Technische Universität Berlin

Bachelorarbeit

zur Erlangung des akademischen Grades
Bachelor of Science (B.Sc.)

Entwicklung einer Strategie für das Mehrpersonenspiel mit Zufallselementen No-Thanks

Dorothee Spitta
Matrikelnr.: 325500

3. August 2012

Fakultät IV: Elektrotechnik und Informatik
Institut für Wirtschaftsinformatik
und Quantitative Methoden
Fachgebiet Agententechnologien in betrieblichen
Anwendungen und der Telekommunikation
Prof. Dr. Sahin Albayrak
Betreuung durch: Dr. Stefan Fricke

Eidesstattliche Erklärung

Ich versichere, die vorliegende Bachelorarbeit selbständig und lediglich unter Benutzung der angegebenen Quellen und Hilfsmittel verfasst zu haben.

Ich erkläre weiterhin, dass die vorliegende Arbeit noch nicht im Rahmen eines anderen Prüfungsverfahrens eingereicht wurde.

Berlin, 3. August 2012

(Dorothee Spitta)

Zusammenfassung

In der vorliegenden Bachelorarbeit wird eine Simulationsumgebung entwickelt, die Spielstrategien für das Kartenspiel No-Thanks evaluieren und deren Leistungsfähigkeit durch maschinelle Lernverfahren erhöhen kann. Bei No-Thanks handelt es sich um ein Mehrpersonen-Kartenspiel mit vollständiger Information und Zufallselementen. Aufgrund seiner Komplexität sind unterschiedlich komplexe Spielertypen und Spielstrategien möglich. Neben der Entwicklung der Simulationsumgebung mitsamt ihren Designentscheidungen und einer spieltheoretischen Betrachtung des Kartenspiels werden in dieser Arbeit ausgewählte Lösungsansätze aus dem Gebiet der Spieltheorie und des Maschinellen Lernens vorgestellt und mithilfe von Beispielen verdeutlicht. Für die Entwicklung der Spielstrategien wird hierbei sowohl die Generierung eines Spielbaums mit Zufallsknoten (Chance Nodes) als auch das Lernen von Parametern einer Bewertungsfunktion unter Verwendung eines Gradientenabstiegsverfahrens untersucht. Die auf diesen beiden Verfahren basierenden Spielstrategien werden schließlich experimentell im Simulator ausgewertet und die Ergebnisse vorgestellt.

Abstract

In this Bachelor thesis, a simulation environment is developed to evaluate game strategies for the game No-Thanks. No-Thanks is a multi-player card game with complete information and chance elements. Due to the game's complexity, a variety of different player types and game strategies exist. Besides the development of the simulation framework with its design choices and a game theoretic analysis of the card game, this work presents a selection of possible game strategies, describing paradigms from game theory and machine learning and using examples for illustration. The game strategies presented in this work take advantage of game trees with chance nodes and apply online parameter learning of a heuristic function by gradient descent. The strategies and learning success are finally evaluated by the simulator and the results are presented.

Inhaltsverzeichnis

Kapitel 1

Einleitung

In dieser Arbeit wurde eine Simulationsumgebung für das Mehrpersonen-Kartenspiel No-Thanks entworfen und implementiert. Die Motivation sowie die Vorgehensweise der Arbeit sollen im Folgenden kurz beschrieben werden.

1.1 Problembeschreibung

Bei No-Thanks handelt es sich um ein Kartenspiel für zwei oder mehr Personen, bei dem die Reihenfolge der Spielkarten und die gegnerischen Spielstrategien den Spielern nicht bekannt sind. Um Spieler in die Lage zu versetzen, in Spielsituationen möglichst optimale Entscheidungen zu treffen, können Methoden aus der Spieltheorie und dem Maschinellen Lernen angewandt werden.

Neben der Möglichkeit, dass Spieler in der Lage sind, das Spiel bis zu einer festgelegten Anzahl von Spielzuständen vorauszuplanen, stellt das Lernen von Bewertungen von Spielzuständen einen weiteren Aspekt zur Entwicklung von Spielstrategien dar. Die Aufgabe des Simulators ist daher, unterschiedliche nicht-lernende und lernende Spielstrategien im Spiel gegeneinander antreten zu lassen und die Spielresultate auszuwerten. Dabei soll der Einfluss des Zufalls untersucht werden, um die Spielstärke einer Strategie von diesem abgrenzen zu können.

1.2 Verwandte Arbeiten

Die in dieser Arbeit genutzten Verfahren sind aus dem Gebiet der Künstlichen Intelligenz, der Spieltheorie sowie dem Maschinellen Lernen bekannt und wurden bereits für die Analyse anderer Spiele genutzt.

Für Zweipersonen-Nullsummenspiele wie Schach wird typischerweise der Min-Max-Algorithmus [von Neumann(1928)], zusätzlich in Verbindung mit

Bewertungsfunktionen genutzt. Hierbei ist insbesondere auch eine Vorausschau über möglichst viele Spielzüge von großer Bedeutung.

Im Gegensatz dazu besitzen die Spieler beim Poker nur unvollständige Information. Zusätzlich dazu spielt auch hier der Zufall eine wichtige Rolle und die einzelnen Spieler versuchen, sich gegenseitig über die Güte ihrer Karten zu täuschen. Ein Ansatz für einen simulierten Pokerspieler findet sich in der Arbeit von Billing et. al. [Billings et al.(2001)Billings, Davidson, Schaeffer, and Szafron].

Der mit diesen und weiteren komplexen Problemstellungen einhergehende hohe Berechnungs- und Speicheraufwand hat zur Entwicklung von probabilistischen Verfahren wie Monte-Carlo-Simulationen [Neal(1993)] sowie Reinforcement-Learning-Algorithmen [Sutton(1988)] geführt. Zu beiden Verfahrensklassen gibt es eine Vielzahl Arbeiten, die sich mit Erweiterungen und Optimierungsmöglichkeiten beschäftigen.

Ein bereits in einer Vielzahl von Arbeiten untersuchtes Spiel, das wie No-Thanks ein Spiel mit vollständiger Information und Zufallselementen darstellt, ist das Brettspiel Backgammon. Als besondere Arbeit ist die des TD-Gammon von Gerry Tesauro [Tesauro(2002)] zu erwähnen. In dieser Arbeit konnte ohne vorausschauendes Spiel mithilfe von Künstlichen Neuronalen Netzen, Reinforcement-Learning und anhand von Trainingsspielen der simulierten Spieler gegeneinander bereits Expertenniveau erreicht werden. Durch die Erweiterung einer geringen Vorausschau (shallow lookahead) stellt die so generierte Spielstrategie eine der drei leistungsfähigsten Backgammon-Programme dar.

1.3 Zielstellung und Beitrag der Arbeit

Ziel dieser Arbeit war die Erstellung einer Simulationsumgebung zur Entwicklung und Evaluierung von Spielstrategien für das Spiel No-Thanks.

Im Einzelnen lässt sich der Beitrag der Arbeit in folgenden Punkten zusammenfassen:

- Entwicklung einer Simulationsumgebung für das Mehrpersonenspiel No-Thanks
- Gleichzeitige Analyse verschiedener Spielstrategien mithilfe der Simulationsumgebung
- Unterstützung von mehrfacher Zugvorausschau für die Zustandsbewertung
- Implementierung eines Lernverfahrens zum Lernen der Heuristikparameter

Die Arbeit ist wie folgt gegliedert:

Zunächst gibt **Kapitel 2** einen Überblick über die genutzten Grundlagen der Spieltheorie und die für diese Arbeit relevanten Verfahren aus dem Gebiet des maschinellen Lernens. In **Kapitel 3** erfolgt eine detaillierte Beschreibung des Spiels. Im Anschluss daran befasst sich **Kapitel 4** mit dem im Rahmen der Arbeit entwickelten Simulator. **Kapitel 5** beschreibt die Experimente, die für die Arbeit ausgewählt wurden und wertet diese aus. Schließlich fasst **Kapitel 6** die Arbeit zusammen und zeigt Ideen für weiterführende Arbeiten auf.

Kapitel 2

Grundlagen

2.1 Spieltheoretische Grundlagen

Die Spieltheorie beschäftigt sich mit der Modellierung von Entscheidungssituationen, insbesondere von Situationen, in denen ökonomische Akteure miteinander interagieren. Sie ist eng mit der Entscheidungstheorie verknüpft, bei der das Entscheidungsverhalten rationaler Teilnehmer an Prozessen analysiert wird. Im Unterschied zur Entscheidungstheorie sind die Konsequenzen von Entscheidungen nicht nur von eigenen Entscheidungen, sondern auch von den Entscheidungen anderer am Prozess beteiligter Akteure abhängig [Neumann and Morgenstern(1961)].

Die Spieltheorie ist in vielen wissenschaftlichen Disziplinen vertreten. In der Wirtschaftswissenschaft beschäftigt sich die strategische Spieltheorie beispielsweise mit der Analyse von Verhandlungen, Wettkämpfen und Spekulationen auf Märkten. In der Biologie dient die Spieltheorie der Analyse von Verhalten und Entwicklungen von Organismen oder Lebensformen von Populationen [Maynard Smith and Price(1973)]. Auch in der Informatik ist die Spieltheorie für die Entwicklung intelligenter dezentraler Agenten von großer Bedeutung, für einen Überblick sei [Halpern(2007)] empfohlen.

Eine grundlegende Annahme der Spieltheorie ist, dass die Akteure rational handeln und primär an eigener Nutzenmaximierung interessiert sind (siehe dazu [Sieg(2010)] S. 133, für eine genauere Betrachtung des Nutzenbegriffs siehe [Neumann and Morgenstern(1961)] S. 15ff). In der Spieltheorie wird der Erwartungsnutzen (der Bernoulli-Nutzen der Ereignisse) maximiert, betrachtet werden aber auch die Risikopräferenzen der Spieler.

Im Folgenden wird zunächst eine Definition eines Spiels gegeben. Anschließend werden verschiedene Gegenüberstellungen von Einteilungen von

Spielen beschrieben und wichtige Algorithmen zur Analyse von Spielen vorgestellt.

2.1.1 Definition eines Spiels

Der Begriff Spiel bezeichnet in der hiesigen Arbeit eine konkurrierende Multiagentenumgebung.

Zu einem Spiel gehören also normalerweise nach [Neumann and Morgenstern(1961)] und [Russell and Norvig(2004)] (S. 213):

- die Menge der beteiligten Spieler,
- die Reihenfolge, in der die Spieler am Zug sind,
- die Zustandsbeschreibung, u.a. mit der Verteilung von Material und der Information, welcher Spieler gerade am Zug ist,
- alle Aktionsmöglichkeiten der Spieler bei jedem Zug,
- die Informationen, die das Wissen der Spieler beschreiben,
- die Beschreibung des Startzustands,
- der Test auf einen Endzustand,
- zum Ende des Spiels die aus den Aktionen resultierenden Konsequenzen, der Nutzen, der sich oftmals mithilfe einer Nutzenfunktion (einer numerischen Bewertung der Endzustände) ergibt. Im weiteren Verlauf dieser Arbeit wird hierfür auch der Begriff Auszahlung verwendet.

Diese Definition bezieht sich auf extensive Spiele, deren Beschreibung im folgenden Abschnitt folgt.

2.1.2 Einteilung von Spielen

Zur Vergleichbarkeit ihrer Eigenschaften können Spiele mithilfe folgender Kriterien eingeordnet werden.

Null- und Konstantsummenspiele

Nullsummenspiele sind Spiele, bei denen die Summe der Auszahlungen aller Spieler in jedem möglichen Spielausgang stets den Wert null ergibt [Sieg(2010)] (S. 29). Nullsummenspiele sind ein Spezialfall von Konstantsummenspielen, bei welchen die Summe der Auszahlungen lediglich konstant sein muss.

Statische und dynamische Spiele

Statische Spiele sind solche, bei denen die Spieler einmal simultan entscheiden, aber die gleichzeitige Entscheidung der anderen Spieler nicht wahrnehmen können.

In dynamischen Spielen, auch sequentielle oder extensive Spiele genannt, entscheiden die Spieler nicht nur einmal simultan, sondern gegebenenfalls mehrmals. Auch eine nacheinander erfolgende Entscheidungsfindung der Spieler bezeichnet ein dynamisches Spiel ([Sieg(2010)] S. 39). Zur grafischen Veranschaulichung wird ein sogenannter Spielbaum herangezogen, dessen Darstellung auch extensive Form genannt wird.

Ein-, Zwei- und Mehrpersonenspiele

Bei Einpersonenspielen gibt es nur einen Spieler, der zwischen einigen Handlungsalternativen entscheidet und sich dementsprechend lediglich in einer Entscheidungssituation mit unterschiedlichen Auszahlungen befindet.

An Zwei- oder Mehrpersonenspielen nehmen zwei oder mehr Spieler teil, welche jeweils definierte Aktionsmöglichkeiten haben. Die Auszahlung von Endzuständen beziehungsweise genereller die Bewertung von Zuständen kann in diesem Fall als ein Vektor dargestellt werden. Bei Null- oder Konstantsummenspielen mit mehreren Spielern ist es möglich, die Auszahlung eines Spielers im Vektor wegzulassen, da sie sich durch die Auszahlungen der anderen Spieler ergibt.

Oftmals wird bei Mehrpersonenspielen angenommen, dass Spieler an ihrer eigenen Nutzenmaximierung interessiert sind, die Zielstellung für den Nutzen kann theoretisch aber durchaus variieren. Beispielsweise kann das Kooperieren für ein gemeinsames Ziel angestrebt werden. Eine detaillierte Betrachtung von Mehrpersonenspielen ist in [Neumann and Morgenstern(1961)] beschrieben worden.

Deterministische und nichtdeterministische Spiele

Viele Spiele wie beispielsweise das Schachspiel basieren auf einem deterministischen Modell. Werden jedoch Spieler, Auszahlungen oder Aktionen vom Zufall bestimmt, spricht man von nichtdeterministischen Spielen.

Je nachdem, ob die den Spielern nicht bekannten Information anderen Spielern oder lediglich der Natur bekannt sind, wird außerdem zwischen Spielen mit vollständiger ([Neumann and Morgenstern(1961)] S. 112ff und S. 627ff) und unvollständiger Information unterschieden. Verfügt lediglich die Natur über die unbekannte Information, spricht man von Spielen mit Zufallselementen.

Folgende Beispiele sind bekannte Illustrationen für die beschriebenen vier Spieleinteilungen:

- Vollständige Information ohne Zufallselemente: Schach
- Vollständige Information mit Zufallselementen: Mensch ärgere dich nicht
- Unvollständige Information ohne Zufallselemente: Schiffe versenken
- Unvollständige Information mit Zufallselementen: Poker

Als Besonderheit wird außerdem zwischen Spielen mit verrauschter bzw. unverrauschter Informationen unterschiede. Dabei wird untersucht, ob die Informationen - eventuell nach vorheriger Kommunikation - korrekt und vollständig gesendet, übermittelt und empfangen wurden. In dieser Arbeit wird von grundsätzlich unverrauschter Information ausgegangen.

2.1.3 Spielbaumalgorithmen

Zur Analyse von Spielsituationen gibt es verschiedene Algorithmen. Im folgenden werden kurz die für diese Arbeit relevanten Algorithmen vorgestellt.

Extensive Spiele mit vollständiger Information ohne Zufallselemente

Bei Zweipersonen-Konstantsummenspielen kann der Min-Max-Algorithmus [Russell and Norvig(2004)] (S. 214) angewendet werden. Er ist optimal für deterministische Spiele bei vollständiger Suche und ist daher nur für überschaubare Suchbäume einsetzbar. Seine Zeitkomplexität beträgt $O(b^m)$ und seine Speicherkomplexität bei Tiefensuche beträgt $O(b \cdot m)$, wobei m die Suchtiefe und b den Verzweigungsgrad bezeichnet.

Ähnlich verhält sich die Berechnung des teilspielperfekten Gleichgewichts (Subgame Perfect Equilibriums), welches bei Mehrpersonenspielen für jeden Spielzustand beim Propagieren der Auszahlungen jeweils die Aktion wählt, die dem aktuellen Spieler den größten Nutzen verschafft. Dabei wird vorausgesetzt, dass Spieler die selben Annahmen über die Auszahlungen (den Nutzen einer Folge von Aktionen eines jeden Spielers am Ende eines Teilspiels) besitzen und zudem davon ausgehen, dass jeder Spieler seine Auszahlungen maximieren möchte. In diesem Zusammenhang wird der MaxN-Algorithmus [Luckhart and Irani(1986)] genannt.

Auch der MaxN-Algorithmus ist linear bezüglich der Anzahl der zu überprüfenden möglichen Züge. In der Regel werden mit steigender Suchtiefe demnach exponentiell viel Zeit und Speicher benötigt.

Mögliche Verbesserungen der Komplexität ergeben sich durch verschiedene Techniken wie das Kürzen von Teilbäumen (Pruning) [Hauk(2004)] [Hauk(2006)] von beispielsweise hoffnungslosen oder bereits berechneten Zügen. Durch dynamisches Programmieren kann die Komplexität zusätzlich verringert werden, beispielsweise kann durch die Nutzung von Transpositionstabellen eine Zeitersparnis durch die Speicherung bereits untersuchter Zustände und deren Bewertungen erfolgen.

Extensive Spiele mit Zufallselementen

Für extensive Spiele mit Zufallselementen kommt die Einführung von Zufallsknoten (Chance Nodes) hinzu. Der nach oben im Baum zu propagierende Wert der Zufallsknoten ergibt sich aus dem Erwartungswert des Wertes der Kindknoten [Russell and Norvig(2004)] (S. 228-231).

Der Expectimax-Algorithmus (ExpMx) ist eine Erweiterung des Min-Max-Algorithmus, der diesen durch die Einführung von Zufallsknoten ergänzt. Seine rekursive Berechnungsformel des Expectimax-Wertes eines gegebenen Knotens n ist in 2.1 gegeben. max und min beschreiben hierbei die Maximum- und Minimum-Funktion, $Nutzen(n)$ ermittelt die Auszahlung eines Endknotens n und $Nachfolger(n)$ liefert alle Nachfolgerknoten des Knotens n.

$$ExpMx(n) = \begin{cases} Nutzen(n), & \text{falls } n \text{ Endknoten} \\ max_{s \in Nachfolger(n)} ExpMx(s), & \text{falls } n \text{ Max-Knoten} \\ min_{s \in Nachfolger(n)} ExpMx(s), & \text{falls } n \text{ Min-Knoten} \\ \sum_{s \in Nachfolger(n)} P(s) \times ExpMx(s), & \text{falls } n \text{ Zufallsknoten} \end{cases} \tag{2.1}$$

Bei einem Verzweigungsfaktor der Zufallsknoten n, einer Suchtiefe m und dem sonstigen Verzweigungsfaktor b liegt die Zeitkomplexität von ExpectiMax bei $O(b^m \dot{n}^m)$

Der ExpectiMaxN-Algorithmus ist auch hier analog zur Bestimmung des teilspielperfekten Gleichgewichts anwendbar: An Knoten, die keine Zufallsknoten sind, wird der aktuelle Spieler die Aktion wählen, die ihm den meisten Nutzen bringt - auch hier wieder mit der Annahme, dass alle anderen Spieler genauso vorgehen.

Es existieren wie beim normalen MaxN-Algorithmus verschiedene Verbesserungen, beispielsweise mithife von Kürzung (Pruning) als eine Erweiterung der Alpha-Beta-Kürzung, bei der mithilfe der Einführung von Intervallen irrelevante Pfade identifiziert und gekürzt werden können. Für einen Über-

blick sei die Arbeit von [Schadd et al.(2009)Schadd, Winands, and Uiterwijk] empfohlen.

Eine andere Herangehensweise zum teilweisen Lösen des Komplexitätsproblems ist die Nutzung von Bewertungsfunktionen, die nun im Folgenden erörtert werden soll.

2.1.4 Heuristiken und Bewertungsfunktionen

Die Berechnung optimaler Aktionen in Echtzeit ist aufgrund begrenzter Ressourcen oftmals schwierig. Als Lösung bietet es sich beispielsweise an, den Suchraum zu beschränken (z.B. bis zu einer gewissen Tiefe im Spielbaum vorauszuschauen) sowie Bewertungsfunktionen zu verwenden, welche die Günstigkeit eines Spielzustandes abschätzen und dadurch Heuristiken darstellen [Russell and Norvig(2004)] (S. 222ff).

Zu den Anforderungen und Vorteilen von Bewertungsfunktionen zählt, dass Bewertungsfunktionen schnell berechenbar sind und die Günstigkeit von Zuständen dennoch in der richtigen Ordnung wiedergeben. Daher werden Bewertungsfunktionen in vielen Arbeiten verwendet, die sich mit Entscheidungen unter begrenzten Systemressourcen beschäftigen. Auch für viele untersuchte Spiele wie z.B. Schach wurde häufig eine lineare gewichtete Summe von bestimmten Spielmerkmalen verwendet.

Allerdings haben Bewertungsfunktionen den Nachteil, dass sie unsicher sind, da sie nur mit beschränkten Systemressourcen arbeiten. Auch das Aufstellen von linearen Bewertungsfunktionen bedeutet nicht immer, dass die identifizierten Merkmale, die in die Bewertungsfunktion einfließen, unabhängig voneinander sind. Desweiteren können sich Gewichte in unterschiedlichen Spielphasen verändern.

Das Finden einer geeigneten Bewertungsfunktion kann durch maschinelle Lernverfahren unterstützt werden. Im nächsten Abschnitt sollen daher einige wichtige Lernverfahren kurz erörtert werden.

2.2 Grundlagen des Maschinellen Lernens

Im Gebiet des Maschinellen Lernens geht es darum, neues Wissen für zukünftige zu klassifizierende Daten oder anzunähernde Funktionen mithilfe von vorherigen Trainingsbeispielen zu generieren. Ein Schwerpunkt liegt dabei darin, bessere Entscheidungen basierend auf den bereits beobachteten Daten zu treffen und die Daten adäquat zu generalisieren.

Ein maschinelles Lernproblem benötigt nach [Mitchell(1997)] eine Lernaufgabe, ein Performanz-Maß und eine Trainingsquelle. Beim Entwickeln

und Anwenden von Lernverfahren ist es nötig, zuvor bestimmte Design-Entscheidungen zu treffen wie über die Art der Lernerfahrung (beispielsweise überwachtes oder unüberwachtes Lernen), die Form und Repräsentation der zu lernenden Zielfunktion und die Auswahl der zu lernenden Merkmale.

Auf dem Gebiet des Maschinellen Lernens wurden viele Algorithmen entwickelt, von denen im Folgenden drei sehr wichtige kurz vorgestellt werden sollen.

2.2.1 Gradientenabstieg

Das Gradientenabstiegsverfahren [Bottou(2010)] ist ein numerisches, iteratives Verfahren für Optimierungsprobleme, bei dem ein optimales Parameter-Tupel für eine bestimmte Bewertungsfunktion gesucht wird. Ausgehend von einem Anfangspunkt werden die einzelnen partiellen Ableitungen für jeden Parameter über der Bewertungsfunktion gebildet. Da die Ableitungen oft nicht analytisch ermittelt werden können, werden dabei kleine Deltaschritte vom Ausgangspunkt in jeder Dimension des Parameterraumes durchgeführt. Das nächste Parametertupel bestimmt sich aus dem größten Abstieg auf der Bewertungsfunktion (ermittelt durch die partiellen Ableitungen). Bei großen Änderungen des Funktionswertes der Bewertungsfunktion werden große Schritte (große Änderungen der Parameterwerte), bei kleinen Änderungen kleine Schritte durchgeführt. Gradientenabstiegsverfahren besitzen eine Vielzahl von Erweiterungen, u.a. um lokale Minima überwinden zu können. So gibt es oftmals eine mit der Zeit kleiner werdende Lernrate, die dafür sorgt, dass das Verfahren anfangs schneller in den Bereich einer guten Lösung gelangt und später das (lokale oder globale) Minimum auch erreicht.

Die Least-Mean-Squares-Methode (LMS) ist ein Gradientenabstiegsverfahren, das die Parameter einer linearen Funktion verändert, um den mittleren quadratischen Fehler zu einer Zielfunktion (z.B. definiert aus Trainingsbeispielen) zu minimieren, bis die Gewichte konvergieren.

2.2.2 Evolutionäre Verfahren

Evolutionäre Verfahren [Koza(1997)] finden sowohl in der Künstlichen Intelligenz als auch in der Robotik breite Anwendung. Sie orientieren sich an biologischen Selektionsprozessen und erfordern nur geringes Problemwissen. Anstatt die Parameter für nur einen Startpunkt der Bewertungsfunktion zu optimieren, werden gleich viele Startpunkte (Anfangsindividuen) meist zufällig ausgewählt. Die Menge der n Startpunkte wird als Anfangspopulation bezeichnet. Im Bewertungsschritt wird nun jedem Startpunkt (= Individuum) durch eine Fitnessfunktion ein Fitnesswert zugeordnet. Ist die Ausgabe

für ein Beispiel korrekt oder nahe am Optimum, so entspricht dies einer hohen Fitness. Im Propagierungsschritt werden die Individuen mit der höchsten Fitness in die nächste Runde, ggf. sogar mehrfach propagiert. Die Parameter werden bei Individuen mit hoher Fitness nur wenig modifiziert, bei solchen mit geringerer Fitness stärker. Diese Abfolge von Bewertung, Propagierung, Modifikation wird solange wiederholt, bis eine Teilmenge der Population eine bestimmte Fitness erreicht hat. Für Evolutionäre Verfahren gibt es eine große Menge von Erweiterungen, auf die an dieser Stelle nicht eingegangen werden soll.

2.2.3 Reinforcement-Learning

Bei Reinforcement-Learning-Verfahren [Sutton(1988)] geht es darum, eine Policy zu lernen, d.h. welche Aktionen sind für gegebene Zustände am gewinnbringendsten und sollten daher gewählt werden, wie bei einem Mehrpersonenspiel. Eine Besonderheit ist, dass anfangs nur die Bewertung der Zielzustände bekannt ist, definiert durch die Reward-Funktion. Die Bewertung von Zustands-Aktionspaaren wird durch die zu lernende Value-Funktion repräsentiert. Reinforcement-Learning-Verfahren sind durch ihre Orientierung auf den Gesamtgewinn für ein gegebenes Problem sehr leistungsfähig, das Training kann durch den oftmals sehr großen Zustandsraum sehr aufwändig sein. Für die Repräsentation der Bewertungsfunktion, welche sehr speicheraufwändig sein kann, werden oftmals Tabellen oder neuronale Netze verwendet. Bekannte Unterarten von Reinforcement-Learning-Verfahren sind Temporal-Difference-Learning und Q-Learning.

Nach der Vorstellung ausgewählter Grundlagen, die für das Verständnis der weiteren Arbeit hilfreich sind, liegt der Kern des folgenden Kapitels auf der Beschreibung des Spiels No-Thanks.

Kapitel 3

Beschreibung des Spiels No-Thanks

No-Thanks ist ein kurzes, leicht zu erlernendes Mehrpersonenkartenspiel - eine Partie dauert in etwa 10 bis 20 Minuten. Das Spiel wurde im Jahr 2004 von Thorsten Gimmler erfunden. No-Thanks besitzt mittleren Bekanntheitsgrad - im deutschen Sprachraum ist es unter dem Titel 'Geschenkt ...ist noch zu teuer!' bekannt, im englischen unter dem Namen No-Thanks. Der Name der internationalen Variante ist 'No Merci!' [Boardgamegeek.com(2012)]. Laut Anleitung [Gimmler(2005)] lässt es sich am besten mit drei bis sieben Personen ab einem Alter von 8 Jahren spielen, da die Beherrschung von Addition und Subtraktion der Zahlen bis 200 sowie das Prognostizieren verdeckter Karten und Entscheidungen anderer Spieler für den Spielerfolg wichtig sind.

Das vorliegende Kapitel beschreibt zunächst die Spielregeln von No-Thanks und geht auf spezifische Besonderheiten aus dem Gebiet der Spieltheorie ein.

3.1 Spielregeln

Benötigt werden für das Spiel No-Thanks spezielle Spielkarten, Spielchips sowie mindestens zwei Spieler, die gegeneinander spielen.

Spielvorbereitung

Zu Beginn werden die 33 Karten des Spiels, welche die Werte von 3 bis 35 haben, gemischt. Anschließend werden zufällig 9 Karten, die im weiteren Spiel nicht mehr von Bedeutung sind, aus dem Kartenstapel entfernt. Von den im Spiel befindlichen 24 Karten wird eine Karte aufgedeckt und für alle sichtbar auf den Stapel gelegt. Außerdem werden an alle Spieler gleich viele

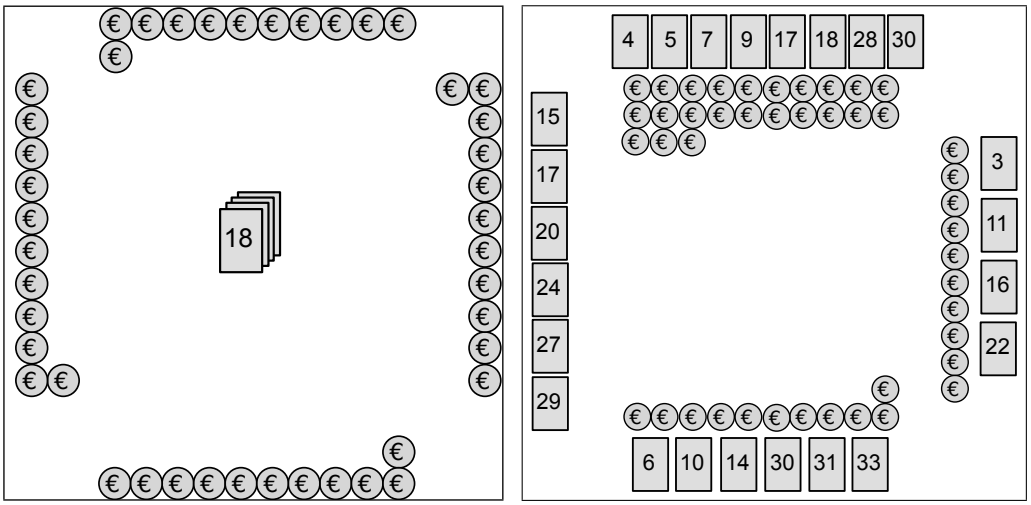
Chips verteilt. Die Chipanzahl pro Spieler ist je nach Spieleranzahl durch die Anleitung [Gimmler(2005)] festgelegt und ist Tabelle 3.1 zu entnehmen.

Tabelle 3.1: Spielvorbereitung

Spieleranzahl	**Chipanzahl**
3-5	11
6	9
7	7

Verteilung der Spielchips in Abhängigkeit der Spieleranzahl

Eine Beispielsituation für einen Spielanfang ist in Abb. 3.1a abgebildet, in der die vier Spieler jeweils gleich viele Chips haben und noch alle Karten auf dem Kartenstapel in den Mitte des Tisches verteilt sind.



(a) Beginn des Spiels (b) Ende des Spiels

Abbildung 3.1: Ein Beispiel einer Spielpartie in No-Thanks zwischen vier Spielern (jeweils an den Rändern befinden sich ihre gesammelten Karten und Chips), in der Mitte befindet sich der Kartenstapel, auf dem sich möglicherweise Chips befinden können.

Spielablauf

No-Thanks ist rundenbasiert. Die Spieler führen ihre Spielzüge hintereinander und entsprechend ihrer Sitzposition um den Spieltisch aus.

Ein Spieler, der aktuell am Zug ist, muss zwischen zwei Spielaktionen entscheiden:

- Der Spieler kann einen seiner Chips auf die Karte legen. Damit erhöht sich die Anzahl der Chips auf dieser Karte um eins. In diesem Fall ist der nächste Spieler am Zug und muss seinerseits eine Entscheidung fällen.

- Alternativ kann der Spieler die aktuell auf dem Stapel liegende Karte mitsamt ihrer Chips nehmen. Dabei erhält er sowohl die Karte, die er für alle sichtbar vor sich legt und bis zum Ende des Spiels behält, als auch die Chips. Danach wird vom Kartenstapel die nächste Karte aufgedeckt und derselbe Spieler ist noch einmal am Zug.

Die beiden Handlungsalternativen sind in Abb. 3.2 für ein reduziertes Beispiel dargestellt.

Spielende und -auswertung

Das Ende des Spiels ist erreicht, sobald der Kartenstapel leer ist. Die Auswertung erfolgt über eine Auszählung der Punkte aller jeweiligen Spieler: Zunächst werden alle Kartenwerte des Spielers summiert. Falls ein Spieler Kartenwerte besitzt, welche in einer Reihe liegen, beispielsweise [4,5,6,7] zählt nur der niedrigste Kartenwert der Reihe, im Beispiel zählt also nur die 4. Von der Kartensumme wird anschließend die Anzahl gesammelter Chips abgezogen.

Für jeden Spieler ergibt sich also am Ende des Spiels eine Punktzahl. Der Spieler, der von allen Spielern die niedrigste Punktzahl hat, gewinnt das Spiel. Alle anderen Spieler mit mehr Punkten als der Gewinner verlieren. Falls nach der Auswertung mehrere Spieler die gleiche Punktanzahl besitzen und diese der niedrigsten Punktzahl entspricht, sind all diese Spieler Gewinner.

Zusammengefasst ist der Gewinner also derjenige Spieler mit der minimalen Differenz aus der Kartenwertsumme und der Anzahl der Chips. Im Beispiel in Abb. 3.1b gewinnt der Spieler rechts mit 42 Punkten gegenüber den anderen (oben: 72 Punkte, links: 132 Punkte, unten: 82 Punkte).

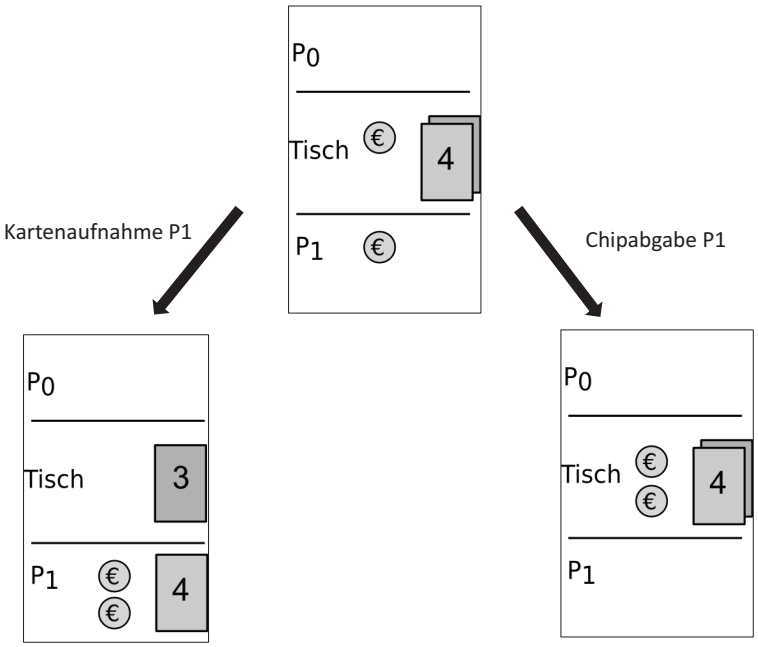


Abbildung 3.2: In den drei dargestellten Spielzuständen ist jeweils der Besitz der beiden Spieler P_0 und P_1 sowie der aktuelle Kartenstapel und die aktuelle Verteilung der Spielchips dargestellt. Im oben dargestellten Spielzustand kann sich der Spieler P_1 für eine Kartenaufnahme entscheiden, woraus sich nach den in 3.1 beschrieben Spielregeln der Zustand unten links ergibt. Entscheidet er sich für eine Chipabgabe, so ergibt sich der unten links dargestellte Zustand.

3.2 Spieleinordnung

Das Spiel No-Thanks ist laut den in 2.1.2 genannten Kategorien von Spielen

- dynamisch. Die Aktionen und Entscheidungen der Spieler finden stets nacheinander und niemals gleichzeitig statt.
- endlich. Das Spiel endet genau dann, wenn die letzte Karte des endlich großen Kartenstapels von einem Spieler genommen wurde. Eine Karte des Stapels muss nach endlich vielen Schritten aufgenommen werden, da die Anzahl der Chips, die das Ablehnen einer Karte ermöglichen, nur endlich ist und ein Spieler somit nach endlich vielen Schritten die Karte aufzunehmen hat. Da das Spiel endlich ist, ist auch der Spielbaum endlich.
- nicht notwendigerweise kooperativ. In einem abgeschlossenen Spielablauf ist es nicht sinnvoll für Spieler zu kooperieren, da jeder Spieler

jeden anderen Spieler ausstechen möchte und Spieler lediglich das Ziel verfolgen, Gewinner der Runde zu sein. Die Anzahl der Punkte, mit der er dies tut, wird nicht berücksichtigt - was zählt, ist, dass er der Gewinner ist. Zweite, dritte oder weitere Plätze werden einheitlich als eine Niederlage gewertet.

- ein Spiel mit vollständiger Information und Zufallselementen. Die Spieler haben keine Kenntnis über die Kartenwerte, die sich auf den Karten auf dem Kartenstapel befinden. Abgesehen davon wissen alle Spieler alles über die Spielsituation.

- kein Nullsummenspiel. Die Punkte, die am Ende unter den Spielern ausgezählt werden, ergeben nicht immer die gleiche Summe. Der Grund hierfür ist, dass einige Kartenwerte nicht gewertet werden, da Spieler ihre Punkte reduzieren können, falls sie die in 3.1 genannten Reihen von Kartenwerten bilden.

- Ein Unentschieden tritt selten auf, ist aber möglich, falls gleiche Punktsummen bei unterschiedlichen Spielern am Ende auftreten.

Auf Grundlage dieser Eigenschaften wurde im Folgenden ein Simulator für das Spiel No-Thanks entwickelt und implementiert, was im folgenden Kapitel näher erläutert werden soll.

Kapitel 4

Simulatordesign

Bei der Entwicklung des hier vorgestellten Simulators standen für den Autor der vorliegenden Arbeit die folgenden Eigenschaften im Vordergrund:

- Startzustand mit beliebiger Spielkarten- und Spielchipverteilung sowie beliebiger Reihenfolge der Spieler,
- Unterstützung beliebig vieler Spieler, die hinzugefügt werden. Neu hinzukommende Spielertypen müssen lediglich ihre Entscheidungslogik selbst implementieren und können die Möglichkeit nutzen, einen Spielbaum zur Vorausschau aufzubauen,
- Verfügbarkeit unterschiedlich komplexer Spielerprototypen, z.B. Random Spieler, 1-Zug und 3-Zug-Vorausschauer mit der Option, Parameter der Bewertungsfunktion zu lernen.

4.1 Struktur des Spielbaums

Im Folgenden soll beschrieben werden, wie Zustände, Aktionen und Spieler für diese Arbeit repräsentiert wurden.

4.1.1 Repräsentation von Spielzuständen

In einem gespielten Spiel von No-Thanks gibt es keine Spielzustände, die durch verschiedene Spielzüge erreicht werden können.

Alle Spieler kennen zu jedem Zeitpunkt die aktuelle Spielsituation, welche sich aus mehreren bekannten und unbekannten Größen zusammensetzt, die im Folgenden erläutert werden.

Abgesehen von den Spielstrategien gibt es keine Größe, die einen aktuellen Spielzustand kennzeichnet und die nicht auch allen Spielern bekannt oder

gleichermaßen unbekannt ist - man spricht in diesem Zusammenhang auch von Common Knowledge [Osborne and Rubinstein(1994)].

Unbekannte Größen

In diesem Spiel gibt es lediglich folgende unbekannte Größen:

- Die Werte der noch auf dem Stapel verbleibenden Karten. Die Spieler wissen nicht, welche der Karten noch im Stapel enthalten sind bzw. welche Karten vor Beginn des Spiels aus dem Stapel aussortiert wurden. Außerdem wissen die Spieler nicht, in welcher Reihenfolge die Karten des Stapels auf dem Stapel liegen. Die Wahrscheinlichkeitsverteilung dafür, dass eine noch nicht gesehene Karte überhaupt noch im Stapel vorkommt, ist eine diskrete Gleichverteilung, bei der jede Einzelwahrscheinlichkeit wie folgt berechnet werden kann:

$$\frac{1}{\#möglicheKarten - \#geseheneKarten}$$

- Außerdem wissen die Spieler nicht voneinander, welche Strategien von den Gegenspielern verfolgt werden. Das Schließen auf diese Größe ist erst nach mehreren gespielten Runden möglich, in dieser Arbeit wird diese Größe nur indirekt über die gewählte zu lernende Heuristik ins Modell eingehen. Zudem wird zur Vereinfachung angenommen, dass Gegner von Spielern die gleichen Strategien wie die Spieler selbst verwenden.

Bekannte Größen

Folgende Informationen sind vollständig in jedem Zustand bekannt:

- Die Anzahl der Spieler. Diese ist von Beginn eines Spiels an fest und allen Spielern bekannt.
- Die möglichen Kartenwerte, die im Spiel auftreten können. Laut der Spielbeschreibung in [Gimmler(2005)] sind in jedem Spiel stets alle Kartenwerte von 3 bis 32 möglich.
- Der Wert der obersten auf dem Stapel liegenden Karte. Dieser ist für alle Spieler sichtbar, sobald eine neue Karte aufgedeckt wird.
- Die Anzahl der Karten auf dem Stapel. Diese wird zu Beginn des Spiels allen Spielern mitgeteilt.

- Die aktuelle und die vergangene Verteilung der Chips und Karten unter den Spielern. Auch diese Größen sind stets allen Spielern gleichermaßen bekannt. Zwar wird in der Spielanleitung darauf hingewiesen, dass die Chips von jedem Spieler mit der Hand verdeckt werden sollten, allerdings kann ein Mensch, der die anfängliche Verteilung der Chips und den bisherigen Spielverlauf kennt, sehr einfach folgern, wie die Chips momentan verteilt sind. Wir nehmen also an, dass Spieler sich die Verteilung der Chips merken können und werden daher die Verteilung der Chips als bekannte Gößе voraussetzen.

- Durch die letzten zwei genannten bekannten Größen ist auch stets bekannt, wie viele Karten noch auf dem Stapel liegen.

4.1.2 Entwicklung des Spielbaums

Wenn den Spielern die Werte der Karten auf dem Stapel bekannt wären, so besäße No-Thanks keine Zufallselemente. Demzufolge würde eine berechenbare Gewinnstrategie existieren, die darin besteht, den gesamten Spielbaum zu erstellen und das teilspielperfekte Gleichgewicht, das in Abschnitt 2.1.3 vorgestellt wurde, zu ermitteln. Zur Illustration ist ein Beispiel eines vereinfachten Spielbaums in Abb. 4.1 gegeben. Die Auszahlungen sind in den Endknoten für alle Spieler angegeben und werden mithilfe des MaxN-Algorithmus bis zur Wurzel des Teilspiels propagiert.

4.1.3 Hinzunahme von Zufallselementen

Das teilspielperfekte Gleichgewicht mit Zufallselementen zu bestimmen ist möglich, wenn Zufallsknoten eingeführt werden. Würden alle Spieler optimal spielen, wäre das Ergebnis allein von den zufälligen auf dem Kartenstapel verteilten Karten sowie den - hier festen - Regeln des Spiels abhängig.

In unserem Fall haben wir allerdings 32! (rund $2 * 10^{35}$) mögliche Anordnungen von Karten auf dem Kartenstapel sowie 24 von 35 möglichen Kartenwerten, sodass die Zufallsknoten einen hohen Verzweigungsgrad haben, der im Spielbaum bei 24 startet und pro Baumhöhe um eins abnimmt, bis er am Ende des Spiels bei 1 liegt.

Da ein Spielbaum im Endspiel eine kleine Tiefe hat, besitzt er eine überschaubarere Zahl von Knoten. Hinzu kommt, dass es am Ende eines Spiels von No-Thanks für Spieler vorteilhaft ist, im Spielbaum bis hin zu den möglichen Endzuständen und den resultierenden Auszahlungen vorauszuschauen. Daher sollte ab einer überschaubaren Höhe der gesamte Baum aufgebaut und die Aktion mit der höchsten Gewinnwahrscheinlichkeit gewählt werden.

Zur Illustration ist ein Beispiel des Spielbaums in Abb. 4.2 dargestellt.

4.2 Bewertungsfunktionen für No-Thanks

In unserem Spiel geht es laut der Spielregeln nur um das relative Gewinnen oder Verlieren in einem einzelnen Spiel - die Höhe des Sieges ist egal. Pro gespieltem Spiel gibt es demnach einen Punkt für den Gewinner (bei mehreren Gewinnern wird der eine Punkt proportional aufgeteilt), alle Verlierer erhalten null Punkte. Jeder Spieler spielt also, wenn Kooperationseffekte vernachlässigt werden, gegen jeden anderen Spieler. Bei No-Thanks ist die Strategie, möglichst selbst wenige Minuspunkte zu bekommen nicht so wichtig wie die, relativ zu den Gegnern die wenigsten Minuspunkte zu bekommen und daher wird es wichtiger, den Gegnern Schaden zuzufügen. Allgemein möchte jeder Spieler möglichst den anderen Spielern schaden und selbst so wenig Schaden wie möglich erleiden.

Aus dem formulierten Ziel, am Ende möglichst wenige Minuspunkte zu bekommen, ergeben sich folgende Formulierungen von Zielen für eine fortlaufende Spielrunde:

- Wenn man der momentan beste Spieler ist, könnte das Ziel sein, den Abstand zu allen anderen Spielern zu maximieren.

- Wenn man momentan nicht der beste Spieler ist, könnte das Ziel sein, den Abstand zum ersten Spieler zu minimieren. Allerdings könnte man auch versuchen, anderen Spielern als dem momentan besten zu schaden.

Des Weiteren können mehrere Spiele hintereinander gespielt werden - genau das ist auch der Fall, wenn menschliche Spieler aufeinander treffen. Daher ist ein Lernen aus vergangenen Spielen oder aus dem bisherigen Spielverhalten innerhalb eines Spieles und Anpassung an jeweilige Gegner möglich.

Eine Option für Spieler ist es, den Spielbaum nur bis zu einer gewissen Tiefe zu generieren, um dann die Spielsituationen mithilfe einer Heuristik zu bewerten. Dabei stellen sich zwei grundsätzliche Fragen: Einerseits nach der Höhe des aufgebauten Spielbaumes, andererseits nach der genutzten Bewertungsfunktion sogenannter Frontierknoten, welche Endknoten eines bereits expandierten Baumes darstellen, selbst aber noch unexpandierte Kindknoten besitzen.

Da bei No-Thanks das Annehmen einer Karte die Spieler in eine neue Situation versetzt, in der sie den Kartenwert der nächsten Karte sowie die finale Zuordnung der vorherigen Karte zu einem Spieler erfahren, ist es günstig, das Annehmen einer Karte als Erhöhung der Baumtiefe zu betrachten.

Das Ablehnen von Karten sollte dagegen immer vorausgeplant werden, da es für Spieler sinnvoll ist, die Situation in Betracht zu ziehen, in der der erste Spieler, welcher keine Spielchips mehr hat, die Spielkarte annehmen muss.

Für das Erstellen einer Bewertungsfunktion kommen verschiedene Modelle in Betracht. In dieser Arbeit soll eine einfache lineare Bewertungsfunktion mit nur wenigen Parametern genutzt werden, zusätzlich soll zur Bestimmung eines Gewichts das in 2.2.1 vorgestellte Gradientenabstiegsverfahren verwendet werden. Gradientenabstiegsverfahren haben dabei den Vorteil, dass sie sich sehr einfach implementieren lassen, auf höherdimensionale Problemräume noch gut anwendbar sind und zugleich bei vielen Problemstellungen sehr gute Lernresultate erzielen. Evolutionäre Verfahren sowie Reinforcement-Learning sollen aufgrund ihres vergleichsweise hohen Implementations- sowie Berechnungsaufwandes im Rahmen dieser Arbeit nicht betrachtet werden, sind jedoch prinzipiell im hier vorgestellten Simulator integrierbar.

Im Besonderen werden in dieser Arbeit die Kartenwerte und das Gewicht der Spielchips betrachtet, was im folgenden Kapitel genauer beschrieben und ausgewertet werden soll.

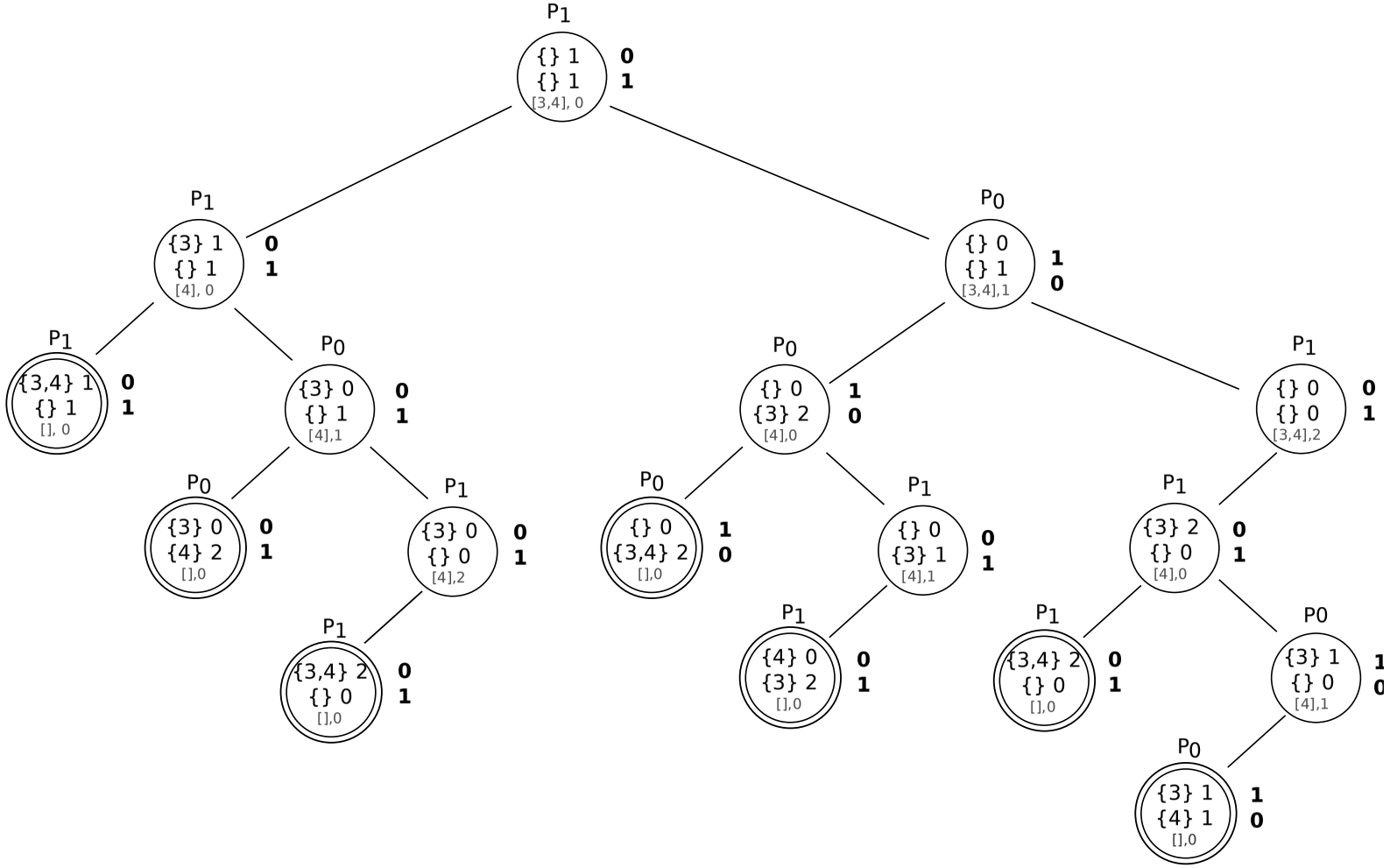


Abbildung 4.1: Beispiel eines Spielausschnitts von No-Thanks ohne Zufallselemente für nur zwei Spieler. Spielzustände werden durch Knoten repräsentiert - die Nummer des jeweiligen Spielers, der am Zug ist, steht über dem jeweiligen Knoten. Kanten nach links unten stellen das Annehmen der Karte vom Kartenstapel dar und nach rechts unten das Legen eines Spielchips. Die erste Zeile innerhalb eines Zustandknotens stellt die gesammelte Menge Karten und die Anzahl der Chips des ersten Spielers, die zweite die des zweiten Spielers dar. Die dritte Zeile beschreibt die Liste der auf dem Stapel befindlichen Karten, wobei die erste Karte die oberste ist, sowie daneben die Anzahl an Chips auf dem Kartenstapel. Rechts neben den Knoten wurden die Gewinnwahrscheinlichkeiten für alle Spieler eingetragen: Nachdem man sie für die Endknoten bestimmt hat, werden sie im Spielbaum mithilfe der Bestimmung des teilspielperfekten Gleichgewichts nach oben propagiert. Bei diesem dargestellten Spielausschnitt kann demnach der Spieler P_1 im an der Wurzel des Teilspiels dargestellten Zustand mit Sicherheit das Spiel gewinnen.

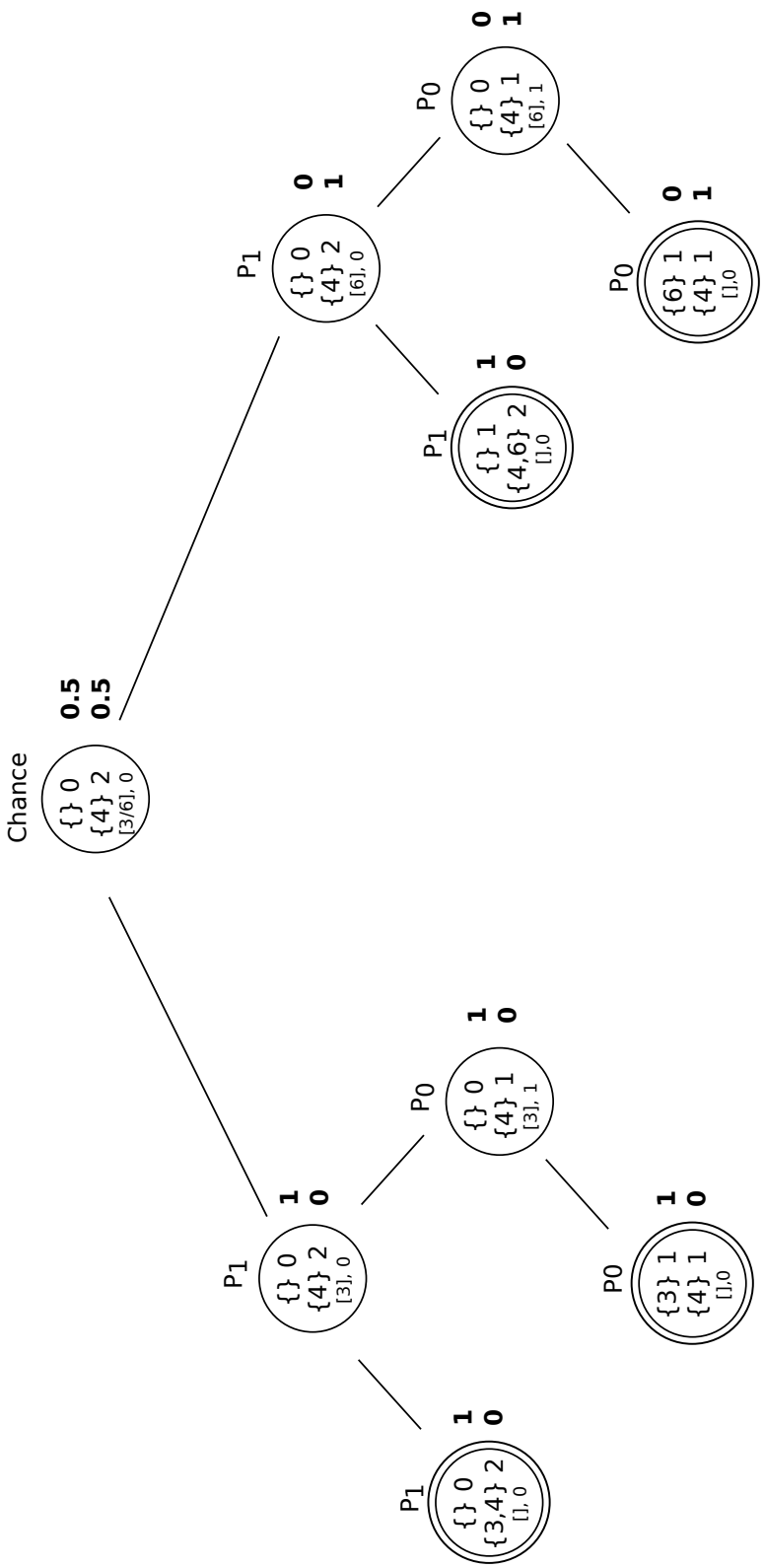


Abbildung 4.2: Beispiel eines Spielausschnitts von No-Thanks mit Zufallselementen und zwei Spielern
Die Beschriftung ist die gleiche wie in Abb. 4.1. Es wurden lediglich Zufallsknoten (Chance Nodes) hinzugefügt, deren Gewinnwahrscheinlichkeiten sich aus dem Erwartungswert der Gewinnwahrscheinlichkeiten Kindknoten ergeben. Aus Darstellungsgründen ist hier ein sehr kurzes Endspiel abgebildet. Wie zu erkennen, haben beide Spieler im Wurzelknoten des dargestellten Teilspiels die gleiche Gewinnchance.

Kapitel 5

Experimente und Auswertung

Im folgenden Kapitel sollen einfache Lernexperimente mit dem in dieser Arbeit vorgestellten Simulator durchgeführt werden. Das Hauptaugenmerk soll dabei auf der Fragestellung liegen, inwieweit sowohl eine Vorausschau im Suchbaum als auch die Lernfähigkeit einer Strategie zu ihrer Leistungsfähigkeit beitragen können. Zur Vereinfachung und insbesondere um die Berechnungszeit für die Simulation zu verringern, wird in den Lernexperimenten nur eine Teilmenge der vorhandenen Karten und eine geringe Anzahl von Spielchips verwendet. Zugleich wird die Simulation für ein Zweipersonenspiel betrachtet. Im folgenden Abschnitt sollen die einzelnen Experimente näher beschrieben werden.

5.1 Kurze Beschreibung der Experimente

Folgende Experimente wurden im Rahmen dieser Arbeit durchgeführt:

1. Da es sich bei der gegebenen Problemstellung um ein Spiel mit Zufallselementen handelt, soll zunächst der Einfluss des Zufalls auf die Spielergebnisse untersucht werden. Dazu werden sowohl die mittlere Gewinnwahrscheinlichkeit als auch ihre Standardabweichung bei einer gegebenen Anzahl von Spieldurchläufen ermittelt. Dieses Experiment dient zur Bestimmung der notwendigen Versuchsdurchläufe der nachfolgenden Experimente, um signifikante Aussagen über die Leistungsfähigkeit unterschiedlicher Spielstrategien treffen zu können.

2. Im zweiten Experiment soll eine einfache Spielstrategie ohne Vorausschau mit Hilfe eines Gradientenabstiegsverfahrens über einem Parameter der Heuristikfunktion gelernt werden. Dabei wird ein lernender

Spieler gegen einen nicht lernenden Standardspieler (Einfachvorausschau) antreten. Besonders interessant ist dabei die Konvergenz des Parameters bei verschiedenen Ausgangswerten als auch die schrittweise Verbesserung der Gewinnwahrscheinlichkeit.

3. Im dritten Experiment soll eine Heuristikfunktion für eine Spielstrategie mit Dreifachvorausschau, ähnlich zum zweiten Lernexperiment, ebenfalls per Gradientenabstiegsverfahren gelernt werden.

5.2 Versuchsaufbau

Die folgenden Experimente wurden mit jeweils zwei Spielern, 6 Spielkarten, welche die Werte 3, 6, 9, 10, 11, 13, 15, 17, 33 annehmen können, sowie einer Chipanzahl von 3 Chips pro Spieler durchgeführt. Pro Experiment ist von einer festen Reihenfolge der Spieler ausgegangen worden; das heißt, dass die Spieler an der gleichen Position sitzen und immer der gleiche Spieler das Spiel beginnt.

Um den Einfluss des Zufalls der Reihenfolge der Spielkarten zu untersuchen, werden die Spielkarten bei jedem Spielversuch neu gemischt.
In den folgenden Experimenten werden lediglich die am Spiel teilnehmenden Spielertypen verändert. Dabei kommen folgende Spielertypen zum Einsatz:

- Random Spieler:
  Der Random Spieler entscheidet sich zu 50 Prozent dafür, die Karte anzunehmen, und zu 50 Prozent dafür, sie weiterzugeben, ohne Berücksichtigung des Spielzustandes.

- Kurzfristiger Maximierer:
  Der Spielertyp des kurzfristigen Maximierers baut den Spielbaum bis zu einer Höhe von eins auf, bewertet die Frontier-Knoten des Spielbaums wie Blattknotens und entscheidet sich für die Aktion mit der höchsten Gewinnwahrscheinlichkeit. Dabei nimmt er an, dass auch seine Gegner kurzfristige Maximierer sind.

  Die Bewertungsfunktion für Frontier-Knoten entspricht dabei der Auswertungsfunktion von Spielendzuständen laut der Spielregeln:

  Gewinner sind der oder die Spieler, bei denen die Differenz aus der berechneten Summe gesammelter Karten und die Anzahl gesammelter Chips minimal ist. Hinzu kommt, dass der kurzfristige Maximierer den Abstand zum aktuell besten Spieler minimieren möchte, als Abstand wird hier die Differenz des oben beschriebenen Wertes des eigenen zum

besten Spieler betrachtet. Aktuell bereits beste Spieler versuchen, den Abstand zum zweitbesten zu maximieren.

- Einfachlerner:
  Der Einfachlerner nutzt keine Vorausschau im Spielbaum, entscheidet aber aufgrund der Information über den Quotient der aktuell auf dem Kartenstapel liegenden obersten aufgedeckten Karte und der sich dort befindlichen Chipanzahl über die Annahme oder Weitergabe der Karte. Der Einfachlerner entscheidet sich für eine Aufnahme der Karte, falls der Quotient aus der Anzahl der Chips und dem Kartenwert größer ist als ein zuvor durch Lernen ermitteltes Gewicht. Das Gewicht soll in einer Folge von Spielen schrittweise mithilfe eines Gradientenabstiegsverfahrens angepasst werden. Das Anfangsgewicht wird dabei zunächst willkürlich gesetzt.

  Es wurde sich in dieser Arbeit aus dem Grund für ein Gradientenabstiegsverfahren entschieden, weil es dank seiner Eigenschaften im Gegensatz zu evoltionären oder Reinforcement-Lernverfahren schnell konvergiert, einfach zu implementieren ist und gut skaliert.

  Wie in 2.2.1 beschrieben, wird in jedem Lernschritt der Parameter jeweils um einen geringen Wert erhöht bzw. verringert und auf erhöhte Gewinnerwartung überprüft. Die in dieser Arbeit verwendete Lernrate wurde experimentell ermittelt, wobei ein guter Kompromiss aus geringer Lerndauer, der anfänglichen Vermeidung kleiner lokaler Maxima sowie einer guten Konvergenz angestrebt wurden. Im Ergebnis stellt sich die Lernrate wie folgt dar:

$$2 \cdot 0.9^{\text{Lernschritt}} \tag{5.1}$$

- Dreifachlerner:
  Der Dreifachlerner nutzt die Vorausschau im Spielbaum bis zur Tiefe 3 und passt die Bewertungsfunktion für Frontierknoten des kurzfristigen Maximierers insofern an, dass die Chips nicht das normale Gewicht 1 haben, sondern ein ab einem Anfangswert gelerntes Gewicht. Somit liegt nach Ansicht des Dreifachlerners derjenige Spieler vorn, dessen Differenz aus gesammelter Kartensumme und Anzahl gesammelter Chips, die zuvor mit einem zu lernenden Gewicht multipliziert wird, minimal ist. Er wählt dann die Aktion, die ihm einen besseren Vorsprung gegenüber dem momentan besten Spieler verschafft. Ein momentan bester Spieler versucht, seinen Abstand zum zweitbesten zu maximieren. Auch der Dreifachlerner geht für Folgezustände, in denen die Gegner

am Zug sind, davon aus, dass jene dieselbe Bewertungsfunktion - mit dem aktuell gelernten Parameter - besitzen.

Um diese Spielertypen miteinander vergleichen zu können, wurde in der Simulation der Einfluss des Zufalls berücksichtigt. Im nächsten Abschnitt folgt daher eine kurze Erläuterung zur Simulationsfehlerabschätzung.

5.3 Simulationsfehlerabschätzung

Die beschriebenen Experimente stellen Zufallsexperimente dar, da sie auf einem stochastischen Simulationsmodell beruhen. Die Simulation benötigt daher eine ausreichende Anzahl unabhängiger Wiederholungen der Simulationsexperimente.

Abb. 5.1 zeigt die Standardabweichung der Gewinnwahrscheinlichkeit für den beschriebenen Versuchsaufbau mit zwei Random Spielern und jeweils 200 durchgeführten Durchläufen der gegebenen Anzahl an Versuchen. Wie zu erkennen ist, besteht ein polynomiell abnehmender Zusammenhang zwischen der Anzahl an Versuchen und der Standardabweichung um den Mittelwert der Gewinnwahrscheinlichkeit.

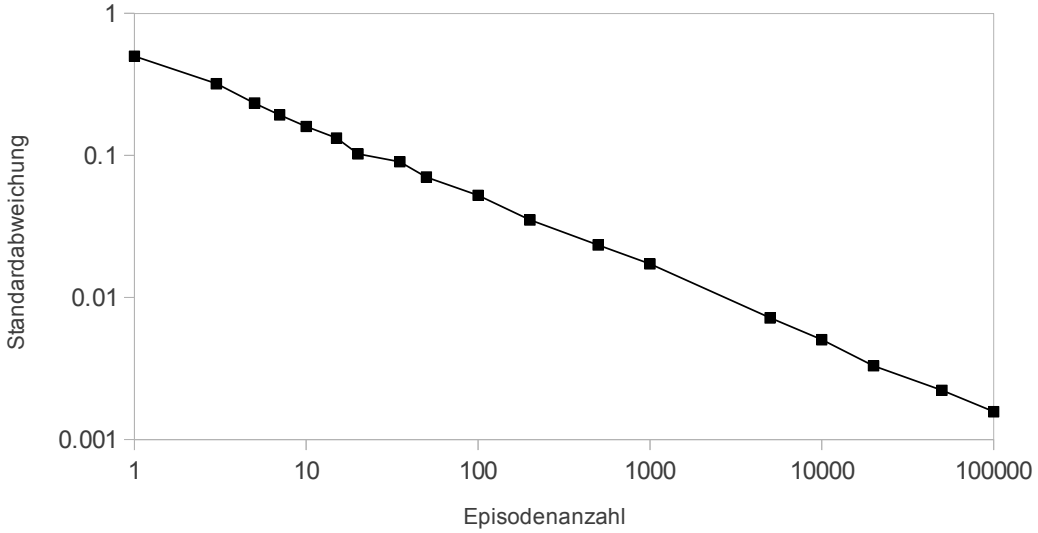


Abbildung 5.1: Experiment 1, Standardabweichung der Gewinnwahrscheinlichkeitsverteilung in Abhängigkeit der Episodenanzahl bei 200 Versuchen (eine Episode entspricht hierbei einem einzelnen Spiel). Für beide Achsen wurde eine logarithmische Skala verwendet.

Die Formeln 5.2 bzw. 5.3 sind eine Annäherung für den dargestellten Zusammenhang zwischen Episodenanzahl x und der Standardabweichung um

die Gewinnwahrscheinlichkeit σ:

$$\sigma = \frac{0.5}{x^{0.5}} \tag{5.2}$$

$$\log \sigma = \log 0.5 - 0.5 \cdot \log x \tag{5.3}$$

Wie zu erkennen, erfordert eine Verringerung der Standardabweichung um 90 Prozent eine Erhöhung des Berechnungsaufwandes um das 30-fache.

Bei 1000 Versuchen beträgt die Standardabweichung zum Mittelwert der Gewinnwahrscheinlichkeitsverteilung weniger als 2 Prozent (1,72 %), was für die folgenden Experimente einen guten Kompromiss zwischen Berechnungsaufwand und Genauigkeit des Ergebnisses darstellt.

5.4 Simulationsergebnisse

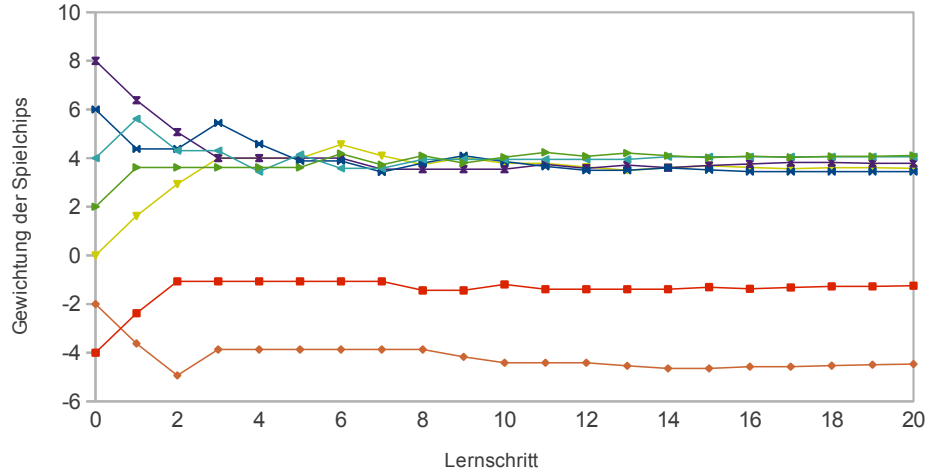
Im zweiten Experiment traten als Spieler ein kurzfristiger Maximierer als anfangender Spieler und ein Einfachlerner gegeneinander an.

Abb. 5.2a stellt die Veränderung des zu lernenden Parameters für das Gewicht der Spielchips in Abhängigkeit der Episodenanzahl für den Einfachlerner dar. Die dazugehörigen Gewinnwahrscheinlichkeiten sind in 5.2b gezeigt.

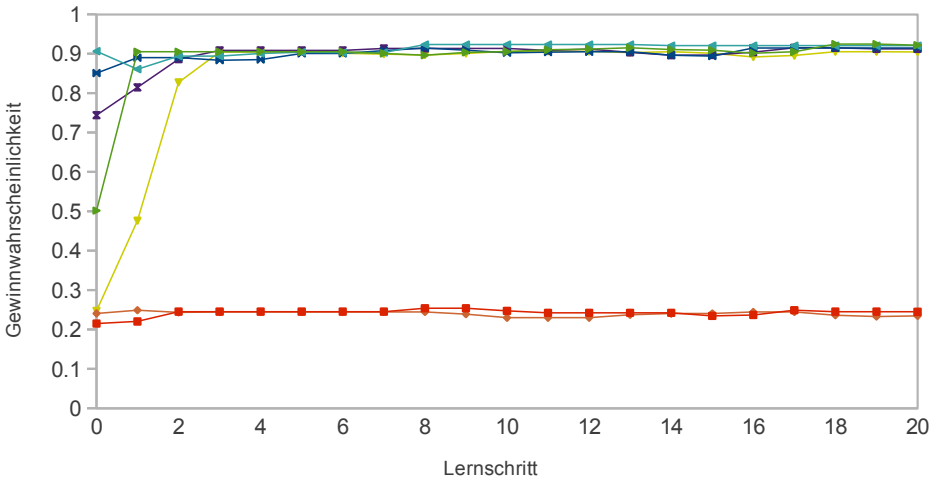
Für diesen Beispielparameter ist bei den Ausgangswerten von 0, 2, 4, 6 und 8 eine Konvergenz gegen den Wert 4 erkennbar, außerdem konvergiert die Gewinnwahrscheinlichkeit für alle Beispielparameter gegen einen Wert von rund 92 Prozent, was ein sehr gutes Ergebnis darstellt. Die Startwerte -2 und -4 jedoch verbessern sich trotz ihrer anfänglichen Änderung nicht, da sie aufgrund der gewählten Lernrate nicht den entsprechenden Bereich erreichen. Für alle anderen Startwerte ist ein Lernen allerdings deutlich zu erkennen - demnach scheint der Quotient zwischen Kartenwert und Chipanzahl durchaus die Gewinnchancen zu beeinflussen.

Im dritten Experiment trat ein Dreifachlerner gegen einen kurzfristigen Maximierer an, wobei auch hier der kurzfristige Maximierer als erster die jeweilige Spielrunde beginnen durfte.

Die Abb. 5.3a zeigt das gelernte Gewicht der Spielchips für den Dreifachlerner, die entsprechenden Gewinnwahrscheinlichkeiten sind wiederum in 5.3b dargestellt. Zu erkennen ist, dass die gewählten verschiedenen Startparameter zu Beginn gegen einen Wertebereich zwischen 1 und 3 konvergieren, sie aber weniger Einfluss auf die Gewinnwahrscheinlichkeit haben als zuvor beim Einfachlerner. Auch hier wird nach einer gewingen Anzahl an Lernepisoden eine Gewinnwahrscheinlichkeit von ca. 92 Prozent erreicht, diesmal für alle Startparameter.

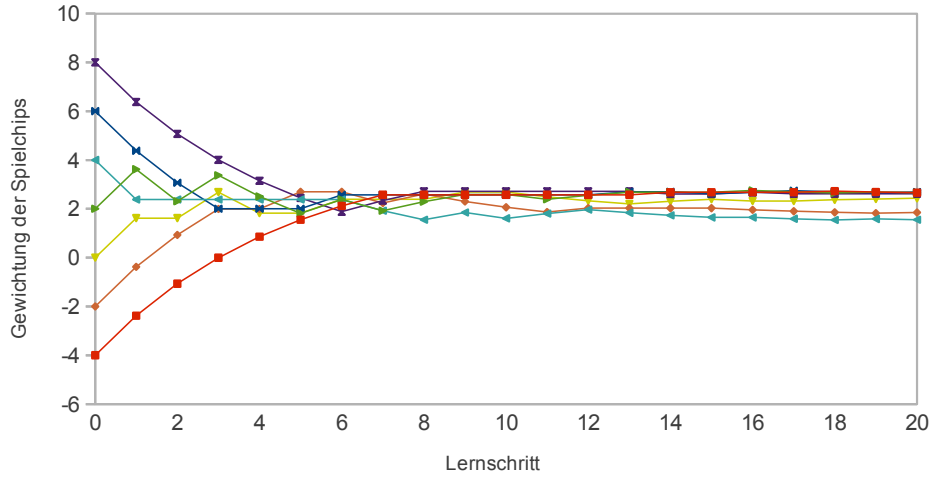


(a) Gewichte der Spielchips

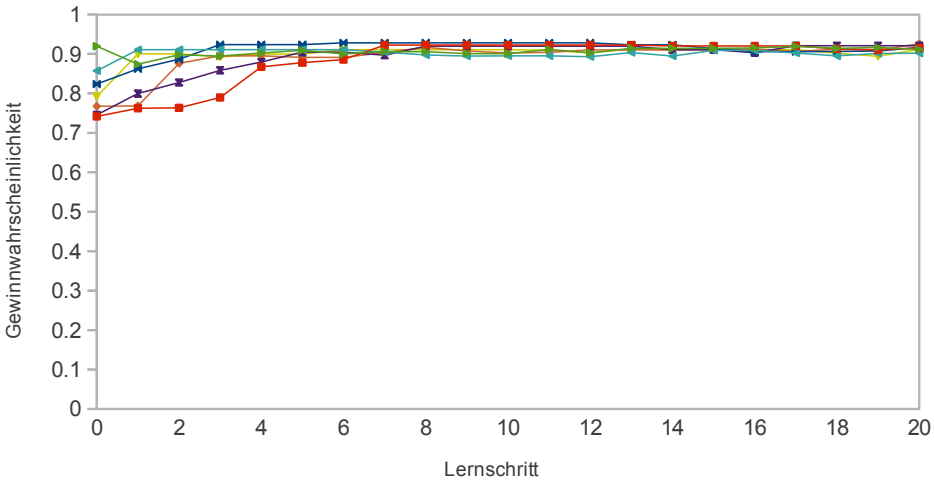


(b) Gewinnwahrscheinlichkeiten

Abbildung 5.2: Experiment 2, Lernen der Gewichtung der Spielchips bei einem Einfachlerner, der gegen einen kurzfristigen Maximierer spielt, mit jeweils verschieden Ausgangsgewichten der Spielchips



(a) Gewichtung der Spielchips



(b) Gewinnwahrscheinlichkeiten

Abbildung 5.3: Experiment 3, Lernen der Gewichtung der Spielchips bei einem Dreifachlerner, der gegen einen kurzfristigen Maximierer spielt, mit jeweils verschieden Ausgangsgewichten der Spielchips

Die beiden letzten durchgeführten Experimente deuten darauf hin, dass eine Vorausschau nur dann sinnvoll ist, wenn die dabei verwendete Heuristik gut gewählt wurde. Anders ausgedrückt: Eine gute Heuristik führt bei einem Spieler, der gar nicht vorausschaut, unter Umständen zur gleichen Gewinnwahrscheinlichkeit wie eine weite Vorausschau.

Kapitel 6

Zusammenfassung und Ausblick

6.1 Zusammenfassung

In der Arbeit wurde ein Simulator für das Spiel No-Thanks vorgestellt, der verschiedene Spiel- und Lernstrategien auswerten kann. Mithilfe des Simulators können beliebige Spielsituationen von No-Thanks simuliert werden. Die für das Spiel No-Thanks genutzten Spielregeln wurden dabei berücksichtigt und aus ihnen ein entsprechendes Modell des Spiels abgeleitet.

Wie gezeigt, ist No-Thanks ein Spiel mit Zufallselementen und relativ unabsehbaren Handlungen der Gegner. Insgesamt kann man nach der in dieser Arbeit vorgestellten spieltheoretischen Betrachtung des Spiels folgern, dass der Einfluss des Zufalls auf den Spielausgang recht hoch ist: Wenn als nächstes eine zufällige Karte kommt, hat dies Auswirkungen auf die künftige Chipverteilung, somit direkte Auswirkungen auf künftige Entscheidungen und demnach wieder auf künftige Chip- und Kartenverteilungen. Die Spieler selbst müssen spekulieren, welchen Wert ihre gesammelten Chips und Karten haben, denn dieser Wert ändert sich je nachdem, welche Karten und wie viele Chips sie noch in Zukunft erhalten werden, und je nachdem, wie ihre Gegner die Spielsituation als auch die anderen Spieler bewerten werden.

Den Schwerpunkt zum Entwickeln von Spielstrategien bilden in dieser Arbeit die Vorausschau im Spielbaum und die Entwicklung einer einfachen Bewertungsfunktion für Spielzustände, die sich stark an den Spielregeln orientiert. Als Variante wurde die Möglichkeit des Lernens von Gewichten der Spielchips vorgestellt. Dabei zeigte sich, dass das Gewicht der Spielchips durchaus konvergiert. Die in dieser Arbeit beschriebenen Experimente weisen darauf hin, dass sowohl eine Spielvorausschau als auch eine gute Bewertungsfunktion für eine Spielstrategie für No-Thanks gegen einen Gegner mit einer sehr einfachen Strategie erfolgversprechend sind. Erstaunlich war dabei, dass

die in Experiment 2 genutzte Entscheidungsfunktion, die nicht den Spielbaum aufbaut und nur konstanten Aufwand in Anhängigkeit ihrer Eingabegrößen hat, ein nahezu genauso gutes Ergebnis wie die in Experiment 3 gezeigte komplexere Strategie erreicht.

6.2 Ausblick

Die in dieser Arbeit gewählten Strategien wurden mit Absicht einfach gehalten, könnten für weitergehende Arbeiten aber optimiert werden.

In den Betrachtungen zur Entwicklung einer Bewertungsfunktion für Spielzustände blieben einige Parameter unberücksichtigt, beispielsweise der Spielschritt oder die verbleibende Anzahl Karten auf dem Kartenstapel. Des Weiteren setzt die Nutzung der Bewertungsfunktion auf Zuständen im expandierten Spielbaum Annahmen über die Aktionen der Gegner voraus. Im Rahmen dieser Arbeit wurde von der nicht realistischen Annahme ausgegangen, dass Gegner die jeweils gleiche Bewertungsfunktion und Entscheidungslogik verwenden. Aus diesem Grund sollte ein detaillierteres Modell über die gegnerischen Spieler und die Reihenfolge, in der die Spieler spielen, entwickelt werden. Beispielsweise könnten aus dem bisherigen Spielverhalten des Gegners Parameter seiner Bewertungsfunktion abgeleitet werden.

Um den Einfluss dieser Parameter automatisiert zu untersuchen, bieten sich Erweiterungen des hier vorgestellten Gradientenabstiegsverfahrens an. Die Einbeziehung von mehr Parametern wie der verbleibenden Kartenanzahl oder des bisher beobachteten Gegnerverhaltens könnte wiederum über nichtlineare Bewertungsfunktionen gelernt werden. Es sollte untersucht werden, ob eine Anwendung des Reinforcement-Learning zu verbesserten Ergebnissen führt. Als Reward-Function könnte die in dieser Arbeit genutzte Abstandsfunktion zum Gegner verwendet werden, ebenso könnten Spieler erst am Ende des Spiels bei einem wirklichen Sieg belohnt werden. Zu untersuchen wäre hierbei, ob die Value-Function für eine so entwickelte Strategie auch in Form eines künstlichen Neuronalen Netzes oder einer Tabelle repräsentiert werden könnte anstatt durch eine einfache lineare Funktion.

Die so entwickelten Strategien könnten möglicherweise in der Lage sein, auch folgenden Spielaspekt zu berücksichtigen: Ein Spieler sollte unter Umständen schon am Spielbeginn viele, auch hohe Karten annehmen, um sich einen Vorteil an Chips und somit Mobilität gegenüber anderen Spielern zu sichern. Solange keine anderen Spieler die gleiche Strategie besitzen, ist diese unter Umständen sehr günstig, da mit ihrer Hilfe auch Reihen von hohen Karten und folglich eine geringe Gesamtkartensumme sowie eine hohe Zahl Chips gesammelt werden kann. Anhand dieses Beispiels wird deutlich, dass

eine komplexere Bewertungsfunktion, die den aktuellen Spielschritt sowie ein Modell der Gegner miteinbezieht, durchaus Vorteile hätte.

Ein weiterer Aspekt, der in dieser Arbeit noch nicht untersucht wurde, aber bereits zu diesem Zeitpunkt mithilfe des entwickelten Simulators realisierbar ist, liegt im simultanen Lernen für mehrere Spieler. Gegeneinander antretende Spieler würden demnach nach einem gespieltem Spiel jeweils ihre Parameter pro Lernschritt anpassen. Neben dem Einfluss der Lernfähigkeit des Gegners auf das Spielergebnis und die Konvergenz der gelernten Parameter wäre ein weiterer Aspekt, inwieweit das in dieser Arbeit genutzte Gradientenabstiegsverfahren geändert werden müsste, um zu vermeiden, dass Spieler sich in den ersten Schritten der Anpassung des gelernten Parameters zu früh auf einen Parameter festlegen, der später aufgrund der immer kleiner werdenden Lernrate nur noch wenig verändert werden kann.

Schließlich können mithilfe des in dieser Arbeit entwickelten Simulators weitere Experimente für beliebige Versuchsaufbauten mit unterschiedlichen Karten- und Chipverteilungen sowie weiteren Spielertypen durchgeführt und analysiert werden.

Literaturverzeichnis

[Billings et al.(2001)Billings, Davidson, Schaeffer, and Szafron] Darse Billings, Aaron Davidson, Jonathan Schaeffer, and Duane Szafron. The challenge of poker. *Artificial Intelligence*, 134:2002, 2001.

[Boardgamegeek.com(2012)] Boardgamegeek.com. Detaillierte Spielbeschreibung von No-Thanks, May 2012. URL `http://boardgamegeek.com/boardgame/12942/no-thanks`. , letzter Zugriff am 30. Juli 2012.

[Bottou(2010)] Léon Bottou. Large-Scale Machine Learning with Stochastic Gradient Descent. In Yves Lechevallier and Gilbert Saporta, editors, *Proceedings of the 19th International Conference on Computational Statistics (COMPSTAT'2010)*, pages 177–187, Paris, France, August 2010. Springer.

[Gimmler(2005)] Thorsten Gimmler. No Thanks Rules, 2005. URL `http://www.zmangames.com/cardgames/files/nothanks/NoThanksRules.pdf`. , letzter Zugriff am 30. Juli 2012.

[Halpern(2007)] Joseph Y. Halpern. Computer science and game theory: A brief survey. *CoRR*, abs/cs/0703148, 2007.

[Hauk(2006)] Hauk. Rediscovering *-minimax search. In *Proceedings of the 4th international conference on Computers and Games*, CG'04, pages 35–50, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-32488-7, 978-3-540-32488-1.

[Hauk(2004)] Thomas Gordon Hauk. Search in trees with chance nodes, 2004.

[Koza(1997)] John R. Koza. Genetic programming, 1997.

[Luckhart and Irani(1986)] Carol Luckhart and Keki B. Irani. An algorithmic solution of n-person games. In *AAAI'86*, pages 158–162, 1986.

[Maynard Smith and Price(1973)] J. Maynard Smith and G. R. Price. The logic of animal conflict. *Nature*, 246(5427):15–18, 1973.

[Mitchell(1997)] Thomas M. Mitchell. *Machine Learning*. McGraw-Hill, Inc., New York, NY, USA, 1 edition, 1997. ISBN 0070428077, 9780070428072.

[Neal(1993)] Radford M. Neal. Probabilistic inference using markov chain monte carlo methods, 1993.

[Neumann and Morgenstern(1961)] John Neumann and Oskar Morgenstern. *Spieltheorie und wirtschaftliches Verhalten*. Physica, 1961.

[Osborne and Rubinstein(1994)] M.J. Osborne and A. Rubinstein. *A Course in Game Theory*. MIT Press, 1994.

[Russell and Norvig(2004)] Stuart Russell and Peter Norvig. *Künstliche Intelligenz. Ein moderner Ansatz*, volume 2. Pearson Studium, 2004. ISBN 3-8273-7089-2.

[Schadd et al.(2009)Schadd, Winands, and Uiterwijk] Maarten P. D. Schadd, Mark H. M. Winands, and Jos W. H. M. Uiterwijk. Chance-probcut: forward pruning in chance nodes. In *Proceedings of the 5th international conference on Computational Intelligence and Games*, CIG'09, pages 178–185, Piscataway, NJ, USA, 2009. IEEE Press. ISBN 978-1-4244-4814-2.

[Sieg(2010)] Gernot Sieg. *Spieltheorie*, volume 3. Oldenbourg Verlag München, 2010.

[Sutton(1988)] Richard S. Sutton. Learning to predict by the methods of temporal differences. In *MACHINE LEARNING*, pages 9–44. Kluwer Academic Publishers, 1988.

[Tesauro(2002)] Gerald Tesauro. Programming backgammon using self-teaching neural nets. *Artificial Intelligence*, 134(1–2):181 – 199, 2002. ISSN 0004-3702.

[von Neumann(1928)] J. von Neumann. Zur Theorie der Gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320, December 1928. ISSN 0025-5831.