 2.2 klassische Lösungsverfahren beschrieben, mit denen eine exakte Lösung für die Klasse der unvollkommenen dynamischen Nullsummenspiel berechnet werden kann.

2.1 Extensive Darstellung

Die in Abschnitt 1.1 vorgestellte Normalform eines Spiels eignet sich nur sehr schlecht zur Darstellung eines dynamischen Spiels, da die Modellierung von zugbasierten Entscheidungen nicht vorgesehen ist. Eine intuitivere Darstellung kann durch einen Spielbaum erreicht werden, was auch als die *extensive Darstellung* eines Spiel bezeichnet wird. Dabei werden alle zu treffenden Entscheidungen durch *Knoten* in einem Baum repräsentiert, denen jeweils genau ein Spieler, nämlich der am Zug befindliche, zugeordnet wird. Die einzelnen Handlungsalternativen, die einem Spieler für jede seiner Entscheidungen zur Verfügung stehen, werden durch die ausgehenden *Kanten* im Baum repräsentiert. Die Auszahlungen an die Spieler notiert man in den *Blättern*, die das Spielende symbolisieren. Abbildung 1 zeigt einen einfachen generischen Spielbaum.

Ein Spiel beginnt immer an der Wurzel, wo der erste Spieler eine Entscheidung treffen muss und so die erste Verzweigung wählt. Im Anschluß trifft der nächste Spieler eine Entscheidung und verzweigt so tiefer im Baum. Dieser Prozess wiederholt sich, bis das Spiel schließlich beim Erreichen eines Blattes endet und die Spieler die entsprechenden Auszahlungen erhalten. Ein vollständig gespieltes Spiel ist also ein Pfad von der Wurzel des Baumes zu einem seiner Blätter. Die Folge von Entscheidungen, die zu diesem Verlauf geführt haben, nennt man eine *Spielinstanz*. Sie ist ein Ausschnitt aus den jeweiligen Strategien der Spieler.

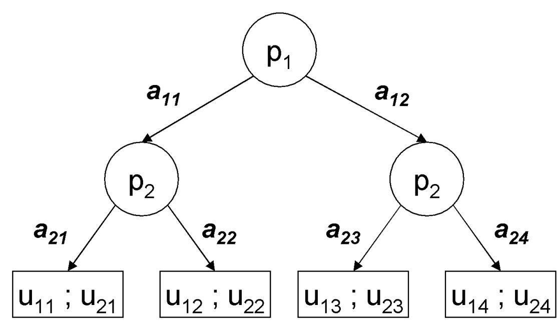

Abbildung 1: Ein einfacher generischer Spielbaum

Ein einfacher Spielbaum ist allerdings noch nicht ausreichend, um auch zufällige Ereignisse sowie unvollkommene Information zu modellieren. Dazu benötigt man weitere Konzepte, die durch einen *erweiterten Spielbaum* dargestellt werden. Für die Modellierung des Zufalls fügt man einen weiteren imaginären Spieler hinzu, gewissermaßen die Natur. Das Verhalten der Natur ist dabei in Form einer gemischten Strategie mit einer vorgegebenen Wahrscheinlichkeitsverteilung gegeben. Die jeweiligen Wahrscheinlichkeiten notiert man an den einzelnen Kanten, die die alternativen Zufallsausgänge repräsentieren.

Für die Modellierung von unvollkommenen Informationen muss zunächst geklärt werden, wodurch diese zustande kommen. Unvollkommene Informationen bedeuten, dass einem Spieler nicht alle Informationen über den aktuellen Status des Spiels zur Verfügung stehen. Der aktuelle Status wird dabei durch die bisherigen im Spiel getroffenen Entscheidungen repräsentiert und ist somit ein Pfad von der Wurzel des Baumes zum aktuellen Knoten. Liegt unvollkommene Information vor, so kennt ein Spieler nicht den aktuellen Status des Spiels und weiß daher nicht, an welchem Knoten er sich befindet um eine Entscheidung zu treffen. In einem Spielbaum modelliert man diese Situation durch *Informationsbezirke*, die eine Menge von Knoten des Spielbaums definieren, zwischen denen ein Spieler nicht unterscheiden kann. Ein Informationsbezirk ist dabei durch die Tatsache gekennzeichnet, dass er immer nur Knoten genau eines Spielers enthält und alle Knoten eine identische Kantenmenge haben, ein Spieler also dieselben Handlungsalternativen bei allen Entscheidungen eines Informationsbezirks hat. Für den Spieler selbst stellt ein Informationsbezirk logisch auch nur eine einzige Entscheidung dar. Er kann zwischen den verschiedenen Spielstati nicht differenzieren und trifft daher genau eine Entscheidung die von allen Knoten im Bezirk weiterverzweigt. Grafisch symbolisiert man Informationsbezirke in einem Spielbaum durch eine umschließende Linie[9].

Beispiel 2 *Wir definieren ein einfaches Pokerspiel für zwei Spieler S_1 und S_2. Der Grundeinsatz für jeden Spieler beträgt dabei eine Geldeinheit. Gespielt wird mit drei Karten, nämlich einem Buben(B), einer Dame(D) und einem König(K). Jeder der Spieler erhält eine Karte, die er seinem Gegner nicht zeigt. Nun entscheidet Spieler 1, ob er den Einsatz auf 2 Geldeinheiten erhöhen möchte. Wenn sich Spieler 1 für eine Erhöhung des Einsatzes entscheidet, hat Spieler 2 die Möglichkeit, entweder mitzugehen oder auszusteigen. Wenn er aussteigt, verliert er das Spiel und Spieler 1 gewinnt seinen Einsatz. Geht er jedoch mit, kommt es zur Aufdeckung und die Spieler legen ihre Karten offen auf den Tisch. Der Spieler mit der höheren Karte gewinnt. Wenn Spieler 1 nicht erhöht hat, kommt es direkt zur Aufdeckung.*

Beispiel 2 beschreibt eine einfache Variante des Pokerspiels[10]. In Abbildung 2 wird genau dieses Spiel durch einen Spielbaum dargestellt. Man erkennt, dass für Spieler 2 drei Informationsbezirke definiert sind (symbolisiert durch die grauen Kasten), an denen jeweils zwei Entscheidungsknoten zusammengefasst werden. Die Informationsbezirke entstehen dadurch, dass Spieler 2 nicht weiß, welche Karte Spieler 1 durch den Zufall im ersten Spielschritt zugeteilt worden ist. Er kann einzig die Karte, die er selber auf der Hand hält, als Ansatzpunkt nehmen, welche Karte Spieler 1 nicht haben kann.

Bemerkenswert ist, dass es für dasselbe Spiel mehrere strukturgleiche Spielbäume geben kann. Der Spielbaum in Abbildung 2 induziert eigentlich, dass Spieler 2 seine Karte erst bekommt, nachdem Spieler 1 bereits seine Entscheidung getroffen hat. Dies ist natürlich nicht der Fall. Dennoch ist der Spielbaum äquivalent zu dem des beschriebenen Spiels, da Spieler 1 über die Information, welche Karte Spieler 2 auf der Hand hat, kein Wissen erlangt. Durch den Trick, die Kartenvergabe für Spieler 2 erst als dritten Spielzug zu modellieren, spart man die Definition der Informationsbezirke für Spieler 1. Diese wären eigentlich notwendig, da ja auch Spieler 1 über unvollkommene Informationen verfügt.

9. Vgl. zur extensiven Form z.B. (Koller & Pfeffer, 1997, S. 6f).
10. Die vereinfachte Pokervariante basiert auf (Kuhn, 1950). Zu Darstellungszwecken wurde sie für diese Arbeit allerdings weiter vereinfacht.

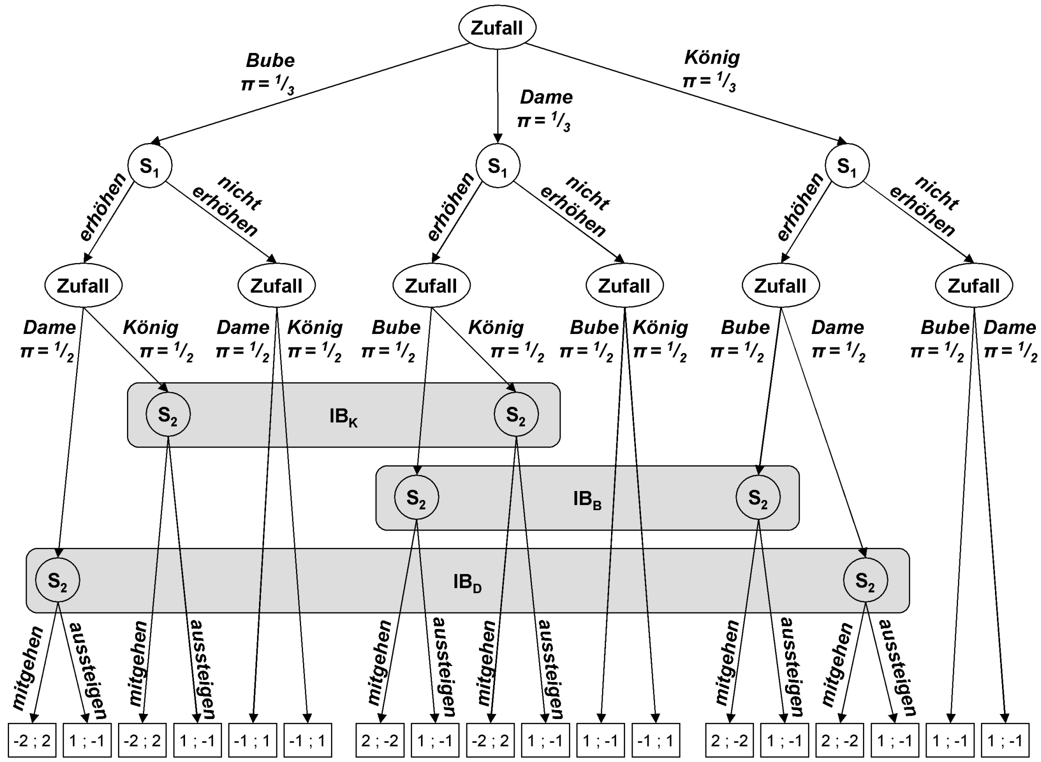

Abbildung 2: Vollständiger Spielbaum eines stark vereinfachten Pokerspiels

2.2 Klassische Lösungsverfahren

Für die Lösung eines unvollkommenen dynamischen Spiels mit Zufallsereignissen gibt es mehrere Lösungsmöglichkeiten[11]. Zunächst wird in Abschnitt 2.2.1 ein einfaches Verfahren vorgestellt, dass darauf basiert, Unsicherheiten aufgrund unvollkommener Informationen aus dem Spiel zu entfernen. Dadurch ist es möglich, ein baumbasiertes Lösungsverfahren für vollkommene Spiele anzuwenden, daß in sehr kurzer Zeit zu einer Lösung führt. Obwohl dieses Verfahren im allgemeinen nicht zu einer Lösung für alle unvollkommenen Spiele führt, kann es dennoch eingesetzt werden, um den zu durchsuchenden Lösungsraum von vorneherein einzuschränken. Ein deutlich aufwändigeres Verfahren, das dafür immer für unvollkommene Nullsummenspiele (und damit auch für Poker) eine exakte Lösung findet, wird im Abschnitt 2.2.2 vorgestellt.

2.2.1 Dominante Strategien und Rückwärtsinduktion

Ein Verfahren zur Ermittlung der Lösung eines vollkommenen dynamischen Spiels ist die *Rückwärtsinduktion*. Dabei analysiert man einen Spielbaum von den Blättern in Richtung Wurzel und antizipiert die Entscheidungen jedes Spielers. Man interpretiert dazu jeden Subbaum des Spielbaums als Teilspiel, in dem man separat nach einer Lösung sucht. Jedes Teilspiel, dass nur eine einzige Entscheidung enthält, ist direkt lösbar. Dies trifft also auf alle Knoten der untersten Ebene zu, wo auf die Entscheidung eines Spielers die Auszahlungen

11. Zur Definition einer Lösung siehe Abschnitt 1.3

folgen. Ein Spieler würde sich in so einer Situation aufgrund des Rationalitätsprinzip für genau die Handlungsalternative entscheiden, die ihm den größten Nutzen liefert.

Da der Ausgang eines Teilspiels auf der untersten Ebene also bekannt ist, kann man im Spielbaum das Teilspiel direkt durch die resultierenden Auszahlungen in Form eines Blattes ersetzen. Die Tiefe des Spielbaums verringert sich dadurch um eine Ebene und man kann dasselbe Vorgehen nun auf die darüberliegende Ebene anwenden. Am Ende des Verfahrens hat man den Spielbaum auf ein einziges Blatt reduziert, der die finale Auszahlung angibt. Die Entscheidungen, die man durch das Ersetzungsverfahren antizipiert hat, bestimmen dabei eindeutig den Spielverlauf[12]. Jeder Spieler hat dadurch eine reine Strategie, die sich im Gleichgewicht mit den Strategien seiner Mitspieler befindet. Dieses Ergebnis wurde erstmalig durch Ernst Zermelo bewiesen, der damit die eindeutige Lösung eines Schachspiel beschrieben hat[13].

In Spielen mit unvollkommener Information ist dieses Verfahren vom Prinzip her nicht anwendbar, da man bei der Auswertung eines Knotens, der sich in einem Informationsbezirk befindet, alle weiteren Entscheidungssituationen innerhalb des Bezirks mitberücksichtigen muss. Unter Umständen ist es aber dennoch möglich, die Entscheidung eines Spielers exakt zu antizipieren. Dazu untersucht man alle möglichen Auszahlungen, zu denen eine bestimmte Entscheidung führen kann. Gibt es eine Entscheidung, die in jedem Ausgang eine niedrigere Auszahlung liefert, als eine alternative Handlung, spricht man von einer *strikt dominierten* Handlung. Mathematisch formuliert wird eine Handlungsalternative $a_i \in \mathcal{A}_p$ innerhalb eines Informationsbezirk B strikt dominiert, wenn gilt: $\exists a_j \in \mathcal{A}_p . u_p(a_i|w_b) < u_p(a_j|w_b) \forall w_b \in B$, wobei w_b die Pfade zu den einzelnen Knoten des Informationsbezirks darstellt. Gibt es eine solche strikt dominierte Handlung a_i, sagt man auch a_j dominiert a_i und streicht die Alternative a_i aus dem Spielbaum, da ein Spieler aufgrund des Rationalitätsprinzips nie eine Handlung wählen wird, die ihm eine niedrigere Auszahlung als eine alternative liefert[14].

Man betrachte nun den Informationsbezirk IB_B von Spieler 2 aus dem vereinfachten Pokerspiel in Abbildung 2, der die Spielknoten gruppiert, bei denen der Spieler einen Buben auf die Hand bekommen hat. Vergleicht man die durch die Strategie Mitgehen möglichen Auszahlungen mit denen der Strategie Aussteigen, fällt sofort ins Auge, dass die Strategie Austieg in jedem Fall eine höhere Auszahlung liefert, als man durch ein Mitgehen erreichen könnte. In beiden Fällen verliert Spieler 2 zwei GE, wenn er die Erhöhung mitmacht, aber nur eine GE, wenn er das Spiel vorzeitig beendet. Dies ist intuitiv klar, da Spieler 2 genau weiß, dass er die niedirgste Karte des Spiels auf der Hand hat, und bei der anschließenden Auszahlung gar nicht gewinnen kann[15].

Analog kann man mit dem Informationsbezirk IB_K verfahren, bei dem Spieler 2 einen König auf die Hand bekommt. In diesem Fall dominiert die Strategie Mitgehen. Nachdem man die beiden Handlungsalternativen aus dem Spielbaum eliminiert hat, können die Informationsbezirke aufgelöst werden, da die Entscheidung von Spieler 2 durch die letzte verbliebene Alternative feststeht. Der Informationsbezirk IB_D kann auf diese Weise jedoch nicht eliminiert werden, daher ist eine vollständige Lösung durch Rückwärtsinduktion wei-

12. Vgl. zu Rückwärtsinduktion z.B. (Helm, 2007, Vl. 6, S. 3ff).
13. Für einen Beweis vgl. (Zermelo, 1929).
14. Zu dominierten Strategien vgl. z.B. (Osborne, 2004, S. 120f).
15. Man beachte, dass dieser Fall nur eintritt, da die Aufdeckung unmittelbar bevorsteht. (Koller & Pfeffer, 1997, S. 5) zeigen, dass auch Bluffen eine optimale Spielstrategie darstellen kann.

terhin nicht möglich. Jedoch kann man die Entscheidung, die Spieler 1 treffen wird, wenn er eine Dame auf die Hand bekommt, nun bestimmen. Da die erwartete Auszahlung für die Strategie Erhöhen bei -$\frac{1}{2}$ liegt, die bei der Strategie Nicht-Erhöhen jedoch bei 0, wird Spieler 1 grundsätzlich den Einsatz nicht erhöhen.

Abbildung 3 zeigt den Spielbaum nach der Eliminiation der dominierten Strategien. Außerdem wurden in dem Spielbaum bereits alle Zufallszüge, die direkt auf Auszahlungen verzweigen, durch ihre Erwartungswerte ersetzt.

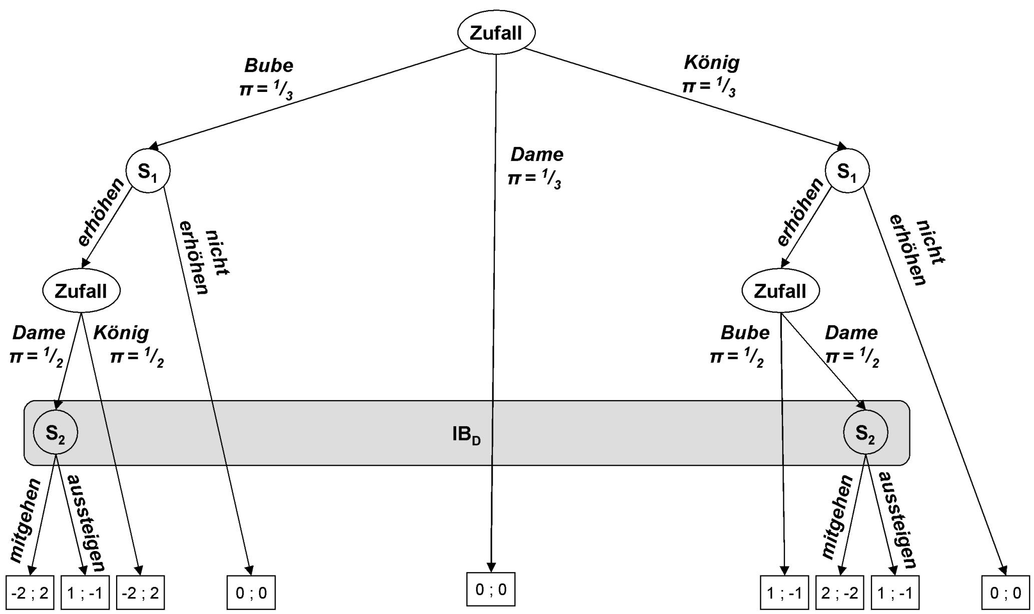

Abbildung 3: Spielbaum des vereinfachten Pokerspiels vollständig reduziert

2.2.2 Lösung durch den MaxiMin-Algorithmus

Ein Lösungsverfahren, das für unvollkommene dynamische Nullsummenspiele immer eine korrekte Lösung liefert, ist der MaxiMin-Algorithmus. Dabei versucht ein Spieler diejenige Strategie zu wählen, bei der er im schlechtesten Fall noch die höchste Auszahlung erhält. Um dies zu erreichen analysiert ein Spieler für alle eigenen Strategien, welche die minimale Auszahlung ist, die er mit dieser Strategie auf jeden Fall erreichen wird. Er wählt im Anschluss diejenige Strategie aus, für die er die höchste minimale Auszahlung erwartet, er *maximiert also seine minimale Auszahlung.*

Bei einem Nullsummenspiel, das ein strikt kompetitives Spiel ist, entspricht die Minimierung der Auszahlung des gegnerischen Spielers gleichzeitig der Maximierung der eigenen Auszahlung. Daraus folgt, dass jede Stragie, die den erwarteten Nutzen eines Spielers durch die minimal erwartete Auszahlung maximiert, auch gleichzeitig den Nutzen des Gegners bei Einhaltung der eigenen Strategie maximiert. Jede Strategie, die durch den MaxiMin-

Algorithmus in einem Nullsummenspiel ermittelt wird, stellt also eine Gleichgewichtsstrategie und damit eine Lösung des Spiels dar[16].

Betrachtet man nun auch gemischte Strategien[17] in einem Nullsummenspiel mit den Spielern p_1 und p_2, kann man das MaxiMin-Problem wie folgt mathematisch formulieren:

Definition 6 Lösung des MaxiMin-Algorithmus
Gesucht werden zwei gemischte Strategien s_1 und s_2 in einem Nullsummenspiel, die die Gleichgewichtsbedingung erfüllen. $\mathcal{A}_1$ und $\mathcal{A}_2$ seien dabei die Handlungsalternativen (reinen Strategien) der beiden Spieler, U_1 die Auszahlungsmatrix[18] für Spieler 1. Eine Lösung erfolgt durch das Optimierungsproblem:

$$\begin{array}{rl} \textit{Löse:} & \max_{s_1} \min_{s_2} (s_1^T U_1 s_2) \\ \textit{unter den Bedingungen:} & \sum_{i=1}^{|\mathcal{A}_1|} s_{1,i} = 1, \\ & \sum_{i=1}^{|\mathcal{A}_2|} s_{2,i} = 1, \\ & x \geq 0, y \geq 0. \end{array}$$

Bei dem in Definition 6 formulierten Optimierungsproblem handelt es sich um ein Problem der linearen Programmierung (LP), dass z.B. mit einem Simplex-Verfahren in polynomieller Zeit gelöst werden kann[19].

Damit man den MiniMax-Algorithmus auf ein Spiel in extensiver Form anwenden kann, muss man jedoch zunächst das Spiel in die Normalform umwandeln. Dafür trägt man die Kombination aller möglichen Handlungsoptionen, die ein Spieler im Spielverlauf hat, als seine Menge an reinen Strategien in der Normalformtabelle ab. Als Auszahlungen notiert man den jeweiligen Erwartungswert der Einzelauszahlungen, die durch die eine Zelle definierende Strategiekombination erreichbar sind. Diesen Erwartungswert erhält man also, wenn man alle durch eine Strategie erreichbaren Blätter das Baumes mit den Wahrscheinlichkeiten entlang des Pfades zu dem Blatt multipliziert[20].

Man betrachte dazu noch einmal das vereinfachte Pokerspiel aus Beispiel 2. Für die Umwandlung in die Normalform eignet sich der in Abschnitt 2.2.1 vereinfachte Spielbaum aus Abbildung 3 am besten. Zunächst werden alle Entscheidungenskombinationen gebildet, die die Spieler treffen können. Für Spieler 2 gibt es nur noch eine einzige Entscheidung zu treffen, für die er zwei mögliche Alternativen hat: Wenn er eine Dame auf der Hand hält, kann er entweder aussteigen oder mitgehen. Alle anderen möglichen Entscheidungen wurden bereits durch die Auswertung dominanter Strategien eliminiert. Wenn er einen König auf der Hand hält, wird er immer mitgehen, hat er hingegen einen Buben, wird er immer aussteigen. Spieler 1 hat hingegen noch zwei Entscheidungen zu treffen: Einmal, wie er sich mit einem Buben auf der Hand verhält und einmal, wie er bei einem König handelt. Er hat jeweils zwei Alternativen, also insgesamt vier (2×2) mögliche Strategien.

16. Für einen ausführlichen Beweis, dass eine MaxiMin-Strategie eine Gleichgewichtsstrategie darstellt, siehe (Osborne, 2004, S. 367f).
17. Zur Erläuterung gemischter Strategien siehe Abschnitt 1.3.
18. Die Auszahlungsmatrix entspicht der Normalformdarstellung eines Spiels, in der nur die Auszahlungen eines Spielers enthalten sind. Bei Nullsummenspielen gilt: $U_1 = -(U_1)^T$.
19. Zum Simplex-Algorithmus siehe z.B. (Domschke & Drexl, 2002, S. 20ff).
20. Für eine Beschreibung der Umwandlung von der extensiven Form in die Normalform vgl. (Koller & Pfeffer, 1997, S. 24f).

Die Berechnung der Erwartungswerte der Auszahlungsfunktion sei beispielhaft demonstriert an der Strategiekombination, an der Spieler 1 sowohl mit einem Buben, als auch mit einem König auf der Hand erhöht und Spieler 2 mit einer Dame auf der Hand mitgeht. Man betrachtet nun für alle erreichbaren Knoten, bzw. für alle möglichen Kartenkombinationen (BD, BK, D*, KB, KD) die Auszahlungen und gewichtet diese mit ihrer Wahrscheinlichkeit:

$$E(u_1(B_e, K_e; D_m)) = \frac{1}{6} \cdot -2 + \frac{1}{6} \cdot -2 + \frac{1}{3} \cdot 0 + \frac{1}{6} \cdot 1 + \frac{1}{6} \cdot 2 = -\frac{1}{6}$$

Tabelle 3 zeigt die Normalform des Spiels, auf der man nun direkt den MaxiMin-Algorithmus anwenden kann(angegeben sind nur die Auszahlungen für Spieler 1). Den eben berechnete Erwartungswert findet man links oben in der ersten Zeile und ersten Spalte wieder. Bemerkenswert ist, dass durch die Vereinfachungen des Spielbaumes in Abschnitt 2.2.1 erhebliche Einsparungen beim Bilden der Normalform erzielt worden sind. Für den nicht vereinfachten Spielbaum hätte man 36 (6×6) mögliche Strategiekombinationen untersuchen müssen, also die vierfache Menge.

Die Möglichkeiten der Reduktion sollten wenn möglich immer genutzt werden, da die Größe der Normalformmatrix exponentiell mit der Größe des Spiels (Runden $\times$ Handlungsalternativen) ansteigt. Für das vereinfachte Pokerspiel von Kuhn, das noch eine weitere Setzrunde zur Erhöhung durch Spieler 2 enthält, besteht die nicht reduzierte Matrix bereits aus 1728 Einträgen (27×64)[21].

		Spieler 2	
		D→mitgehen	D→aussteigen
Spieler 1	B→erhöhen; K→erhöhen	$-\frac{1}{6}$	$\frac{1}{6}$
	B→erhöhen; K→lassen	$-\frac{2}{3}$	$-\frac{1}{6}$
	B→lassen; K→erhöhen	$\frac{1}{2}$	$\frac{1}{3}$
	B→lassen; K→lassen	0	0

Tabelle 3: Normalform des reduzierten Spielbaums für Beispiel 2

Als Lösung für das modellierte vereinfachte Pokerspiel erhält man nach der Berechnung, dass Spieler 1 bei einem Buben nie und bei einem König immer erhöhen wird. Spieler 2 wird mit einer Dame auf der Hand immer aussteigen. Es handelt sich also um ein Gleichgewicht mit reinen Strategien. Diese Lösung erhält man auch direkt, wenn man die Normalform-Tabelle des Spiels untersucht. Es fällt ins Auge, das die dritte Strategie von Spieler 1 alle anderen Strategien dominiert, da sie für jede mögliche Strategie des Gegners immer die höchste Auszahlung liefert. Die erwarteten Auszahlungen für die Spieler liegen im Gleichgewicht also bei $\left(\frac{1}{3}; -\frac{1}{3}\right)$.

3. Verbesserte Lösungsstrategien

Nachdem wir im vorherigen Kapitel allgemeinere Konzepte einer spieltheoretischen Berechnung beschrieben haben, gehen wir nun auf weiterentwickelte aber auch speziellere Konzepte ein.

21. Vgl. (Kuhn, 1950).

3.1 Optimierung

Berechnungen für Spiele sind meistens sehr komplex. Der Rechenaufwand für reale Spiele ist enorm und oft gar nicht nur mit den bisher vorgestellten Methoden zu lösen. Es gibt viele Möglichkeiten die Berechnung an sich zu optimieren während das Ergebnis weiterhin eine exakte Lösung bleibt. Zwei weitreichende Methoden werden hier vorgestellt.

D. Koller und A. Pfeffer[22] haben bis 1997 die Sprache „Gala“ entwickelt um realistische und große Spiele unter einer spieltheoretischen Betrachtung analysieren zu können. In diesem Kapitel werden wir es häufiger für Beispiele verwenden.

3.1.1 Kompakte regelbasierte Darstellung

Wird einem Menschen ein Spiel beigebracht, so werden ihm in den meisten Fällen die Regeln erklärt. Bei Computern ist das häufig anders, hier werden Spielmodelle oft anders dargestellt, etwa mit Spielbäumen. Bei der Regelbasierten Darstellung wird ein System geschaffen, das normale Spielregeln versteht. Diese können oft schneller in linear zu lösende Probleme umgewandelt werden.

Wird das Spiel verändert, dann gewöhnlich durch neue oder veränderte Spielregeln. Diese lassen sich in vielen Fällen schneller in einer regelbasierten Darstellung umsetzen, weil der Entwickler keine ganzen Daten- und Baumkonstrukte per Hand abändern muss.

Man kann die Regeln in zwei Kategorien einteilen. In der ersten werden alle Objekte des Spieles wie etwa die Karten, die Spieler deklariert. In der zweiten Kategorie werden Handlungsabfolgen der Spieler und des Spielbetreibers (bei Poker der Dealer) beschrieben. Dies kann beispielsweise die Kartenausgabe und das Setzen eines Betrages sein.

Konkret wurde in unserer Beispiel-Programmiersprache Gala wie auch in vielen anderen Programmiersprachen eine Datenflusskontrolle *(flow control)* implementiert. Die drei wichtigsten Elemente davon heißen *choose*, *reveal* und *outcome*. Choose definiert jeweils einen Entscheidungspunkt, reveal ändert den Informationszustand der Spieler und outcome bestimmt den tatsächlichen Gewinn oder Verlust jedes Spielers. Die Syntax für Entscheidungspunkte bei Gala für den choose Ausdruck ist *chosse(Player, Template, Constraint)*, wobei ein Anwendungsbeispiel *choose(peter, InitialBet, between(0, $money(peter), Bet))* wäre. Man kann gut erkennen, dass der InitalBet zwei Voraussetzungen erfüllen muss: zum Einen muss der Betrag positiv und maximal soviel, wie der Spieler peter zu Verfügung hat, sein. Zum Anderen unterliegt das erste Setzen den allgemeinen Regeln eines Setzens *(Bet)*. Die Erklärung weiterer Beispiele die in Gala genutzt werden würde hier zu weit führen.

Den Vorteil von dieser regelbasierten Darstellung ist, dass in dem choose Ausdruck schon die Möglichkeiten als Antwort enthalten sind. Das Gleiche gilt analog für reveal und outcome. Schließlich kann man also sagen, dass diese Darstellung angenehm für die Softwareentwicklung ist und Rechenaufwand spart.

3.1.2 Sequentieller Lösungsalgorithmus

Das exponentielle Wachstum, das mit der Normalform verbunden ist, macht Standardlösungsalgorithmen für viele Spiele unrealistisch. 1994 wurde daher der sequentielle Lösungsansatz

22. Der Artikel (Koller & Pfeffer, 1997) beschreibt Gala ausführlich.

von Koller, Megiddo und von Stengel (Koller, Megiddo, & von Stengel., 1994, S. 750-759) entwickelt.

Der Algorithmus, der vermeidet den Rechenaufwand exponentiell ansteigen zu lassen, basiert auf strategischen Variablen. Statt Wahrscheinlichkeiten von Einzelaktionen wie in der extensiven Form, oder Wahrscheinlichkeiten von deterministischen Strategien - repräsentieren einzelne Realisationswerte ganze Abfolgen bzw. Sequenzen von Aktionen. Eine solche Sequenz eines Spielers kann anschaulich in einem Spielbaum als Pfad von der Wurzel vertikal bis in ein Blatt des Baumes angesehen werden.

Sei k ein Spieler und p ein Knoten des Spielbaumes, so gibt es genau einen Weg von der Wurzel zu p . Auf diesem Pfad kann es einige Entscheidungspunkte die der Spieler k zu entscheiden hat und wird als $\delta^k(p)$ definiert. In manchen Fällen ist eine solche Sequenz eine leere Menge, in allen anderen Fällen gibt $\delta^k(p)$ an, wie sich der Spieler zu entscheiden hat um zu p zu gelangen.

Wir beschreiben nun eine zufällige Spielsstrategie, die aus einer Menge von deterministischen Strategien besteht mit μ_k . Natürlich hängt das Erreichen des Knotens p auch von den Entscheidungen der anderen Spieler ab, daher berechnen wir die Wahrscheinlichkeit mit μ_k δ_k spielen zu können. Dies ist dann der Realisationswert und wir bezeichnen ihn mit $\mu_k(\delta_k)$.

Definition 7 Realisationswert
Wird eine Sequenz δ_k nach μ_k gespielt, so bezeichnet $\mu_k(\delta_k)$ den Realisationswert dieser. Hierbei ist k der Spieler und der Realisationswert gibt die Wahrscheinlichkeit an, mit der er den entsprechenden Informationsbezirk von δ_k erreicht.

Der Realisationsplan ist die Menge aller möglichen Realisationswerte $\mu_k(\delta_k^1)$, ..., $\mu_k(\delta_k^m)$ wenn δ_k^1 ... δ_k^m alle möglichen Sequenzen sind.

Wir suchen nun unter den Realisationswerten einen, der unsere Optimierungsbedingungen, etwa minimax, erfüllt. Koller und Megiddo führen weiter aus, dass mit der Matrix E , einem Vektor e und dem positiven Vektor x durch Lineare Optimierung $Ex = e$ die optimal Aktion berechnet werden kann.

Der Vektor x repräsentiert dabei eine zufällige Strategie die den Bedingungen der Matrix E und des Vektors e entsprechen. Die Bedingungen sind dabei aufaddierte Wahrscheinlichkeiten von einzelnen Entscheidungen. $Ex = e$ läßt sich als lineares Problem lösen.

Abschließend kann dies mit der intuitiven Spielweise eines Menschen verglichen werden. Etwa wenn der Spieler in der ersten Biet-Runde passen möchte und in der Letzten Bieten. Dann überlegt er sich, wie es möglich ist, dass dieser Fall eintreten kann, ohne dass die Gegner etwa durch ein Wegwerfen oder Erhöhen bis zum All-in. Der Spieler wählt dann die erste Strategie, die ihm einfällt und seinen Ansprüchen entspricht.

3.1.3 Resümee GaLa-System

Das in den vorhergehenden Kapiteln behandelte System „Gala“ hat den Anspruch Spiele, die in der Welt auch echt gespielt werden, unter spieltheoretischen Gesichtspunkten berechenbar zu machen. Dieser Anspruch wird auch weitgehend erfüllt. Die Entwickler D. Koller und A. Pfeffer haben bis 1997 unter Ausnutzung verschiedener Optimierungen Spielstrategien von verschienden Spielen berechnet. Gala baut auf der Logik Programmiersprache auf, dies wurde vorallem für die Regelbasierte Darstellung sehr passend gewählt.

Leider wird Gala nicht weiterentwickelt und die Autoren geben auf der ehemaligen Projekt-Homepage[23] an, dass das System vermutlich nie ein komplettes Poker-Programm (mit allen Bietrunden) berechnen werden kann.

Zur Entscheidung zwischen Handlungsalternativen greift jedoch Gala auf das noch weiterhin exsitierende Programm Gambit zurück.[24]

3.2 Approximation

In diesem Abschnitt werden Annäherung an gewünschte Optimums behandelt. Wie schon beschrieben ist es nötig Wege zu finden, die enorme Komplexität von spieltheoretischen Aufgaben in berechenbare Probleme umwandeln. Approximation bedeutet das wir die in geringem Maße von den gesuchten Werten abweichen, diese aber dafür in akzeptabler Zeit finden können.

Die Autoren Peter B. Miltersen und Troels B. Sorensen haben im Artikel „*A Near-Optimal Strategy for a Heads-Up No-Limit Texas Hold'em Poker Tournament*" (Miltersen & Sorensen, 2007) eine fast optimale Strategie für eine Poker Variante berechnet. Im Folgenden werden wir anhand dieser Variante drei Approximations Möglichkeiten erläutern.

Das realistische Beispiel gilt für *No-Limit Texas Hold'em Poker* mit zwei Spielern und wird auch bei Partypoker.com angeboten. Der Small Blind ist fest 300, der Big Blind 600. Insgesamt ist die Summe beider Spieler 8000, der jeweilige Stack muss jedoch nicht zu Beginn gleich groß sein. Wir betrachten den *Payoff* als 1 oder 0, je nachdem ob der betrachtete Spieler gewinnt oder verliert.

3.2.1 Grundsatz

Die zugrunde liegende Approximation besteht daraus, dass wir die Berechnung vereinfachen. Das Spiel an sich soll weiterhin die reale Spielsituation darstellen, wir schränken jedoch die Möglichkeiten unseres Spielers ein. Gegner können weiterhin nach den normalen Turnierregeln gegen diesen Spieler agieren.

Indem wir die Alternativenmenge verkleinern, verkleinert sich gleichzeitig die zu berechnende Aufgabe, ebenso der Spielbaum.

In unserem Turnier-Beispiel, auf das wir noch eingehen werden, wird dem Spieler nur noch erlaubt entweder die Hand weg zu werfen oder all-in zu gehen. Wir nennen diese Strategie wie auch im Artikel von Miltersen und Sorensen „*jam/fold*"-Strategie.

3.2.2 Knotengruppierung

Durch die eben angekündigte Regeleinschränkung muss nur noch die Pre-Flop Situation ausgewertet werden. Da der Spieler entweder die Hand weg wirft und damit die Runde beendet oder all-in geht und keiner weiteren Aktionsmöglichkeiten mehr in dieser Runde hat.

Dadurch müssen nur noch die Handkarten berechnet werden. Die Wertigkeit der Zahlen sowohl und ob sie von der gleichen Farbe sind, spielt eine Rolle. Dadurch erhält man 169 verschiedene Möglichkeiten. Da der eigentliche Spielbaum nicht diese Einschränkung hat,

23. Die Projekt-Homepage befindet sich unter: http://robotics.stanford.edu/ koller/gala.html

24. Eine freie Sammlung von Bibliotheken für Anwedungen der Spieltheorie http://gambit.sourceforge.net/

sondern nur der Spieler, haben wir alle möglichen Spielstrategien zu 169 so genannten Informationsbezirken zusammengefasst.

Eine zweite Approximation um die Berechnung noch mehr zu vereinfachen entsteht indem wir die Aktionen des Gegners nur in Kategorien einteilen. Der Gegner hat an sich die Möglichkeiten die Hand weg zu werfen, mit zugehen und um einen beliebigen Betrag zu erhöhen.

Da jeder unterschiedliche Betrag einen anderen Zustand generiert, und das für jeden einzelnen Informationsbezirk, ergeben sich über hunderttausend weitere Zustände. *In unserem Beispiel* wird die Aktion des Gegners daher nur in die Kategorie Fold und Check/Raise geteilt. Im Folgenden wird illustriert warum wir dadurch zwar die Knotenmenge stark verkleinern aber nicht schwächer gegen gute Spieler spielen:

Haben wir eine fast-optimale Strategie für den Spieler gefunden der auf das Mitgehen des Gegners reagiert, so befinden wir uns im Nash-Gleichgewicht. Diese Strategie liefert bei unserem jam/fold-Beispiel als Ergebnis, das der Spieler die Hand wegwerfen oder ihr all-in gehen soll.

Der Gegner muss nun ebenfalls mitgehen oder die Hand wegwerfen. Im ersten Fall macht es nun keinen Unterschied für unsere Auszahlung (Gewinn oder Verlust) mehr um wieviel der Gegner schon vor dem all-in erhöht hat. Im zweiten Fall wirft der Gegner die Karten weg – er verliert seinen Einsatz. Ein optimaler Gegner würde daher gegen unseren Spieler nur mitgehen aber nie zuerst erhöhen. Ein suboptimaler Gegner verliert einfach einen höheren Betrag gegen uns. Mit dieser Approximation wird unser Spieler nicht schlechter gegen (spieltheoretisch) optimale Gegner. Gegen suboptimale Gegner ist er überlegen aber reizt seine Möglichkeiten nicht ganz aus.

Diese Einschränkung führt dazu, dass der binäre Spielbaum, wenn der Spieler an der Position mit dem Big Blind sitzt, nur eine Tiefe von 3 besitzt, beim Small Blind eine Tiefe von 4.

Die letzte Approximation, die wir in diesem Artikel behandeln, fasst ebenfalls Knoten in Gruppen zusammen. Hierbei wird die Tatsache, dass die Spielstrategie von der Größe des Stacks eines Spielers abhängt betrachtet und Intervalle gebildet.

Als Grundvorraussetzung ist es wichtig zu verstehen, dass bei einem Turnier die Größe des stacks eine große Auswirkung auf die optimale Spielstrategie hat. Es erscheint intuitiv, dass es für einen Spieler mit der überwiegenden Mehrheit an Chips gut ist, etwas mehr zu riskieren um das Spiel zu beenden.

Die Spielstrategien ändern sich also auch bei gleichen Händen aber unterschiedlicher Chip Verteilung. Der Spielbaum hat daher für jede Hand und jeden stack eine optimale Strategie. Wir nähern uns an die optimalen Strategien an, indem wir den stack eines Spielers in Intervalle einteilen. Je mehr Intervalle es gibt, um so geringer ist die Abweichung der berechneten Strategie von der Optimalen.

In unserem Beispiel liegt die Summe der Chips, die beide Spieler zusammen haben, 8000, wir reduzieren den Berechnungsaufwand, indem wir Intervalle von 50 bilden. Daraus ergeben sich noch 158 nicht triviale Fälle. Zu jedem Fall wird eine Tabelle mit allen verschiedenen Starthänden gebildet, jeweils wird angegeben, ob der Spieler die Hand wegwerfen, all-in setzen oder eine zufällige Aktion durchführen soll.

Für die meisten Hände ist festgelegt, in welcher Situation sie gespielt werden. Wenige werden jedoch nur zu einem bestimmten Prozentsatz gespielt, als Beispiel dient die Starthand [sechs, acht] von verschiedenen Farben.

Die Autoren Koller und Pfeffer stellen dazu folgendes Theorem auf: Bei einer optimalen jam/fold Spielweise, definiert durch minimax, muss nur genau mit [6,8] verschiedener Farben mixed gespielt werden, wenn beide Spieler einen stack von 4000 haben. Dies läßt sich einfach beweisen, indem man die Strategie gegen die beiden Teilstrategien jam/fold als deterministische Strategie spielen lässt.

Eine Auffälligkeit dabei ist, dass ein Spieler der an der Position des Small Blind mit einer stack-Größe von 1800 die Starthand [3,4] von der gleichen Farbe wegwirft, [Bube, 2] jedoch spielt. Ist der stack größer und zwar bei 3600 ist dies genau umgekehrt.

Die rationale Erklärung ist leicht nach zu vollziehen, ein Gegner mit einer schlechten Hand (Trash-hand) würde bei einem größeren stack eher bei all-in mitgehen um das Spiel für sich zu entscheiden. Gegen eine Trash-hand ist besser mit dem Jack auf eine Highcard hoffen, als auf seltenere Strasse oder Flush.

3.2.3 Ober- und Unterschranken

Um eine Gewinnwahrscheinlichkeit zu berechnen - gehen wir wie folgt vor: Everett (Everett, 1957) hat in gezeigt, dass im rekursiven Spiel für alle Spielelemente „critical values“ berechnet werden können. Diese Werte sind analog zu den Werten einer minimax Berechnung von garantiert terminierenden Spielen und geben die Wahrscheinlichkeit an, zu der ein Spieler in seiner aktuellen Position mit einer optimalen Spielweise das gesamte Spiel gewinnt.

Häufig können jedoch keine optimalen Spielweisen berechnet werden, sondern nur Approximationen wie die weiter oben beschriebenen. Um nun die Gewinnwahrscheinlichkeit eines Spielers zu berechnen der nur eine fast-optimale Strategie spielt werden Ober- und Untergrenzen (*upper* bzw. *lower bound*) eingeführt. Die Obergrenze gibt an, um bis zu wie viel Prozent die eingesetzte Strategie verbessert werden kann, bis sie die Optimale Strategie wäre.

Damit haben wir das entscheidende Werkzeug an der Hand um zu vergleichen, wie gut verschiedene Spielweisen sind. Je kleiner die Prozentzahl der Abweichung ist, um so besser ist eine Strategie.

Um zurück zu Poker zu kommen: Ein Turnier mit festen Blinds findet unter Umständen kein Ende. Etwa wenn beide Spieler fortgehend ihre Hände wegwerfen. Daher sind die „critical values“ für uns von Bedeutung. Je nachdem wie groß der stack eines Spielers ist, um so größer ist die Wahrscheinlichkeit, dass er das Turnier gewinnen kann. Die „critical values“ geben diese Werte in Prozent an, wenn ein Spieler optimal spielen würde. Aber da es in diesem Artikel ja genau darum geht, dass die optimale Strategie nicht berechnet werden kann sind für Ober- und Untergrenzen für uns interessant.

3.2.4 Genauigkeitsverlust

Durch die Approximationen werden einige Probleme erst in angemessener Zeit berechenbar, dafür nähern sich die gefundenen Ergebnisse lediglich den exakt gesuchten Werten an.

Bei Poker sind Strategien mit *Starting Hand Charts* weit verbreitet. David Sklansky hat ein einfaches System „*The System*“ entwickelt, das ähnlich zu unserem Beispiel vorgibt, ob ein Spieler nichts oder all-in setzen sollte. Es werden jedoch nur die Starthände und Erhöhungen der Gegner betrachtet.

Sklansky meinte 2003 „I am extremly curious as to how strong it [the system] really might be. And if it is strong, how much stronger yet a more complex, move all-in system would be?“ (Sklansky, 2003, S. 14)

Eine verbesserte Version des Systems, nammte er dann *revised system*, dieses hat zusätzlich noch die Größe des Blinds und die Anzahl der Gegner in der Kalkulation. Berechnet man hierfür den Genauigkeitsverlust nach wie im vorherigen Kapitel beschrieben kommt man immer noch auf einen Wert von 5,9%.

Das ist deutlich deutlich schlechter als jam/fold aus unseren Beispielen mit 1,4% und zieht gerade einmal mit der Strategie bei jedem Zug all-in zu setzen bei unserem Turnier gleich.

Eine noch geringere Abweichung vom Optimum konnte mit dem System von Giplin und Sandholm (Gilpin & Sandhold, 2004, S. 160-169) erreicht werden. Hier konnte man die maximale Abweichung in Voraus angeben, der Rechenaufwand stieg dementsprechend bei sehr kleinen Werten. Außerdem waren die fast optimalen Strategien mittels Gleichgewichten nur für die einfache Rhode Island Poker Variante und auch nur für die ersten drei Biet-Runden berechenbar.

Daher stellen die Autoren Koller und Pfeffer die Frage, ob es nicht sinnvoller ist Systeme wie in unseren Beispielen zu entwickeln. Die Abweichungen sind zwar höher, etwa bei jam/fold 1,4%, aber taugen wenigstens für eine echtes und populär Spiel.

3.2.5 Resümee Jam/Fold

Das System von Koller und Avi nutzt viele Möglichkeiten aus, durch Approximation ein Partypoker.com Turnier nahezu optimal spielen zu können. Erstaunlich erscheint, das man selbst wenn man nur in der Pre-Flop Situation agiert nur mit maximal 1,4% von der Gewinnwahrscheinlichkeit der optimalen Strategien abweicht, die diese Einschränkung nicht haben. Das Verhältnis zwischen den Blinds und der Gesamtsumme an Chips macht die jedoch wieder greifbarer: alle Chips, also 8000, ergeben weniger als 14 Big Blinds.

Ein ähnlich optimales System für ein No-Limit Cash-Game ist wiederum um ein vielfaches komplexer und daher zur Zeit nicht berechenbar. Das Gleiche gilt für Limit-Poker, welches eine noch höhere Berechnungskomplexität hat.

4. Umsetzung bei der Poker Challenge

Die Variante Limit Texas Hold'em Tunier mit bis zu 6000 Händen werden wir nicht im spieltheoretischen Sinne optimal lösen können. Das Spiel ist zu komplex für eine vollständige Berechnung selbst mit allen möglichen Optimierungen werden wir noch zusätzlich auf Approximationen angewiesen sein.

Die bisher behandelten Beispiele und Poker Varianten weisen alle entscheidende Unterschiede zu der Poker Variante bei unserem Poker Challenge auf, in diesem Kapitel werden jedoch Lösungen und Hilfen abstrahiert und auf das AAAI Modell abgebildet.

4.1 Knoten des AAAI-Modells

Bei dem AAAI Multi-Table Limit-Turnier gibt es eine viele Faktoren welche die Knotenanzahl eines Poker Spielbaumes in die Höhe treiben.

Dazu gehören die Hand- sowie Tischkarten, die Anzahl der Spieler und sämtliche Handlungsmöglichkeiten, die Verteilung der Chips, und wie viele Hände noch gespielt werden.

Würde man nicht optimieren und approximieren so gäbe es für die verschiedenen Kartenkombination des Spielers mit Hand und Tischkarten alleine $169! - 162! \approx 3,4 \cdot 10^{15}$ Knoten. Die anderen genannten Faktoren würden ohne Optimierungen diese Zahl in eine Höhe katapultieren, jenseits der $3,4 \cdot 10^{26}$ Knoten. In heutigen Zeiten nicht berechenbar.

Im Gegensatz dazu haben Darse Billings et al. (Billings, Burch, A., Holte, Schaeffer, Schauenberg, & Szafron, 2003) gefolgert, dass man zur optimierten Berechnung von Texas Hold'em Limit Poker einen Spielbaum mit $\sim 10^{18}$ Knoten braucht um es ganz analysieren zu können. Billings ist unter anderem Entwickler der derzeit führenden Poker-Bot Software Polaris die verschiedene Strategien der aktuellen AAAI Poker Challenge Gewinnern implementiert.

4.2 Operationalisierungsansätze

Wir haben viele Optimierungen und Approximationen besprochen die auch für unsere Poker Challenge anwendbar sind. Der Schwerpunkt liegt eindeutig bei der Knotenmengenreduzierung. Aber auch bevor wir die Knoten festlegen und wenn wir sie auswerten gibt es gut Verbesserungsmöglichkeiten.

4.2.1 Sequentielle Form

Die besprochene sequentielle Form hilft uns, dass exponentielle Wachstum der Normal Form zu umgehen. Der Ansatz ist recht allgemein und kann vollständig auf unser Poker Turnier angewendet werden. Hiermit ist es möglich die Berechnung einfach in Lineare Gleichungssysteme umzuwandeln, was sicher das Ziel von den meisten unserer Poker-Bots sein wird.

4.2.2 Regelbasierte Darstellung

Daneben ist die regelbasierte Darstellung sehr interessant. Auch sie ist so weit offen gestaltet, dass sich die meisten Spiele mit ihr Modellieren lassen. Hat man ein Framework wie Gala, das die Regeln versteht und interpretieren kann, ist die Umsetzung unseres Limit Texas Hold'em Turnieres ein einfaches. Wählt man diese Darstellung für die eigene Software wird man vermutlich auf schon vorhandene Framework zurückgreifen. Ansonsten ist eine Implementierung in Java sinnvoll, die Teilaspekte davon umsetzt. Die Spielelemente lassen sich wegen Javas Objektorientierung gut als Klassen erstellen. Auch der Einsatz von choose, reveal und payout sollten abgewogen werden.

4.2.3 Evaluation der Strategien

Neben diesen Verbesserungsmöglichkeiten haben wir in diesem Artikel auch die Evaluation von Spielstrategien behandelt. Mit den angegebenen Schranken, der Abweichung vom Optimum, kann der Genauigkeitsverlust berechnet werden. So erhält man einen Überblick, wie gut oder schlecht die eigenen Strategien sind. Der Vergleich mit anderen Poker-Bots unserer

Gruppe oder des AAAI Wettbewerbs lässt sich hier auch einfach mit einer signifikanten Zahl anstellen.

4.2.4 Zusammenfassung von Knoten

Die Verdichtung der Information ist anschaulich und sehr effektiv. Die Spiele und Poker Varianten in den beschrieben Beispielen waren zwar alle sehr unterschiedlich zu unserem Multi-Table Limit Texas Hold'em Poker Turnier, dennoch lassen sich viele Aspekte auf unsere Challenge übertragen.

Die Ausnutzung von Symmetrien erscheint fast schon selbst verständlich. Die Farben haben beispielsweise bei Poker keine unterschiedlichen Bedeutungen und es gibt auch keine Rangfolge. Etwa bei der jam/fold Variante wurde wurden die Starthände in Informationsbezirke geteilt. Die Information ob die zwei Karten von der gleichen Farbe sind und ihre Wertigkeit reichten aus. Dies gilt auch für unsere Spielvariante für den Pre-Flop.

Die Einteilung von Aktionen in Kategorien sollten wir ebenso berücksichtigen. Für uns ist es möglich häufig wiederholte reraises von Gegnern in Bereiche einzuteilen.

Genauso können wir auch bei der Betrachtung der Größe der gegnerischen Stacks Intervalle bilden, wie es am Beispiel beschrieben wurde. Da neben gibt es natürlich noch eine Reihe andere Möglichkeiten Knoten zusammenzufassen.

Allgemein läßt sich sagen, dass wir zur Berechnung von komplexen Spielen wie unserem Turnier eine gute Kontenverdichtung durch Optimierung und Approximation genauso, wie ein effizienter Berechnungsalgorithmus brauchen.

References

Billings, D., Burch, N., A., D., Holte, R., Schaeffer, J., Schauenberg, T., & Szafron, D. (2003). Approximating game theoretic optimal strategies for full-scale poker. In *18th International Joint Conference on AI*, Vol. 18.

Domschke, W., & Drexl, A. (2002). *Einführung in Operations Research* (5. edition). Springer (Berlin Heidelberg New York).

Everett, H. (1957). Recursive games. In *Contributions to the Theory of Games Vol. III*, Vol. 39.

Gilpin, A., & Sandhold, T. (2004). Finding equilibria in large sequential games of incomplete information. *Electronic Commerce*, *94*, 160–169.

Helm, C. (2007). Skript zur vorlesung: Angewandte spieltheorie..

Koller, D., Megiddo, N., & von Stengel., B. (1994). Fast algorithms for finding randomized strategies in game trees. In *26th Annual ACM Symposium on the Theory of Computing*, Vol. 26.

Koller, D., & Pfeffer, A. (1997). Representations and solutions for game-theoretic problems. *Artificial Intelligence*, *94*, 167–215.

Kuhn, H. W. (1950). A simplified two-person poker. In Kuhn, H. W., & Tucker, A. W. (Eds.), *Contributions to the Theory of Games I*. Princeton University Press (Princeton).

Miltersen, P. B., & Sorensen, T. B. (2007). A near-optimal strategy for a heads-up no-limit texas hold'em poker tournament. In *6th international joint conference on Autonomous agents and multiagent systems*, Vol. 6.

Nash, J. F. J. (1951). Non-cooperative games. *Annals of Mathematics*, *54*, 286–295.

Osborne, M. J. (2004). *An introduction to Game Theory*. Oxford University Press (New York).

Sklansky, D. (2003). The system. *Card Plaxer Magazine*, *94*.

von Neumann, J., & Morgenstern, O. (1944). *Theory of Games and Economic Behavior*. Princeton University Press (Princeton).

Zermelo, E. F. F. (1929). Die berechnung der turnier-ergebnisse als ein maximumproblem der wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, *29*, 436–460.