

Institut für Parallele und Verteilte Systeme  
Abteilung Simulation großer Systeme

Universität Stuttgart  
Universitätsstraße 38  
D-70569 Stuttgart

Studienarbeit Nr. 2398

# Eine künstliche Intelligenz für das Kartenspiel Tichu

David Pfander

**Studiengang:** Informatik  
**Prüfer/in:** Prof. Dr. Dirk Pflüger  
**Betreuer/in:** Dipl.-Inf. Patrick Diehl

**Beginn am:** 2.11.2012

**Beendet am:** 4.5.2013

**CR-Nummer:** I.2.8





## **Kurzfassung**

Tichu ist ein Kartenspiel, bei dem klassische Suchalgorithmen aufgrund der Unkenntnis der Verteilung der Karten nicht effizient angewendet werden können. In dieser Arbeit wird eine künstliche Intelligenz vorgestellt, die trotz dieser Schwierigkeit stark spielt und alle Entscheidungen in kurzer Zeit trifft. Dies wird mit einem einfachen Suchalgorithmus und einer komplexen Bewertungsfunktion erreicht, wobei der Bewertungsfunktion insbesondere ein Modell der Initiative einer Aktion zugrunde liegt. Die mit diesem Ansatz erzielte Spielstärke der künstlichen Intelligenz ist mit der Spielstärke durchschnittlicher menschlicher Spieler vergleichbar. Es zeigt sich, dass mittels komplexer, hinter der Bewertungsfunktion stehender Modelle auf einen Suchalgorithmus mit hoher Suchtiefe bei Tichu verzichtet werden kann.



# Inhaltsverzeichnis

<b>1. Einleitung</b>	<b>9</b>
<b>2. Das Kartenspiel Tichu</b>	<b>11</b>
2.1. Überblick . . . . .	11
2.2. Karten und Kombinationen . . . . .	12
2.3. Spielablauf . . . . .	15
2.4. Einige strategische Überlegungen . . . . .	16
2.4.1. Über das Erzielen von Punkten . . . . .	16
2.4.2. Erkennen eines guten Blattes . . . . .	17
2.4.3. Unterstützen des Partners . . . . .	18
<b>3. Eine künstliche Intelligenz für Tichu</b>	<b>19</b>
3.1. Tichu aus Sicht der künstlichen Intelligenz . . . . .	19
3.2. Der Basialgorithmus . . . . .	20
3.3. Reduktion der Entscheidungen . . . . .	22
3.4. Analyse der Handkarten . . . . .	27
3.4.1. Bestimmung aller Kombinationen einer Kartenmenge . . . . .	27
3.4.2. Bestimmung aller Partitionen einer Menge von Kombinationen . . . . .	30
3.5. Die Berechnung der Initiative . . . . .	37
3.5.1. Berechnung der benötigten Wahrscheinlichkeiten . . . . .	39
3.5.2. Einige Anmerkungen zur Initiativberechnung . . . . .	42
3.6. Die zentrale Bewertungsfunktion . . . . .	42
3.6.1. Die Initiativeeigenschaft . . . . .	43
3.6.2. Die Wenige-Kombinationen Eigenschaft . . . . .	46
3.6.3. Die Hohe-Partner-Kombination Eigenschaft . . . . .	48
3.6.4. Die Stich-Punkte Eigenschaft . . . . .	48
3.6.5. Die Tichu-Endspiel Eigenschaft . . . . .	50
3.6.6. Die Kleinste-Kombination Eigenschaft . . . . .	51
3.6.7. Die Tichu Schwellwerte . . . . .	52
3.6.8. Die Gewichte der Bewertungsfunktion . . . . .	53
3.6.9. Kalibrierung der Gewichte . . . . .	54
3.7. Die Behandlung der weiteren Entscheidungssituationen . . . . .	55
3.7.1. Der Kartentausch . . . . .	55
3.7.2. Das Wünschen . . . . .	57
3.7.3. Das Verschenken des Drachens . . . . .	57

<b>4. Ergebnisse</b>	<b>59</b>
4.1. Diskussion der Spielstärke und des Verhaltens . . . . .	59
4.2. Bewertung der Performance . . . . .	61
4.3. Schwächen und Verbesserungsvorschläge . . . . .	63
<b>5. Zusammenfassung und Ausblick</b>	<b>67</b>
<b>A. Anhang</b>	<b>69</b>
A.o.1. Eindeutigkeit einer Kartenmenge . . . . .	69
A.o.2. Der Fragebogen zur Evaluierung der KI . . . . .	70
<b>Literaturverzeichnis</b>	<b>73</b>

## Tabellenverzeichnis

---

3.1. Wert der Wenige-Kombinationen Eigenschaft abhängig von der Größe der Partition . . . . .	47
3.2. Gewichte der Eigenschaften und Parameter der Bewertungsfunktion . . . . .	53
4.1. Die Fragen der Studie zusammen mit der Häufigkeit der gewählten Antworten.	60
4.2. Performance der Implementierung auf verschiedener Hardware . . . . .	62
A.1. Fragen und Antwortmöglichkeiten der durchgeführten Umfrage . . . . .	71

## Verzeichnis der Algorithmen

---

3.1. Der Minimax-Algorithmus . . . . .	21
3.2. Idee des Suchalgorithmus zur Entscheidungsfindung . . . . .	22
3.3. Der Suchalgorithmus der Tichu KI . . . . .	25
3.4. Vorläufiger Algorithmus für den Kartenaustausch . . . . .	26
3.5. Algorithmus für die Tichu-Ankündigung . . . . .	26
3.6. Berechnung der Paare . . . . .	28
3.7. Berechnung der Treppen . . . . .	29
3.8. Berechnung der Partitionen . . . . .	32
3.9. Fortsetzung der Berechnung der Partitionen . . . . .	33
3.10. Die verwendete Bewertungsfunktion . . . . .	43
3.11. Die Initiativeigenschaft . . . . .	44
3.12. Der Algorithmus für den Kartenaustausch . . . . .	56





# 1. Einleitung

Tichu ist ein von Urs Hostettler entwickeltes und bei Fata Morgana erschienenes Kartenspiel [Fat]. Ein Spiel besteht aus mehreren Runden, in denen die Spieler anfangs jeweils neue Karten bekommen und diese ausspielen müssen, wobei sie Punkte sammeln. Tichu wird von vier Spielern gespielt, die auf zwei Mannschaften aufgeteilt werden. Ein zentrales Spielelement ist das Ankündigen eines "Tichu", mit dem ein Spieler ansagt, dass er in der aktuellen Runde als Erster alle Karten ausgespielt haben wird. Erfolgreiche Tichus bringen der Mannschaft des Spielers Punkte. Wird der Spieler allerdings nicht als Erster fertig, dann verliert die Mannschaft des Spielers Punkte.

Tichu ist aus Sicht der künstlichen Intelligenz ein interessantes Spiel, da es einige Merkmale besitzt, die den Entwurf einer künstlichen Intelligenz mittels klassischer Suchalgorithmen erschweren. Da in Tichu die Karten am Anfang einer Runde vollständig verteilt werden und der Spieler keine Information über die Karten der anderen Spieler hat, müssen Entscheidungen bei hoher Unsicherheit getroffen werden. Klassische Suchalgorithmen können dabei nicht verwendet werden, da eine übliche Modellierung der unsicheren Elemente mittels Zufallsknoten [RN05, S.177] zu einem sehr hohen Verzweigungsfaktor im Suchbaum führen würde, was das Treffen einer Entscheidung bei vertretbarem Zeitaufwand unmöglich macht. Trotzdem ist es für einen Spieler erforderlich abzuschätzen, ob er die Runde gewinnen kann oder nicht, um gegebenenfalls ein Tichu anzukündigen.

In dieser Arbeit werden grundlegende Konzepte zur Entwicklung einer künstlichen Intelligenz für Tichu dargestellt. Dafür wird gezeigt, wie von den Handkarten des Spielers zu Plänen für das Ausspielen der Karten abstrahiert werden kann. Wesentlich ist es dabei vorherzusagen, ob und wie die Handkarten ausgespielt werden können. Dafür wird der Begriff der "Initiative" eingeführt werden. Das wichtigste Element zur Entscheidungsfindung ist eine zentrale Bewertungsfunktion, auf die die meisten zu treffenden Entscheidungen reduziert werden können. Es wird detailliert dargestellt, welche Eigenschaften zur Bewertung einer Spielsituation verwendet werden und wie diese algorithmisch repräsentiert werden können. Dabei wird eine Spielstärke der KI angestrebt, die sich auf dem Niveau von menschlichen Spielern befindet. Auch erfahreneren menschliche Spieler sollen durch die KI noch gefordert werden.

Eine weitere Anforderung ist eine schnelle Entscheidungsfindung der KI. Dadurch soll es ermöglicht werden, dass auch bei Verwendung von schwächeren Hardwareplattformen lediglich kurze Wartezeiten entstehen, was eine interaktive Verwendung der KI ermöglicht. Konkret ist eine Rechenzeit für jede zu treffende Entscheidung von maximal einer Sekunde im Durchschnitt angestrebt.

Die ist nach Wissen des Autors die erste Arbeit, die sich mit Tichu beschäftigt, weswegen grundlegende Abstraktionen eingeführt werden, die erstmals weitergehende theoretische Untersuchungen von Tichu ermöglichen.

### **Gliederung**

Die Arbeit ist in folgender Weise gegliedert:

**Kapitel 2 – Das Kartenspiel Tichu:** In diesem Kapitel wird in das Spiel Tichu eingeführt. Es werden die Spielregeln erklärt und einige strategische Überlegungen vorgestellt, um eine Vorstellung zu vermitteln, wie sich Tichu Partien üblicherweise entwickeln.

**Kapitel 3 – Eine künstliche Intelligenz für Tichu:** Es werden theoretische Konzepte eingeführt, mit deren Hilfe eine KI für Tichu aufgebaut werden kann. Es wird weiter beschrieben, wie in Tichu Entscheidungen mit einer zentralen Bewertungsfunktion getroffen werden können. Die in dieser Bewertungsfunktion verwendeten Eigenschaften werden vorgestellt.

**Kapitel 4 – Ergebnisse:** Hier wird die erreichte Spielstärke diskutiert und die Ergebnisse von Laufzeitmessungen vorgestellt. Anschließend werden einige Vorschläge präsentiert, wie die KI weiter verbessert werden kann.

**Kapitel 5 – Zusammenfassung und Ausblick** fasst die Ergebnisse der Arbeit zusammen und stellt Anknüpfungspunkte vor.

## 2. Das Kartenspiel Tichu

Tichu ist ein von Urs Hostettler entwickeltes Kartenspiel, basierend auf einem traditionellen chinesischen Kartenspiel. Tichu erschien 1991 beim Schweizer Fata Morgana Verlag [Tica]. Die Spielregeln von Tichu sind auch beim Fata Morgana Verlag online einsehbar [Ticb]. Im Folgenden werden die Spielregeln und einige grundlegende strategische Überlegungen vorgestellt.

### 2.1. Überblick

Tichu wird mit 4 Spielern gespielt, diese teilen sich auf 2 Mannschaften auf. Die Spieler sitzen im Kreis und spielen nacheinander, üblicherweise gegen den Uhrzeigersinn. Dabei ist jeder zweite Spieler in der gleichen Mannschaft, was für gewöhnlich bedeutet, dass sich einander gegenüberstehende Spieler in der gleichen Mannschaft befinden.

Es gewinnt die Mannschaft, die zuerst 1000 Punkte erreicht. Punkte können im Rahmen von Runden erworben werden. Am Anfang jeder Runde werden die Karten auf die Spieler verteilt. Die Spieler versuchen nun ihre Karten möglichst schnell auszuspielen, und Stiche zu erzielen. Ein Spieler macht einen Stich, wenn die von ihm gelegten Karten von keinem anderen Spieler überspielt werden können oder wenn kein Spieler zu diesem Zeitpunkt spielen möchte. Sowohl ein schnelles Ausspielen der eigenen Karten als auch das Gewinnen möglichst vieler Stiche kann zu einem hohen Punktestand führen.

Zentrales Spielelement ist dabei die Ankündigung eines "Tichu". Am Anfang jeder Runde kann ein Spieler ein Tichu ankündigen, solange er noch keine seiner Handkarten ausgespielt hat. Wenn ein Spieler "Tichu" ankündigt, dann sagt er an, dass er als Erster alle Handkarten ausgespielt haben wird. Wenn er tatsächlich als Erster fertig wird, dann erhält seine Mannschaft 100 Punkte zusätzlich zu den Punkten, die die Mannschaft in dieser Runde durch die Stiche der beiden Spieler gewonnen hat. Wird ein anderer Spieler jedoch vorher fertig, dann verliert die Mannschaft, deren Spieler Tichu gerufen hat, 100 Punkte.

Da 100 Punkte bereits ein Zehntel eines gesamten Spiels sind, ist das korrekte Ankündigen von Tichus äußerst wichtig. Wenn eine Mannschaft sehr auf Sicherheit bedacht spielt und selten oder nie Tichu ankündigt, wird sie normalerweise gegen eine Mannschaft verlieren, die zumindest selten Tichu ruft. Diese wiederum verliert gegen eine dritte Mannschaft, die Tichu noch besser ankündigt. Aufgrund dieser Dynamik werden Tichu Partien vor allem dadurch gewonnen, dass ein Spieler korrekt vorhersagt, wann er selbst ein erfolgreiches Tichu ankündigen kann. Außerdem sollten gegnerische Tichus nach Möglichkeit verhindert werden.

## 2. Das Kartenspiel Tichu

---

Neben gewöhnlichen Tichus, auch "kleine" Tichus genannt, gibt es noch große Tichus. Dieses ist dem gewöhnlichen Tichu sehr ähnlich, muss aber bereits beim Austeilen der Spielkarten vor der Aufnahme der neunten Karte angesagt werden. Ein erfolgreiches großes Tichu gibt 200 Punkte, ein falsch angesagtes großes Tichu kostet jedoch die Mannschaft 200 Punkte.

Einen weiteren Bonus erhält eine Mannschaft, falls beide ihrer Spieler vor einem Spieler der gegnerischen Mannschaft fertig wurden. In diesem Fall erhält diese Mannschaft 200 Punkte, zuzüglich möglicher weiterer Punkte durch ein erfolgreiches großes oder kleines Tichu. Diese Situation wird "Doppelsieg" genannt. Stiche werden bei einem Doppelsieg nicht gezählt.

In jeder Runde Tichu werden weiterhin durch Stiche insgesamt 100 Punkte verteilt. Selbst bei vergleichsweise guten Tichu Spielern ist die Bedeutung des Punkterwerbs durch Stiche eher gering. Dies wird, neben dem Ankündigen von Tichus und der Möglichkeit von Doppelsiegen, noch dadurch verstärkt, dass der Spieler, der als letztes noch Karten auf der Hand hält, diese an die gegnerische Mannschaft abgeben muss. Falls der letzte Spieler Stiche erworben hat, werden diese dem Spieler gegeben, der als Erster alle Handkarten ausgespielt hatte.

Grundsätzlich ist ein Spieler in fast jeder Spielsituation bestrebt, möglichst schnell seine Karten auszuspielen:

- Man will Erster werden, um angekündigten Tichus nachzukommen und um die Möglichkeit eines Doppelsieges zu schaffen. Außerdem um gegnerische Tichus zu verhindern. Eine wichtige Ausnahme besteht, wenn der eigene Partner ein Tichu angekündigt hat. In diesem Fall sollte man ihn unterstützen und dann möglichst als Zweiter fertig werden.
- Wenn man nicht erster wird, will man als zweiter Spieler fertig werden, um einen Doppelsieg zu erreichen oder einen gegnerischen Doppelsieg zu verhindern.
- Schafft man es auch nicht als zweiter Spieler fertig zu werden, so will man wenigstens dritter Spieler werden, damit man nicht seine Handkarten, und eventuell auch seine Stiche, an die gegnerische Mannschaft verliert.

Es gibt in Tichu einige seltene Situationen, in denen auf Basis der Regeln unklar ist, wie die Regeln genau anzuwenden sind. So wird zum Beispiel nicht erklärt, wie vorzugehen ist, wenn beide Mannschaften gleichzeitig über die 1000 Punkte Grenze gelangen. Für einige dieser Probleme hat der Fata Morgana Verlag Antworten zur Verfügung gestellt [Fat].

### 2.2. Karten und Kombinationen

Tichu wird mit einem Deck aus 56 Karten gespielt. Davon sind 52 Karten im Wesentlichen identisch zu einem normalen Skatblatt. Es gibt vier Farben: Jade, Schwert, Pagode und Stern. Die Bezeichnung der Farben ist jedoch im Spiel nicht weiter wichtig, da für praktische Zwecke nur zwischen gleicher und ungleicher Farbe unterschieden werden muss. Aus diesem Grund werden in Beispielen die Farben der Karten auch nicht angegeben. Weiterhin

besitzen die Karten Werte wie bei einem gewöhnlichen Skatblatt, also Zahlen von 2 bis 10 und die Bildkarten Bube, Dame, König und Ass. Bei Tichu sind die Asse stets oberhalb des Königs und können nicht als 1 gespielt werden. Von jedem Wert existieren 4 Karten, für jede Farbe und Wert eine Karte.

Neben diesen 52 "normalen" Karten gibt es 4 Sonderkarten. Den Mah Jong, gelegentlich auch als 1 bezeichnet, den Drachen, den Phönix und den Hund. Bevor die Sonderkarten erklärt werden können, muss erläutert werden, wie Karten in Tichu kombiniert und ausgespielt werden können.

Karten werden in einer Runde in Form von Kombinationen ausgespielt. Eine Kombination ist eine Menge von Karten, die gewissen Eigenschaften genügt. In Tichu existieren die folgenden Typen von Kombinationen:

- Eine beliebige einzelne Karte stellt eine **Einzelkarte** dar, zum Beispiel eine einzelne 7. Der Wert einer Einzelkartenkombination entspricht dem Wert der Karte.
- Zwei Karten von beliebigem, aber gleichem Wert sind ein **Paar**. Ein Beispiel sind zwei 3er. Der Wert eines Paares entspricht dem Wert der Karten.
- Drei Karten von beliebigem, aber gleichem Wert stellen ein **Trippel**, auch Drilling, dar. Ein Beispiel hierfür sind drei 10er. Der Wert eines Trippel entspricht dem Wert der Karten.
- Ein **Fullhouse** besteht aus einem Drilling und einem Paar. Ein Beispiel sind ein Drilling bestehend aus drei 7ern sowie ein Paar bestehend aus zwei 8ern. Für den Wert eines Fullhouse wird immer nur der Drilling gezählt. Der Drilling darf nicht vom gleichen Wert sein wie das Paar<sup>1</sup>.
- Eine **Straße**, auch Reihe, besteht aus mindestens 5, sonst beliebig vielen Karten, deren Wert direkt aufeinanderfolgt. Ein Beispiel wäre die Straße 7-8-9-10-Bube. Der Wert eine Straße entspricht dem Wert der höchsten Karte.
- Eine Menge von aufeinanderfolgenden Paaren wird **Treppe**, oder Folge von Paaren, genannt. Ein Beispiel für eine Treppe ist ein 2er-Paar zusammen mit einem 3er-Paar. Eine Treppe kann aus beliebig vielen Paaren bestehen, solange alle direkt aufeinanderfolgen. Somit ist 3-3-4-4-5-5-6-6 eine gültige Treppe. Der Wert einer Treppe entspricht dem Wert des höchsten Paares.
- Wenn ein Spieler alle vier Karten eines bestimmten Wertes hält, dann besitzt er eine **Bombe**. Bomben haben die besondere Eigenschaft, dass sie zum einen jede andere Kombination, mit Ausnahme höherer Bomben und Straßenbomben schlagen. Zum anderen dürfen Bomben auch dann gespielt werden, wenn der Spieler nicht an der Reihe ist. Bomben können jederzeit eingeworfen werden. Der Wert einer Bombe entspricht dem Wert der Karten.

<sup>1</sup>Dies wäre theoretisch möglich z.B. durch 2-2-2-2-Phönix.

## 2. Das Kartenspiel Tichu

---

- Eine **Straßenbombe** ist eine Straße, bei der alle Karten dieselbe Farbe haben. Eine Straßenbombe kann nur von einer höheren Straßenbombe geschlagen werden. Außerdem können Straßenbomben, wie gewöhnliche Bomben, jederzeit eingeworfen werden. Wie bei einer Straße entspricht der Wert einer Straßenbombe dem Wert der höchsten Karte. Dabei gilt, dass längere Straßenbomben höher als kürzere Straßenbomben sind, unabhängig vom Wert der kürzeren Straßenbombe.

In Tichu dürfen nur Kombinationen von gleichem Typ und gleicher Länge aufeinander gespielt werden. Lediglich Bomben und Straßenbomben stellen eine Ausnahme dar, das sie jederzeit eingeworfen werden können.

Jetzt kann die Rolle der Sonderkarten erklärt werden. Der Spieler, der am Anfang einer Runde den Mah Jong besitzt, darf als Erster eine Kombination anspielen. Der Wert des Mah Jong entspricht dem einer Eins. Er kann damit nur als Einzelkarte gespielt werden oder als Teil einer Straße, die bei Eins anfängt. Sobald der Spieler den Mah Jong spielt, darf er einen Wunsch äußern. Dieser Wunsch ist ein Kartenwert, also 2 bis 10 oder eine der Bildkarten, Sonderkarten dürfen nicht gewünscht werden. Der Wunsch muss von dem ersten Spieler erfüllt werden, der dies aufgrund seiner Handkarten tun kann. Dafür muss der Spieler gegebenenfalls auch Sonderkarten nutzen. Wenn der Mah Jong beispielsweise in einer Straße gespielt wurde und der Spieler wünscht sich einen Buben, dann muss ein Spieler, abhängig von seinen Handkarten, eine 10-Bube-Phönix-König-Ass Straße ausspielen, obwohl er dies in der Regel nicht möchte.

Der Phönix ist in seiner Funktion ein Joker. Er kann anstelle jeder anderen Karte in einer Kombination gespielt werden und wird meistens verwendet, um lange und seltene Kombinationen auszuspielen, die der Spieler ohne den Phönix nicht spielen könnte. Der Phönix kann nie im Rahmen einer Bombe oder Straßenbombe gespielt werden. Wird der Phönix als Einzelkarte gespielt, was gelegentlich vorkommt, dann zählt er als einen halben Wert über der Karte, auf die er gespielt wird. Das heißt, wenn eine 5 liegt, und der Phönix wird gespielt, dann ist er höher als die 5, kann aber durch eine 6 wiederum überspielt werden. Wird der Phönix einzeln angespielt, dann zählt er als höher als der Mah Jong, jedoch niedriger als eine 2. Der Phönix zählt für die Mannschaft, die ihn als Stich besitzt, mit  $-25$  Punkten. Trotz des negativen Wertes ist der Phönix aufgrund der Flexibilität, die er einer Hand bringt, eine sehr wertvolle Karte. Dies gilt besonders, falls ein Tichu angekündigt wurde.

Der Drache ist eine Art besseres Ass. Er schlägt jede andere Einzelkarte, inklusive Phönix und Ass. Außerdem kann er nicht von einem einzeln ausgespielten Phönix überspielt werden. Die einzige Möglichkeit dem Drachen einen Stich zu verwehren ist das Ausspielen einer Bombe. Wenn mit einem Drachen ein Stich gewonnen wird, dann muss dieser einem der gegnerischen Spieler gegeben werden. Da der Drache mit 25 Punkten die wertvollste Karte nach Punkten ist, ist es eine gängige Strategie, den Drachenstich dem Spieler aus der gegnerischen Mannschaft zu geben, von dem man annimmt, dass er am ehesten als Letzter fertig wird. Dadurch besteht eine Chance, dass man die 25 Punkte zurückerhält. Auch wenn der Drachenstich der gegnerischen Mannschaft übergeben werden muss, behält der Spieler, der den Drachen ausgespielt hat, das Anspielrecht für den nächsten Stich.

Der Hund erlaubt es einem Spieler, das Anspielrecht an seinen Partner zu übergeben. Der Hund darf auf keine andere Karte gespielt werden. Es ist also notwendig, dass der Spieler zuerst selbst das Anspielrecht gewinnt, um es dann an seinen Partner zu übertragen. Der Hund ist vor allem nützlich, um seinen Partner zu unterstützen, wenn dieser Tichu angekündigt hat.

### 2.3. Spielablauf

Wie einleitend erwähnt, besteht das Spielziel darin, dass die eigene Mannschaft zuerst 1000 Punkte erreicht. Das Spiel ist dabei in mehrere Runden aufgeteilt.

Am Anfang einer Runde werden alle Karten an die Spieler ausgeteilt. Während des Austeilens können vor Aufnahme der neunten Karte große Tichus angesagt werden.

Nachdem alle Karten ausgeteilt wurden, wählt jeder Spieler drei Karten zum Austausch mit den anderen Spielern aus. Eine für seinen Partner, eine Karte für den links von ihm sitzenden Gegner und eine für den recht von ihm sitzenden Gegner. Sobald alle Spieler ihre Karten ausgewählt haben, wird der Austausch durchgeführt. Diese Phase des Spiels wird auch als "Schupfen" bezeichnet.

Es beginnt der Spieler mit dem Mah Jong, der Eins, auf der Hand. Dieser kann den Mah Jong als Einzelkarte oder als Teil einer Straße ausspielen und anschließend einen Wunsch äußern. Der Spieler muss dabei nicht mit dem Mah Jong anfangen, sondern kann auch eine andere Karte anspielen, allerdings kann ein Wunsch nur geäußert werden, wenn der Mah Jong direkt am Anfang der Runde gespielt wird. Des Weiteren ist der Spieler nicht gezwungen einen Wunsch zu äußern, der Mah Jong kann auch ohne Wunsch gespielt werden.

Als Nächstes dürfen die Spieler reihum über die gespielte Kombination spielen, falls sie es wollen und können<sup>2</sup>. Passen ist ebenfalls erlaubt. Grundsätzlich dürfen Kombinationen nur auf Kombinationen von gleichem Typ, aber mit einem höheren Wert gespielt werden. Das heißt, über ein 2er-Paar darf nur ein Paar mit einem Wert höher als 2 gespielt werden, zum Beispiel ein Paar bestehend aus zwei Assen. Bei manchen Kombinationen spielt zusätzlich noch die Länge der Kombination eine Rolle. Hier gilt, dass über eine Kombination nur mit einer Kombination der gleichen Länge gespielt werden kann. Wenn eine Straße der Länge 5 mit einer 10 als Wert der höchsten Karte liegt, dann darf eine Straße der Länge 8 mit einem Ass als höchsten Wert nicht gespielt werden, wohl aber die Teilstraße der Länge 5 mit einem Buben als höchste Karte.

Wenn eine Kombination von einem Spieler gespielt wird, auf die alle anderen Spieler passen, dann geht der Stich an den Spieler, der die letzte Kombination gespielt hat. Eine Ausnahme stellt der Drache dar. Wenn dieser einen Stich gewinnt, dann muss der Stich an einen der gegnerischen Spieler übergeben werden. Das Anspielrecht verbleibt jedoch bei dem Spieler,

<sup>2</sup>Oder im Falle eines Wunsches müssen.



## 2. Das Kartenspiel Tichu

---

der den Drachen gespielt hat. Der Spieler darf nun einen neuen Stich mit einer beliebigen Kombination starten.

Es werden solange weitere Stiche durchgespielt, bis nur noch ein Spieler Karten auf seiner Hand hält. Dann ist die Runde zu Ende und es wird ausgezählt. Dabei werden alle Punktgewinne und Punktverluste durch Ankündigungen von Tichu oder Großem Tichu verrechnet und gegebenenfalls der Doppelsiegbonus. Ebenfalls übergibt der letzte Spieler seine Handkarten dem gegnerischen Team und seine Stiche an den Spieler, der als Erster fertig wurde. Falls kein Doppelsieg vorliegt, dann werden jetzt die Karten ausgezählt. Die Karten besitzen dabei die folgenden Punktzahlen:

- Eine Fünf hat einen Wert von 5 Punkten.
- Eine Zehn hat einen Wert von 10 Punkten.
- Ein König hat einen Wert von 10 Punkten.
- Der Drache hat einen Wert von 25 Punkten.
- Der Phönix kostet den Besitzer 25 Punkte.
- Alle anderen Karten haben einen Wert von 0 Punkten.

Es werden solange weitere Runden gespielt, bis eine der Mannschaften 1000 Punkte erreicht hat.

### 2.4. Einige strategische Überlegungen

Um auch Lesern mit weniger Erfahrung im Spielen von Tichu einen Eindruck zu vermitteln, wie Tichu üblicherweise gespielt wird, werden einige grundlegende Strategien erklärt. Falls der Leser bereits mit Tichu vertraut ist, kann dieser Abschnitt übersprungen werden.

#### 2.4.1. Über das Erzielen von Punkten

Gerade als Anfänger ist es eine gute Idee, die Punkte durch Stiche weitgehend zu ignorieren und stattdessen zu versuchen, als Erster fertig zu werden. Die Abwägung zwischen einem Stich, der eine gewisse Menge an Punkten bringt, und einem schnelleren, vollständigen Ausspielen des Blattes, geht fast immer zugunsten eines schnelleren Ausspielens der Karten aus. In Tichu wird der erste Spieler häufig sehr überraschend fertig, da es gängig ist, eine längere Kombination als letzte oder vorletzte Kombination zu spielen, um dem Gegner nicht anzukündigen, dass man bald fertig sein wird. Überhaupt wird der erste Spieler häufig nach nur wenigen Zügen fertig, sodass es schon fatal sein kann, wenn man nur einen oder zwei Züge später als optimal möglich fertig wird, insbesondere wenn der Gegner Tichu angekündigt hat.

Wenn der erste Spieler bereits fertig ist, will man in der Regel einen Doppelsieg der eigenen Mannschaft erreichen oder einen gegnerischen Doppelsieg verhindern, da die 200 Punkte,

die die Mannschaft dadurch, von einem einzelnen Stich im Allgemeinen nicht aufgewogen werden. In Situationen, in denen der Spieler vermutet, dass er es nicht schaffen wird, als Zweiter fertig zu werden, kann es jedoch durchaus eine gute Idee sein, Punkte über Stiche zu erzielen.

Auch wenn nur noch zwei Spieler im Spiel sind, ist es in der Regel ratsam so schnell wie möglich fertig zu werden, damit die eigenen verbleibenden Handkarten nicht an das Gegnersteam verloren gehen. Falls der erste fertig gewordene Spieler aus dem Gegnersteam war, ist dies umso wichtiger, da der letzte Spieler seine Stiche dann an das Gegnersteam geben muss und damit möglicherweise Punkte an das Gegnersteam verloren gehen.

In jedem dieser Fälle wiegt die Anzahl verlorener oder gewonnener Punkte aufgrund von Tichus, Doppelsiegen oder der Platzierung des Spielers die Punkte eines einzelnen Stiche tendenziell auf. Es ist sehr schwer korrekt einzuschätzen, wann ein Kombination tatsächlich aufgrund der Punkte des Stichs gespielt werden sollte. Es wird daher zunächst empfohlen, den Wert eines Stichs zu ignorieren.

Da jede Runde, in der ein Spieler als Erstes fertig wurde und kein Tichu angekündigt hatte, für das Team einen Punkteverzicht darstellt, bietet es sich an, schnell zu lernen, wann es möglich ist, ein erfolgreiches Tichu zu rufen. Dies wiederum hängt vor allem von der Frage ab, ob man ein gutes Blatt besitzt.

### 2.4.2. Erkennen eines guten Blattes

In der Regel ist es möglich, ein Blatt auf viele unterschiedliche Arten nach Kombinationen aufzuteilen. Ein gutes Blatt besteht aus möglichst wenig Kombinationen, da pro Zug immer eine Kombination gespielt werden kann und der Spieler damit seine Chance verbessert, als Erster fertig zu werden.

Wenige Kombinationen allein reichen allerdings noch nicht aus. Der Spieler muss genügend Kombinationen auf seiner Hand halten, die es ihm erlauben das Anspielrecht zu erlangen. Die Anzahl der Kombinationen, die es dem Spieler erlauben das Anspielrecht zu erlangen, wird an dieser Stelle als die Initiative des Spielers bezeichnet. In 3.5 wird später ein genauerer Initiativbegriff eingeführt.

Gerade sehr lange Kombinationen, zum Beispiel Straßen, die aus mehr als 7 Karten bestehen, setzen für ihre Spielbarkeit meistens voraus, dass der Spieler das Anspielrecht besitzt, da es unwahrscheinlich, dass eine derartige Kombination von einem Gegner angespielt wird. Andererseits ist es wiederum unwahrscheinlich, dass ein weiterer Spieler ebenfalls eine derartige Kombination hält, was bedeutet, dass man das Anspielrecht in der Regel zurückerhält.

Bei den meisten Blättern gibt es allerdings auch Kombinationen, bei denen es unwahrscheinlich ist, dass sie auf andere Kombinationen gespielt werden können und die gleichzeitig selbst das Anspielrecht nicht wieder zurückholen. Dies sind vor allem niedrige Einzelkarten und niedrige Paare. Eine einzelne 2 kann beispielsweise lediglich auf den Mah Jong gespielt werden oder wenn der Partner den Hund spielt. Damit muss für derartige Kombinationen

## 2. Das Kartenspiel Tichu

---

in der Regel einmal das Anspielrecht geopfert werden. Ein gutes Blatt sollte einen genügend großen Initiativwert besitzen, um alle derartigen Kombinationen ausspielen zu können. Noch besser ist es, wenn niedrige Einzelkarten oder niedrige Paare überhaupt nicht vorhanden sind.

Zentral ist damit die Abwägung zwischen einem möglichst schnell ausspielbaren Blatt und genügend vielen guten Kombinationen, um das Blatt überhaupt ausspielen zu können. Diese Abwägung kann gut am Besitz zweier Asse veranschaulicht werden. Diese können als ein Paar oder als zwei Einzelkarten gespielt werden. Im einen Fall wird der Spieler einen Zug schneller fertig, erhält aber nur einmal das Anspielrecht. Im anderen Fall kann er das Anspielrecht zweimal gewinnen. Gerade zwei Asse werden nur selten als Paar gespielt. Dagegen werden zwei 2er nur selten als Einzelkarten gespielt.

Trotz der negativen Punkte ist der Phönix die wahrscheinlich beste Karte des Spiels, da er viele neue Kombinationsmöglichkeiten in einem Blatt schafft. Damit erhöht er die Flexibilität des Spielers enorm. Wenn ein Spieler den Phönix auf der Hand hält, dann sollte er sich gut überlegen, ob er nicht Tichu ankündigen möchte. Kommt noch der Drache und ein paar Bildkarten dazu, sollte fast sicher Tichu angekündigt werden.

### 2.4.3. Unterstützen des Partners

Wenn der Partner Tichu gerufen hat, dann sollte der Spieler auf jeden Fall versuchen, seinen Partner möglichst zu unterstützen. Dabei ist das erfolgreiche Tichu wichtiger als jeder andere Punkteerwerb. Selbst wenn alle Kartenpunkte an die gegnerische Mannschaft gehen, was äußerst selten der Fall ist, haben immerhin beide Mannschaften 100 Punkte erworben.

Ein Tichu kann man vor allem unterstützen, indem man niedrige Einzelkarten oder Paare anspielt, da Einzelkarten und Paare in nahezu jeder Hand zu finden sind. Von dieser Strategie absehen sollte man, wenn der nachfolgende Gegner ebenfalls Tichu gerufen hat. In diesem Fall sollte man sein Verhalten tendenziell genau umkehren und eher hohe und seltene Kombinationen spielen. Vor allem sollte man es vermeiden, dass ein Gegner, der Tichu angekündigt hat, das Ausspielrecht mittels eigentlich niedriger Kombinationen erhält.

Falls der Spieler den Hund besitzt, sollte dieser möglichst schnell gespielt werden. Dies wird aus zwei Gründen empfohlen. Zum einen wird ein Spieler, der Tichu gerufen hat, häufig recht schnell fertig. Wenn der Partner also Tichu gerufen hat, ist eine frühe Unterstützung notwendig, um ihn überhaupt zu unterstützen. Und selbst wenn sein Blatt nicht optimal ist, kann die Wahrscheinlichkeit, dass ein Spieler des anderen Teams vor ihm fertig wird, mithilfe des Hundes reduziert werden. Zum anderen geht das Anspielrecht durch den Hund an den nächsten Spieler, der noch Karten auf der Hand hat, falls der Partner schon fertig ist. Es kommt daher vor, dass der Hund dem Gegner das Anspielrecht gibt, wenn der Hund nicht früh genug gespielt wird.

## 3. Eine künstliche Intelligenz für Tichu

Tichu ist kein Spiel im gänzlich klassischen Sinne der künstlichen Intelligenz. Klassische Suchalgorithmen können deswegen nicht verwendet werden. Darum wird ein sehr einfacher Suchalgorithmus zusammen mit einer komplexen Bewertungsfunktion verwendet. Dies ist notwendig, um trotz einer sehr geringen Suchtiefe Aussagen über den Verlauf des Spiels machen zu können. Dieser Ansatz wird in diesem Kapitel näher erläutert.

### 3.1. Tichu aus Sicht der künstlichen Intelligenz

Klassische Spiele der künstlichen Intelligenz sind Spiele, bei denen zwei Spieler abwechselnd ziehen, die Umgebung deterministisch und vollständig beobachtbar ist [RN05, S.161ff]. Zudem sind klassische Spiele Nullsummenspiele, das heißt, dass die Summe von Gewinn und Verlust der Spieler null beträgt. Was dabei Gewinn und Verlust sind, hängt vom Spiel ab. Dies können Punkte sein, oder, wie in Tichu, dass eine Mannschaft das Spiel gewinnt oder verliert. Die Spieler werden auch Agenten genannt. Dabei ist ein Agent in der KI ein Objekt, das seine Umwelt mittels Sensoren wahrnimmt und seine Umgebung mittels Aktuatoren manipuliert [RN05, S.32].

Tichu weicht von dieser Klassifikation insofern ab, als dass in Tichu vier Agenten aktiv sind, üblich sind nur 2 Agenten. Weiter verhält sich die Umgebung beim Einwurf von Bomben nicht diskret, da die Entscheidung zum Einwurf einer Bombe jederzeit gefällt und eine Bombe eingeworfen werden kann. Zudem ist die Umgebung in Tichu nicht vollständig beobachtbar, da dem Spieler die Information fehlt, welche Karten die anderen Spieler auf ihrer Hand halten. Tichu ist ein Nullsummenspiel, da am Ende einer Partie immer eine der Mannschaften gewonnen und die andere verloren hat.

Aus Sicht der künstlichen Intelligenz ist ein Tichuspieler ein Agent, die anderen Spieler werden ebenfalls als Agenten modelliert<sup>1</sup>. Das Brett, auf das die Stiche gespielt werden, die erworbenen Stiche, die Handkarten des Spielers selbst, sowie die Handkarten der anderen Spieler stellen die Umgebung dar.

Umgebungen lassen sich nach bestimmten Kriterien klassifizieren [RN05, S.40ff]:

- Die Umgebung ist nur teilweise beobachtbar, da der Spieler die Karten der Mitspieler nicht kennt, mit Ausnahme der Karten, die er getauscht hat.

<sup>1</sup>In manchen Spielen ist es möglich, andere Spieler als Teile der Umgebung zu modellieren.

### 3. Eine künstliche Intelligenz für Tichu

---

- Tichu hat deterministische Aspekte, da die Aktionen des Spielers immer zu den gleichen vorhersehbaren Ergebnissen führen. Die Aktionen der anderen Spieler können jedoch nicht vorhergesehen werden, was eine Behandlung als stochastisches Spiel erforderlich macht.
- Tichu wird über mehrere Runden gespielt und ist insofern ein episodisches Spiel. Innerhalb einer Runde hängt jedoch die Aktion eines Spielers von seinen vorherigen Aktionen ab, daher ist Tichu innerhalb einer Runde ein sequenzielles Spiel.
- Die Zeit in Tichu verläuft getaktet entsprechend den Aktionen der Spieler, damit ist Tichu in dieser Hinsicht ein diskretes Spiel.
- Ein Spieler in Tichu interagiert mit anderen Agenten, wodurch er sich in einer Multi-Agenten Umgebung befindet.

Aus Gründen der Einfachheit der Modellierung und dem geringen angenommenen Einfluss auf die Spielstärke wurde das Einwerfen von Bomben außerhalb der eigentlichen Zugreihenfolge im Rahmen der hier vorgestellten künstlichen Intelligenz nicht modelliert.

#### 3.2. Der Basisalgorithmus

Aus der Klassifikation von Tichu im letzten Abschnitt ergeben sich einige Herausforderungen an das Design einer künstlichen Intelligenz. Künstliche Intelligenzen für klassische Spiele nutzen häufig Varianten des Minimax-Algorithmus [RN05, S.165].

Dessen Idee besteht darin, dass der Spieler seinen Nutzen in einer Aktion immer maximal hoch wählen sollte, während der gegnerische Spieler versucht, den Nutzen für den Spieler zu minimieren. Theoretisch kann für eine Spielsituation eines klassischen Spiels die Menge der Aktionen und der sich daraus ergebenden Folgezustände aufgezählt werden. Aus Menge der Aktionen wählt der Spieler den Zug, der seinen Nutzen maximiert. Um herauszufinden, welcher Zug den Nutzen des Spielers maximiert, muss untersucht werden, welche Züge dem Gegner in den Folgezuständen zur Verfügung stehen. Da angenommen wird, dass der Gegner optimal spielt, werden alle Folgezustände untersucht und der Zustand ausgewählt, der für den ursprünglichen Spieler den Nutzen minimiert. An dieser Stelle sieht man, dass sich der Prozess wiederholen muss, da sich der Nutzen der Folgezustände wiederum aus einer Aktion des ursprünglichen Spielers ergibt. Die Anzahl der Wiederholungen ist jedoch endlich, falls sich das Spiel nach einer endlichen Anzahl von Zügen immer in einem Endzustand befindet. Die Bewertungsfunktion, die den Nutzen des Spielers im Endzustand berechnet, ist spielabhängig, für manche Spiele reicht jedoch eine einfache Codierung von Sieg, Niederlage und Unentschieden<sup>2</sup> oder eine relative Punktzahl der Spieler. Da das Ergebnis der Endzustände über die Rekursion zurück propagiert wird, kann theoretisch schon im aktuellen Zug festgestellt werden, welcher Spieler gewinnt, angenommen, dass der Gegner optimal spielt.

<sup>2</sup>Zum Beispiel mit Sieg=1, Niederlage=-1, Unentschieden=0.

**Algorithmus 3.1** Der Minimax-Algorithmus

$$\text{Minimax}(n) = \begin{cases} \text{Utility}(n) & n \text{ ist Endzustand} \\ \max_{s \in \text{Nachfolger}(n)} \text{Minimax}(s) & \text{bei } n \text{ ist der Spieler am Zug} \\ \min_{s \in \text{Nachfolger}(n)} \text{Minimax}(s) & \text{bei } n \text{ ist der Gegner am Zug} \end{cases}$$

Der Algorithmus kann als Baum dargestellt werden, wobei die Wurzel der aktuelle Spielzustand ist. Die Knoten modellieren die aus den Aktionen resultierenden Spielzustände, wobei auf jeder Ebene des Baumes auf allen Knoten der Nutzen der Nachfolgerknoten entweder minimiert oder maximiert wird, abhängig davon, welcher Spieler am Zug ist.

Für praktische Anwendungen ist der Minimax-Algorithmus wenig geeignet, da der vollständige Baum für interessante Spiele wie Schach oder Tichu nicht in praxisnaher Zeit durchsucht werden kann.

Eine Möglichkeit dieses Problem zu lösen, besteht in der Erweiterung des Minimax Algorithmus zur Cutting-Off-Suche [RN05, S.173]. Hier wird der Baum nur bis zu einer gewissen Tiefe aufgebaut. Die sich ergebenden Blattknoten sind noch keine Endzustände des Spiels. Anstatt zu überprüfen, wer an einem Blattknoten das Spiel gewonnen hat, wird stattdessen der Zustand des Spiels aus Sicht des Spielers bewertet. Dies erfolgt mithilfe einer Bewertungsfunktion, die einem Spielzustand eine reelle Zahl, eine Abschätzung des erwarteten Nutzens der Situation, zuordnet.

Die Situation in Tichu ist noch weiter erschwert, da die Karten der anderen Spieler nicht bekannt sind. Grundsätzlich kann diese fehlende Information in der KI modelliert werden, indem man nicht nur alle Aktionen betrachtet, die einem anderen Spieler in einer Situation zu Verfügung stehen, sondern alle Aktionen innerhalb aller möglichen Verteilungen der Handkarten auf die anderen Spieler. In diesem Fall würde man an einem Knoten eine diskrete Wahrscheinlichkeitsverteilung erhalten. Der Wert eines Knotens ließe sich dann errechnen, indem man die Wahrscheinlichkeit der angenommenen Verteilung mit dem Wert der besten Aktion multipliziert und dies über alle möglichen Verteilungen aufsummiert. Aus dieser Überlegung ergibt sich der Expectiminimax-Algorithmus [RN05, S.177].

Diese Strategie kann für Tichu nicht gewählt werden, da gerade am Anfang einer Runde sehr viele unbekannte Karten existieren und es deswegen sehr viele Möglichkeiten gibt, die unbekanntes Karten auf andere Spieler zu verteilen. Da eine der Anforderungen an die KI darin besteht, dass auch eine interaktive Nutzung durch menschliche Spieler möglich ist, wurde von dieser Möglichkeit abgesehen. Stattdessen wird lediglich über die eigenen Aktionen gesucht und die sich daraus ergebenden Spielzustände bewertet. Das heißt, es wird eine Suche mit einer Suchtiefe von 1 verwendet. Algorithmus 3.2 zeigt den aus diesen Überlegungen resultierenden Algorithmus.

Beim Minimax-Algorithmus reicht es noch aus, wenn die Nutzenfunktion lediglich wiedergibt, wie gut dieser erreichte Endzustand für den Spieler ist. Bereits bei einer Cutting-Off

---

**Algorithmus 3.2** Idee des Suchalgorithmus zur Entscheidungsfindung

---

```
procedure FINDBESTACTION(state, actions)
  bestAction  $\leftarrow$  NULL
  bestUtility  $\leftarrow$   $-\infty$ 
  for all a  $\in$  actions do
    s  $\leftarrow$  MAKESUCCESSOR(state, a)
    utility  $\leftarrow$  GETUTILITY(s)
    if utility > bestUtility then
      bestUtility  $\leftarrow$  utility
      bestAction  $\leftarrow$  action
    end if
  end for
  return bestAction
end procedure
```

---

Suche ist eine aufwendigere Nutzenfunktion notwendig, da diese modellieren muss, ob der Zustand auch gut für zukünftig erreichbare Zustände ist. Dabei hilft bei einer Cutting-Off Suche die Simulation der Züge der Gegner, um besonders schlechte Aktionen auszuschließen und besonders gute Aktionen auszuwählen. In der hier verwendeten trivialen Cutting-Off Suche werden die Aktionen der anderen Spieler überhaupt nicht simuliert, weswegen eine präzise Nutzenfunktion besonders wichtig ist. Diese wird in 3.6 vorgestellt werden. Zunächst wird jedoch der verwendete Ansatz weiter präzisiert.

### 3.3. Reduktion der Entscheidungen

Im letzten Abschnitt wurde erklärt, dass ein Suchalgorithmus mit Suchtiefe 1 verwendet wird, bei dem sich ergebende Spielzustände mittels einer Bewertungsfunktion bewertet werden. Es steht noch aus zu erklären, was genau bewertet wird. Dafür ist zunächst zu bemerken, dass Spieler unterschiedliche Entscheidungen zu fällen haben. Die KI soll nicht nur den Standardfall abdecken, die Entscheidung, ob und wenn ja welche Kombination als Nächstes gespielt werden soll. Die KI soll bei allen anderen in Tichu auftretenden Situationen starke Entscheidungen treffen.

In Tichu gibt es folgende Szenarien, die eine Entscheidung erfordern:

- Standardsituation: Spielen oder passen auf eine bereits liegende Kombination oder Anspielen eines neuen Stichs. Beim Anspielen darf der Spieler nicht passen.
- Kartenaustausch: Auswählen, welche Karten am Anfang des Spiels mit den anderen Spielern ausgetauscht werden sollen.
- Tichu-Ankündigung: Es muss entschieden werden, ob ein Tichu gespielt werden soll.

- Großes-Tichu-Ankündigung: Es muss entschieden werden, ob ein großes Tichu gespielt werden soll.
- Wünschen: Der Spieler darf einen Wunsch äußern, nachdem er den Mah Jong gespielt hat.
- Drachen-Verschenken: Wenn der Drachen einen Stich gewonnen hat, dann muss dieser an einen Gegner verschenkt werden.

Apriori scheint für jede Situation eine eigene Behandlung erforderlich. Allerdings sind sich einige der Entscheidungen teils sehr ähnlich, weswegen sie, mit Ausnahme des Wünschens und dem Drachen-Verschenken, mit einer einzelnen zentralen Bewertungsfunktion und spezifischen ergänzenden Heuristiken durchgeführt werden können.

Um zu verstehen, wie dies erreicht werden kann, wird das Konzept der Partition benötigt. Eine Partition in Tichu folgt der gewöhnlichen mathematischen Auffassung des Begriffs der Partition und stellt eine Untergliederung einer Menge von Karten in mehrere Teilmengen dar, die vereinigt wiederum die Ausgangsmenge geben. Keine Karte darf in zwei der Untermengen auftreten.

#### **Definition 1 (Gültigkeit einer Partition)**

*Eine Partition von Tichukarten ist gültig genau dann, wenn alle ihre Elemente gültige Kombinationen entsprechend den Spielregeln von Tichu sind.*

Im Folgenden wird "Partition" ausschließlich im Sinne von "gültige Partition" verwendet, wenn nicht anders angegeben. Mithilfe des Partitionsbegriffs kann der Begriff der "Aktion" in Tichu definiert werden.

#### **Definition 2 (Aktion eines Spielers)**

*Eine Aktion eines Spielers ist ein 2-Tupel  $(p, c)$ , wobei  $p$  eine gültige Partition der Handkarten des Spielers ist. Die Kombination  $c$  ist Element der Partition und zudem im aktuellen Spielzustand ausspielbar.*

Dabei darf die Kombination auch die leere Menge sein, wenn der Spieler passen will und darf. Wenn der Spieler eine Entscheidung zu fällen hat, dann muss er eine Aktion auswählen, die von ihm gespielt wird.

Es gibt einige Gründe, die für dieses Modell einer Aktion sprechen. Dass die zu spielende Kombination bewertet werden muss, scheint offensichtlich, da sie dem Spieler selbst unter Umständen das Anspielrecht verschafft, die Zugmöglichkeiten der Gegner einschränken kann oder, wenn der Spieler nur noch eine Kombination auf der Hand hält, die Runde für diesen Spieler beendet. Die zu spielende Kombination sollte also Teil des Aktionsmodells sein.

Die Auswahl einer Partition löst das Problem, dass bei der Auswahl einer zu spielenden Kombination zunächst unklar ist, wie die verbleibenden Karten gespielt werden. Es ist unklar,



### 3. Eine künstliche Intelligenz für Tichu

---

welche Kombinationen durch die Karten gebildet werden sollen. Und da die Kombinationen unklar sind, ist es schwer abzuschätzen, ob man sich in einer guten oder schlechten Spielsituation befindet, das heißt unter anderen, ob man ausreichend Initiative zum hinreichend schnellen Ausspielen der eigenen Karten besitzt. Über das Konzept der Partition ist es dagegen möglich, die nächsten Züge des Spielers vor auszuplanen.

Neben der Partition und der zu spielenden Kombination wird auch noch der restliche Weltzustand berücksichtigt, insbesondere die von den anderen Spielern bereits gespielten Karten. Da die Suchfunktion jedoch nur über den nächsten eigenen Zug sucht und der Weltzustand damit für alle möglichen Partition-Kombination Tupel zu einem Entscheidungszeitpunkt gleich ist, wird er hier nicht separat in der Bewertungsfunktion modelliert.

Aus diesem Modell einer Aktion in Tichu ergibt sich, dass die Bewertungsfunktion folgende Form besitzt:

$$(3.1) \quad \text{GetUtility} : (p, c) \rightarrow \mathbb{R}$$

Dabei ist  $p$  eine Partition und  $c$  eine Kombination. Zusammen bilden sie das Aktionstupel. Die Bewertungsfunktion soll so konstruiert werden, dass sie für die beste Aktion ihr Maximum annimmt. Außerdem sollen mit der Bewertungsfunktion nicht nur Zustände untereinander verglichen werden. Es sollen auch absolute Aussagen getroffen werden können. Es kann zum Beispiel festgestellt werden, ob einer Partition der Karten, die ein Spieler am Anfang einer Runde bekommen hat, besonders gut oder schlecht ist.

Angesichts dieser Definitionen können die vom Spieler zu treffenden Entscheidungen neu formuliert werden.

- Standardsituation: Finde die beste Aktion. Beim Anspielen darf die Kombination nicht leer sein. In Algorithmus 3.3 ist der Algorithmus für die Standardsituation ohne Berücksichtigung des Anspielens dargestellt.
- Kartenaustausch: Bestimme für jedes 3-Tupel an Karten den Nutzen der verbleibenden Karten. Wähle das 3-Tupel, für das die verbleibenden Karten den höchsten Nutzen darstellen. Hier werden zusätzliche Heuristiken genutzt, um den Partner zu unterstützen und den Gegner zu sabotieren.
- Tichu-Ankündigung: Finde ein Aktionstupel mit einem Wert größer als ein vordefinierter Schwellwert. Dies ergibt dann den in Algorithmus 3.5 dargestellten Algorithmus. Dieser berücksichtigt noch zusätzlich, dass kein Tichu angekündigt werden sollte, wenn der Partner bereits ein Tichu spielen will. Ebenfalls sollte ein Tichu nur selten gespielt werden, wenn der Gegner ein Tichu angekündigt, da daraus folgt, dass wahrscheinlich zumindest einer der Gegner gute Karten hat.
- Großes-Tichu-Ankündigung: Finde ein Aktionstupel mit einem Wert größer als ein vordefinierter Schwellwert. Der Schwellwert kann von dem Schwellwert abweichen, der für Ankündigung eines gewöhnlichen Tichus verwendet wird. Da das Blatt des Spielers zum Zeitpunkt der Entscheidung noch unvollständig ist, wird die Bewertungsfunktion von zusätzlichen Heuristiken unterstützt.

**Algorithmus 3.3** Der Suchalgorithmus der Tichu KI

---

```

procedure FINDBESTACTION(handCards, playerId)
  partitions  $\leftarrow$  GETPARTITIONS(handCards)
  maxUtility  $\leftarrow -\infty$ 
  bestAction  $\leftarrow$  NULL
  for all  $p \in$  partitions do
    combos  $\leftarrow$  GETCOMBINATIONS( $p$ )
    for all  $c \in$  combos do
       $\triangleright$  Erzeuge das Aktions-2-Tupel
      action  $\leftarrow$  MAKEACTION( $p, c$ )
       $\triangleright$  Berechne den Nutzen des aus der Aktion erreichten Folgezustands
      utility  $\leftarrow$  GETUTILITY(action)
      if utility > maxUtility then
        maxUtility  $\leftarrow$  utility
        bestAction  $\leftarrow$  action
      end if
    end for
     $\triangleright$  Passen berücksichtigen
    action  $\leftarrow$  MAKEACTION( $p, \text{NULL}$ )
    utility  $\leftarrow$  GETUTILITY(action)
    if utility > maxUtility then
      maxUtility  $\leftarrow$  utility
      bestAction  $\leftarrow$  action
    end if
  end for
  return bestAction
end procedure

```

---

- Wünschen: Das Wünschen wird gesondert in 3.7.2 behandelt.
- Drachen-Verschenken: Das Verschenken eines Drachenstichs wird gesondert in 3.7.3 behandelt.

Die Tichu Schwellwerte werden als Parameter der Bewertungsfunktion behandelt. Aus diesem Grund werden sie im Zusammenhang mit der Erklärung der Bewertungsfunktion in 3.6.7 näher erläutert. Dort wird auch auf die Heuristiken, die für die Ankündigung eines großen Tichus verwendet werden, näher eingegangen.

Für die Standardsituation und den Kartentausch reicht es aus, wenn die Bewertungsfunktion einen relativen Vergleich zwischen Aktionen zulässt. Für die Bewertung beim Ankündigen eines großen oder kleinen Tichu ist es notwendig, dass das Ergebnis der Bewertungsfunktion auch ein absolutes Maß für die Güte des Spielzustandes darstellt.

Beim Anspielen ist passen nicht erlaubt, daher wird *bestAction* zunächst nicht initialisiert und *bestUtility* auf  $-\infty$  gesetzt. Da jeder Spieler, der noch Karten hat, noch mindestens eine

### 3. Eine künstliche Intelligenz für Tichu

---

---

#### Algorithmus 3.4 Vorläufiger Algorithmus für den Kartenaustausch

---

```
procedure FINDBESTEXCHANGE(handCards)
  bestUtility  $\leftarrow -\infty$ 
  bestExchange  $\leftarrow NULL$ 
  ▷ Iteriere über alle Kartenaustausch 3-Tupel (linker Spieler, rechter Spieler, Partner)
  for all exchange = (l, r, p)  $\in$  handCard do
    remainingCards  $\leftarrow$  REMOVECARDS(handCards, exchange)
    utility  $\leftarrow$  COMBINE(GETMAXUTILITY(remainingCards), GETEXCHANGEUTILITY(exchange))
    if utility > bestUtility then
      bestUtility  $\leftarrow$  utility
      bestExchange  $\leftarrow$  exchange
    end if
  end for
  return bestExchange
end procedure
```

---

---

#### Algorithmus 3.5 Algorithmus für die Tichu-Ankündigung

---

```
procedure CALLTICHU(handCards)
  bonus = 0
  if HASENEMYCALLEDTICHU() then
    bonus = ENEMY_TICHU_THRESHOLD_BONUS
  end if
  bestUtility  $\leftarrow$  GETMAXUTILITY(handCards)
  if  $\neg$  HASPARTNERCALLEDTICHU()  $\wedge$  bestUtility > TICHU_THRESHOLD + bonus then
    return true
  else
    return false
  end if
end procedure
```

---

Partition mit einer Kombination auf seiner Hand hält und jeder Wert von *GetUtility* größer als  $-\infty$  ist, wird immer tatsächlich eine Aktion gewählt.

Algorithmus 3.4 zeigt den vorläufigen Algorithmus für den Kartentausch. Dieser wird in 3.7.1 weiter verfeinert. Die Funktion *GetMaxUtility* ist dabei identisch zu *FindBestAction*, es wird lediglich *maxUtility* zurückgegeben, anstelle von *bestAction*. Des Weiteren muss noch das Austausch tupel selbst bewertet werden, dazu dient *GetExchangeUtility*. Und der Nutzen des Austausch tupels muss mit dem Nutzen der verbleibenden Karten zu einem Gesamtnutzen kombiniert werden, diesen errechnet *Combine*.

Der Kartenaustausch erfordert eine zusätzliche Iteration über alle 3-Tupel. Da jeder Spieler mit 14 Karten startet, ergeben sich  $14 \cdot 13 \cdot 12 = 2184$  Möglichkeiten Karten mit den anderen Spielern auszutauschen. Diese hohe Anzahl an Bewertungsvorgängen wird durch die etwas

geringere Anzahl an zu bewertenden Karten nicht aufgewogen. Die Bewertung dauert für eine praktische Verwertbarkeit zu lange. Daher ist für den Kartentausch eine heuristische Reduzierung der 3-Tupel auf die Tupel notwendig, die tatsächlich Kandidaten für den Austausch darstellen.

Die einzigen Aktionen, die gesondert behandelt werden, sind das Wünschen und das Verschenken des Drachens. Da das Wünschen jedoch nur geringen Einfluss auf das Spiel hat und es nicht besonders schwierig ist, hier eine gute Aktion auszuwählen, wird diese Entscheidung mittels einem einfachen Satz an Regel getroffen, wie in 3.7.2 näher ausgeführt wird. Auch das Verschenken des Drachens hat nur einen geringen Einfluss auf den Verlauf des Spiels, weswegen auch hier auf eine einfache Heuristik zurückgegriffen wird, diese wird in 3.7.3 vorgestellt.

Mit dieser Konstruktion reicht eine einzige Bewertungsfunktion, die teilweise von zusätzlichen Heuristiken unterstützt wird, aus, um alle wichtigen Entscheidungen in Tichu treffen zu können.

### 3.4. Analyse der Handkarten

Die Analyse der Handkarten eines Spielers erfolgt in zwei Schritten. Zuerst wird die Menge aller Kombinationen berechnet, die aus den Handkarten konstruiert werden kann. Basierend auf dieser Menge von Kombinationen können die Partitionen aufgebaut werden. Der Umweg über die Menge der Kombinationen ist notwendig, um sicherzustellen, dass später tatsächlich nur gültige Partitionen aufgebaut werden, beziehungsweise, um überflüssige Nachprüfungen zu vermeiden.

#### 3.4.1. Bestimmung aller Kombinationen einer Kartenmenge

Um alle Kombinationen zu erhalten, die aus einer Menge von Karten konstruiert werden können, wird diese Kartenmenge auf Kombinationen der einzelnen Typen von Kombinationen untersucht. Dabei werden Einzelkarten, Paare, Drillinge, Bomben und Straßen jeweils unabhängig von allen anderen Typen untersucht. Die Kombinationen vom Typ Fullhouse kann aus den Mengen der Paare und Drillinge errechnet werden. Treppen können ebenfalls aus der Menge der Paare errechnet werden. Das Auffinden von Straßenbomben kann in die Berechnung der Straßen integriert werden.

Bei den hier vorgestellten Algorithmen wird die Einbeziehung des Phönix jeweils nicht aufgeführt, da die Behandlung des Phönix die Übersichtlichkeit der Algorithmen deutlich verringert. Grundsätzlich wird er immer mit derselben Strategie behandelt. Bei der Wahl der nächsten Karte für eine Kombination wird immer geprüft, ob der Phönix vorhanden ist und nicht bereits in der Konstruktion der Kombination verwendet wird. Ist der Phönix verfügbar, dann wird er als in Benutzung markiert und die Konstruktion der Kombination fortgesetzt.

---

**Algorithmus 3.6** Berechnung der Paare

---

```
procedure CALCULATEALLPAIRS(handCards)
  pairs ← MAKELIST()
  ▷ Iteriere über die Kartenwerte 2 bis Ass.
  for all value ∈ Card.VALUES do
    ▷ Jeder Farbe wird ein Index zugeordnet, über diesen wird iteriert
    for firstColor = 1 → 4 do
      ▷ Suche nach der durch Farbe und Wert spezifizierten Karte im Deck
      firstCard ← Card.GET(firstColor, value)
      if not handCards.CONTAINS(firstCard) then
        continue
      end if
      for secondColor = firstColor + 1 → 4 do
        secondCard ← CARD.GET(secondColor, value)
        if not handCards.CONTAINS(value, secondColor) then
          continue
        end if
        pairs.ADD(MAKEPAIR(firstCard, secondCard))
      end for
    end for
  end for
  return pairs
end procedure
```

---

Bei Kombinationen, die aus anderen Kombinationen zusammengesetzt werden, wird zusätzlich berücksichtigt, ob der Phönix bereits in einer der Unterkombinationen verbaut wurde.

Die Berechnung der Einzelkartenkombination ist trivial, es wird für jede der Handkarten eine Einzelkartenkombination erzeugt.

Algorithmus 3.6 zeigt die Berechnung aller Paare. Dabei wird für alle Kartenwerte über alle Farben iteriert, da eine Farbe und ein Wert zusammen eine bestimmte Karte darstellen. Es wird überprüft, ob die Karte auf der Hand des Spielers zu finden ist. Durch den Offset in der inneren Schleife wird sichergestellt, dass alle 2-Tupel an möglichen Karten desselben Wertes genau einmal ausgewählt werden. Hält der Spieler beide ausgewählten Karten auf seiner Hand, so wird das dazu passende Paar zur Liste der Paare auf der Hand des Spielers hinzugefügt.

Die Berechnung der Drillinge kann analog zu der Berechnung der Paare durchgeführt werden, es wird lediglich eine dritte innere Schleife in Algorithmus 3.6 hinzugefügt.

Die Berechnung der Bomben ist besonders einfach, da hierzu einfach für jeden Wert geprüft wird, ob der Spieler die Karten aller vier Farben auf seiner Hand hält. Da der Phönix

**Algorithmus 3.7** Berechnung der Treppen

---

```

pairs ← CALCULATEALLPAIRS(handCards)
stairs ← MAKELIST()
CALCULATEALLSTAIRS(MAKELIST(), SORTBYVALUE(pairs), stairs)
procedure CALCULATEALLSTAIRS(prefix, suffix, stairs)
  lastPair ← GETLASTPAIR(prefix)
  ▷ Suche alle Paare mit Wert um 1 größer als das letzte Paar der aktuellen Suffix
  followers ← GETFOLLOWINGPAIRS(suffix, lastPair)
  newSuffix ← CLONE(suffix)
  newSuffix.REMOVE(followers)
  for all follower ∈ followers do
    newPrefix ← CLONE(prefix)
    newPrefix.ADD(follower)
    if newPrefix.Length ≥ 2 then
      stairs.ADD(MAKECOMBO(newPrefix))
    end if
    ▷ Aktuelle Treppe rekursiv fortsetzen
    CALCULATEALLSTAIRS(newPrefix, newSuffix, stairs)
  end for
  ▷ Eine neue Treppe nach dem aktuellen Wert starten.
  if prefix.Length == 0 then
    CALCULATEALLSTAIRS(MAKELIST(), newSuffix, stairs)
  end if
end procedure

```

---

nicht Teil einer Bombe sein darf, muss hier auch keine Sonderbehandlung für den Phönix eingebaut werden.

Etwas komplizierter ist die Berechnung aller möglichen Treppen. Der dafür verwendete Algorithmus ist in Algorithmus 3.7 abgebildet. Zur Berechnung der Treppen wird zunächst die Menge der Paare nach Wert aufsteigend sortiert. Die Treppenberechnung baut eine Liste aufeinanderfolgender Paare rekursiv auf, die teilweise aufgebaute Treppe wird dabei in der Variablen *prefix* gespeichert. Die im aktuellen Rekursionsschritt noch verbleibenden Paare mit einem höheren Wert sind in der Variablen *suffix* gespeichert. Sobald die *prefix* einen ersten Eintrag hat, wird nach dem Einfügen jeden weiteren Paares aus der Menge der direkt folgenden Paare eine neue Treppe erzeugt. Es werden vom ersten Paar aus solange neue Paare hinzugefügt, bis es kein passendes nächstes Paar mehr gibt. Eine zweite Rekursion ist nötig, da ein Paar nicht nur eine Fortsetzung einer angefangenen Treppe darstellen kann, sondern auch eine neue Treppe einleiten kann.

Die Berechnung der Straßen erfolgt analog zu der Berechnung der Treppen. Anstatt die Paare des nächsten Wertes zu berechnen, werden die verfügbaren Einzelkarten des nächsten Wertes gesucht. Sobald die Präfix mindestens 4 Karten enthält, wird beim Einfügen jeder weiteren Karte eine neue Straßenkombination erzeugt.

### 3. Eine künstliche Intelligenz für Tichu

---

Die Berechnung der Straßenbomben kann in die Berechnung der Straßen integriert werden, indem vor dem eigentlichen Erzeugen der Straßen überprüft wird, ob alle Karten dieselbe Farbe besitzen. Wenn dies der Fall ist, wird anstatt einer Straße eine Straßenbombe erzeugt.

Die Menge aller Kombinationen erhält man durch Vereinigung der Mengen von Kombinationen der einzelnen Kombinationstypen.

Es ist bei diesem Algorithmus zu beachten, dass weder besonders effiziente Algorithmen noch eine effiziente Implementierung erforderlich ist, da die Menge der Kombinationen einmal am Anfang einer Runde<sup>3</sup> berechnet werden kann. Danach wird für weitere Berechnungen fast ausschließlich die Menge der Kombinationen verwendet. Zudem ist die Anzahl der Handkarten am Anfang einer Runde mit 14 immer noch relativ klein und fällt mit jeder gespielten Kombination weiter ab.

#### 3.4.2. Bestimmung aller Partitionen einer Menge von Kombinationen

Für die Bestimmung aller Partitionen ist es wünschenswert, nicht tatsächlich alle Partitionen zu bestimmen, sondern, im Fall von äquivalenten Partitionen, nur eine der Partition aus einer Äquivalenzklasse. Ein Beispiel für eine derartige Äquivalenzklasse ist eine Hand, die drei Buben enthält. Weiter sollen passende Karten vorhanden sein, um mit einem der Buben eine Straße zu bauen. Da für eine Straße und für ein Paar die Farbe der Karten keine Rolle spielt und nur der Wert ausschlaggebend ist, können die Buben beliebig auf das Paar und die Straße verteilt werden. Dabei bleiben Wert, Farbe und Länge der beiden Kombinationen immer gleich, weswegen die verschiedenen Möglichkeiten, die Karten zu verteilen, beim Ausspielen der Kombination nicht berücksichtigt werden müssen.

##### **Definition 3 (Relation = auf der Menge der Kombinationen)**

Für zwei Kombinationen  $c$  und  $d$  gilt:  $c = d$  genau dann, wenn  $Typ(c) = Typ(d) \wedge Länge(c) = Länge(d) \wedge Wert(c) = Wert(d)$

Die Äquivalenzklasse einer Kombination sei definiert als die Menge der Kombinationen, bei denen Länge, Typ und Wert gleich sind.

##### **Definition 4 (Äquivalenzklasse $\langle \rangle$ einer Kombination)**

Für zwei Kombinationen  $c$  und  $d$  gilt:  $c \in \langle d \rangle$  genau dann, wenn  $c = d$ . Wo nötig, gibt ein Subskript die Bezugsmenge  $s$  an.

Auf Basis dieser Überlegungen kann die Äquivalenz zweier Partitionen definiert werden.

<sup>3</sup>Genauer müssen die Kombinationen dreimal pro Spiel berechnet werden. Bei der Frage nach einem großen Tichu, vor und nach dem Kartentausch.

**Definition 5 (Relation = auf der Menge der Partitionen)**

Für die Partitionen  $p$  und  $q$  gilt:  $p = q$  genau dann, wenn  $\forall c \in p : \exists d \in q : |\langle c \rangle_p| = |\langle d \rangle_q|$

Die Erwähnung der Äquivalenzklassen in der Definition der Äquivalenz zweier Partitionen ist notwendig, falls eine Partition beispielsweise zwei Einzelkartenkombinationen vom selben Wert besitzt. Aus der Äquivalenz zweier Partitionen kann wiederum die Äquivalenzklasse der Partitionen definiert werden.

**Definition 6 (Äquivalenzklasse  $\langle \rangle$  einer Partition)**

Für zwei Partitionen  $p$  und  $q$  gilt:  $p \in \langle q \rangle$  genau dann, wenn  $p = q$

Aus eine Menge von Karten kann häufig eine Vielzahl unterschiedlicher Partitionen erzeugt werden. Eine Partitionsmenge ohne gleiche Partitionen wird "eindeutig" genannt. Dabei enthält jede Äquivalenzklasse nur genau ein Element und alle Äquivalenzklassen der Ausgangsmenge sind vorhanden. Dies kann wie folgt formal ausgedrückt werden.

**Definition 7 (Eindeutigkeit einer Menge von Partitionen)**

Sei  $m$  eine Menge von Partitionen. Dann gilt für  $e \subseteq m$ :  $e$  ist "eindeutig" genau dann, wenn  $\forall p \in e : |\langle p \rangle| = 1 \wedge \forall q \in m : \exists p \in e : p = q$

Aus Sicht des hier vorgestellten Ansatzes ist nur die Menge der eindeutigen Partitionen interessant. Äquivalente Partitionen enthalten die gleichen Kombinationen, lediglich die Karten sind anders auf die Kombinationen verteilt. Da die konkreten Karten bei der folgenden Untersuchung der Aktions-2-Tupel nicht mehr benötigt werden, können äquivalente Partitionen ausgelassen werden. Zwei äquivalente Kombinationen können beispielsweise aus der folgenden Kartenmenge gewonnen werden:

(3.2)  $\{2, 2, 3, 4, 5, 6, \text{Phönix}\}$

Hier existieren insbesondere zwei Partitionen, die aus einer Straße von 2 bis 6 und einem 2er-Paar besteht. Dabei kann der Phönix wahlweise zur Vervollständigung des Paares oder der Straße genutzt werden, wodurch zwei äquivalente Partitionen entstehen.

In Algorithmus 3.8 ist der verwendete Algorithmus zur Berechnung aller eindeutigen Partitionen abgebildet. Bei diesem handelt es sich um einen Backtracking Ansatz, bei dem die Partitionen rekursiv aufgebaut werden. Die Variable *current* speichert dabei die aktuell im Aufbau befindliche Partition. Mit jedem weiteren Rekursionsschritt wird eine weitere Kombination hinzugefügt. Eine Partition ist vollständig, sobald jede der Handkarten in genau einer Kombination zu finden ist. In Algorithmus 3.8 zählt *cardCount* die Anzahl der Karten, die im aktuellen Konstruktionsversuch durch die Wahl der Kombinationen verbaut wurden.

Dieser Algorithmus muss dabei einige Schwierigkeit korrekt behandeln. So muss sichergestellt sein, dass sich die verbauten Kombinationen bezüglich ihrer Karten nicht überlappen. Dies wird erreicht, indem nach dem Hinzufügen einer Kombination alle Kombinationen



#### Algorithmus 3.8 Berechnung der Partitionen

---

▷ Bestimmung aller Kombinationen wie in 3.4.1 vorgestellt  
combos  $\leftarrow$  CALCULATEALLCOMBINATIONS(*handCards*)  
partitions  $\leftarrow$  MAKELIST()  
CALCULATEALLPARTITIONS(*partitions*, MAKELIST(), *combos*, 0)  
**procedure** CALCULATEALLPARTITIONS(*partitions*, *current*, *combos*, *cardCount*)  
  **if** *cardCount* == *handCards.Length* **then**  
    *partitions.ADD*(MAKEPARTITION(*current*))  
    **return**  
  **end if**  
  ▷ Speichert die Menge der gültigen Nachfolger in diesem Rekursionsschritt  
  *validHashes*  $\leftarrow$  MAKEHASHMAP()  
  *validDontFollow*  $\leftarrow$  MAKELIST()  
  ▷ Berechnung aller noch einfügbarer Kombinationen  
  **for all** *combo*  $\in$  *combos* **do**  
    ▷ *GetHash* berechnet den Partitionshashwert  
    **if** *combo.GETHASH*() > *current.GETLASTCOMBO().GETHASH*() **then**  
      ▷ *GetUniqueHash* berechnet den Eindeutigkeithashwert  
      *uniqueHash*  $\leftarrow$  *combo.GETUNIQUEHASH*()  
      **if** *validHashes.CONTAINS*(*uniqueHash*) **then**  
        **if** *combo.GETHASH*() < *validHashes.GET*(*uniqueHash*).*GETHASH*() **then**  
          *validDontFollow.ADD*(*validHashes.GET*(*uniqueHash*))  
          *validHashes.PUT*(*uniqueHash*, *combo*)  
        **else**  
          *validDontFollow.ADD*(*combo*)  
        **end if**  
      **else**  
        *validHashes.PUT*(*uniqueHash*, *combo*)  
      **end if**  
    **end if**  
  **end for**  
  ▷ Fortsetzung siehe Algorithmus 3.9

---

aussortiert werden, die sich mit der hinzugefügten Kombination überlappen. Diese reduzierte Kombinationsmenge wird im nächsten Schritt der Rekursion verwendet und ist im Algorithmus in der Variablen *nextCallCombos* gespeichert.

Des Weiteren ist nur die eindeutige Partitionsmenge interessant, äquivalente Partitionen sollen nach Möglichkeit nicht erzeugt werden. Dieses Problem kann mit der Definition einer Ordnung auf die Menge der Kombinationen weitgehend gelöst werden. Dafür wird jeder Kombination eine natürliche Zahl zugeordnet.

**Definition 8 (Partitions hashwert)** *n* ist der Partitions hashwert einer Kombination *c*, wenn die folgenden Bedingungen erfüllt sind:

**Algorithmus 3.9** Fortsetzung der Berechnung der Partitionen

---

```

▷ Alle entsprechend der >-Relation einfügbaren Kombinationen durchprobieren
for all combo ∈ validHashes.VALUES() do
  nextCallCombos ← MAKELIST()
  for all nextCallCombo ∈ validHashes.VALUES() do
    ▷ Überlappende Kartenmengen ausschließen
    if not nextCallCombo.CONTAINSANY(combo) then
      nextCallCombos.ADD(nextCallCombo)
    end if
  end for
  ▷ Gleiche Kombinationen mit höherer Farbe wieder hinzufügen
  for all nextCallCombo ∈ validDontFollow do
    if not nextCallCombo.CONTAINSANY(combo) then
      nextCallCombos.ADD(nextCallCombo)
    end if
  end for
  current.ADD(combo)
  newCardCount ← cardCount + combo.LENGTH()
  CALCULATEALLPARTITIONS(partitions, current, nextCallCombos, newCardCount)
  current.REMOVE(combo)
end for
end procedure

```

---

- Die Ziffern 1 und 2 entsprechen dem Index des Typs der Kombination.
- Die Ziffern 3 und 4 entsprechen der Länge der Kombination.
- Die Ziffern 5 und 6 entsprechen dem Index des Werts der Kombination.
- Die Ziffern 7 und 8 eines Fullhouse entsprechen dem Wert des Paares, für alle anderen Kombinationen werden sie auf null gesetzt. Hier wird vorausgesetzt, dass der Index der Werte mit 1 beginnt.
- Die Ziffer 9 entspricht der Farbe der ersten Karte der Kombination.
- In den Ziffern 10 und 11 wird der Wert des Phönix eingetragen, falls der Phönix in der Kombination verwendet wird. Hier beginnt der Wertebereich der Indizes der Kartenwerte bei 1. Die Null für Kombinationen vorgesehen, die den Phönix nicht enthalten.

Es wird gegebenenfalls eine führende Null ergänzt.

Der Partitionshashwert ordnet jeder Kombination eine natürliche Zahl zu. Dabei werden alle gleichen Partitionen unterschieden, da die oben definierte =-Relation auf Kombinationen in den Ziffern 1-6 eingebaut ist. Zudem sind in den anderen Ziffern weitere Bedingungen

eingebaut, die dazu führen, dass eigentlich gleiche Kombination als, im Sinne des Partitions-hashwerts, unterschiedlich behandelt werden. Mithilfe des Partitions-hashwerts kann eine Ordnung auf die Menge der Kombinationen definiert werden.

**Definition 9 (>-Relation für Kombinationen)**

Für die Kombinationen  $c$  und  $d$  gilt:  $c > d$  genau dann, wenn  $\text{Partitionhash}(c) > \text{Partitionhash}(d)$ .

Eine weitere Kombination wird der unvollständigen Partition hinzugefügt, wenn sie entsprechend der auf den Kombinationen definierten Ordnung  $>$ -größer als die letzte hinzugefügte Kombination ist. Fall es zudem Kombinationen gibt, die im Sinne des Partitions-hashwerts identisch sind, mit Ausnahme der Farbe der ersten Karte, dann wird die Kombination nur hinzugefügt, falls sie die Kombination mit dem niedrigsten Index der Farbe der ersten Karte ist. In Algorithmus 3.8 werden alle Kombinationen, die diesen Kriterien genügen, in der Variablen *validHashes* gespeichert. Alle Kombinationen, die diesen Anforderungen mit Ausnahme der Farbe der ersten Karte genügen, werden in *validDontFollow* gespeichert.

Aufgrund des unter der Relation liegenden Partitions-hashwerts bedeutet dies, dass die Partitionen sortiert konstruiert werden. Da die Ziffern 1 und 2 des Partitions-hashwerts dem Typ der Kombination entsprechen wird, muss die einzufügende Kombination entweder einen Typ von höherem oder gleichem Index haben. Falls der Typ den gleichen Index hat, muss die Partition, entsprechend den Ziffern 3 und 4 des Partitions-hashwerts, länger oder gleich lang sein. Falls die Kombinationen auch gleich lang sind, werden die in den weiteren Ziffern codierten Kriterien überprüft. Falls eine Kombination einen kleineren Partitions-hashwert als die zuletzt eingefügte Kombination besitzt, wird diese Kombination in der aktuellen Partitions-konstruktion übersprungen.

Diese Kriterien könnten auch mit einer Menge an Fallunterscheiden überprüft werden. Mit dem Partitions-hashwert ist jedoch nur ein einziger Vergleich notwendig, zudem muss der Partitions-hashwert nur ein einziges Mal für eine Kombination verwendet werden und kann stets wiederverwendet werden.

Dadurch, dass die Partitionen sortiert konstruiert werden, werden alle Duplikate, die sich durch Einfügen von Kombinationen in einer anderen Reihenfolge ergeben, ausgeschlossen.

Eine besondere Rolle spielt die Farbe der ersten Karte im Partitions-hashwert. Falls der Spieler zwei gleiche Kombinationen  $b_1$  und  $b_2$  besitzt, beispielsweise zwei Buben als Einzelkarten, und er fügt eine der Kombinationen in den aktuellen Konstruktionsversuch ein, dann gilt ohne Berücksichtigung der Ziffer für die Farbe der ersten Karte  $\text{Partitionhash}(b_1) = \text{Partitionhash}(b_2)$ , womit die Kombinationen nicht  $>$ -vergleichbar sind. Daraus folgt, dass egal welche Kombination zuerst gewählt wurde, die zweite Kombination nicht mehr in die Partition als Einzelkarte eingefügt wird. Durch die Berücksichtigung der Farbe der ersten Karte gilt dagegen wieder  $b_1 > b_2$  oder  $b_2 > b_1$ .

Dabei ist zu beachten, dass die Farbe der ersten Karte niemals gleich sein kann. Es wurde vorausgesetzt, dass der Hashwert gleich ist, wenn die Farbe nicht berücksichtigt wird. Daraus

folgt, dass beide Kombinationen eine Karte von gleichem Wert an erster Stelle besitzen. Da beim Einfügen der letzten Kombination alle verbleibenden Kombinationen gefiltert wurden, bei denen es zu Überlappungen mit den Karten kam, muss die Farbe der ersten Karte der einzufügenden Kombination eine andere sein.

Die Farbe in den Partitionshashwert einzufügen bringt jedoch eine Schwierigkeit mit sich. Wenn sich der Algorithmus in einem Zustand befindet, indem zwei, bis auf die Farbe der ersten Karte, gleiche Kombinationen gültige nächste Kombinationen sind, dann würden Duplikate erzeugt werden, falls der Algorithmus beide Kombinationen weiter verfolgt. Das heißt, der Algorithmus sollte lediglich den Unterbaum einer der Kombinationen verfolgen. Wenn sich der Algorithmus für die Kombination mit der Farbe der ersten Karte mit dem höheren Index entscheidet, dann wird die andere Kombination später nie mehr gewählt werden. Aus diesem Grund muss sichergestellt werden, dass bei zwei oder mehr gültigen nächsten Kombinationen immer die Kombination mit der niedrigsten Farbe ausgewählt wird, da alle anderen Kombinationen einen höheren Partitionshashwert besitzen und folglich noch später der Partition hinzugefügt werden können. Dazu wird ein zweiter Hashwert definiert.

**Definition 10 (Eindeutigkeitshashwert)** *Der Eindeutigkeitshashwert  $n$  einer Kombination ist identisch zum Partitionshash, allerdings ist die Ziffer für die Farbe der ersten Karte immer 0.*

Mithilfe des Partitionshashwerts wird immer die Menge der gültigen Nachfolger durchprobiert. Die als gültig betrachteten Kombinationen werden in der Hashmap *validHashes* gespeichert. Dabei wird der Eindeutigkeitshashwert als Schlüssel verwendet. Wird für eine Kombination festgestellt, dass der Eintrag ihres Eindeutigkeitshashwerts schon belegt ist, dann wird überprüft, welche Kombination den niedrigeren Partitionshashwert besitzt, also bei welcher Kombination der Index der ersten Karte niedriger ist. Gegebenenfalls wird die bereits gespeicherte Kombination ersetzt. Hierdurch wird sichergestellt, dass bei zwei gleichen Kombinationen immer die niedrigere zuerst weiterverfolgt wird. Die höhere Kombination wird in der Liste *validDontFollow* gespeichert. Alle Kombinationen in dieser Liste können grundsätzlich noch in die unvollständige Partition eingefügt werden. Allerdings erst im nächsten Rekursionsschritt. Hierdurch wird sichergestellt, dass eine Kombination mit einem höheren Index der Farbe der ersten Karte erst nach der Kombination mit dem niedrigeren Index der Farbe der ersten Karte eingefügt wird.

Eine weitere Ausnahme stellt der Phönix dar. Im Partitionshashwert wurde der Wert des Phönix miteinbezogen. Wenn es zwei Kombinationen gibt, die gleich sind, bis auf dass die eine Kombination den Phönix enthält, die andere nicht, dann müssen diese Kombinationen als unterschiedlich behandelt werden. Handelt es sich bei der Kombination mit dem Phönix beispielsweise um ein Paar 2-Phönix, wobei noch eine zweite Karte mit Wert zwei verfügbar ist, dann würde daraus folgen, dass jede Partition, die ein 2er-Paar enthält, den Phönix in dem 2er-Paar verbaut hat und nicht in einer anderen Kombination. Zum Beispiel könnte eine Straße erzeugbar sein, wenn der Phönix nicht in dem Paar verbaut wird. Wenn der Partitionshashwert den Phönix nicht berücksichtigen würde, dann gäbe es keine Partition,

### 3. Eine künstliche Intelligenz für Tichu

---

die sowohl ein 2er-Paar als auch die Straße mit dem Phönix enthält, wenn das falsche 2er-Paar ausgewählt wurde.

Derselbe Sachverhalt liegt auch bei einem Fullhouse vor. Der Wert eines Fullhouse wird nur durch den Wert des Drillings bestimmt. Falls mehrere Paare für das Fullhouse zur Auswahl stehen, dann kann nicht nur ein beliebiges Paar ausgewählt werden, da die gewählten Paare unterschiedliche Implikationen auf die verbleibende Kombinationsmenge haben können. Zum Beispiel könnte ein 3er-Paar, ein 6er-Paar und ein 7er-Paar zur Auswahl stehen. Wenn nur das 7er-Paar gewählt wird, dann würde es keine Partition geben, in der sowohl das Fullhouse als auch die Treppe 6-6-7-7 vorhanden sind.

Diese beiden Probleme werden durch diesen Algorithmus nicht abgefangen. Um trotzdem die eindeutige Partitionsmenge zu bestimmen, kann die resultierende Partitionsmenge gefiltert werden. Dabei für alle möglichen 2-Tupel an Partitionen überprüft, ob die Äquivalenzklassen jeder Kombination jeweils gleich groß sind. Ist dies der Fall, und die Partitionen besitzen zusätzlich eine unterschiedliche Verteilung der Karten, dann handelt es sich um ein Duplikat. Experimentell hat sich gezeigt, dass auch bei ungünstigen Kartenmengen nur wenige Duplikate ohne diese Filterung erzeugt werden. Aus diesem Grund ist die Filterung nicht unbedingt notwendig.

Ein weiteres Problem ergibt sich für Straßenbomben. Straßenbomben sind der einzige Kombinationstyp, für den es relevant ist, welche Farbe eine Karte hat. Es kann sich folgende Situation ergeben: Ein Spieler besitzt passende Karten für eine Straßenbombe und ein Paar, falls er für das Paar die Karten so wählt, dass die Straßenbombe noch möglich ist. Nun wird aber zuerst das Paar in die Partition hinzugefügt und die Wahl des Paares verhindert, dass noch passende Karten für die Straßenbombe vorliegen. In diesem Fall gäbe es keine Partition, die sowohl das Paar als auch die Straßenbombe enthält. Diese Situation kann verhindert werden, indem der Straßenbombentyp den niedrigsten Index im Partitionshash erhält, also eine Straßenbombe, wenn verfügbar, immer vor einem Paar in Partition eingefügt wird.

Dieser Algorithmus berechnet tatsächlich die eindeutige Partitionsmenge, wenn die erwähnte Nachfilterung durchgeführt wird. Dazu muss, entsprechend der obigen Definition, jede Äquivalenzklasse der Menge der Partitionen genau ein Element enthalten und jede Äquivalenzklasse vorhanden sein.

In jeder Äquivalenzklasse der Partitionsmenge findet sich mindestens ein Element, das entsprechend dem Partitionshashwert angeordnet ist. Da der Algorithmus die Partition entsprechend dem Partitionshashwert erzeugt, enthält jede Äquivalenzklasse dieses Element. Dadurch, dass der Partitionshashwert noch andere Kriterien als Typ, Wert und Länge überprüft, liegt in bestimmten Fällen mehr als eine Anordnung der Kombinationen einer Partitionsmenge vor, die der durch den Partitionshash definierten Ordnung genügt. Diese Partitionen können jedoch mit der vorgestellten Nachfilterung entfernt werden.

Weiter muss argumentiert werden, dass alle Äquivalenzklassen vorhanden sind. Wäre eine Äquivalenzklasse nicht vorhanden, dann müsste es eine Partition geben, deren Kombinationen sich nicht entsprechend dem Partitionshashwert anordnen lassen. Da der Partitionshashwert jedoch jeder Kombination bijektiv eine natürliche Zahl zuordnet und die gewöhnliche

$<$ -Relation der natürlichen Zahlen eine Totalordnung darstellt<sup>4</sup>, kann eine derartige Partition nicht existieren. Folglich entspricht die berechnete Menge der Partitionen der eindeutigen Partitionsmenge.

Durch die Berechnung der eindeutigen Partitionsmenge wird die für die Berechnung der Aktionen zu durchsuchenden Partitionen bereits stark verringert. Da jedoch immer noch zu viele Partitionen übrig blieben, um den Anforderungen an die Laufzeit zu genügen, wurden Partitionen, deren Länge um einen Schwellwert größer ist als die kürzeste bekannte Partition, bei der Berechnung ausgelassen. Dies ist in Algorithmus 3.8 nicht dargestellt, kann jedoch durch einen einfachen Abbruch der Rekursion realisiert werden. Die Tiefe des Abbruchs ist aktuelle Tiefe – Schwellwert  $\geq$  Länge der kleinsten Partition. Als Schwellwert wurde dabei 3 gewählt. In Tichu ist die Länge der kürzesten Partition ein wichtiges Merkmal für die Güte eines Blattes. Grundsätzlich sind lange Partitionen eine schlechte Wahl beim Ausspielen und es konnte keine Verringerung der Spielstärke durch die Einführung dieser Heuristik beobachtet werden.

Zu bemerken ist, dass die Laufzeit der Berechnung der Partition an sich auch ohne diese Heuristik akzeptabel ist. Da jedoch der in 3.2 vorgestellte Basisalgorithmus über die Menge aller Partitions-Kombinationen 2-Tupel sucht, ist für eine möglichst kurze Rechenzeit dieser Suche eine möglichst kleine Partitionsmenge gewünscht.

Hilfreich für eine effiziente Implementierung ist eine schnelle Überprüfung auf Überlappung zwischen zwei Kombinationen. Hierzu kann für jede Kombination anfangs einmal ein Bitfeld erstellt werden, bei dem eine 1 bedeutet, dass die korrespondierende Karte in der Kombination enthalten ist. Da Tichu nur 56 Karten verwendet, kann dieses Bitfeld mit einer einzigen Variablen dargestellt werden. Der Überlappungstest kann dann durch eine And-Operation der Felder und einem Test auf null des Resultats erfolgen. Ist dieses ungleich null, dann liegt eine Überlappung vor.

### 3.5. Die Berechnung der Initiative

Die Initiative einer Partition ist eine wichtige Kenngröße für die Suche nach der besten Aktion in einem gegebenen Spielzustand und wird im Rahmen mehrerer Merkmale der zentralen Bewertungsfunktion verwendet.

Die Initiative einer Partition ist aus zwei Gründen besonders wichtig. Wenn ein Spieler das Anspielrecht nicht oft genug erwerben kann, um alle schlechten Kombinationen auszuspielen, dann ist eine Partition nicht ausspielbar und der Spieler wird mit großer Wahrscheinlichkeit Letzter in der aktuellen Runde.

Zudem ist die Initiative eine wichtige Kenngröße dafür, wie schnell ein Spieler sein Blatt im Kontext einer gewählten Partition vollständig ausspielen kann. Eine hohe Initiative bedeutet,

<sup>4</sup>Hier wird verwendet, dass jede Menge an Zahlen in einer Kette angeordnet werden kann, zum Beispiel  $2 < 3 < 5 < 10$ .

### 3. Eine künstliche Intelligenz für Tichu

---

dass der Spieler viele Stiche machen kann, seine schlechten Kombinationen damit schnell ausspielen kann. Oder auch, dass der Spieler nur wenig schlechte Kombinationen besitzt, was ebenfalls auf eine schnell ausspielbare Partition und damit gute Partition hinweist.

Zur Veranschaulichung der Initiative kann man alle Kombinationen in 4 idealisierte Kategorien aufteilen:

- **Initiativkombinationen:** Können mit hoher Wahrscheinlichkeit gespielt werden und geben dem Spieler das Anspielrecht. Ein Beispiel hierfür ist der Drache.
- **Abwurfkombinationen:** Können nur mit geringer Wahrscheinlichkeit gespielt werden und der Spieler muss das Anspielrecht dafür aufgeben. Diese Kategorie besteht vor allem aus niedrigen Paaren und Einzelkarten.
- **Verkettete Kombinationen:** Diese können nur mit geringer Wahrscheinlichkeit gespielt werden, geben dem Spieler aber mit hoher Wahrscheinlichkeit das Anspielrecht zurück. Die verketteten Kombinationen spielen bei der Initiativanalyse nur insofern eine Rolle, dass einmal das Anspielrecht erworben werden muss. Anschließend können alle verketteten Kombinationen hintereinander gespielt werden. Ein Beispiel hierfür wäre eine Straße von 2 bis 10.
- **Einwurfkombinationen:** Diese Kombinationen können mit großer Wahrscheinlichkeit gespielt werden können, sie geben dem Spieler das Anspielrecht jedoch nicht zurück. Diese Klasse spielt in der Regel kaum eine Rolle, da sie einfach auf laufende Stiche gespielt werden können und weder das Anspielrecht gewonnen wird noch Initiativkombinationen geopfert werden müssen. In dieser Klasse finden sich vor allem mittelhohe Einzelkarten und Paare, wie zum Beispiel ein Bube.

Diese Kategorien stellen eine stark idealisierte Unterteilung dar. Ob eine Kombination einem Spieler tatsächlich das Anspielrecht verschafft, ist stark vom jeweiligen Spiel abhängig. Auch lassen sich die meisten Kombinationen nicht klar einer Kategorie zuordnen, sondern liegen zwischen den Kategorien. Deswegen werden Kombinationen nicht starr klassifiziert, sondern es werden für jede Kombination zwei Wahrscheinlichkeiten berechnet:

$$(3.3) \quad P_{\text{Spiel}}(k) := P(\text{k ist ausspielbar})$$

$P_{\text{Spiel}}$  gibt die Wahrscheinlichkeit an, mit der eine Kombination  $k$  auf einen laufenden Stich zu irgendeinem Zeitpunkt in der aktuellen Runde gespielt werden kann.

$$(3.4) \quad P_{\text{An}}(k) := P(\text{k erwirbt Anspielrecht})$$

Gibt die Wahrscheinlichkeit an, mit der die Kombination  $k$  dem Spieler das Anspielrecht verschafft, unter der Voraussetzung, dass die Kombination gespielt werden kann. Wie diese Wahrscheinlichkeiten berechnet werden können, wird für den Moment offen gelassen. Eine Darstellung folgt in 3.5.1.

Mithilfe dieser Wahrscheinlichkeiten lässt sich ein einfaches Initiativmodell definieren, indem für jede Kombination angegeben wird, mit welcher Wahrscheinlichkeit sie als ein Initiativ- oder Abwurfkombination gespielt werden wird.

$$(3.5) \quad P_{Init}(k) := P(k \text{ wird als Initiativkombination gespielt}) = P_{Spiel}(k) \cdot P_{An}(k)$$

$$(3.6) \quad P_{Abwurf}(k) := P(k \text{ wird als Abwurfkombination gespielt}) = (1 - P_{Spiel}(k)) \cdot (1 - P_{An}(k))$$

Die anderen Kombinationstypen müssen für die Initiativberechnung nicht beachtet werden. Zum Spielen von Einwurfskombinationen muss keine Initiative geopfert werden. Für das Spielen der verketteten Kombinationen muss lediglich sichergestellt sein, dass das Anspielrecht einmal gewonnen wird.

Aus  $P_{Init}(k)$  und  $P_{Abwurf}(k)$  der einzelnen Kombinationen einer Partition lässt sich durch Aufsummieren die Anzahl der erwarteten Initiativkombinationen und der erwarteten Abwurfkombinationen bestimmen:

$$(3.7) \quad E_{Init}(p) = \sum_{k \in p} (P_{Init}(k))$$

$$(3.8) \quad E_{Abwurf}(p) = \sum_{k \in p} (P_{Abwurf}(k))$$

Jetzt kann die Initiative einer Partition im Kontext dieses Modells auch formal definiert werden.

**Definition 11** Die Initiative  $I$  einer Partition  $p$  sei gegeben durch  $I(p) := E_{Init}(p) - E_{Abwurf}(p)$

Wie die Initiative für die Wahl der besten Aktion verwendet wird, wird bei der Darstellung der Eigenschaften der zentralen Bewertungsfunktion in 3.6 dargelegt.

### 3.5.1. Berechnung der benötigten Wahrscheinlichkeiten

Bislang wurde offengelassen, wie die Wahrscheinlichkeiten  $P_{Spiel}$  und  $P_{An}$  berechnet werden können. Dies wird jetzt nachgeholt. Als Voraussetzung für die Berechnung der Wahrscheinlichkeiten werden statistische Daten benötigt.

Da alle statistischen Daten von der jeweiligen Spielsituation abhängen, wird ein Kontext benötigt. In der Praxis bewährt hat sich dabei der folgende Ansatz:

$$(3.9) \quad \textit{Kontext} := (\text{höchste verbleibende Karte der anderen Spieler, Nummer des Sticks})$$

Es könnte problemlos ein feinerer Kontext gewählt werden, dies vergrößert jedoch Anzahl der Einträge in der Statistikdatenbank. Dies wiederum erschwert es, genügend Daten für



### 3. Eine künstliche Intelligenz für Tichu

---

aussagekräftige Statistiken zu sammeln. Diese Wahl scheint Qualität der Initiativabschätzung und die Verwaltung und Berechnung der Statistiken gut zu balancieren.

Als Datenbanken wurden persistente Hashtabellen verwendet. Insgesamt werden vier Datenbanken angelegt, dabei sind initial alle Datenbankeinträge in allen Datenbanken auf null gesetzt. Alle Datenbanken sind mit einem bestimmten Ereignis verknüpft. Wird dieses Ereignis ausgelöst, dann wird der zum aktuellen Kontext passende Tabelleneintrag um eins erhöht. Die Statistiken wurden durch viele Spiele vorberechnet. Die dabei verwendete Vorgehensweise wird in 3.5.2 erklärt.

Die verwendete Hashfunktion der Datenbanken ist überall dieselbe:

$$(3.10) \text{ hash} := \text{Statistikhashwert}(k, \text{Kontext})$$

Dabei ist  $k$  die Kombination, zu der statistische Daten gesammelt werden sollen, beziehungsweise für die  $P_{\text{Spiel}}$  und  $P_{\text{An}}$  berechnet werden sollen. Diese Hashfunktion kann analog zum in 3.4.2 beschriebenen Partitionshashwert konstruiert werden.

**Definition 12 (Statistikhashwert)**  $n$  ist der Statistikhashwert einer Kombination  $c$ , wenn die folgenden Bedingungen erfüllt sind:

- Die Ziffern 1 und 2 entsprechen dem Index des Typs der Kombination.
- Die Ziffern 3 und 4 entsprechen der Länge der Kombination.
- Die Ziffern 5 und 6 entsprechen dem Index des Werts der Kombination.
- Die Ziffern 7 und 8 entsprechen dem Index des Werts der höchsten verbleibenden Karte.
- Die Ziffern 9 und 10 entsprechen der Nummer des Stichs.

Es wird gegebenenfalls eine führende Null ergänzt.

Die vier Datenbanken sind im Einzelnen:

- $D_{\text{Gespielt}}$ : Der Tabelleneintrag an der Stelle von  $\text{hash}$  wird immer dann erhöht, wenn  $k$  gespielt wurde. Diese Datenbank sammelt Daten darüber, bei welchem Spielzustand eine Kombination gespielt wurde.
- $D_{\text{Initiative}}$ : Der Tabelleneintrag an der Stelle von  $\text{hash}$  wird immer dann erhöht, wenn  $k$  gerade einen Stich gewonnen hat, also dem Spieler, der  $k$  gespielt hatte, das Anspielrecht verschafft hat. Wichtig im Zusammenhang mit  $D_{\text{Gespielt}}$  ist, dass jeder Eintrag von  $D_{\text{Initiative}}$  höchstens so groß wie der zugehörige Eintrag in  $D_{\text{Gespielt}}$  ist.
- $D_{\text{Hält}}$ : Der Tabelleneintrag an der Stelle von  $\text{hash}$  wird immer dann erhöht, wenn ein Spieler die Kombination  $k$  durch seine Handkarten besitzt. Ob die Kombination gehalten wird, wird bei jeder Standardsituation des Spielers überprüft. Das heißt, immer dann, wenn der Spieler auf eine Kombination spielen soll oder eine neue Kombination startet, wird über seine Kombinationen iteriert und der zugehörige Tabelleneintrag in  $D_{\text{Hält}}$  erhöht.

- $D_{\text{Spielbar}}$ : Der Tabelleneintrag an der Stelle von  $hash$  wird immer dann erhöht, wenn  $k$  gespielt werden kann. Dies wird, wie bei  $D_{\text{Hält}}$ , in jeder Standardsituation des Spielers überprüft.

Mithilfe der Daten aus  $D_{\text{Gespielt}}$  und  $D_{\text{Initiative}}$  kann wie Wahrscheinlichkeit, dass der Spieler mit der aktuellen Aktion im aktuellen Kontext das Anspielrecht erwirbt, abgeschätzt werden:

$$(3.11) P(k \text{ erwirbt Anspielrecht}) := D_{\text{Initiative}}(hash) / D_{\text{Gespielt}}(hash)$$

Dabei wird  $hash$  wie oben beschrieben aus der Kombination und dem aktuellen Kontext errechnet.

Die Berechnung von  $P(k \text{ ist ausspielbar})$  ist etwas aufwendiger. Zunächst einmal wird definiert:

$$(3.12) P(k \text{ kann in Stich } n \text{ nicht gespielt werden}) := 1 - D_{\text{Spielbar}}(hash) \cdot D_{\text{hält}}(hash)$$

Der  $hash$  wird wieder wie oben definiert berechnet, als Nummer des Stichs wird allerdings  $n$  verwendet. Die Nummer des aktuellen Stichs wird mit  $s$  bezeichnet. Damit lässt sich die gewünschte Abschätzung darstellen:

$$(3.13) P_{\neg An}(k) := P(k \text{ ist innerhalb den nächsten } N \text{ Runden nicht ausspielbar})$$

$$(3.14) P_{\neg An}(k) = \prod_{n=s}^{s+N} P(k \text{ kann in Stich } n \text{ nicht gespielt werden})$$

$$(3.15) P(k \text{ ist ausspielbar}) := 1 - P_{\neg An}(k)$$

Oder in geschlossener Form:

$$(3.16) P(k \text{ ist ausspielbar}) := 1 - \prod_{n=s}^{s+N} (1 - D_{\text{Spielbar}}(hash) \cdot D_{\text{Hält}}(hash))$$

Zu beachten ist, dass hier  $hash$  von  $n$  abhängt.

$N$  wird dabei wie folgt gewählt:

$$(3.17) N := \text{Max}(10 - s, 2)$$

Die Wahl modelliert, dass davon ausgegangen wird, dass eine Runde etwa 10 Stiche lang dauert. Mindestens jedoch noch zwei weitere Stiche, um sicherzustellen, dass  $P_{\text{Spiel}}$  auch dann noch errechnet wird, wenn sich die Annahme, dass ein Spiel 10 Runden dauert, im Einzelfall als falsch erweist. Diese Wahl hat sich experimentell bewährt.

#### 3.5.2. Einige Anmerkungen zur Initiativberechnung

In dem hier vorgestellten Modell zur Berechnung der Initiative wurden nicht immer korrekte Annahmen getätigt. Insbesondere wurde Folgendes gesetzt:

- Der Wahl des Kontexts bei den Statistiken.
- Es wurde angenommen, dass  $P_{Init}(k)$  und  $P_{Abwurf}(k)$  nur von  $k$  abhängen und nicht von den anderen Kombinationen aus der gewählten Partition.

Ein besseres Modell sollte nach Möglichkeit ohne diese Annahmen auskommen, zumindest sollte die Wahl von  $N$  durch eine bessere Heuristik erfolgen.

Allerdings wird die Initiative mit diesen Annahmen für die Anwendung in der KI hinreichend genau abgeschätzt. Zudem ist eine Berechnung der Initiative in konstanter Zeit möglich. Eine effiziente Berechnung der Initiative ist außerordentlich wichtig, da diese für jedes Aktionstupel berechnet wird. Zudem werden auch die Einzelwahrscheinlichkeiten  $P_{An}$  und  $P_{Spiel}$  von den Eigenschaften der Bewertungsfunktion verwendet.

Ein weiteres Problem ist die Initialisierung der Statistiken. Wenn noch keine geeigneten Statistiken vorhanden sind, könnte vermutet werden, dass die Kombinationen zu zufällig gespielt werden, um nützlich zu sein. Experimentell zeigte sich jedoch, dass die anfallenden Daten auch bei leeren Statistiken bereits brauchbar sind und im Laufe vieler Spiele besser werden. Daher können die Statistiken initialisiert werden, indem einfach eine große Zahl von Spielen durchgespielt werden, wobei alle Spieler KI-Spieler sind. Dabei werden die Statistiken parallel aufgebaut werden. Empirisch hat sich für diese Wahl des Kontexts gezeigt, dass sich die Spielstärke der Spieler bei mehr als 20000 Spielen nicht weiter verbessert.

Limitierend an diesem Ansatz ist insbesondere die Wahl des Kontexts. Wird zum Beispiel noch eine einzige weitere boolsche Information über die Spielsituation hinzugenommen, so verdoppelt sich die Anzahl der Einträge in den Statistiken annäherungsweise. Mit dem gewählten Kontext benötigen die gespeicherten Statistiken weniger als 1 Megabyte an Speicherplatz.

#### 3.6. Die zentrale Bewertungsfunktion

Bereits in 3.2 wurde angesprochen, dass für die Bewertung der Aktionen eine Bewertungsfunktion verwendet wird, die einem Aktionstupel, welches aus einer Partition und einer Kombination besteht, auf eine reelle Zahl abbildet. Dabei konnte gezeigt werden, dass sich fast alle Entscheidungssituationen in Tichu so formulieren lassen, dass eine einzelne Bewertungsfunktion für die Entscheidungsfindung verwendet werden kann. Diese Bewertungsfunktion wird in diesem Kapitel dargestellt.

Der algorithmische Ansatz ist in Algorithmus 3.10 zu sehen. Dabei handelt es sich um eine gewöhnliche Bewertungsfunktion zur Berechnung des erwarteten Nutzens, wie zum Beispiel in [RN05, S.171] beschrieben. Hierbei wird ein Spielzustand durch eine Menge von

**Algorithmus 3.10** Die verwendete Bewertungsfunktion

---

```

procedure GETUTILITY(partition, combo, gameState, properties, weights)
  value  $\leftarrow$  0
  for all property  $\in$  properties do
    value+ = weights[property]  $\cdot$  property.GETVALUE(partition, combo, gameState)
  end for
   $\triangleright$  Auf Intervall 0..1 normiertes Ergebnis zurückgeben
  return value/SUM(weights)
end procedure

```

---

Eigenschaften bewertet. Die Eigenschaften bewerten die einzelnen Aspekte der Spielsituation und des Zuges. Der Gesamtwert ergibt sich durch die gewichtete Summe der Eigenschaften. Dabei wird vorausgesetzt, dass der Wertebereich der Eigenschaften das Intervall  $[0, 1]$  ist. Ebenso wird das Resultat, also der Nutzen der Aktion, auf dasselbe Intervall normiert.

Eine derartige Bewertungsfunktion löst das Problem nur, insofern die zur Bewertung hinzugezogenen Eigenschaften qualitativ ausreichend sind und alle wesentlichen Aspekte der Situation und des Kandidaten für den nächsten Zug abbilden. Ebenfalls müssen die Gewichte passend eingestellt werden. Im Folgenden werden die einzelnen Eigenschaften und die Wahl der Gewichte vorgestellt.

### 3.6.1. Die Initiativeigenschaft

Die Initiativeigenschaft basiert auf der in 3.5 vorgestellten Berechnung der Initiative. Dort wurde die Initiative als die Differenz zwischen der erwarteten Anzahl der Initiativkombinationen abzüglich der erwarteten Anzahl der Abwurfkombinationen definiert.

Die Initiativeigenschaft wurde aus mehreren Gründen eingeführt. Ein hoher Initiativwert erlaubt es für gewöhnlich eine Partition schnell auszuspielen, da der Spieler oft das Anspielrecht gewinnen kann und damit den Typ der nächsten Kombination festlegt. Außerdem erlaubt es eine Initiativkombination in der Regel, dass der Spieler die Kombination ausspielen kann, bevor der erste Spieler seine Hand komplett ausgespielt hat.

Ein hoher Initiativwert kann auch bedeuten, dass generell wenig Abwurfkombinationen vorliegen. Auch in diesem Fall kann der Spieler die Partition tendenziell schnell ausspielen, da er nicht darauf zurückgreifen muss, Initiativ- und Abwurfkombinationen abwechselnd zu spielen. Das Problem beim Ausspielen von Abwurfkombinationen besteht darin, dass den anderen Spielern eine Chance gegeben wird, ihrerseits Initiativkombination zu spielen. Sind in einer Partition nur wenige Abwurfkombinationen vorhanden, dann kann die gesamte Partition unter Umständen als eine Verkettung gespielt werden, das heißt, es wird eine Kombination gespielt, die dem Spieler wiederum direkt das Anspielrecht zurückgibt. Dies wird wiederholt, bis der Spieler sein Blatt komplett ausgespielt hat.

---

**Algorithmus 3.11** Die Initiativeeigenschaft

---

```
procedure GETINITIATIVVALUE(partition, combo, gameState)
  ▷ GetInitiative beachtet, dass die gewählte Kombination jetzt spielbar ist
  rawInitiative ← GETINITIATIVE(partition, combo)
  if Partner wurde Hund geschoben then
    rawInitiative ← rawInitiative + DOG_BONUS
  end if
  if rawInitiative ≥ UPPER_BOUND then
    value ← 1.0
  else if rawInitiative ≤ LOWER_BOUND then
    value ← 0.0
  else
    range ← UPPER_BOUND – LOWER_BOUND
    value ← (rawInitiative – LOWER_BOUND)/range
  end if
  if partition.LENGTH() == 2 then
    if  $P_{Init}(combo) > HIGH\_FIRST\_THRESHOLD \wedge P_{Init}(combo) > value$  then
      value ←  $P_{Init}(combo)$ 
    end if
  end if
  return value
end procedure
```

---

Ein niedriger Initiativwert deutet darauf hin, dass der Spieler nicht als Erster fertig wird. Dies ist notwendig, um in diesem Fall kein Tichu anzukündigen und kann selbst bei Partitionen auftreten, die nur wenige und hohe Kombinationen enthalten.

Zur tatsächlichen Berechnung der Initiativeeigenschaft reicht der Initiativbegriff noch nicht ganz aus. Der verwendete Algorithmus für die Berechnung der Initiativeeigenschaft wird in Algorithmus 3.11 dargestellt. Dieser implementiert einige weitere Anforderungen an den Initiativwert, die jetzt erläutert werden.

- Wenn bekannt ist, dass der Partner den Hund besitzt, erhält die Initiative einen Bonus.
- Der Initiativwert muss nach oben und unten beschränkt werden. Dazu werden passende Schwellwerte eingeführt.
- Die Kombination der aktuellen Aktion muss gesondert berücksichtigt werden, da diese Kombination einen besonders großen Einfluss auf die Initiative im nächsten Spielzustand besitzt.
- Situationen, in denen der Spieler nur noch zwei Kombinationen besitzt, erfordern eine gesonderte Behandlung.

Ein Bonus für den Hund wird vergeben, falls der Hund in der Kartenaustauschphase an den Partner geschoben wurde. Dabei wird vorausgesetzt, dass der Partner den Hund auch

tatsächlich rechtzeitig spielen wird. Je nachdem, wie gut der Partner den Spieler unterstützt, fällt der Bonus unterschiedlich groß aus. Für die hier vorgestellte Implementierung wurde eine Bonusinitiative von 0,5 gewählt.

Des Weiteren ist es wünschenswert, dass der Wert der Initiativeeigenschaft nach oben und unten begrenzt ist. Die Grenze nach unten ist notwendig, weil bereits eine negativ Initiative darauf hindeutet, dass der Spieler nicht alle seine Abwurfkombinationen ausspielen kann. Ein deutlich negativer Wert liefert keine weitere Information. Für die hier vorgestellte Implementierung wurde ein Wert von  $-1,0$  als untere Grenze gewählt.

Eine Begrenzung nach oben ist noch deutlich wichtiger. Zwar ist es wichtig, eine etwas höhere als die notwendige Initiative von 0 zu besitzen, da die Karten der anderen Spieler und deren Entscheidungsprozesse nicht bekannt sind und deswegen immer eine gewisse Unsicherheit verbleibt, wie gut die berechneten Erwartungswerte das aktuelle Spiel approximieren. Allerdings wird ein besonders hoher Initiativwert häufig durch eine Vergrößerung der Partition erkaufte.

Ein Beispiel hierfür ist ein Blatt, bei dem der Spieler zwei Asses hält. Diese können insbesondere als ein einzelnes Ass-Paar oder als zwei Einzelkarten gespielt werden. Sowohl Paare als auch Einzelkarten werden oft genug angespielt, sodass der Spieler in der Regel die Möglichkeit hat, entweder einmal oder zweimal das Anspielrecht zu gewinnen. Wenn er die Paare als Einzelkarten spielt, benötigt er allerdings einen Zug mehr, um sein Blatt vollständig auszuspielen. Gerade wenn der Spieler nur noch eine Abwurfkombination auf seiner Hand hält und er die Möglichkeit hat, die Asses als Paar zu spielen, dann ist ein weiteres Abwarten unnötig, da er zuerst das Ass-Paar und danach die Abwurfkombination spielen kann. Genauso ist ein Spielen der einzelnen Karten als einzelne Stiche, bei denen der Spieler direkt das Anspielrecht zurückgewinnt, unnötig.

Zuletzt wächst der Initiativwert häufig stärker, wenn mehrere Kombinationen mit hoher Initiativwahrscheinlichkeit enthalten sind, als wenn eine Kombination mit noch höherer Initiativwahrscheinlichkeit vorhanden ist. Dies ist nützlich, wenn noch viele Abwurfkombinationen gespielt werden müssen. Andererseits ist eine einzige Kombination mit einer höheren Initiativwahrscheinlichkeit tendenziell besser, wenn der Spieler nur noch eine verbleibende Abwurfkombination besitzt.

Die obere Grenze muss damit balancieren, dass etwas mehr als die mindestens benötigte Initiative vorhanden ist, aber nicht zu viel, damit das Blatt schnell ausgespielt werden kann. Als Wert für die obere Grenze hat sich in der Praxis 2,0 als gute Wahl erwiesen.

Ein weiteres essenzielles Merkmal der Initiativeeigenschaft ist, dass die Kombination, die gerade gespielt werden soll, gesondert behandelt werden muss. Die in 3.5 eingeführte Wahrscheinlichkeit  $P(k \text{ ist ausspielbar})$  beträgt für diese Kombination 1,0, da sie jetzt gespielt werden kann. Daraus folgt, dass die Wahrscheinlichkeit, dass die Kombination das Anspielrecht gewinnt, wenn die Kombination jetzt gespielt wird, stark erhöht sein kann:

$$(3.18) \quad P_{Init}(k) = P_{Spiel}(k) \cdot P_{An}(k) = 1 \cdot P_{An}$$

Dies gilt besonders für Kombinationen, die eher selten angespielt werden, wie zum Beispiel lange Straßen oder lange Treppen. In diesen Fall kann eine Kombination, die erwartet nur als verkettete Kombination gespielt werden kann, als Initiativkombination gespielt werden. Ebenfalls zählt die auszuspielende Kombination der Aktion nicht zu der Menge der erwarteten als Abwurfkombinationen zu spielenden Kombinationen hinzu, da sie mit dieser Aktion ausgespielt würde. Dies ist besonders nützlich, wenn überraschend eine eigentlich sehr niedrige Kombination gespielt werden kann, ohne dass eine Initiativkombination geopfert werden muss.

Die letzten beiden Merkmale ergeben zusammen, wie sich der Initiativwert voraussichtlich im nächsten Spielzug entwickeln wird. Eine gewisse Unsicherheit bleibt, da die Gegner weitere Karten spielen und insbesondere der Wert der höchsten bekannten Karte der anderen Spieler weiter sinken kann.

Des Weiteren ist noch ein Spezialfall zu behandeln. Angenommen die gewählte Partition besteht nur noch aus zwei Kombinationen, wobei eine der Kombinationen eine Initiativkombination ist und die andere eine Abwurfkombination. Dann ist es erwünscht, dass der Spieler die Initiativkombination zuerst spielt und danach die Abwurfkombination. Dies führt dazu, dass der Spieler schnell fertig wird, ohne dass die andere Mannschaft noch eingreifen kann. Das Verhalten der Initiative steht dem jedoch entgegen. Wenn die Initiativkombination und die Abwurfkombination spielbar sind, dann maximiert es den Initiativwert, wenn der Spieler zuerst die Abwurfkombination spielt, da die erwartete Anzahl der Abwurfkombinationen nach Ausspielen der letzten Abwurfkombination 0 ist. Es verbleibt als Initiativwert die Initiative der Initiativkombination. Wird dagegen die Initiativkombination zuerst gespielt, dann bekommt der Spieler die Initiative der Initiativkombination, allerdings abzüglich der erwarteten Abwurfwahrscheinlichkeit der Abwurfkombination. Dieses Phänomen ist darauf zurückzuführen, dass die Wahrscheinlichkeit  $P_{\text{Spiel}}$  der Initiativkombination bereits nahe 1 ist, sich am Initiativwert also viel nicht ändert, unabhängig davon, wann die Initiativkombination gespielt wird. Dagegen ist der Abwurfwert nur dann 0, wenn die Abwurfkombination jetzt gespielt wird. Dieses Problem kann behoben werden, indem ein Schwellwert festgelegt wird, der bestimmt, wann die Initiativkombination eine genügend hohe Initiative besitzt, damit die Abwurfkombination anschließend gespielt werden kann. Experimentell hat sich dabei ein Schwellwert von 0,8 als sinnvoll erwiesen.

#### 3.6.2. Die Wenige-Kombinationen Eigenschaft

Ein weiteres essenzielles Merkmal einer Partition ist die Anzahl der Kombinationen, aus denen diese besteht. Da höchstens eine Kombination pro Spielzug gespielt werden kann, bestimmt die Anzahl der Kombination, wie lange es noch dauert, bis der Spieler frühestens das Spiel beendet hat. Zudem deutet eine Partition, die aus wenigen Kombinationen besteht, darauf hin, dass die einzelnen Kombinationen länger sind. Dies wiederum bedeutet, dass wahrscheinlich mehr Kombinationen enthalten sind, die in Form einer Verkettung gespielt werden können. Aus diesen Argumenten ergibt es sich, dass kleine Partitionen tendenziell deutlich stärker sind.

Anzahl Kombinationen	Wert
0	0,10
1	0,75
2	0,70
3	0,65
4	0,60
5	0,55
6	0,5
7	0,4
8	0,3
9	0,2
10	0,1
11	0,0
12	0,0
13	0,0
14	0,0

**Tabelle 3.1.:** Wert der Wenige-Kombinationen Eigenschaft abhängig von der Größe der Partition

Besonders wichtig ist diese Eigenschaft auch bei der Entscheidung, ob ein Tichu angekündigt werden soll. Für erfolgreiche Tichu-Ankündigungen ist es wichtig, dass das Blatt schnell ausgespielt werden kann, also eine Partition mit wenig Kombinationen gespielt werden kann.

Zum Anderen soll modelliert werden, dass eine Reduktion der Anzahl der Kombinationen um eins den Wert einer Aktion erhöht, da der Spieler nun einen Zug näher am vollständigen Ausspielen seines Blattes ist.

Algorithmisch wurde diese Eigenschaft in Form einer Lookup-Tabelle implementiert. Dabei wird jeder Partitionsgröße ein Wert zugeordnet, wobei eine längere Partition ein niedriger Wert und eine kürzere Partition mit einem höheren Wert verknüpft sein sollte. Die genauen Werte sind in Tabelle 3.1 gelistet.

Dabei fällt vor allem der Sprung beim Abwurf der letzten Kombination auf. Dieser stellt sicher, dass die KI ihre letzte Kombination auch tatsächlich ausspielt, wobei dies zusätzlich verstärkt wird, dass diese Eigenschaft ein relativ hohes Gewicht erhält, wie in 3.6.8 näher ausgeführt wird. Dies sicherzustellen ist notwendig, da insbesondere die Initiativeigenschaft nach dem Ausspielen der letzten Kombination einen Wert von 0 besitzt. Falls die letzte Kombination eine Initiativekombination ist, wird der Wert der Initiativeigenschaft, wenn die



KI passt, deutlich größer als Null sein, was jedoch trotzdem fast immer eine schlechte Aktion ist<sup>5</sup>.

#### 3.6.3. Die Hohe-Partner-Kombination Eigenschaft

Es ist wichtig den eigenen Partner zu unterstützen, insbesondere wenn dieser Tichu gerufen hat. Diese Eigenschaft überprüft, ob die vom Partner gespielte Kombination eine hohe Chance hat, das Anspielrecht zu gewinnen. Wenn dies der Fall ist, dann sollte der Spieler normalerweise nicht darüberspielen.

Zur Berechnung des Werts der Eigenschaft wird die Wahrscheinlichkeit  $P_{Init}(k)$  für die vom Partner gespielte Kombination  $k$  als Basis gewählt. Es sind dann einige Fälle zu unterscheiden:

- Es gibt aktuell keinen laufenden Stich, der Spieler hat das Anspielrecht. Hier spielt diese Eigenschaft keine Rolle, der Wert kann beliebig, aber konstant, gewählt werden.
- Die letzte Kombination gehört nicht dem Partner, sondern einem der Gegner. Auch in diesem Fall kann der Wert der Eigenschaft beliebig, aber konstant gewählt werden.
- Die letzte Kombination gehört dem Partner und der Spieler bewertet die Passen-Aktion. In diesem Fall wird als Wert der Eigenschaft  $P_{Init}(k)$  zurückgegeben, da Passen umso besser ist, je höher die Wahrscheinlichkeit ist, dass die Kombination des Partners dem Partner das Anspielrecht verschafft.
- Die letzte Kombination gehört dem Partner und der Spieler passt nicht. Hier wird 0 als Wert zurückgegeben, da dem Partner die Chance auf den Gewinn des Anspielrechts genommen wird. Daraus folgt, dass das Überspielen einer Kombination des Partners umso schlechter ist, je höher deren Initiativwahrscheinlichkeit ist.

#### 3.6.4. Die Stich-Punkte Eigenschaft

Auch wenn in Tichu Punkte vor allem über erfolgreiche Tichus und Doppelsiege erzielt werden, so spielen auch die über die Stiche erzielten Punkte eine gewisse Rolle. Insbesondere sollte es die KI miteinbeziehen, wenn über einem einzelnen Stich besonders viele Punkte zu gewinnen sind, zum Beispiel weil ein Spieler ein Fullhouse bestehend aus drei 10ern und zwei Königen gespielt hat.

Die über die Aktion voraussichtlich gewonnenen oder verlorenen Punkte werden über diese Eigenschaft abgebildet. Dafür wird zunächst modelliert, um welchen Anteil der Punkte der Runde gespielt wird, mit:

$$(3.19) \text{ Wert}(\text{Stich}) = \text{Punkte}(\text{Stich}) / \text{Punkte der Runde}$$

<sup>5</sup>Eine Ausnahme liegt vor, wenn der Partner ein Tichu angekündigt hat. Dann sollte die letzte Kombination nicht ausgespielt werden, solange noch kein Spieler fertig ist.

Dabei ist zu bemerken, dass in einer Runde eines Tichu Spiels immer 100 Punkte über die Stiche verteilt werden, wenn kein Doppelsieg vorliegt.  $Wert(Stich)$  beschreibt damit, um welchen Teil der Punkte einer Runde gespielt wird.

Zur Bestimmung des Werts der Eigenschaft wird die durch die Aktion gewählte erwartete relative Anzahl gewonnener Punkte und die erwartete relative Anzahl verlorener Punkt berechnet. Diese werden als  $E(gewonnenePunkte)$  respektive  $E(verlorenePunkte)$  dargestellt. Wie genau  $E(gewonnenePunkte)$  und  $E(verlorenePunkte)$  berechnet werden, hängt von der Spielsituation ab.

- Wenn der Spieler eine Kombination spielt, dann lässt sich der erwartete Punktergebnis wie folgt darstellen:

$$(3.20) \quad E(\text{gewonnene Punkte}) = P_{Init}(k) \cdot Wert(\text{Stichs})$$

Analog kann der erwartete Verlust beschrieben werden mit:

$$(3.21) \quad E(\text{verlorene Punkte}) = (1 - P_{Init}(k)) \cdot Wert(\text{Stichs})$$

- Wenn der Spieler passt, aber die zuletzt gespielte Kombination  $l$  seinem Partner gehört, dann gilt folgendes:

$$(3.22) \quad E(\text{gewonnene Punkte}) = P_{Init}(l) \cdot Wert(\text{Stichs})$$

$$(3.23) \quad E(\text{verlorene Punkte}) = (1 - P_{Init}(l)) \cdot Wert(\text{Stichs})$$

- Wenn der Spieler passt und einem gegnerischen Spieler gehört die zuletzt gespielte Kombination  $l$ , dann gilt, wenn auch der Partner schon gepasst hat:

$$(3.24) \quad E(\text{gewonnene Punkte}) = 0$$

$$(3.25) \quad E(\text{verlorene Punkte}) = Wert(\text{Stichs})$$

Falls der Partner noch spielen kann, wird angenommen, dass das eigene Passen keinen Einfluß auf die Initiativwahrscheinlichkeit der liegenden Kombination hat:

$$(3.26) \quad E(\text{gewonnene Punkte}) = (1 - P_{Init}(l)) \cdot Wert(\text{Stichs})$$

$$(3.27) \quad E(\text{verlorene Punkte}) = P_{Init}(l) \cdot Wert(\text{Stichs})$$

Aus den Erwartungswerten lässt sich der Wert der Eigenschaft berechnen:

$$(3.28) \quad Wert = (E(\text{gewonnenePunkte}) - E(\text{verlorenePunkte}) + 1,25) / 2,5$$

Die Verschiebung wird benötigt, damit die Eigenschaft folgende Bedeutung erhält. Ein Wert von 0,5 bedeutet, dass erwartet weder Punkte verloren noch gewonnen werden. Ein Wert von 0 bedeutet, dass alle Punkte des Spiels mit diesem Stich an den Gegner gehen, ein Wert von 1 analog, dass das eigene Team alle Punkte des Spiels erhält. Die 1,25 ergibt sich daraus, dass es durch Drache und Phönix möglich ist, dass ein Team 125 Punkt gewinnt, während das andere Team 25 Punkte verliert.

#### 3.6.5. Die Tichu-Endspiel Eigenschaft

Es ist wichtig seinen Partner zu unterstützen, falls dieser Tichu angekündigt hat. Hierzu wird versucht, dem Partner niedrige Kombinationen vorzuspielen, die ihm eine Chance geben, das Anspielrecht zu erlangen oder es im zumindest ermöglichen, schneller weitere Kombinationen auszuspielen. Insbesondere ist dies in Spielsituationen wichtig, in denen der Partner nur noch wenige Karten auf seiner Hand hält und das Anspielrecht nicht besitzt. Ebenfalls kommt es gelegentlich vor, dass die eigenen Initiativkombinationen von Gegner überspielt werden. Dadurch kann es passieren, dass der Spieler, der Tichu angekündigt hat, nicht mehr genügend Initiative besitzt, um das Tichu erfolgreich auszuspielen. Auch in dieser Situation kann die Runde durch Unterstützung des Partners noch für das Team gewonnen werden.

Andererseits bedeutet eine aktive Unterstützung des Partners im Allgemeinen, dass der Spieler sein eigenes Blatt nicht optimal schnell ausspielt. Er nutzt Situationen, in denen er das Anspielrecht erhalten könnte, nicht. Er spielt schwache Kombinationen an, auch wenn dies für ihn nicht notwendig wäre, zum Beispiel weil er nur noch eine Initiativkombination und eine Abwurfkombination auf seiner Hand hält. Dies wurde dahin gehend beachtet, dass die Unterstützung des Tichus des Partners erst dann stattfindet, wenn dieser die meisten seiner Karten ausgespielt hat. Dabei hat es sich in der Praxis als nützlich erwiesen, diesen Schwellwert so zu setzen, dass der Partner unterstützt wird, sobald er weniger als 6 Karten auf seiner Hand hält. Es ist dabei zu beachten, dass der Partner auch dann eingeschränkt unterstützt wird, wenn er noch mehr Karten besitzt, da die in 3.6.3 vorgestellte Hohe-Partner-Kombination Eigenschaft bereits zu einer eingeschränkten Kooperation führt.

Im Einzelnen werden zur Berechnung des Werts der Eigenschaft wieder unterschiedliche Spielsituationen unterschieden. Dabei wird in allen Situationen vorausgesetzt, dass der Partner Tichu gerufen hat und nur noch wenige Karten auf seiner Hand hält.

- Wenn eine Passen-Aktion untersucht wird, dann hängt der Wert der Eigenschaft von der zuletzt gespielten Kombination ab. Wenn der Partner noch nicht auf die liegende Kombination, oder eine schlechtere Kombination vom selben Typ, gepasst hat, dann gilt für den Wert der Eigenschaft:

$$(3.29) \text{ Wert} = 1 - \text{Wert}(\text{Kombination}) / \text{Wert}(\text{Ass})$$

Dies führt dazu, dass der Wert der Eigenschaft umso höher ist, je niedriger die Kombination ist, die aktuell auf dem Brett liegt.

Wenn der Partner bereits auf die liegende oder eine niedrigere Kombination von gleichem Typ gepasst hat, dann ist zu vermuten, dass er wiederum passen wird. In diesem Fall sollte der Spieler nicht passen, sondern stattdessen versuchen, das Anspielrecht zu gewinnen, um dem Partner dann eine andere niedrige Kombination vorzulegen. Der Wert der Eigenschaft ist dementsprechend hier 0, wenn der Spieler passt.

- Wenn der Spieler nicht passt und der Partner hat über die Kombination der in Untersuchung befindlichen Aktion schon einmal gepasst, dann wird angenommen, dass er wiederum passen wird, also keine Unterstützung des Partners stattfindet. Folglich wird der Wert der Eigenschaft auf 0 gesetzt.
- Wenn der Partner nur noch eine Karte auf der Hand hält, dann wird der Wert wieder auf

$$(3.30) \text{ Wert} = 1 - \text{Wert}(\text{Kombination}) / \text{Wert}(\text{Ass})$$

gesetzt. Aber nur, wenn die Kombination der Aktion eine Einzelkarte ist, ansonsten wird als Wert 0 zurückgegeben.

- Wenn der Spieler eine Einzelkarte oder ein Paar als Kombination seiner Aktion untersucht und der Partner noch mindestens 2 Karten hält, dann wird der Wert wieder auf

$$(3.31) \text{ Wert} = 1 - \text{Wert}(\text{Kombination}) / \text{Wert}(\text{Ass})$$

gesetzt.

In allen anderen Situationen, die nicht zu den beschriebenen Fällen gehören, wird der Wert auf 0 gesetzt. Eine Ausnahme existiert für den Hund. Da der Hund ausgespielt werden sollte, solange der Partner noch Karten hat, wird der Wert dieser Eigenschaft fest auf 1,0 gesetzt, solange der Spieler noch den Hund auf der Hand hält.

### 3.6.6. Die Kleinste-Kombination Eigenschaft

Beim Ausspielen einer Partition spielt die Reihenfolge eine Rolle, in der die Partition ausgespielt wird. Dabei ist es hilfreich, wenn längere Kombinationen zuletzt gespielt werden, da der Spieler, wenn er am Ende kurze Kombinationen spielt, relativ schnell nur noch wenige Karten auf seiner Hand hält. In dieser Situation wird die andere Mannschaft verstärkt versuchen zu verhindern, dass der Spieler fertig wird, besonders um ein angekündigtes Tichu zu verhindern. Besser ist es, wenn der Spieler seine Hand überraschend vollständig ausspielt. Dabei gilt als grobe Heuristik, je mehr Handkarten, desto besser und je weniger Kombinationen, desto besser. Eine ausreichende Initiative wird vorausgesetzt.

Dieses Verhalten wird dadurch modelliert, dass eine kürzere Kombination einen kleinen Bonus durch diese Eigenschaft erhält. Der Wert der Eigenschaft kann dabei folgendermaßen berechnet werden, falls der Spieler nicht passt:

$$(3.32) \text{ Wert} = 1 - \text{Länge}(k) / 14$$

$k$  ist dabei die Kombination der Aktion. Die Konstante 14 dient der Normierung auf das Intervall  $[0, 1]$  und bezieht sich auf die maximale Länge einer Kombination, die der Anzahl der Handkarten zu Beginn einer Runde entspricht. Passen wird als das Ausspielen der kleinstmöglichen Kombination betrachtet, weswegen der Wert der Eigenschaft in diesem Fall auf 1.0 gesetzt wird.

#### 3.6.7. Die Tichu Schwellwerte

Neben den vorgestellten Eigenschaften gibt es noch zwei wichtige Parameter. Dies sind die beiden Schwellwerte, die eine Partition überschreiten muss, damit der Spieler Tichu beziehungsweise Großes Tichu ankündigt. Auch wenn beide Schwellwerte grundsätzlich verschieden gewählt werden können, so hat sich für beide Schwellwerte ein Wert von 0,425 als gute Wahl erwiesen. Tichu wird dabei, wie in Algorithmus 3.5 dargelegt, angekündigt, das heißt, es wird überprüft, ob der Nutzen der besten Partition besser als der gegebene Schwellwert ist.

Zusätzlich wird bei beiden Arten von Tichus überprüft, ob bereits ein anderer Spieler ein Tichu angekündigt hat. In diesem Fall ruft der Spieler kein Tichu. Dies ist bei einem Tichu des Partners offensichtlich, da dann selbst im besten Fall durch die Tichus 0 Punkte erzielt würden. Bei einem gegnerischen Tichu ist eine eigene Ankündigung grundsätzlich denkbar. Allerdings kann ein gegnerisches Tichu als ein Indiz gewertet werden, dass mindestens einer der Spieler der gegnerischen Mannschaft gute bis sehr gute Karten besitzt. In dieser Situation möchte man dann bestenfalls noch Tichu ankündigen, wenn die eigenen Karten herausragend gut sind. Diese Möglichkeit wurde untersucht, indem bei einem angekündigten Tichu der Gegner ein höherer Schwellwert verwendet wurde. Es konnte jedoch kein Schwellwert gefunden werden, bei dem eine Mannschaft Tichus zuverlässig verhindert hat und selbst erfolgreiche Tichus erzielen konnte. Zudem ist bereits die Verhinderung gegnerischer Tichus nützlich, da sie, ohne Berücksichtigung der sonstigen Punkte, den aktuellen Punktestand um 100 Punkte im Sinne der Mannschaft des Spielers verschiebt.

Für die Ankündigung eines großen Tichus werden zusätzliche Informationen ausgewertet, da das Blatt des Spielers zum Zeitpunkt der Ankündigung noch unvollständig ist. Dabei wird nicht nur überprüft, ob der Nutzen der besten Partition höher als der Schwellwert ist. Es wird sichergestellt, dass mindestens drei Partitionen existieren, und deren durchschnittlicher Nutzen oberhalb des Schwellwerts liegt. Dabei werden mehrere Partitionen hinzugezogen, um sicherzustellen, dass die Karten nicht nur gut sind, sondern auch ein Mindestmaß an Flexibilität besitzen, was die Chance erhöht, dass die noch fehlenden Karten sinnvoll in die Partition integriert werden können. Insbesondere würde zum Beispiel eine Partition mit einer einzelnen langen Straße einen hohen Nutzen besitzen, da die Länge der Partition sehr gering ist. Ist zudem noch ein Ass vorhanden, ist auch die Initiative hoch. Trotzdem wäre mit einer Partition aus diesen beiden Kombinationen eher unwahrscheinlich, dass das große Tichu erfolgreich ist, da nur eine einzige Initiativkombination vorhanden ist und deswegen zusätzliche niedrige Karten die Initiative stark absenken würden. Wünschenswert sind eher hohe Paare und hohe Einzelkarten. Aus diesem Grund wird für die Ankündigung eines großen Tichus weiter vorausgesetzt, dass sowohl Drache als auch Phönix vorhanden sind. Gerade diese letzte Voraussetzung ist wahrscheinlich als eigentlich zu restriktiv anzusehen. Um große Tichus zukünftig besser abschätzen zu können, würde sich vermutlich eine abgewandelte Initiativeeigenschaft anbieten, die eine sehr hohe Initiative positiv berücksichtigt.

Generell können große Tichus allerdings nur relativ selten gespielt werden, weswegen bei dem hier vorgestellten Ansatz eher versucht wurde, gewöhnliche Tichus zu erkennen und erfolgreich auszuspielen. Zumal es immer noch möglich ist, mit Karten, die eigentlich ein

Eigenschaft	Gewicht
Initiative	0,18
Wenige-Kombinationen	0,38
Hohe-Partner-Kombination	0,12
Stich-Punkte	0,045
Tichu-Endspiel	0,22
Kleinste-Kombination	0,04
Tichu Schwellwert	0,425
Großes Tichu Schwellwert	0,425

**Tabelle 3.2.:** Gewichte der Eigenschaften und Parameter der Bewertungsfunktion

großes Tichu ermöglichen, ein gewöhnliches Tichu zu spielen. In diesem Fall verzichtet die Mannschaft auf zusätzliche 100 Punkte, erhält dafür aber, da der Spieler die weiteren Karten aufnehmen darf, zusätzliche Sicherheit, dass das Tichu auch erfolgreich ausspielbar ist. Aus diesem Grund wird vermutet, dass die zusätzliche Spielstärke durch ein besseres Ankündigen von großen Tichus eher gering ist.

### 3.6.8. Die Gewichte der Bewertungsfunktion

Da die Bewertungsfunktion eine gewichtete Summe der Werte der Eigenschaften berechnet, wird neben den Werten der Eigenschaften auch noch ein Gewicht für jede Eigenschaft benötigt. In Tabelle 3.2 sind die Gewichte der einzelnen Eigenschaften der vorgestellten Bewertungsfunktion zu sehen.

Die Gewichte der Eigenschaften entsprechen aus den folgenden Gründen den in sie gesetzten Erwartungen. Die Wenige-Kombinationen Eigenschaft spielt erwartungsgemäß eine wichtige Rolle, da sie ein Indikator für Güte der Kombinationen und für die Geschwindigkeit ist, mit der das Blatt voraussichtlich ausgespielt werden kann.

Ebenfalls besonders wichtig für die Bewertung ist die Initiativeigenschaft, da sie hilft zu entscheiden, wann eine Kombination gespielt werden kann und in welcher Reihenfolge.

Die Tichu-Endspiel Eigenschaft hat zwar ebenfalls einen hohen Wert, wird aber erst dann aktiv, wenn der Partner Tichu gerufen hat und nur noch wenige Karten auf der Hand hält. In dieser Situation ist diese Eigenschaft dann besonders wichtig, was gewünscht war, da nur so die Unterstützung des Partners angemessen berücksichtigt werden kann.

Ähnlich verhält es sich bei der Hohe-Partner-Kombination Eigenschaft. Sie sollte einen relativ starken Einfluss auf das Spiel haben, da es durch sie ermöglicht wird, dass sich die Spieler einer Mannschaft nicht gegenseitig schwächen, wodurch wiederum die Chance beider Spieler steigt, früh alle Karten ausspielen zu können.

Wie bereits in 2.4 bei den grundlegenden Strategien zu Tichu erklärt, ist es wichtiger als Erster fertig zu werden, am besten mit einem angekündigten Tichu, als zu versuchen, Stiche

mit vielen Punkten zu machen. Dementsprechend hat die Stich-Punkte Eigenschaft einen relativ niedrigen Wert.

Das Ausspielen der kleinsten Kombination spielt eine untergeordnete Rolle bei der Auswahl einer geeigneten Kombination. Auch dies ist sinnvoll, da es zwar gewünscht ist, dass eine kürzere Kombination vor einer längeren gespielt werden soll. Häufig sprechen jedoch Gründe dagegen, zum Beispiel weil gerade eine seltene Kombination angespielt wurde und sich die seltene Möglichkeit ergibt, eine lange Straße direkt auszuspielen, also eine verkettete Kombination als Initiativkombination zu spielen. Die Kleinste-Kombination Eigenschaft soll vor allem dann zum Zuge kommen, wenn die Wahl zwischen einer kürzeren und einer längeren Kombination aufgrund der anderen Eigenschaften unentschieden ist. Hierfür scheint ein eher kleiner Gewichtswert sinnvoll.

#### **3.6.9. Kalibrierung der Gewichte**

Die Gewichte der Bewertungsfunktion wurden mittels eines automatisierten Kalibrierungsprozesses gefunden. Dieser benötigt einen manuell erstellten oder zufälligen erzeugten Startvektor. Anschließend wird versucht, diesen Startvektor mittels eines iterativen Prozesses zu verbessern. Dafür wird ein Satz neuer Vektoren aus dem Startvektor erzeugt, bei dem alle Komponenten vom ursprünglichen Vektor kopiert werden, bis auf eine einzelne Komponente. Diese wird um den Ausgangswert variiert. Diese Menge an Vektoren sind die Kandidaten für eine Verbesserung der Gewichte.

Für jeden Vektor aus der Menge wird eine für die Bewertung des Vektors ausreichende Anzahl von Spielen gespielt, hier wurden 40 Spiele gewählt. Anschließend wird der Vektor, der am besten abgeschnitten hat, als neuer Startvektor verwendet. Der einzelne Kalibrierungsdurchlauf endet, sobald für alle Komponenten versucht wurde, einen besseren Wert zu finden.

Da es sich bei diesem Algorithmus um eine einfache lokale Suche handelt, müssen zwei Probleme gelöst werden. Zum einen wird durch einen einzelnen Durchlauf häufig nur eine geringfügige Verbesserung im Vergleich zum Ausgangsvektor erreicht und der neue beste Vektor stellt weder ein lokales noch ein globales Maximum unter den möglichen Parametervektoren dar. Dies liegt vor allem daran, dass die Werte der Gewichte voneinander abhängen. Zumindest ein lokales Maximum kann dadurch gefunden werden, dass der Kalibrierungsprozess wiederholt wird.

Zweitens ist es gewünscht, dass der verwendete Vektor der beste mögliche Vektor aus dem Raum der möglichen Vektoren ist, also hinsichtlich der Spielstärke einem globalen Maximum im Parameterraum entspricht. Dies kann der verwendete Suchalgorithmus nicht garantieren. Aus diesem Grund wurden unplausibel erscheinende Gewichtswerte manuell variiert und die Auswirkungen auf die Spielstärke untersucht. Dabei haben sich jedoch keine Verbesserungen eingestellt. Da die Verhältnisse zwischen den Parametern zudem dem entsprechen, was aufgrund von strategischen Überlegungen erwartet wird, wird vermutet, dass dieser Parametervektor zumindest nur wenig schlechter als das globale Maximum ist.

## 3.7. Die Behandlung der weiteren Entscheidungssituationen

Neben den wichtigsten Entscheidungssituationen, der Wahl einer Kombination in der Standardsituation und dem Ankündigen eines Tichu, sollen auch Strategien zur Behandlung der weniger wichtigen Spielsituationen angegeben werden. Dabei wird für den Kartentausch einmal auf die zentrale Bewertungsfunktion zurückgegriffen. Aufgrund des geringen Einflusses auf die Spielstärke wird das Wünschen und das Verschenken des Drachens mittels einfacher Heuristiken gelöst.

### 3.7.1. Der Kartentausch

Die Berechnung der Karten, die am Anfang des Spiels mit den anderen Spielern getauscht werden, kann auf die vorgestellte Bewertungsfunktion reduziert werden, wie bereits in 3.3 beschrieben wurde. In 3.3 wurde ein Algorithmus vorgeschlagen, der über alle Karten-3-Tupel iteriert und die verbleibenden Karten bewertet. Dies hat sich in der Praxis als zu aufwendig hinsichtlich der Rechenzeit herausgestellt.

Algorithmus 3.12 zeigt den tatsächlich verwendeten Algorithmus. Zunächst wird die beste Partition der ausgeteilten Karten berechnet. Aus der Menge der Karten, die sich aus den Karten der Einzelkartenkombinationen und den Karten aller Paare mit einem Wert von höchstens 6 ergeben, werden alle 3-Tupel konstruiert, die sich aus dieser Menge zusammensetzen lassen. Dabei müssen einige Einschränkungen beachtet werden, damit ein 3-Tupel für den Austausch zulässig ist.

So wie der Algorithmus bisher vorgestellt wurde, ist davon auszugehen, dass der Nutzen für den Austausch Kandidaten maximiert wird, der 3 für den Spieler sehr schlechte Karten aus dessen Blatt entfernt. Dieses Verhalten ist jedoch nicht immer gewünscht. Dabei sind zwei Fälle zu unterscheiden. Wenn der Spieler ein Tichu ankündigen will, dann ist es akzeptabel, sein eigenes Blatt rücksichtslos zu verbessern. Falls der Spieler aber kein Tichu ankündigen will, dann besteht die Möglichkeit, dass der Partner ein Tichu ankündigen will. Dieses sollte dann auch unterstützt werden.

Diese Unterstützung wird realisiert, indem der Spieler zuerst prüft, ob er Tichu ankündigen will. Anschließend wird diese Information bei der Erzeugung der Kandidaten verwendet. Wenn der Spieler kein Tichu ankündigen will, dann ist die Wahl der Partnerkarte auf die Bildkarten, Drache und Phönix eingeschränkt. Andernfalls unterliegt die Auswahl der Karte für den Partner keinerlei Beschränkungen. Die Auswahl der Karten für die Gegner ist immer eingeschränkt auf die Karten mit dem Wert 2 bis 10.

Als Nächstes werden alle Kandidaten bewertet. Dazu werden die Karten des Kandidaten aus der ursprünglichen Menge der Handkarten entfernt. Für die verbleibenden Karten wird wieder die beste Partition berechnet. Das Austausch-3-Tupel, dessen verbleibende Karten den größten Nutzen über alle Mengen an verbleibenden Karten aller Austausch Kandidaten hat, wird schließlich zurückgegeben.



---

**Algorithmus 3.12** Der Algorithmus für den Kartenaustausch

---

```
procedure FINDBESTEXCHANGE(handCards)
  combos ← CALCULATEALLCOMBINATIONS(handCards)
  partitions ← CALCULATEALLPARTITIONS(combos)
  ▷ Die beste Partition maximiert den Nutzen für das (Partition, Passen)-Aktionstupel
  bestPartition ← GETBESTPARTITION(partitions)
  ▷ Überprüfen, ob der Spieler voraussichtlich Tichu ankündigen wird
  callsTichu ← CALLSTICHU(partitions)
  bestUtility ←  $-\infty$ 
  bestExchange ← NULL
  ▷ Iteriere über alle Kartenaustausch 3-Tupel (linker Spieler, rechter Spieler, Partner)
  for all exchange = (l, r, p) ∈ GETCANDIDATES(bestPartition, callsTichu) do
    remainingCards ← REMOVECARDS(bestPartition, exchange)
    remainingCombos ← CALCULATEALLCOMBINATIONS(remainingCards)
    remainingPartitions ← CALCULATEALLPARTITIONS(remainingCombos)
    for all p ∈ remainingPartitions do
      utility ← GETUTILITY(p)
      if utility > bestUtility then
        bestUtility ← utility
        bestExchange ← exchange
      end if
    end for
  end for
  return bestExchange
end procedure
```

---

Ein Sonderfall liegt vor, wenn einer der anderen Spieler ein großes Tichu angekündigt hat. Wenn der Partner des Spielers das große Tichu angekündigt hat, dann ist die Auswahl der Karten auf Drache, Phönix und alle Asse beschränkt, soweit mindestens eine dieser Karten vorhanden ist. Wenn ein Spieler der gegnerischen Mannschaft ein großes Tichu angekündigt hat und der Spieler besitzt den Hund, dann wird der Hund immer an diesen weitergegeben. Der Hund muss immer als Abwurfkombination gespielt werden, wodurch er die Initiative des Spielers um 1 absenkt. Dies reduziert die Wahrscheinlichkeit, dass das große Tichu erfolgreich ist, erheblich.

Gelegentlich kann es vorkommen, dass dieser Algorithmus überhaupt kein gültiges Austausch-3-Tupel findet. Unter Umständen hat der Spieler weniger als drei Einzelkartenkombinationen in seiner besten Partition und keine Paare. Oder die Karten sind nicht zulässig, weil sie für den Partner einen zu niedrigen oder für die Gegner zu hohen Wert besitzen. In diesem Fall wird auf eine einfache Heuristik zurückgegriffen. Falls der Spieler kein Tichu ankündigen möchte, wird die höchste Karte des Blattes an den Partner weitergegeben, die niedrigsten beiden Karten werden mit den Gegnern ausgetauscht. Falls der Spieler ein Tichu ankündigen will, werden die drei niedrigsten Karten mit den anderen Spielern ausgetauscht. Dies ist in Algorithmus 3.12 aus Übersichtsgründen nicht dargestellt.

### 3.7.2. Das Wünschen

Das Wünschen spielt in Tichu keine allzu wichtige Rolle für den Ausgang des Spiels. Aus diesem Grund wurde lediglich eine einfache Heuristik implementiert.

Grundsätzlich wünscht sich die KI den niedrigsten Wert, den sie bisher noch nie gesehen hat. Für diesen Wert ist die Wahrscheinlichkeit, dass ein Gegner eine Bombe besitzt, etwas erhöht. Zwar sind Bomben selten, wenn sie aber auftreten, dann können sie die eigene Initiativberechnung erheblich erschweren, da scheinbar sichere Kombinationen überraschend abgestochen werden. Durch diesen Wunsch wird dieses Risiko etwas verringert.

Falls es keinen Wert gibt, den die KI noch nicht gesehen hat, dann wünscht sie sich den Wert der Karte, den sie an den nachfolgenden Gegner geschoben hat. Auch dieser Wunsch zielt darauf ab, das Bombenrisiko etwas zu verringern.

Weiter wünscht sich die KI zufällig ausgewählt entweder einen Buben oder eine Dame, falls der Mah Jong als Teil einer Straße gespielt wurde. Dieser Wunsch zielt darauf ab, dass ein Gegner eine hohe Straße spielen muss, falls er aus seinen Karten eine hohe Straße zusammenspielen kann. Sehr hohe Straßen werden in Tichu ungern gespielt, da ein Spieler gerade hohe Einzelkarten häufig als Initiativkombinationen verwenden möchte. In seltenen Fällen wird ein Spieler gezwungen, seinen Phönix zum Erzeugen einer passenden Straße zu verwenden. Dies allein führt gelegentlich dazu, dass ein Spieler ein angekündigtes Tichu nicht erfolgreich ausgespielt werden kann.

### 3.7.3. Das Verschenken des Drachens

Wenn der Drache einen Stich gewinnt, so ist dieser Stich entsprechend den Regeln an einen der Mitspieler der gegnerischen Mannschaft zu verschenken. Auch wenn der Drache mit 25 Punkten die, nach Punkten, wertvollste Karte ist, so ist es wichtiger, den Drachen für erfolgreiche Tichus zu nutzen. Grundsätzlich ist es deswegen gewünscht, dass man den Drachen ausspielt, auch wenn man der gegnerischen Mannschaft damit Punkte gibt.

Wenn der Drache jedoch nicht in der Kartenaustauschphase getauscht wird, dann gibt es nur eine Möglichkeit vom Verschenken des Drachens zu profitieren. Man schätzt ab, welcher der Gegner als letzter Spieler fertig wird und gibt diesem den Drachen. Als Annäherung dieser Idee wird der Drache deshalb immer an den gegnerischen Spieler gegeben, der derzeit am meisten Karten besitzt. Dabei wird angenommen, dass ein Spieler mit mehr Karten durchschnittlich länger benötigt, seine Karten auszuspielen, als ein Spieler mit weniger Karten.



## 4. Ergebnisse

Mit dem vorgestellten Ansatz waren zwei Zielen verknüpft. Zum einen sollte eine Spielstärke auf dem Niveau menschlicher Spieler erreicht werden. Zum anderen wurde angestrebt, dass eine schnelle Entscheidungsfindung eine interaktive Nutzung der KI ermöglicht. Aus diesem Grund wird im Folgenden zunächst die erreichte Spielstärke und anschließend die zum Treffen der Entscheidungen notwendige Rechenzeit untersucht.

### 4.1. Diskussion der Spielstärke und des Verhaltens

Ein Vergleich zwischen unterschiedlichen künstlichen Intelligenzen ist immer nur dann einfach, wenn diese direkt gegeneinander gesetzt werden können oder wenn es eine objektive Metrik für die Bewertung des Verhaltens einer KI gibt. Im ersteren Fall könnten einfach die Siege und Niederlagen festgestellt werden und statistisch ausgewertet werden, wobei kompatible Implementierungen vorausgesetzt werden. Da dem Autor keine offenen Tichu-Frameworks bekannt sind, die hätten genutzt werden können, ist die hier vorgestellte KI auf Basis eines eigens dafür entwickelten Frameworks entstanden. Entsprechend ist ein objektiver Vergleich mit anderen künstlichen Intelligenzen schwierig. Ebenfalls existieren keine sonstigen Metriken für die Bewertung des Verhaltens eines Spielers in den unterschiedlichen Spielsituationen.

Um zumindest einen subjektiven Eindruck der Spielstärke anbieten zu können, wurde eine sehr einfach gehaltene Studie durchgeführt. Die Teilnehmer spielten dabei mit einer zu der hier vorgestellten KI kompatiblen Android-App. Anschließend beantworteten sie einige Fragen zu der App und zur Spielstärke der Computerspieler. Die verwendete Android-App wurde im Rahmen einer anderen studentischen Arbeit parallel zu dieser KI entwickelt.

Es konnten 12 Teilnehmer mit unterschiedlicher Erfahrung in Tichu gewonnen werden. Damit ist diese Untersuchung in keiner Weise repräsentativ. Die Ergebnisse werden hier vorgestellt, um eine grobe Einordnung der Spielstärke zu ermöglichen.

In die Tabelle 4.1 sind die für die Spielstärke relevanten Fragen der Studie zu sehen, zusammen mit der Häufigkeit, mit der eine bestimmte Antwort gewählt wurde. Alle Antworten, hinter denen keine Zahlen stehen, wurden nicht gewählt. Für alle Fragen, bei denen die Summe der Antworten kleiner ist als die Anzahl der Probanden, haben einer oder mehrere der Probanden nicht auf diese Fragen geantwortet. Um zu zeigen, dass dies auch tatsächlich alle Fragen sind, die im Zusammenhang mit der Spielstärke relevant sind, ist eine vollständige Auflistung der Fragen in A.0.2 zu finden.

#### 4. Ergebnisse

Frage	Wie viel Erfahrung hast du mit dem Spielen von Tichu?
Antwort	“sehr wenig”(1), “wenig”(2), “ein bisschen”(3), “viel”(1), “sehr viel”(5)
Frage	Welchen Schwierigkeitsgrad hast du in den Optionen gewählt?
Antwort	“wie voreingestellt”(5), “einfach”, “normal”, “schwer”, “sehr schwer”, “extrem”(7)
Frage	Wie stark haben die Computerspieler gespielt?
Antwort	“sehr schwach”, “schwach”(1), “akzeptabel”(3), “stark”(6), “sehr stark”(2)
Frage	Wie stark haben die Computerspieler im Vergleich zu einer Partie mit menschlichen Spielern gespielt?
Antwort	“viel schwächer”, “etwas schwächer”(3), “ähnlich”(7), “stärker”(1), “viel stärker”
Frage	Wie oft haben die Computerspieler unsinnige Züge gespielt?
Antwort	“nie”(4), “gelegentlich”(6), “häufig”(2), “fast immer”
Frage	Wie oft kam dir dein Partner in die Quere?
Antwort	“nie”(5), “gelegentlich”(7), “häufig”, “fast immer”

**Tabelle 4.1.:** Die Fragen der Studie zusammen mit der Häufigkeit der gewählten Antworten.

Die Probanden wurden gebeten, mit dem Schwierigkeitsgrades “extrem” zu spielen, da nur dieser der stärksten Version der hier vorgestellten KI entspricht. Da diese Studie jedoch nicht nur die KI, sondern auch die verwendete Android-App zum Gegenstand hatte, wurde den Probanden auch die Möglichkeit gegeben, gegen andere Schwierigkeitsstufen zu spielen. In der App voreingestellt war ebenfalls der Schwierigkeitsgrad “extrem”. Erfreulicherweise nutzen damit alle Probanden den gewünschten Schwierigkeitsgrad.

Positiv hervorzuheben ist, dass lediglich einer der 12 Teilnehmer die Spielstärke der KI mit “schwach” bewertet. 8 der 12 Teilnehmer bewerteten die Spielstärke der KI mit “stark” oder “sehr stark”. Dies weist darauf hin, dass sich die KI auch gegen menschliche Spieler zu behaupten weiß.

In eine ähnliche Richtung zeigt auch die zweite für die Einschätzung der Spielstärke besonders wichtige Frage. Es wurde gefragt, wie stark die Computerspieler im Vergleich zu einer Partie mit menschlichen Spielern gespielt haben. Auf diese Frage antworteten 8 Teilnehmer mit “ähnlich” oder “stärker” und drei Teilnehmer mit “schwächer”. Auf Basis der Antworten auf diese beiden Fragen kann vermutet werden, dass die KI auf einem ähnlichen Niveau wie ein menschlicher Spieler spielt.

Neben der Spielstärke ist es, gerade wenn es menschliche Mitspieler gibt, auch wichtig, dass das Verhalten der KI nicht offensichtlich unsinnig ist. Unabhängig davon, ob sich unsinniges Verhalten negativ auf die Spielstärke auswirkt oder nicht, sind menschliche Spieler stets irritiert, wenn ein Computerspieler auf eine eigentlich klare Situation unsinnige Aktionen spielt. Entsprechend wurde gefragt, wie oft die Computerspieler unsinnige Züge gespielt haben. Hier wurde bei 10 von 11 Antworten “nie” oder “gelegentlich” genannt, was darauf schließen lässt, dass unsinnige Züge vorkommen, jedoch zumindest nicht in störender Häufigkeit.

Ebenfalls ist es wichtig, dass ein Computerspieler gut mit einem menschlichen Spieler kooperiert, da es zu Frustration bei menschlichen Spielern führen dürfte, wenn der eigene Partner eigentlich erfolgreiche Tichus regelmäßig verhindert oder auch nur klare Stiche des Spielers regelmäßig absticht. Um dies zu untersuchen wurde gefragt, wie oft dem menschlichen Spieler der von der KI gesteuerte Partner in die Quere kam. Auf diese Frage antworteten alle Teilnehmer der Studie mit "nie" oder "gelegentlich", was darauf hindeutet, dass der Partner noch gelegentlich Züge des Spielers stört. Allerdings ist auch hier davon auszugehen, dass dies ausreichend selten der Fall ist, um den Spaß am Spielen mit der KI nicht deutlich zu verringern.

Aufgrund der Antworten auf die im Rahmen der Studie gestellten Fragen kann die KI als auf dem Niveau von menschlichen Spielern betrachtet werden, was eines der wichtigsten Ziele dieser Arbeit war. Dabei ist zu beachten, dass erfahrene und sehr erfahrene Spieler deutlich stärker als die hier vorgestellte KI sein dürften. Das Verhalten der KI ist des Weiteren für menschliche Mitspieler akzeptabel, da die KI wenige Züge spielt, die menschliche Spieler irritieren dürften und die Unterstützung des menschlichen Spielers durch einen von der KI gesteuerten Partner gut zu funktionieren scheint.

## 4.2. Bewertung der Performance

Da parallel zur Entwicklung dieser Arbeit im Rahmen einer weiteren studentischen Arbeit eine zu dieser KI kompatible Android-App entstanden ist, war es ein wichtiges Ziel dieser Arbeit, dass auch die relative schwache in Smartphone und Tablets verwendete Hardware in der Lage ist, die von der KI zu treffenden Entscheidungen in ausreichender Geschwindigkeit berechnen zu können.

Als Minimalanforderungen wurden dabei ein mit einem Gigahertz getakteter Arm Cortex A9 mit einem Kern gewählt. Weiter wird vorausgesetzt, dass diesem 512 Megabyte an Arbeitsspeicher zur Seite stehen, die sich die Tichu App allerdings mit anderen Apps und dem Betriebssystem teilt.

Aus Gründen der Bedienbarkeit ist es erstrebenswert, dass die durchschnittlichen Antwortzeiten der KI bei deutlich unter einer Sekunde liegen. Dies ist wichtig, weil nach dem Zug eines menschlichen Spieler bis zu 3 KI-Spieler ziehen, die teilweise mehrere Entscheidungen treffen müssen. Auf der anderen Seite sind einzelne Ausreißer in der Berechnungszeit grundsätzlich akzeptabel, solange sich auch hier die Rechenzeit im Rahmen weniger Sekunden bewegt.

Zur Evaluierung der Performance der Implementierung der KI konnte nicht auf die App zurückgegriffen werden, die für die Evaluierung der Spielstärke benutzt wurde, da diese zum Zeitpunkt der Tests aufgrund von aufwendigen Animationen eine gewisse Menge an Rechenzeit für die Darstellung der App verwendet. Aus diesem Grund wurde im Rahmen dieser Arbeit eine weitere App erstellt, die lediglich eine rudimentäre Bedienung erlaubt, dafür aber auch keine zusätzliche Rechenzeit für Animationen oder Ähnliches verwendet.

#### 4. Ergebnisse

---

Aktion	Typ der Messung	A9 1,5 GHz	i5 3427
Standardsituation	Maximum	5329ms	179ms
	Durchschnitt	184ms	3ms
Kartenaustausch	Maximum	7072ms	159ms
	Durchschnitt	1262ms	19ms
Tichu-Ankündigung	Maximum	4981ms	170ms
	Durchschnitt	726ms	22ms
Großes-Tichu-A.	Maximum	154ms	13ms
	Durchschnitt	15ms	< 1ms
Wünschen	Maximum	1ms	< 1ms
	Durchschnitt	< 1ms	< 1ms
Drachen-Verschenken	Maximum	< 1ms	< 1ms
	Durchschnitt	< 1ms	< 1ms
Runden		100	100

**Tabelle 4.2.:** Performance der Implementierung auf verschiedener Hardware

Zudem ist es mit dieser App möglich, alle Spieler durch Computerspieler zu ersetzen. Zur Messung der Performance wurde von dieser Möglichkeit Gebrauch gemacht.

Um präzise Rechenzeiten zu messen, wurde jeweils vor und nach einer Aktion die aktuelle Zeit in Millisekunden gespeichert. Aus der Differenz ergibt sich die zum Errechnen einer Aktion notwendige Zeit. Parallel wurde geprüft, ob es sich bei der Differenz um die bisher höchste bekannte Differenz handelt. Gegebenenfalls wurde die Rechenzeit der Aktion in diesem Fall als maximale Rechenzeit der Aktion gespeichert.

Getestet wurde mithilfe eines Samsung Galaxy Note 10.1, welches einen Prozessor mit vier Cortex A9 Kernen, die mit 1,5 GHz takten, besitzt. Dabei muss betont werden, dass die KI lediglich einen Thread nutzt. Um zu demonstrieren, dass die Implementierung auf herkömmlichen PCs sehr performant ist, wurde ein Intel i5 3427 mit 2 Kernen und einem Takt von 2,8 GHz<sup>1</sup> für Messungen verwendet. Die Zeiten wurden dabei für jede Plattform über 100 Runden gemessen.

Die Ergebnisse dieser Messungen sind in Tabelle 4.2 zu sehen. Dabei zeigt sich, dass die durchschnittliche Zeit, bis eine Entscheidung getroffen wurde, relativ kurz ist, die maximale Rechenzeit dafür bei einigen Aktionen aber stark abweicht.

Die erhöhte maximale Rechenzeit kommt dadurch zustande, dass am Anfang eines Spiels die Anzahl der Karten hoch ist, was die Anzahl der Kombinationen und damit wiederum die Anzahl der Partitionen erhöht. Aus diesem Grund sind am Anfang eines Spiels sehr viele Aktionstupel zu untersuchen. Im weiteren Verlauf einer Runde nimmt die Zahl der zu überprüfenden Aktionstupel aber schnell stark ab, wodurch sich die relativ geringe durchschnittliche Rechenzeit ergibt.

<sup>1</sup>Trifft nur bei der Belastung mit einem Thread zu. Die KI nutzt allerdings lediglich einen Thread.

Dies erklärt die hohe maximale Rechenzeit aber nur zum Teil. Ebenfalls eine wichtige Rolle spielt der Besitz des Phönix, da dieser die Anzahl der Kombinationen, die aus einer Kartenmenge gebildet werden können, stark erhöht. Daraus wiederum folgt eine vergrößerte Menge an Partitionen. Da allerdings nur ein Phönix im Kartendeck existiert, ist die besonders hohe maximale Rechenzeit nur in seltenen Fällen und nur bei einem einzelnen Spieler zu erwarten.

Wichtig ist auch, dass, um zu entscheiden, ob ein Tichu angekündigt werden soll, die beste Aktion berechnet wird, die der Spieler in diesem Spielzustand wählen kann. Dafür wurde die Funktion *GetMaxUtility* aus Algorithmus 3.5 so modifiziert, dass sie zwar den höchsten Nutzen zurückgibt, aber als Seiteneffekt die beste Aktion in einer Variablen speichert. Die beste Aktion ist dadurch in der auf die Tichu-Ankündigung direkt folgende Standardsituation bereits bekannt und muss nicht erneut berechnet werden. Dadurch ist ausgeschlossen, dass auf eine maximal lange Tichu-Ankündigung eine maximal lange Standardsituation folgt.

### 4.3. Schwächen und Verbesserungsvorschläge

Wenn einige Schwächen behoben werden, ist noch eine deutliche Verbesserung der Spielstärke mit dem hier vorgestellten Ansatz zu erwarten. Die derzeit vorhandenen Schwächen und einige Verbesserungsvorschläge für die KI sollen hier kurz vorgestellt werden.

Zentral für eine weitere Verbesserung der Spielstärke dürfte ein verbessertes Modell zur Initiativberechnung sein, da auf die Initiativwerte in den Eigenschaften der Bewertungsfunktion häufig zurückgegriffen wird. Falls der vorgeschlagene Ansatz beibehalten werden soll, wird insbesondere vorgeschlagen einen besseren Kontext zu wählen, für den eine Wahrscheinlichkeitsaussage getroffen wird.

Die wahrscheinlich größte Schwäche, die aus dem aktuellen Modell für die Initiativberechnung resultiert, ist eine Abnahme der Spielstärke der KI, je weniger Karten noch im Spiel sind. Dies betrifft vor allem alle Spieler, die kein Tichu angekündigt haben, was dazu führt, dass die KI es zwar schafft, mit guten Karten korrekt Tichu anzusagen und als Erster fertig zu werden. Allerdings ist es als Spieler relativ einfach, gegen die KI nicht Letzter zu werden.

Derzeit verwendet die KI die Statistiken, auf denen die Initiativberechnung aufbaut, auch noch in den späten Phasen einer Runde. Aufgrund der relativ schlechten Entscheidungen, die zu diesem Zeitpunkt getroffen werden, wird vermutet, dass die statistischen Aussagen zu wenig spezifisch für den jeweiligen Spielzustand sind. Dieses Problem lässt sich auf zwei Wegen lösen. Entweder sollte der Kontext für die Initiativberechnung erweitert werden, oder stärker versucht werden, auf Basis der wenigen verbleibenden Karten zu schließen, welche Kombinationen ein Spieler überhaupt noch besitzen kann. Daraus müsste direkt abgeleitet werden, was ein starker Zug für die Gegner wäre. Ein recht eleganter Ansatz wäre es dabei, wenn an dieser Stelle wieder auf einen Minimax-Algorithmus zurückgegriffen werden würde. Ab wann dies aus kombinatorischen Gründen möglich ist, müsste experimentell untersucht werden.



#### 4. Ergebnisse

---

Der Umgang mit Bomben ist verbesserungsfähig. Die KI neigt dazu, Bomben relativ früh zu spielen, was eher schwach ist. Bomben sind vor allem dann stark, wenn sie einen praktisch sicheren Stich eines Spielers schlagen, der optimalerweise Tichu angekündigt hat. Dadurch, dass Bomben selten sind, besteht damit eine reelle Chance, dass die Initiativberechnung des Gegners soweit ungünstig ist, dass es ihm nicht mehr möglich ist, sein Blatt vollständig auszuspielen.

Des Weiteren neigt die KI dazu, zu früh zu hohe Karten auszuspielen. Zwar wird dieses Problem eigentlich durch die Initiativabschätzung gehandhabt, aber gerade wenn die KI ein hohes Blatt mit hohem Initiativwert hat, dann spielt sie häufig vor allem Asse und den Drachen früher als notwendig. Das frühe Spiel dieser Initiativkombinationen kann besonders von menschlichen Gegnern relativ einfach ausgenutzt werden, indem sie mit dem Ausspielen ihres Blattes warten, bis die KI-Spieler ihre wichtigsten Initiativkombinationen ausgespielt haben. Anschließend kann er dann sein eigenes Blatt relativ schnell vollständig ausspielen.

Sowohl die zu früh gespielten Bomben als auch das Problem von zu früh gespielten sonstigen Kombinationen ist darauf zurückzuführen, dass die KI derzeit noch nicht abschätzt, wie lange es noch dauert, bis der nächste Spieler alle seine Handkarten ausgespielt hat. Dies ist in Tichu dadurch erschwert, dass Spieler tendenziell versuchen, längere Kombinationen als Letztes zu spielen, um den anderen Spielern nicht die Möglichkeit zu geben, ein fertig werden des Spielers zu verhindern. Könnte mit akzeptabler Präzision abgeschätzt werden, wie lange es dauert, bis der nächste Spieler fertig wird, dann könnte versucht werden, auf gewisse Kombinationen noch zu passen. Dies würde es erlauben, zum Beispiel mit dem Drachen ein Ass anstatt eines Königs zu stechen oder mit einer Bombe eine lange Straße oder ein Fullhouse zu schlagen.

Unabhängig von diesen Schwächen wird vermutet, dass mit einer Verteilung von Rollen an die Spieler eine Verbesserung der Spielstärke erreicht werden könnte. Dabei übernehmen Spieler in unterschiedlichen Spielen und Spielphasen unterschiedliche Rollen, die beispielsweise wie folgt untergliedert werden könnten:

- Tichu-Ankündigen-Rolle: Am Anfang eines Spiels übernehmen alle Spieler diese Rolle.
- Tichu-Ausspielen-Rolle: Ein Spieler befindet sich in dieser Rolle, wenn er Tichu angekündigt hat.
- Partner-Unterstützen-Rolle: Ein Spieler erhält diese Rolle, wenn sein Partner Tichu angekündigt hat.
- Punkte-Spiel-Rolle: Wenn keiner der Spieler der eigenen Mannschaft ein Tichu angekündigt hat, dann wird lediglich um Punkte gespielt.
- (Eventuell) Tichu-Verhindern-Rolle: Wenn kein Spieler der eigenen Mannschaft ein Tichu angekündigt hat, jedoch einer der Spieler der gegnerischen Mannschaft.

Durch die Verteilung derartiger Rollen würde eine höhere Flexibilität bei der verwendeten Bewertungsfunktion erreicht werden, da es möglich wäre, jeder Rolle zumindest einen eigenen Gewichtsvektor oder gar eine eigene Bewertungsfunktion zuzuordnen. Derzeit wird mittels expliziter Prüfungen sichergestellt, dass eine Eigenschaft nur in einem bestimmten

Kontext einen bestimmten Wert erhält, besonders deutlich ist dies bei der Tichu-Endspiel Eigenschaft. Durch derartige Rollenverteilungen könnte ein Teil dieser Prüfungen implizit in die Gewichtsvektoren codiert werden.

Die Verteilung der Rollen bringt allerdings auch eine Schwierigkeit mit sich. Mit der Anzahl der Gewichtsvektoren steigt auch die Zeit, die zum Kalibrieren der Gewichtsvektoren notwendig ist. Da der aktuelle Ansatz zur Gewichtskalibrierung bereits eine Laufzeit von einigen Stunden besitzt, müsste der Kalibrierungsprozess deutlich verbessert werden.



## 5. Zusammenfassung und Ausblick

In dieser Arbeit wurde eine künstliche Intelligenz für das Kartenspiel Tichu entwickelt, die mehrere Erwartungen erfüllen sollte.

Zum einen war angestrebt, dass die KI eine Spielstärke auf dem Niveau menschlicher Spieler erreicht. Wie mithilfe einer sehr einfach gehaltenen Studie in 4.1 ermittelt wurde, wurde die KI von einer nicht-repräsentativen Menge an Testpersonen entsprechend bewertet. Auch wenn die Untersuchung einfach gehalten war, lässt sich diese als starkes Indiz werten, dass die KI die angestrebte Spielstärke erreicht hat.

Ein weiteres Ziel war die Nutzung hinreichend effizienter Algorithmen, die es ermöglichen, dass die KI auch auf schwacher Hardware interaktiv genutzt werden kann. Wie in 4.2 gezeigt wurde, ist es möglich, die KI auf einer verhältnismäßig wenig leistungsstarken mobilen Plattform zu benutzen. Dabei muss ein menschlicher Spieler in am Anfang einer Runde höchstens wenige Sekunden warten. Die durchschnittlichen Wartezeiten liegen in den restlichen Phasen des Spiels bei deutlich unter einer Sekunde. Aufgrund der kurzen Rechenzeiten ist eine interaktive Nutzung der KI ohne störende Wartezeiten möglich. Wird zudem eine stärkere Hardwareplattform verwendet, bleiben die Wartezeiten selbst am Anfang einer Runde deutlich unter einer Sekunde.

Diese Ziele konnten durch die Einführung einer Vielzahl von Abstraktionen und Heuristiken erreicht werden, die eine effiziente Behandlung der in Tichu auftretenden Spielsituationen ermöglichen. Eine besonders wichtige Rolle spielt dabei der Begriff der Partition, der zentral für die Bewertung von Spielsituationen ist und in etwa einen Plan in anderen Spielen ersetzt. Weiter wurde ein Initiativbegriff eingeführt, mit dessen Hilfe es möglich ist abzuschätzen, ob und wann eine Kombination ausgespielt werden kann. Diese Abstraktionen wurden in einer zentralen Bewertungsfunktion im Rahmen mehrerer Eigenschaften verwendet. Mit dieser Bewertungsfunktion ist eine Untersuchung der möglichen Aktionen eines Spielzustandes möglich und damit eine Wahl der besten Aktion.

### Ausblick

Tichu ist insofern ein ungewöhnliches Spiel, als dass die Verwendung klassischer Suchalgorithmen sehr unattraktiv erscheint. Die Anzahl der unbekannt Karten und deren Verteilung auf die Spieler führen zu einem Suchbaum, der aufgrund seiner Größe von einer praktischen Verwendung ausgeschlossen werden muss.

Dieses Problem kann dadurch gelöst werden, dass die üblicherweise erst in einer gewissen Tiefe zu findende Bewertungsfunktion, die den Nutzen einer Spielsituation ermittelt, bereits zur Untersuchung der unmittelbaren Aktionen des Spielers verwendet wird. Wie der hier vorgestellte Ansatz zeigt, kann trotzdem eine vergleichsweise hohe Spielstärke erreicht werden, indem eine komplexere Bewertungsfunktion verwendet wird.

Diese erreicht ihre Spielstärke vor allem durch die Modellierung der Aspekte der Spielsituationen in Tichu. Darunter ist das Modell für die Initiative einer Aktion besonders wichtig. Das heißt, dass ein gangbarer Weg für weitere Verbesserungen in der Konstruktion besserer Modelle besteht. Einige Schwächen der KI und daraus resultierende Verbesserungsvorschläge wurden in 4.3 vorgestellt.

In dieser Arbeit wurde eine einzelne Bewertungsfunktion verwendet, die das zentrale Element zum Treffen aller wichtigen Entscheidungen darstellte. Zukünftige künstliche Intelligenzen für Tichu werden wohl eher auf mehrere Bewertungsfunktionen zurückgreifen, wobei sich diese allerdings einen Großteil der Eigenschaften teilen werden. Wenn mehrere Bewertungsfunktionen verwendet werden, steigt allerdings auch die Anzahl der zu kalibrierenden Gewichte, weswegen eine besserer Ansatz notwendig sein wird, um die Gewichte zu kalibrieren.

Die erreichte Spielstärke und die durch bessere Modelle vermuteten weiteren Verbesserungsmöglichkeiten zeigen wiederum die Validität des grundlegenden Ansatzes. Insbesondere die Modellierung einer Aktion als 2-Tupel einer Partition mit der zu spielenden Kombination zusammen mit einer einfachen Maximierung über den lokalen Nutzen. Dabei kann insbesondere die effiziente Berechnung der eindeutigen Partitionsmenge auch für abgewandelte Ansätze verwendet werden.

# A. Anhang

## A.0.1. Eindeutigkeit einer Kartenmenge

Eine Menge von Tichukarten hat eine wesentliche Eigenschaft. Dies ist die Eindeutigkeit der Kombination, die von den Karten dieser Menge dargestellt wird, falls sie überhaupt eine Kombination darstellt.

Der Beweis erfolgt über die Aufzählung über die Typen.

- Eine Einzelkarte kann gleichzeitig kein anderer Typ sein, da alle anderen Typen von Kombinationen mehr als eine Karte erfordern.
- Ein Paar kann gleichzeitig von keinem anderen Typ sein, da ein Paar genau 2 Karten erfordert. Alle anderen Typen aber mehr als zwei oder genau eine Karte.
- Ein Trippel kann gleichzeitig von keinem anderen Typ sein, da ein Trippel genau 3 Karten erfordert. Alle anderen Typen erfordern mehr als 3 Karten oder genau eine oder genau zwei Karten.
- Eine Bombe kann gleichzeitig von keinem anderen Typ sein, da eine Bombe genau 4 Karten erfordert. Alle anderen Typen, bis auf die Treppe, erfordern mehr als 4 Karten oder genau eine, zwei oder drei Karten. Eine Treppe kann nicht mit einer Bombe verwechselt werden, da sie aus zwei unterschiedlichen aufeinanderfolgenden Paaren besteht, eine Bombe jedoch nur Karten eines bestimmten Wertes besitzt.
- Ein Fullhouse kann von keinem anderen Typ sein. Alle Typen, die 4 oder weniger Karten erfordern, scheiden aus, da ein Fullhouse aus genau 5 Karten besteht. Da eine Treppe aus beliebig vielen Paaren besteht, ist die Anzahl ihrer Karten immer gerade, also niemals 5. Eine Straße und eine Straßenbombe bestehen nur aus Karten von unterschiedlichem Wert, ein Fullhouse besitzt genau einem Drilling, also drei Karten gleichen Werts.
- Eine Straße besteht nur aus unterschiedlichen, aber aufeinanderfolgenden Werten. Der einzige Typ, der dieses Kriterium ebenfalls erfüllt ist die Straßenbombe. Hier gilt jedoch per Definition, dass eine Straßenbombe niemals als bloße Straße gespielt werden darf [Fat].
- Eine Straßenbombe besteht nur aus unterschiedlichen, aber aufeinanderfolgenden Karten derselben Farbe. Der einzige Kandidat, der dieses Kriterium auch erfüllen würde, wäre die Straße. Wie erwähnt, ist dies jedoch per Definition ausgeschlossen.

Weiter gilt, dass innerhalb eines Typs eine Kombination genau einen Wert einnimmt. Da eine Menge von Karten damit nicht gleichzeitig eine Kombination von einem anderen Typ oder eine Kombination vom gleichen Typ, aber mit anderem Wert darstellt, gilt, dass es höchstens eine solche Kombination gibt.

Die Kombination, die alle Karten einer Kartenmenge umfasst, wird die charakteristische Kombination der Menge genannt. Eine charakteristische Kombination muss nicht existieren, zum Beispiel enthält  $\{3, 3, 4, 3, 3\}$  keine charakteristische Kombination.

Die charakteristische Kombination einer Menge ist insbesondere nützlich, um zu überprüfen, ob eine ausgespielte Menge an Karten auch tatsächlich ein gültiger Zug darstellt.

### **A.0.2. Der Fragebogen zur Evaluierung der KI**

Zur Untersuchung der KI wurde eine sehr einfach gehaltene Studie durchgeführt. Dabei wurden nicht nur Fragen direkt zur KI gestellt, sondern auch zu einer Android-App, zu deren Implementierung die Implementierung der vorgestellten KI kompatibel ist.

Zwar wurden die Antworten auf diese Fragen schon in 4.2 für die KI ausgewertet. Um allerdings noch zu zeigen, dass die restlichen Fragen keinen Bezug zur Spielstärke der KI besitzen, werden die gestellten Fragen und den zugehörigen Antwortmöglichkeiten in Tabelle A.1 vollständig aufgelistet.

Frage	Name und Modell des Android Geräts:
Antwort	Freitext
Frage	Wurde die App auf deinem Android Gerät richtig dargestellt?
Antwort	“ja”, “nein”, “nein, App ist abgestürzt”
Frage	Wie flüssig ließ sich die App bedienen?
Antwort	“störende Wartezeiten”, “lange Wartezeiten”, “geringfügige Wartezeiten”, “flüssig”, “sehr flüssig”
Frage	Wie hilfreich fandest du das Tutorial?
Antwort	“nicht hilfreich”, “wenig hilfreich”, “hilfreich”, “sehr hilfreich”, “äußerst hilfreich”
Frage	Wie viel Erfahrung hast du mit dem Spielen von Tichu?
Antwort	“sehr wenig”, “wenig”, “ein bisschen”, “viel”, “sehr viel”
Frage	Welchen Schwierigkeitsgrad hast du in den Optionen gewählt?
Antwort	“wie voreingestellt”, “einfach”, “normal”, “schwer”, “sehr schwer”, “extrem”
Frage	Wie stark haben die Computerspieler gespielt?
Antwort	“sehr schwach”, “schwach”, “akzeptabel”, “stark”, “sehr stark”
Frage	Wie stark haben die Computerspieler im Vergleich zu einer Partie mit menschlichen Spielern gespielt?
Antwort	“viel schwächer”, “etwas schwächer”, “ähnlich”, “stärker”, “viel stärker”
Frage	Wie oft haben die Computerspieler unsinnige Züge gespielt?
Antwort	“nie”, “gelegentlich”, “häufig”, “fast immer”
Frage	Wie oft kam dir dein Partner in die Quere?
Antwort	“nie”, “gelegentlich”, “häufig”, “fast immer”
Frage	Wie wichtig wäre dir ein Multiplayermodus über Bluetooth?
Antwort	“sehr wenig wichtig”, “wenig wichtig”, “wichtig”, “sehr wichtig”
Frage	Sonstige Anmerkungen (optional)
Antwort	Freitext

**Tabelle A.1.:** Fragen und Antwortmöglichkeiten der durchgeführten Umfrage





## Literaturverzeichnis

- [Fat] Tichu bei Fata Morgana. URL <http://www.fatamorgana.ch/tichu/tichu.asp>. (Zitiert auf den Seiten 9, 12 und 69)
- [RN05] S. J. Russell, P. Norvig. *Artificial intelligence: a modern approach*. Prentice-Hall, New Delhi, 2nd ed. Auflage, 2005. (Zitiert auf den Seiten 9, 19, 20, 21 und 42)
- [Tica] Tichu in der Spieledatenbank Luding. URL <http://sunsite.informatik.rwth-aachen.de/luding/Skripte/GameData.py/DEgameid/2239>. (Zitiert auf Seite 11)
- [Ticb] Tichu Spielregeln. URL [http://www.fatamorgana.ch/tichu/ti\\_regel.pdf](http://www.fatamorgana.ch/tichu/ti_regel.pdf). (Zitiert auf Seite 11)

Alle URLs wurden zuletzt am 30.04.2013 geprüft.



## **Erklärung**

Ich versichere, diese Arbeit selbstständig verfasst zu haben. Ich habe keine anderen als die angegebenen Quellen benutzt und alle wörtlich oder sinngemäß aus anderen Werken übernommene Aussagen als solche gekennzeichnet. Weder diese Arbeit noch wesentliche Teile daraus waren bisher Gegenstand eines anderen Prüfungsverfahrens. Ich habe diese Arbeit bisher weder teilweise noch vollständig veröffentlicht. Das elektronische Exemplar stimmt mit allen eingereichten Exemplaren überein.

---

Ort, Datum, Unterschrift