

JUNGE

wissenschaft

JungforscherInnen publizieren
online | *peer reviewed* | original

Verlag:
Physikalisch-
Technische
Bundesanstalt



Mathematik &
Informatik

Die Würfel sind gefallen

Strategieoptimierung des Spieles
Qwix durch maschinelles Lernen

Für das Würfelspiel Qwixx wurde eine KI entwickelt, die mittels eines evolutionären Algorithmus KI-Spieler trainiert und so eine starke Strategie findet. Dazu werden auch verschiedene Trainingsmethoden verglichen und bewertet. Auf diese Weise wurde ein KI-Spieler („AlphaQwixx“) gefunden, der mit menschlichen Spielern konkurrieren und höhere Punktzahlen erreichen kann.

DIE JUNGFORSCHERIN



Susann Janetzki (2002),

Landesschule Pforta,
Naumburg (Saale)

Eingang der Arbeit:

30.6.2021

Arbeit angenommen:

5.8.2021





Die Würfel sind gefallen

Strategieoptimierung des Spieles Qwixx durch maschinelles Lernen

1. Einleitung

Selbstlernende Algorithmen, die zur künstlichen Intelligenz (KI) gehören, werden heutzutage vielseitig verwendet [1]. Sie ermöglichen beispielsweise bei Spielen, dass ein Mensch auch ohne menschlichen Gegenspieler auf bzw. gegen einen Computer spielen kann: Man sagt dazu auch, der Mensch spiele gegen eine KI. So gewann im Jahr 1997 Deep Blue gegen den damals amtierenden Schach-Weltmeister Garry Kasparow [2] und AlphaGo besiegte 2015 den Europameister Fan Hui im Spiel Go [3].

Die vorliegende Arbeit befasst sich mit dem Würfelspiel „Qwixx“, das vom Nürnberger-Spielkarten-Verlag 2012 herausgegeben wurde. Forschungsziel ist, mithilfe von maschinellem Lernen eine KI zu entwickeln, die eine starke Strategie für Qwixx findet.

1.1 Spielregeln

Eine Runde Qwixx mit zwei bis fünf Spielern dauert etwa 15 Minuten [4]. Jeder Spieler erhält einen Spielzettel vom Spielblock (siehe Abb. 1 [8]). Zusätzlich

werden sechs Würfel benötigt, von denen vier den Farben der bunten Reihen entsprechen und zwei weiß sind. Die Spieler sind nacheinander an der Reihe zu würfeln. Der Spieler, der würfelt, ist der „aktive Spieler“, alle anderen Spieler sind „passive Spieler“. Der aktive Spieler kann pro Zug zwei Kreuze auf seinen Spielzettel setzen, indem er zuerst die Summe der weißen Würfel in einer beliebigen Reihe ankreuzt und dann die Summe aus einem weißen und einem farbigen Würfel in der entsprechend farbigen Reihe. Dem aktiven Spieler ist auch erlaubt, nur ein Kreuz zu setzen oder einen Fehlwurf anzukreuzen. Passive Spieler dürfen in derselben Runde die Summe der weißen Würfel bei sich in einer beliebigen Reihe ankreuzen. Grundsätzlich kann in jeder Reihe nur von links nach rechts angekreuzt werden, entstandene Lücken können daher nachträglich nicht mehr mit Kreuzen gefüllt werden. Außerdem darf die Zahl ganz rechts (12 oder 2) in jeder Reihe nur angekreuzt werden, wenn mindestens fünf Kreuze in der Reihe platziert wurden. Nach dem Ankreuzen der Zahl ganz rechts muss auch das Schloss daneben angekreuzt werden, und die Reihe gilt für alle Spieler als „abgeschlossen“, d. h. es können für den Rest des Spiels keine Kreuze mehr in dieser Reihe gesetzt werden. Das Spiel ist zu Ende, wenn ein Spieler vier Fehlwürfe gesammelt hat oder mindestens zwei verschiedenfarbige Reihen abgeschlossen sind. Mittels der Punkteberechnung, die unten auf dem Spielblock abgebildet ist, können die Punktestände der Spieler ermittelt werden. Der Spieler mit den meisten Punkten ist Sieger [4].

Für die Punktzahl in einer Reihe gilt dabei, dass sie quadratisch mit der Anzahl der Kreuze wächst: Formel 1.

So erhält man bspw. mit zwölf Kreuzen in einer Reihe

$$\text{Punktezahl} = \frac{(\text{Anzahl der Kreuze in einer Reihe})^2 + \text{Anzahl der Kreuze in einer Reihe}}{2}$$

Formel (1)

$$\frac{(12)^2 + 12}{2} = \frac{156}{2} = 78 \text{ Punkte} . \quad (2)$$

1.2 Der Spieler mit der „besten“ Strategie

Bevor eingeschätzt werden kann, welche die „beste“ Strategie für das Spiel Qwixx ist, steht die Fragestellung, welche Erwartungen an einen Spieler mit der „besten“ Strategie gestellt werden. Dazu gibt es im Wesentlichen zwei Ansätze. Natürlich können auch KI-Spieler existieren, die eine Strategie aus einer Kombination der beiden unten genannten Ansätze verfolgen.

1.2.1 Gewinnhäufigkeit maximieren

Der erste Ansatz ist, dass die beste Strategie so konzipiert sein soll, dass der KI-Spieler möglichst oft gewinnt. Dieser Spieler legt keinen Wert auf seine absolute Punktzahl, sondern nur auf den Punktevorsprung zu den anderen Spielern. Dieser KI-Spieler würde versuchen, das Spiel zu beenden, sobald er diesen Punktevorsprung gesichert hat. Dabei ginge er beim Ankreuzen auch das Risiko größerer Lücken ein.

1.2.2 Punktzahl maximieren

Der zweite Ansatz geht von einer Strategie aus, bei der der KI-Spieler eine möglichst hohe Punktzahl erreicht. Dieser Spieler ließe nur wenige und relativ kleine Lücken beim Ankreuzen zu und könnte dadurch lange spielen.

Hat ein Mensch einen solchen KI-Spieler als Gegner, ist der Mensch – im Vergleich zu einem Spiel gegen einen KI-Spieler, der seine Gewinnhäufigkeit maximiert – mehr gefordert. Der Mensch muss selbst eine Taktik entwickeln, wie er möglichst lange dazu in der Lage bleibt, weitere Felder anzukreuzen. Das Spielende ist schwerer abzusehen und das Würfelglück beeinflusst die Länge des Spiels in einem höheren Maße als beim Spiel gegen den KI-Spieler, der möglichst schnell gewinnen möchte. Insgesamt ist die Spannung im Spiel gegen die KI, die versucht viele Punkte zu erhalten, größer.

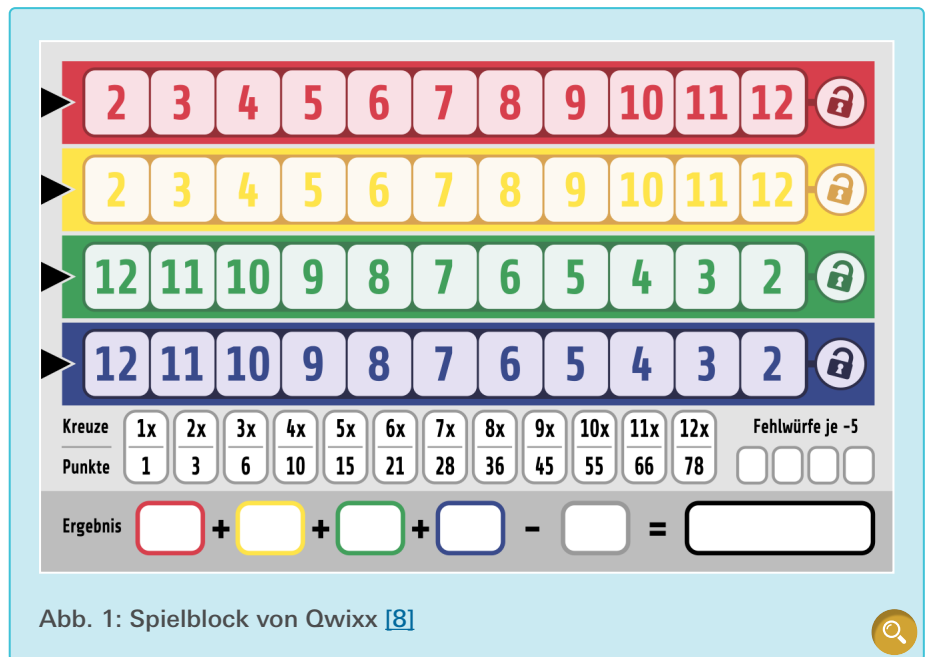


Abb. 1: Spielblock von Qwixx [8]

1.2.3 Vergleich

Wenn der Zweck des Spielers mit der besten Strategie sein soll, gegen Menschen zu spielen, so wäre eine KI, die nach dem zweiten Ansatz spielt, erstrebenswerter.

Ungeachtet dessen gewinnt in der Regel der KI-Spieler, dessen Ziel es ist, schnell zu gewinnen im Spiel gegen den KI-Spieler, der viele Punkte zu bekommen versucht, da dieser zum Zeitpunkt des Spielendes noch nicht so viele Kreuze gesammelt hat, wie der KI-Spieler, der das Spiel schnell beendet.

1.3 Umsetzung und Ziel der Arbeit

Der Code zur Umsetzung der KI wurde eigenständig entwickelt und in der Programmiersprache Python geschrieben. Python hat unter anderem den Vorteil, dass es objektorientiert, modern und gut lesbar ist. Der Code ist unter <https://github.com/sjanetzki/qwixx-trainer> zu finden.

Zum einen ist das Ergebnis dieses Projekts die tatsächlich ermittelte Strategie, die zum Sieg im Qwixx-Würfelspiel führt. Zum anderen werden auch verschiedene Trainingsmethoden in Hinblick auf die Strategiestärken und die

Reproduzierbarkeit der Strategien verglichen und die Zuverlässigkeit der verwendeten Methoden bewertet.

2. Theoretische Grundlagen

2.1 Evolutionärer Algorithmus

Für die Erstellung der Qwixx-KI wird als Methode ein evolutionärer Algorithmus verwendet. Dieser gehört zum Bereich des maschinellen Lernens.

Im Gegensatz zum bestärkenden Lernen muss bei dieser Methode nicht jede mögliche Situation primär auswendig gelernt werden. Eine solche Datenmenge wäre bei Qwixx von enormer Größe. Anstelle dessen soll der KI-Spieler – mit dem evolutionären Algorithmus – ähnliche Situationen erkennen und in diesen auch ähnlich handeln.

Evolutionäre Algorithmen sind stochastische Optimierungsalgorithmen, die der Biologie, speziell der Genetik, angenähert sind. Das Vokabular ist dementsprechend auch der Biologie entnommen. So gibt es Individuen mit Genen, die eine Generation bilden. Diese werden nach ihrer Leistungsfähigkeit (Fitness) sortiert und mittels Mutation, Rekombination und Selektion entsteht eine neue Generation. Dieser Prozess wird so lange wiederholt, bis ein sehr

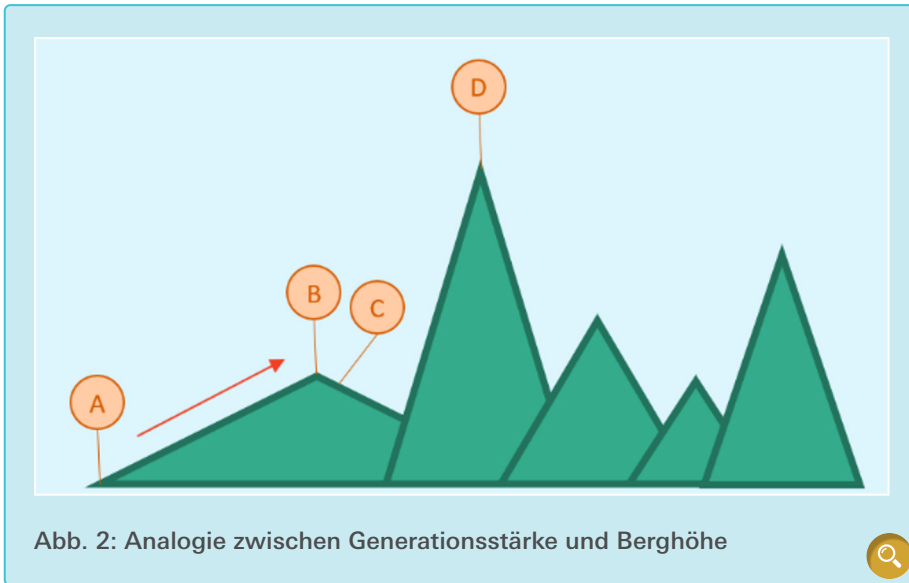


Abb. 2: Analogie zwischen Generationsstärke und Berghöhe

gutes, bzw. das bestmögliche Individuum geschaffen ist [5].

Auf Qwixx bezogen bedeutet das Folgende: Der sogenannte „Trainer“ wendet den evolutionären Algorithmus an. Er erstellt eine Population aus zufällig generierten Individuen. Ein Individuum entspricht dabei genau einem KI-Spieler mit genau einer zufälligen Strategie. Das Ziel des Trainers ist es, die Strategien zu optimieren. Die KI-Spieler treten in Gruppen mehrfach gegeneinander an und werden nach ihrem Erfolg sortiert. Die besten werden miteinander gekreuzt (Rekombination), Individuen werden gegebenenfalls mutiert und einige werden unverändert in die nächste Generation übernommen. Die schwächsten KI-Spieler scheiden aus. Anschließend wird die Population auf ihre ursprüngliche Größe mit völlig neuen, zufällig generierten KI-Spielern aufgefüllt.

2.2 Strategie als künstliche Intelligenz

Der Trainer, der den evolutionären Algorithmus anwendet, ist eine künstliche Intelligenz. Neben dem Trainer existiert auch eine weitere Art von künstlicher Intelligenz: die Spieler mit ihren Strategien. Die Strategie bestimmt, wie sich ein Spieler in Abhängigkeit von der Spielsituation entscheidet. Dabei ist die Strategie starr, das bedeutet, dass ein

Spieler sich in gleichen Situationen mit der gleichen Strategie stets gleich verhält. Für die Spieler wäre es sinnvoll, den quadratischen Punktezuwachs interpretieren zu können, also eine Strategie zu besitzen, die auf den Punktezuwachs eingeht.

Es findet kein Lernprozess statt. Deshalb sind die Spieler zwar eine künstliche Intelligenz, fallen aber nicht in den Bereich des maschinellen Lernens.

2.3 Angemessenheit elitärer Evolution

Eine Frage, die sich beim evolutionären Algorithmus stellt, ist, inwieweit die Evolution „elitär“ sein darf oder ob auch Rückschritte in einer Evolution erlaubt sein sollten. Zum einen soll größtenteils nur die Elite in die nächste Generation übernommen werden, zum anderen kann es kontraproduktiv sein, nur gleich gute und bessere Generationen als Folgegenerationen zu akzeptieren.

Das Problem lässt sich am besten mit einer Analogie erklären, die in [Abb. 2](#) dargestellt wird. Man stelle sich vor, an Stelle A würde man eine blinde KI setzen und sie solle den höchsten Punkt finden. Sie geht also entlang der Pfeilrichtung immer bergauf, denn solange sie aufwärts geht, ist sie noch nicht am höchsten Punkt. Die KI kommt an B an, versucht einen weiteren Schritt Richtung C zu machen und merkt, dass es bergab geht. Da sie nichts sehen kann, kann sie nicht wissen, dass sie den höchsten Punkt, also D, noch nicht erreicht hat und erklärt deshalb B für den höchsten Gipfel.

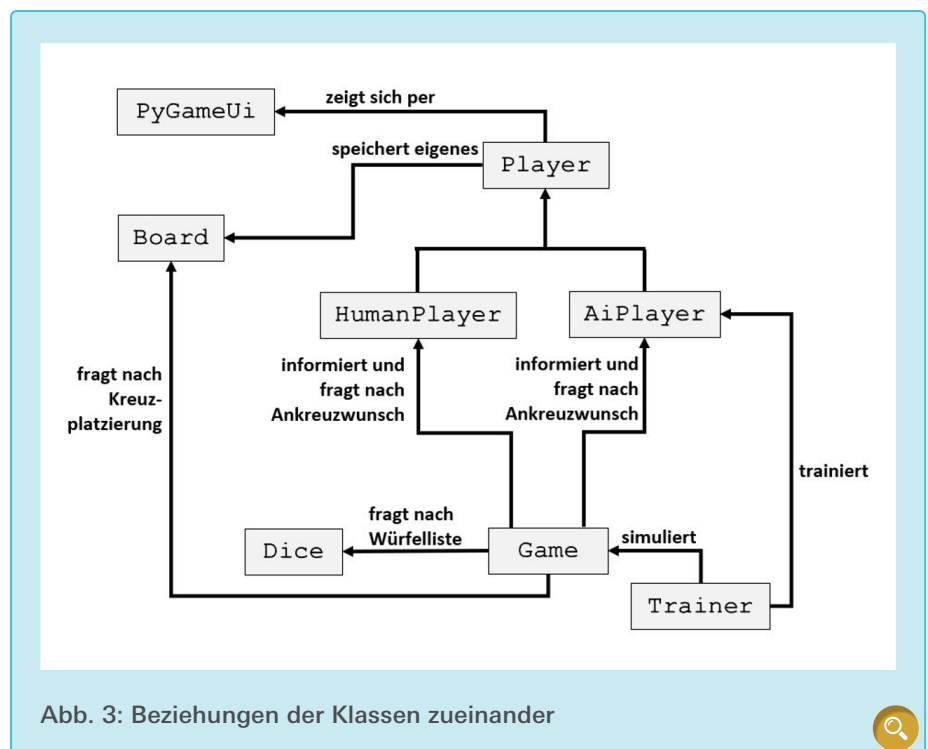


Abb. 3: Beziehungen der Klassen zueinander

Was in dem Beispiel die Höhe der Berge ist, ist bei der Evolution die Stärke einer Generation. Es ist möglich, dass es in einer Evolution – dargestellt durch den Weg von links nach rechts – auch Rückschritte gibt, aus denen möglicherweise bessere Generationen (wie der höhere Berg D) hervorgehen. Deshalb lässt der Trainer auch Rückschritte in der Evolution zu.

3. Aufbau des Codes

3.1 Dateien und Hierarchie

Der Code des Projekts besteht aus mehreren Klassen, die in [Abb. 3](#) dargestellt sind.

Das *Game* (Spiel) hat für den Code die zentrale Rolle und lenkt die meisten Abläufe. Es erhält von der Klasse *Dice* (Würfel) eine Liste von Würfelergebnissen und teilt den Spielern (*HumanPlayer* oder *AiPlayer*) darauf basierend mögliche Ankreuzmöglichkeiten mit. Dann fragt das Game die Spieler so lange nach ihrem Ankreuzwunsch (der bei aktiven Spielern aus einem oder zwei Kreuzen und bei passiven Spielern aus einem Kreuz oder dem Passen bestehen kann), den sie tätigen wollen, bis ein gewählter Ankreuzwunsch gültig ist.

Um später verschiedene Evolutionszyklen besser miteinander vergleichbar zu machen, wird beim Abfragen der Wür-

fellisten eine Seed-Funktion verwendet, wobei jede Spielrunde ihren eigenen Seed-Wert zugeteilt bekommt [6]. Somit ist z.B. die Würfelliste in der 35. Generation im ersten Spiel in der siebten Runde in allen Evolutionszyklen gleich und Unterschiede durch Würfelglück werden verringert.

Der gewählte Zug wird auf dem *Board* eines Spielers gespeichert und gegebenenfalls via *PyGameUi* (Benutzeroberfläche) für einen *HumanPlayer* (menschlicher Spieler) visualisiert.

Der *Trainer* wiederum interagiert mit dem *AiPlayer* (KI-Spieler) und dem *Game*. Dazu simuliert der *Trainer* ein Spiel mittels der *Game*-Klasse und lässt mehrere *AiPlayer* gegeneinander spielen.

Formel (4)

$$\begin{aligned}
 \text{Qualität} = & (a_0 \cdot x_0^2 + b_0 \cdot x_0 + c_0) + (a_1 \cdot x_1^2 + b_1 \cdot x_1 + c_1) + \\
 & (a_0 \cdot x_2^2 + b_0 \cdot x_2 + c_0) + (a_1 \cdot x_3^2 + b_1 \cdot x_3 + c_1) + (a_0 \cdot x_4^2 + b_0 \cdot x_4 + c_0) + \\
 & (a_2 \cdot x_5^2 + b_2 \cdot x_5 + c_2) + (a_0 \cdot x_6^2 + b_0 \cdot x_6 + c_0) + (a_2 \cdot x_7^2 + b_2 \cdot x_7 + c_2) + \\
 & (a_3 \cdot x_8^2 + b_3 \cdot x_8 + c_3)
 \end{aligned}$$

$$\begin{aligned}
 \text{Qualität} = & (1 \cdot 1^2 + 2 \cdot 1 + 0) + (0 \cdot 2^2 + 1 \cdot 2 + 0) + \\
 & (1 \cdot 2^2 + 2 \cdot 2 + 0) + (0 \cdot 5^2 + 1 \cdot 5 + 0) + (1 \cdot 1^2 + 2 \cdot 1 + 0) + \\
 & (0 \cdot 12^2 - 1 \cdot 12 - 1) + (1 \cdot 2^2 + 2 \cdot 2 + 0) + (0 \cdot 10^2 - 1 \cdot 10 - 1) + \\
 & (0 \cdot 0^2 - 5 \cdot 0 + 0)
 \end{aligned}$$

$$\text{Qualität} = 3 + 3 + 8 + 5 + 3 - 13 + 8 - 11 + 0 = 6$$

3.2 AI Player

AiPlayer (KI-Spieler) ist eine Unterklasse von *Player* (Spieler). Sie ist für diese Arbeit von besonderer Relevanz. Diese Klasse bewertet die Qualität von möglichen Zügen. Basierend darauf wird der mutmaßlich beste Zug ausgewählt.

Wie der Spielzettel nach jedem möglichen Zug aussieht, wird als *hypothetische Situation x* bezeichnet. Diese ist ein Array aus neun Werten (siehe [Abb. 4](#) „Situation“): Vier Werte beschreiben, wie viele Kreuze jeweils in den farbigen Reihen gesetzt wurden, vier Werte beschreiben, was in jeder dieser Reihen das Limit (also die am weitesten rechts angekreuzte Zahl) ist und ein Wert ist die Anzahl an Fehlwürfen.

Der Maßstab, nach welchem ein individueller KI-Spieler eine hypothetische Situation *x* bewertet, ist die Strategie. Sie ist der einzige Aspekt, indem sich KI-Spieler voneinander unterscheiden und setzt sich in der Regel aus den drei Koeffizienten-Arrays *a*, *b* und *c* zusammen. Diese Arrays haben jeweils vier Stellen, so hat bspw. der Koeffizienten-Array *a* die Form $[a_0; a_1; a_2; a_3]$.

Dabei bezieht sich a_0 auf die vier Werte der hypothetischen Situation, welche die Anzahl an Kreuzen pro Reihe ausdrücken, a_1 bezieht sich auf die Limits in der roten und gelben Reihe, a_2 bezieht sich auf die Limits in der grünen

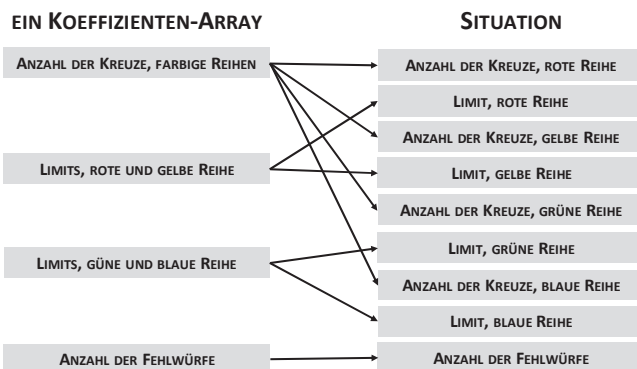


Abb. 4: Beziehung zwischen einem Koeffizienten-Array der Strategie und dem Array einer hypothetischen Situation



$$\text{Qualität} = \sum_{i=0}^8 b'_i \cdot x_i + c'_i$$

Formel (5)

$$\text{Qualität} = \sum_{i=0}^8 z'_i \cdot x_i^3 + a'_i \cdot x_i^2 + b'_i \cdot x_i + c'_i$$

Formel (6)

$$\text{Varianz} = \frac{1}{n} \sum_{i=0}^n (\text{Punkte}_i - \text{durchschnittliche Punkte})^2$$

mit $n = \text{Anzahl der Runden} \cdot \text{Gruppengröße}$

Formel (7)

und blauen Reihe und a_3 bewertet die Anzahl der Fehlwürfe (siehe [Abb. 4](#)).

Der Anschaulichkeit beim Rechnen zugunsten, kann ein Koeffizienten-Array auch auf die Länge des Arrays der hypothetischen Situation angepasst werden, sodass dieser verlängerte Koeffizienten-Array a' von a die Form $[a'_0; a'_1; a'_2; a'_3; a'_4; a'_5; a'_6; a'_7; a'_8]$ hat, was $[a_0; a_1; a_0; a_1; a_0; a_2; a_0; a_2; a_3]$ entspricht.

In der Entscheidungsfunktion wird mithilfe der Strategie und einer hypothetischen Situation nach der Formel

$$\text{Qualität} = \sum_{i=0}^8 a'_i \cdot x_i^2 + b'_i \cdot x_i + c'_i \quad (3)$$

die Qualität eines Zuges berechnet. Der höchste Exponent ist 2, da so dem KI-Spieler ermöglicht werden soll, den quadratischen Punktezuwachs zu interpretieren.

Beispiel:

In einem Beispiel hat der Array der Ausgangssituation die Form $[1; 2; 1; 3; 1; 12; 2; 10; 0]$ und es wird die hypothetische Situation $[1; 2; 2; 5; 1; 12; 2; 10; 0]$ nach Ankreuzen der 5 in der gelben Reihe betrachtet. Die Strategie sei dabei aus den ausgedachten Koeffizienten-Arrays a $[1; 0; 0; 0]$,

b $[2; 1; -1; -5]$ und c $[0; 0; -1; 0]$ zusammengesetzt. Die Qualität des Zuges wird also wie folgt berechnet:

Formel 4

Die Qualität des Zuges, die gelbe 5 anzukreuzen, würde für den KI-Spieler mit der fiktiven Strategie 6 betragen.

Um den mutmaßlich besten nächsten Zug auszuwählen, setzt der KI-Spieler nacheinander jede mögliche hypothetische Situation an Stelle x in die Entscheidungsfunktion ein. Derjenige Zug, der nach dieser Berechnung die größte Qualität zugewiesen bekommt, wird als „bester“ Zug bewertet und anschließend ausgeführt.

Die Entscheidungsfunktion kann auch genutzt werden, wenn die Strategie eines KI-Spielers aus mehr oder weniger als drei Koeffizienten-Arrays besteht. Für einen KI-Spieler, dessen Strategie nur aus zwei Koeffizienten-Arrays besteht, sähe die Formel wie folgt aus:

Formel 5

Andersherum können auch mehr Koeffizienten-Arrays hinzugefügt werden:

Formel 6

Zur Unterscheidung von KI-Spielern anhand der Anzahl ihrer Koeffizienten-Arrays werden verschiedene „Grade“ unterschieden, die durch den höchsten Exponenten in der Formel bestimmt werden. So ist also eine Strategie, die aus den drei Koeffizienten-Arrays a , b und c besteht, eine Strategie zweiten Grades, eine Strategie mit zwei Koeffizienten-Arrays ersten Grades und eine Strategie, die aus vier Koeffizienten-Arrays besteht, eine Strategie dritten Grades.

Wie sich die verschiedenen Grade von Strategien auf die Trainingsergebnisse auswirken, wird in dieser Arbeit untersucht und ausgewertet (siehe Abschnitt 5.3).

4. Ablauf des Trainings

Vor dem Training werden die Werte der notwendigen Parameter festgelegt. Solche Parameter sind z. B. absolute Zahlen (z. B. „Populationsgröße“), Anteile bzw. Raten (z. B. „Mutationsrate“) oder Wahrheitswerte (z. B. „Population laden“).

4.1 Vorbereitung

Ein Trainingszyklus beginnt in der ersten Generation mit dem Erstellen einer Ausgangspopulation. Ist der Parameter „Population laden“ „Wahr“ („True“), so wird als Ausgangspopulation eine Population geladen, die am Ende eines bereits durchgeführten Trainingszyklus gespeichert wurde. Ist dieser Parameter „Falsch“ („False“), so wird eine Population aus KI-Spielern mit zufällig erstellten Strategien erstellt. Das Laden einer Population ist dann sinnvoll, wenn die Trainingsparameter innerhalb der Evolution einer Population verändert werden sollen und bspw. die Mutationsrate nach Erreichen einer relativ hohen Durchschnittsstärke auf einen niedrigeren Wert gesetzt wird.

Der Parameter „Populationsgröße“ bestimmt dabei, wie viele KI-Spieler eine Population in jeder Generation ausmachen. Für diesen Parameter ist 100 bis 250 ein guter Mindestwert.

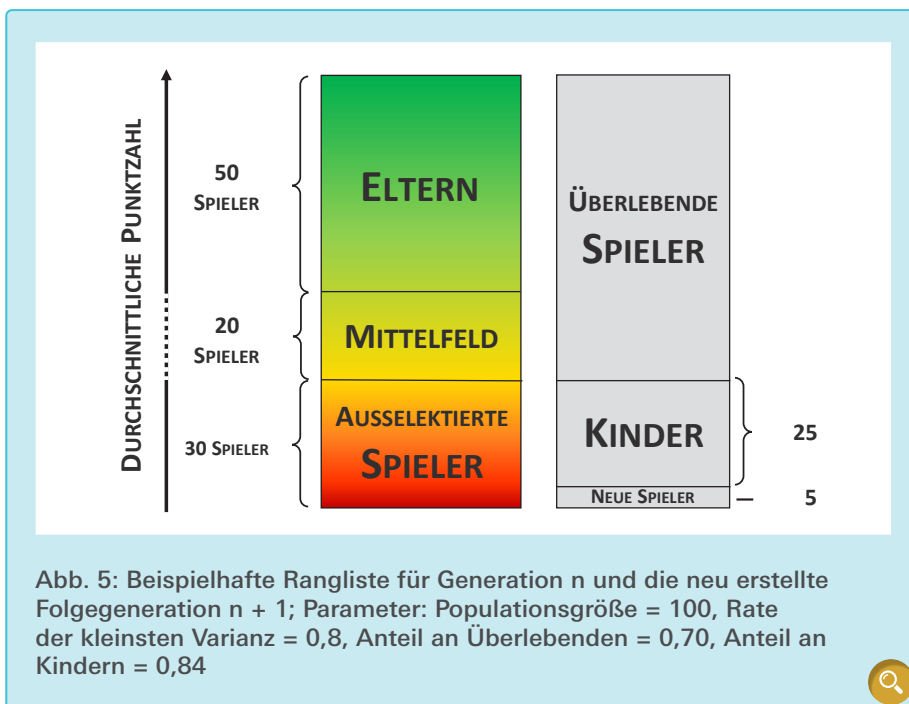


Abb. 5: Beispielfangliste für Generation n und die neu erstellte Folgegeneration n + 1; Parameter: Populationsgröße = 100, Rate der kleinsten Varianz = 0,8, Anteil an Überlebenden = 0,70, Anteil an Kindern = 0,84

„Gruppengröße“ ist der Parameter, der festlegt, wie groß die Gruppen sind, in denen gespielt wird. Im Regelwerk sind zwei bis fünf Spieler vorgesehen.

Ist der Parameter „gegen eigene Kopien spielen“ Wahr, so sind die Gegenspieler eines KI-Spielers in seiner Gruppe Kopien seiner selbst. Diese Spielweise hat zum Vorteil, dass alle Spieler einer Gruppe gleich stark sind. Ist der Parameter „gegen eigene Kopien spielen“ Falsch, so sind die Gegenspieler eines KI-Spielers in einer Gruppe echte KI-Spieler aus der Population.

Die KI-Spieler spielen in verschiedenen Gruppenkonstellationen mehrere Runden gegeneinander. Der Parameter „Anzahl der Runden“ bestimmt dabei, wie viele solcher Runden ein Spieler pro Generation spielt. Darauf basierend wird für jeden Spieler eine durchschnittlich erreichte Punktzahl berechnet. Ein guter Wert für die Anzahl der Runden ist 10, darüber hinaus ändern sich die Ranglisten (sortiert nach durchschnittlichen Punkten) nicht mehr deutlich.

4.2 Auswertung der Spiele

Nachdem alle Spielrunden stattgefunden haben, werden die KI-Spieler nach ihrer Fitness sortiert. Diese ergibt sich aus ihren durchschnittlich erreichten Punkten und der korrigierten Stichprobenvarianz [7] der erreichten Punkte in den einzelnen Spielrunden. Diese ergibt sich nach der Formel:

Formel 7

Die Variable „Rate der kleinsten Varianz“ im Trainer-Algorithmus bestimmt, bei welchem Anteil der Population die Varianz eines einzelnen KI-Spielers im Rahmenbereich liegt. Beträgt die Rate der kleinsten Varianz z. B. 0,8, so liegen die ersten 80 % der nach ihrer korrigierten Stichprobenvarianz sortierten Population im Rahmenbereich. Die korrigierte Stichprobenvarianz der restlichen 20 % wäre also zu hoch.

Der Trainer ordnet diesen restlichen Anteil, der eine höhere Varianz aufweist, ins Mittelfeld, da ein Spieler mit hoher Varianz in den Punkten nicht zuverlässig als stark oder schwach beurteilt werden kann (siehe [Abb. 5](#)). Ein guter Wert für die Rate der kleinsten Varianz ist 0,95 bis 0,98.

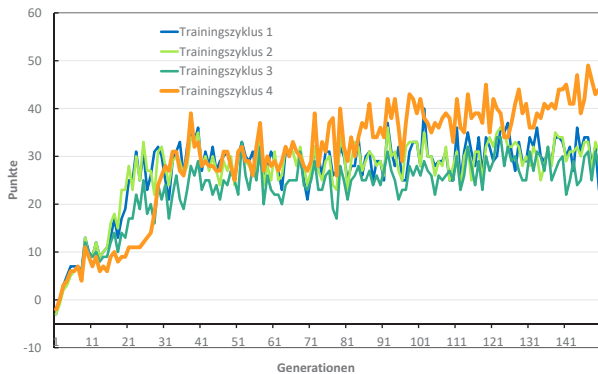
Die KI-Spieler, deren Varianz klein genug ist, werden nach ihren durchschnittlich erreichten Punkten sortiert, sodass die Individuen mit den meisten Punkten am oberen Ende der Rangliste stehen.

Der Parameter „Anteil an Überlebenden“ drückt aus, welchen Anteil die KI-Spieler in der Rangliste oberhalb des Mittelfeldes und des Mittelfeldes selber an der Population ausmachen. Beträgt der Anteil an Überlebenden bspw. 0,70 (70 %) bei einer Populationsgröße von 100, so überleben 70 KI-Spieler. Die restlichen 30 Spieler stünden in der Rangliste unterhalb des Mittelfeldes und würden ausselektiert werden (siehe [Abb. 5](#)).

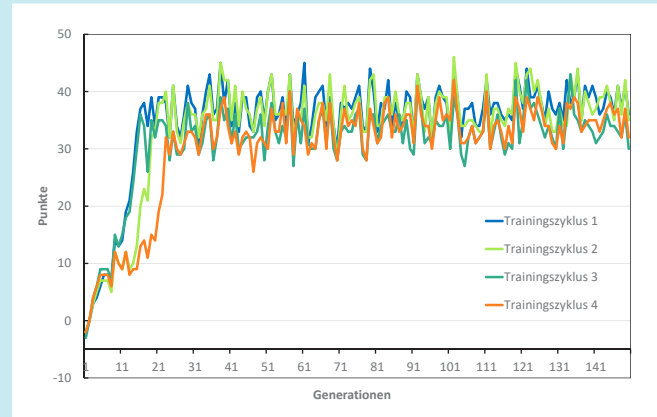
Eine mögliche, alternative Fitness-Funktion könnte die KI-Spieler anstatt nach ihren durchschnittlich erreichten Punkten und der Varianz in diesen nach der Gewinnhäufigkeit sortieren, da es neben dem Gewinn durch Erreichen hoher Punktzahlen auch eine Gewinnstrategie sein kann, das Spiel möglichst schnell mit einem ausreichend großen Punktevorsprung zu gewinnen (siehe [Abschnitt 1.2.1](#)).

4.3 Erstellen der neuen Generation

Da die Populationsgröße über alle Generationen konstant gehalten wird, werden für die nächste Generation so viele neue KI-Spieler hinzugefügt, wie zuvor ausselektiert wurden. Der Parameter „Anteil an Kindern“ drückt aus, wie groß der Anteil an Kindern innerhalb dieser Gruppe an neuen KI-Spielern ist. Die „Kinder“ entstehen dabei durch Rekombination der Strategien von jeweils zwei KI-Spielern am oberen Ende der Rangliste. So entsteht z. B. die Strategie des zweiten Kindes, das erstellt wird, indem jeweils aus den Koeffizienten-Arrays der KI-Spieler an dritter und vierter Stelle der Rangliste der Durchschnitt gebildet wird. Dabei nimmt das Mittelfeld nicht an der Rekombination teil.



a)



b)

Abb. 6: Entwicklung der durchschnittlichen Punktzahlen in vier Trainingszyklen unter gleichen Bedingungen: Populationsgröße=100, Gruppengröße=5, Anzahl an Generationen=150, Rate der kleinsten Varianz=0,98, Anteil an Überlebenden=0,74, Anteil an Kindern=1, wobei a) eine Mutationsrate von 0,05 aufweist und in b) keine Mutation stattfindet.

Ist der Anteil an Kindern kleiner 1, so wird der restliche Teil der fehlenden KI-Spieler mit neu erstellten KI-Spielern mit zufälligen Strategien aufgefüllt.

Anschließend kann es bei jedem KI-Spieler zu Mutationen kommen, wenn der Parameter „Mutationsrate“ größer 0 ist. Beim Mutieren einer Strategie betrachtet der Trainer jede Stelle der Koeffizienten-Arrays der Strategie. Die Mutationsrate ist die Wahrscheinlichkeit, dass der Trainer die Strategie an dieser Stelle mutiert, das heißt, die Zahl dort verändert.

Eine kleinere Mutationsrate führt direkt proportional zu weniger Mutationen.

Das Wirken von Mutation ist dann sinnvoll, wenn im Genpool durch wiederholte Rekombination verwandter KI-Spieler kaum noch Variation vorhanden ist, sodass durch die Evolution kaum neue Strategien mehr entstehen. Da durch Kreuzung von Strategien viel gezielter neue Strategien geschaffen werden als durch rein zufällige Mutationen, sollte die Mutationsrate sehr klein gehalten werden. So sollte eine Mutationsrate von ca. 0,05 (5 %) nicht überschritten werden, wie Experimente zeigen, da ansonsten die durchschnittlich erreichten Punkte der Populati-

on deutlich abnehmen. Grund hierfür ist, dass auch die Strategien der besten Spieler durch Mutation verändert werden, wodurch auch durch Rekombination kaum noch gute neue Strategien zustande kommen.

5. Evaluation und Ergebnisse

5.1 Reproduzierbarkeit der Trainingsergebnisse

Um zu zeigen, dass der evolutionäre Algorithmus des Trainers tatsächlich zuverlässig funktioniert, wird der Trainer mehrmals mit denselben Parametereinstellungen ausgeführt. Die Bedingungen wie etwa Gruppengröße, Anteil an Überlebenden, Rate der kleinsten Varianz und ähnliches sind bei jedem Trainingszyklus gleich. Die Entwicklung der durchschnittlichen Punkte einer Population, die der Trainer findet, sollten ähnlich sein. Da der Zufall auch oft eine Rolle spielt, wird kein Anspruch auf identische Ergebnisse erhoben.

Die jeweils höchsten Punkte werden hier nicht berücksichtigt, da die Ergebnisse einer einzelnen KI viel stärker vom Zufall abhängig sind als die Ergebnisse der gesamten Population und somit der Verlauf der maximalen Punkte eines Evolutionszyklus nicht aussagekräftig genug wäre.

In [Abb. 6](#) sind die durchschnittlich erreichten Punktzahlen über 150 Generationen von jeweils vier Trainingszyklen mit gleichen Parametereinstellungen dargestellt. Dabei beträgt in [Abb. 6a](#) die Mutationsrate 0,05, in [Abb. 6b](#) findet keine Mutation statt. Die Rate der kleinsten Varianz hat den Wert 0,98, sodass die 2 % der Population mit der höchsten Varianz ins Mittelfeld geordnet werden. Von der Population werden in jeder Generation 26 KI-Spieler ausselektiert und durch Kinder ersetzt.

In den Trainingszyklen 1 bis 3 mit Mutation (siehe [Abb. 6a](#)) verlaufen die Kurven ähnlich. Der Trainingszyklus 4 unterscheidet sich allerdings vom Rest. Am Anfang steigen die Punktzahlen langsamer an und ca. ab Generation 35 liegen die Punktzahlen überdurchschnittlich hoch. Im Gegensatz zu den Trainingszyklen 1 bis 3 stagnieren die Punktzahlen nicht, denn ca. ab Generation 81 lässt sich ein aufsteigender Trend im Durchlauf 4 erkennen. Diese Abweichungen sind sehr wahrscheinlich an Mutationen – eine gewollte Zufallsgröße – gekoppelt. Durch Glück sind so im vierten Trainingszyklus relativ „gute“ Strategien entstanden.

[Abb. 6b](#) zeigt wieder vier Trainingszyklen unter gleichen Versuchsbedingungen, allerdings wurde die Mutationsrate

auf 0 gesetzt, sodass weniger Zufall die Evolution beeinflusst. In [Abb. 6b](#) verlaufen die Kurven anfangs wieder etwas unterschiedlich. Dieser Unterschied ist damit zu erklären, dass der Trainer die Ausgangspopulation in allen vier Fällen zufällig erstellt hat. Ungefähr ab Generation 35 zeichnen sich keine wesentlichen Veränderungen mehr ab. In allen vier Fällen verlaufen die Kurven ähnlich.

Demnach liefert der Trainer-Algorithmus reproduzierbare Ergebnisse. Nur Mutation kann den Verlauf der durchschnittlichen Punktzahlen signifikant verändern.

5.2 Gruppengröße

Für das Spiel Qwixx sind nach dem Regelwerk zwei bis fünf Spieler vorgesehen. In [Abb. 7](#) ist die Punkteentwicklung von zwei Trainingszyklen über 100 Generationen zu sehen. Die Rate der kleinsten Varianz beträgt 0,98, der Anteil an Überlebenden 0,74 und der Anteil an Kindern unter den neuen Spielern beträgt 1. Mutationen treten nicht auf. Die Trainingszyklen unterscheiden sich in ihren Bedingungen nur in der Gruppengröße.

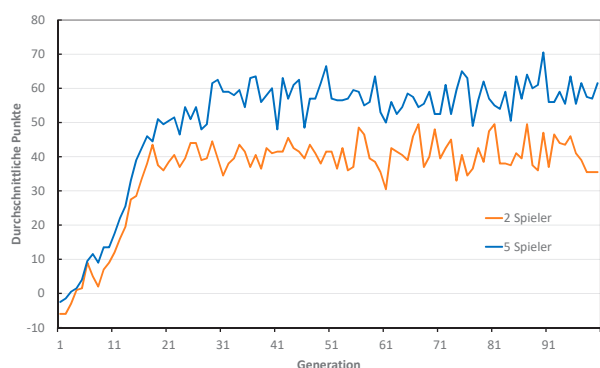


Abb. 7: Beispielhafte Entwicklung der Punkte in Trainingszyklen mit verschiedenen Gruppengrößen, Populationsgröße=100, Anzahl an Generationen=100, Rate der kleinsten Varianz=0,98, Anteil an Überlebenden=0,74, Anteil an Kindern=1, Mutationsrate=0.



Es ist zu erkennen, dass die durchschnittlichen Punktzahlen bei zwei Spielern unterhalb der durchschnittlichen Punktzahl bei fünf Spielern liegen.

Die Beobachtung, dass bei einer größeren Gruppengröße bessere Ergebnisse erzielt werden, hat zwei Gründe. Zum einen sind Spieler in kleineren Gruppen öfter aktiver Spieler, also dazu gezwungen, ein Kreuz zu setzen, auch wenn dadurch auf dem Spielzettel große Lücken entstehen. Zum anderen haben bei größeren Gruppengrößen die Spielerstärken einen geringeren Einfluss auf die Spieldauer, sodass ein Spieler, der bspw. immer Fehlwürfe ankreuzt, länger braucht, um das Spiel zu beenden, weil er seltener aktiver Spieler ist.

5.3 Vergleich von Strategien

5.3.1 KI-Spieler mit Strategien ersten und zweiten Grades

Der Trainings-Algorithmus wird darauf getestet, ob eine Strategie „ersten Grades“ oder „zweiten Grades“ zu besseren Ergebnissen führt. Der Vergleich (siehe [Abb. 8](#)) zeigt, dass die KI-Spieler mit einer Strategie zweiten Grades tendenziell schlechter abschneiden als sol-

che KI-Spieler mit einer Strategie ersten Grades.

Die Punktzahlen von KI-Spielern mit einer Strategie ersten Grades steigen nicht nur in den ersten Generationen schneller an, sondern bleiben auch auf einem höheren Niveau als die durchschnittlichen Punkte von KI-Spielern mit einer Strategie zweiten Grades. Nichtsdestotrotz kann auch beobachtet werden, dass ca. ab Generation 181 die grüne und die orangene Kurve sehr dicht beieinander liegen.

Wenn die Spielzettel verschiedener KI-Spieler beobachtet werden, ist ersichtlich, dass es den KI-Spielern, die eine Strategie ersten Grades haben, vorrangig darum geht, die Lücken so klein wie möglich zu halten. KI-Spieler mit einer Strategie zweiten Grades lassen hingegen auch größere Lücken zu. Durch kleine Lücken können mehr Felder angekreuzt werden, was zu einer höheren Punktzahl führt. Auf diese Weise erzielt ein KI-Spieler mit einer Strategie ersten Grades auch hohe Punktzahlen, ohne in der Lage sein zu müssen, den quadratischen Punktezuwachs interpretieren zu können.

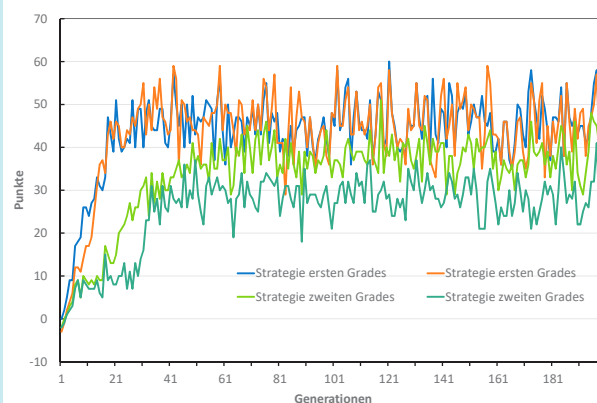


Abb. 8: Durchschnittliche Punkte von zwei Trainingszyklen mit einer Strategie ersten Grades und zwei Trainingszyklen mit einer Strategie zweiten Grades. (Parametereinstellungen: Populationsgröße=100, Gruppengröße=5, Anzahl an Generationen=200, Mutationsrate=0,05, Rate der kleinsten Varianz=0,98, Anteil an Überlebenden=0,74, Anteil an Kindern=1)



Darüber hinaus ist eine Strategie zweiten Grades fünfzig Prozent länger als eine Strategie ersten Grades. Da mehr Werte optimiert werden müssen, bedeutet der Besitz eines Koeffizienten-Arrays a , dass die Evolution länger dauert, um eine ähnlich starke Strategie zu entwickeln. Dies erklärt auch den schnelleren Anstieg der Punkte von KI-Spielern ohne quadratischen Koeffizienten zu Beginn der Trainingszyklen und die Annäherung der Datenreihen gegen Ende.

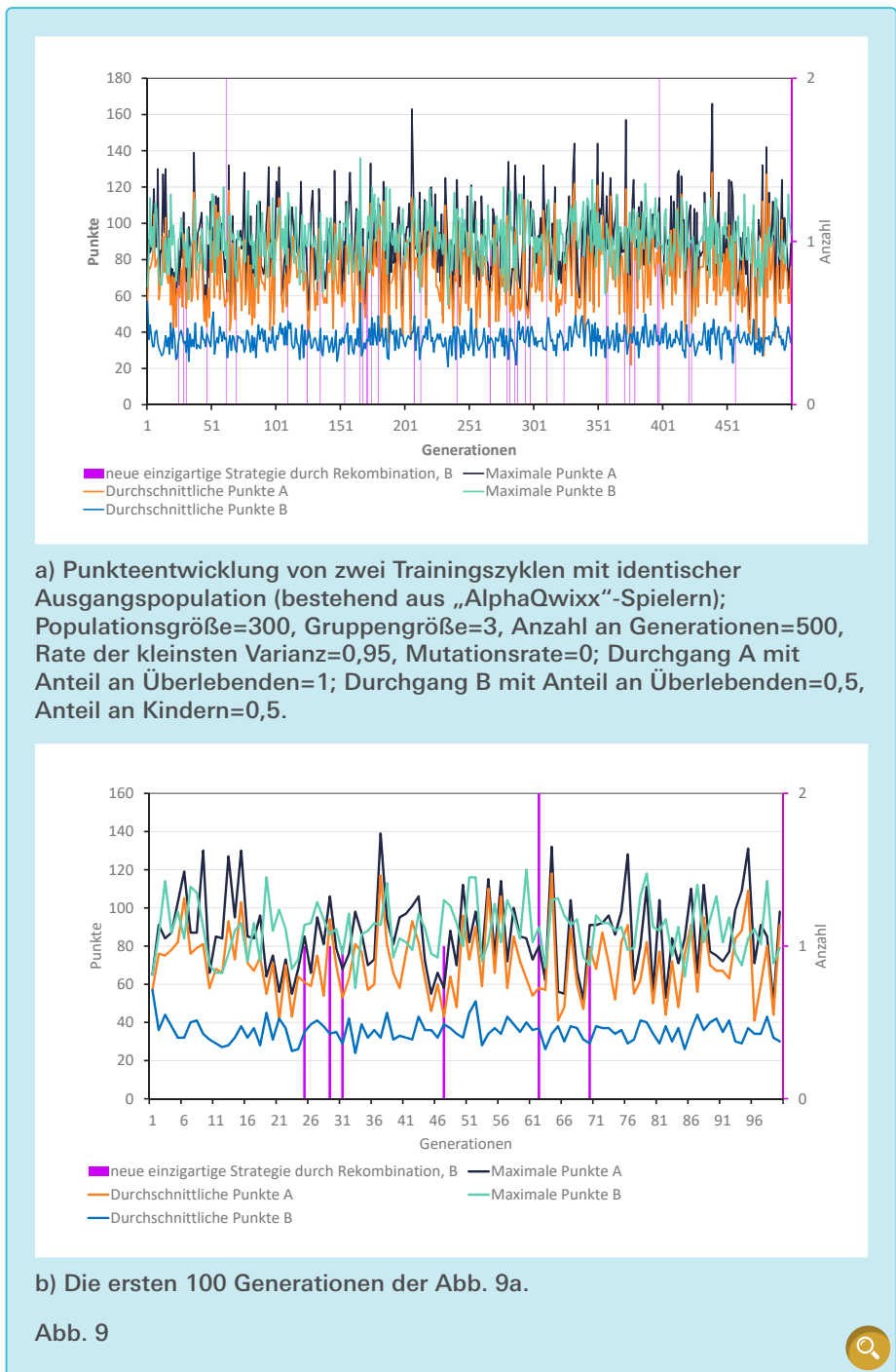
5.3.2 KI-Spieler mit Strategie dritten Grades

Je höher der Grad der Strategie zur Bewertung einer Situation ist, desto differenzierter kann diese Bewertung erfolgen. Es lässt sich allerdings beobachten, dass zufällige KI-Spieler mit einer Strategie dritten Grades in der Evolution schlechtere Punktzahlen erreichen als Spieler mit einer Strategie zweiten Grades und die Punktzahlen schnell stagnieren.

5.4 Zusammenhang zwischen Ergebnis und Stärke

Erreichte Punktzahlen hängen nicht nur von einem Spieler an sich ab, sondern auch von seinen Mitspielern. Ein Beispiel hierfür wäre, wenn ein Spieler nur Fehlwürfe ankreuzt und so das Spiel zu einem schnellen Ende bringt. Das Spielen in verschiedenen Gruppenkonstellationen zur Bewertung der Spielerstärke sollte deshalb repräsentative Ranglisten hervorbringen. Die KI-Spieler, deren Strategien vom Trainer auf diese Weise als „am stärksten“ bewertet werden, sollten auch in weiteren Spielen gewinnen.

In einem Experiment soll untersucht werden, ob KI-Spieler, die sehr hohe Punktzahlen erreichen und somit eine Strategie der Maximierung der Punktzahl verfolgen, gegen KI-Spieler standhalten können, die eher eine Strategie der Maximierung der Gewinnhäufigkeit anstreben und die Spielrunden sehr schnell beenden.



Dazu werden die Ergebnisse zweier Trainingszyklen (A und B) über jeweils 500 Generationen betrachtet. In beiden Trainingszyklen besteht die Ausgangspopulation aus 300 Kopien des bisher besten KI-Spielers, der hier als „AlphaQwixx“ bezeichnet wird. In Durchgang A sind alle Parameter, die die Strategien verändern, deaktiviert: Die Rate an Überlebenden beträgt 1 und die Mutationsrate 0. In Durchlauf B beträgt der Anteil an Überlebenden und der Anteil an Kindern jeweils 0,5. Auch hier

beträgt die Mutationsrate 0, damit die AlphaQwixx-Spieler nicht durch Zufall schlechter werden. In beiden Durchläufen beträgt außerdem die Gruppengröße 3 und die Rate der kleinsten Varianz hat den Wert 0,95.

In Durchlauf A (siehe [Abb. 9a](#)) schwanken die maximal erreichten Punkte der einzelnen KI-Spieler zwischen 29 und 166 und betragen durchschnittlich 88,5 Punkte. Die durchschnittlich erreichten Punkte einer Generation desselben Durch-

laufs betragen zwischen 22 und 127 und ergeben im Durchschnitt 72,4 Punkte.

In Durchlauf B schwanken die maximal erreichten Punkte der einzelnen KI-Spieler zwischen 55 und 136 und betragen durchschnittlich 91,0 Punkte. Die durchschnittlich erreichten Punkte einer Generation im Durchlauf B liegen zwischen 21 und 56 mit einem Durchschnitt von 36,1 Punkten.

Die durchschnittlichen Punkte, die in den Trainingszyklen A und B erzielt wurden, weichen deutlich voneinander ab. Einerseits sind die durchschnittlichen Punkte im Durchlauf A im Schnitt um 36,3 Punkte höher als in Durchlauf B, da in Durchlauf A die Population nur aus AlphaQwixx-Spielern besteht, wohingegen in Durchlauf B in fast jeder Population 75 KI-Spieler sind, deren Strategien zufällig erstellt sind. Diese KI-Spieler erreichen meist sogar negative Punktzahlen (bis zu -20 Punkte). Andererseits ist auch die Amplitude der durchschnittlich erreichten Punkte im Trainingszyklus B geringer als im Trainingszyklus A, da die Diversität unter den Spielern höher ist und somit die durchschnittlichen Punkte einer Population weniger vom Würfelglück beeinflusst werden. Im Durchlauf A hingegen lässt sich der hohe Einfluss des Würfelglücks sehr deutlich erkennen, da die Entwicklung der Maximal- und Durchschnittspunkte weitgehend parallel verläuft (siehe [Abb. 9b](#)).

Die maximal erreichten Punkte in Durchlauf B liegen durchschnittlich 2,5 Punkte über denen von Durchlauf A und die Amplitude der maximalen Punkte in Durchlauf B ist kleiner als die des Durchlaufs A. Diese Beobachtung scheint zunächst widersinnig zu sein, da der wesentliche Unterschied zwischen den Trainingszyklen A und B ist, dass in B scheinbar „dumme“ KI-Spieler hinzukommen, deren zufällig erstellte Strategien bewirken, dass die Spielrunden schnell beendet werden. Spielen AlphaQwixx-Spieler gegen solche „dummen“ KI-Spieler, so erzielen sie deshalb

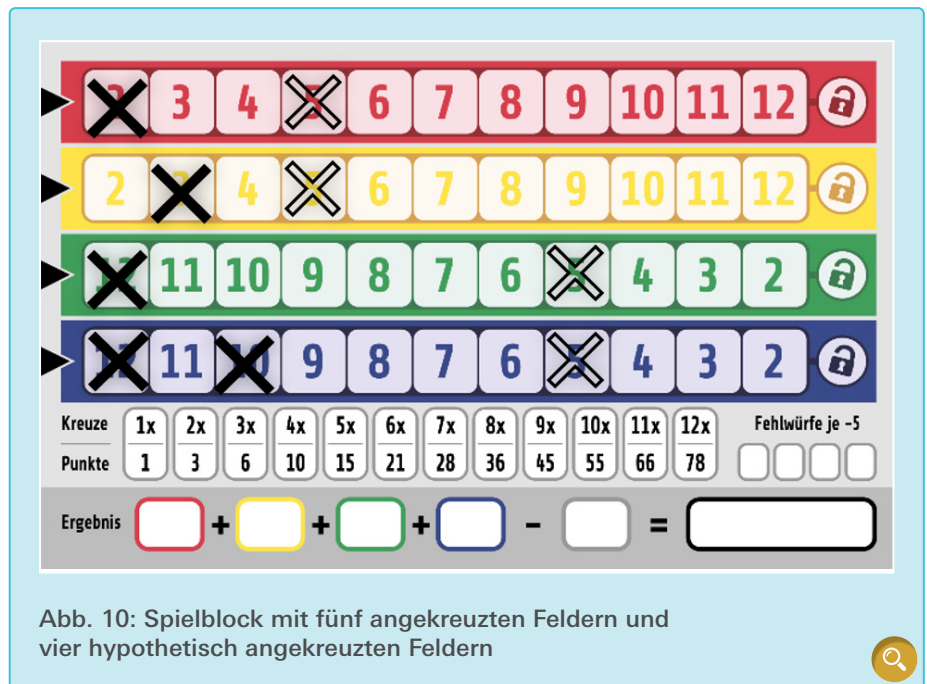


Abb. 10: Spielblock mit fünf angekreuzten Feldern und vier hypothetisch angekreuzten Feldern

weniger Punkte als im Spiel gegen andere AlphaQwixx-Spieler.

Die Beobachtung zeigt allerdings das Gegenteil: Die KI-Spieler mit ihren zufälligen Strategien bewirken teilweise, dass höhere maximale Punktzahlen erreicht werden. Dies ist nur möglich, wenn ein zufällig erstellter KI-Spieler hin und wieder eine relativ gute Strategie hat, d. h. der zufällig erstellte KI-Spieler selbst diese hohe Punktzahl (in Verbindung mit Würfelglück) erreicht. Da die besten 150 KI-Spieler in jeder Generation auch Eltern werden, wird untersucht, wann durch Rekombinationen völlig neue Strategien entstehen, die es so in der Population noch nicht gibt (siehe [Abb. 9a/b](#): „neue einzigartige Strategie durch Rekombination, B“). Sie sind ein Indikator dafür, dass zufällig erstellte KI-Spieler gute Punktzahlen erzielen. Liegen die maximalen Punkte von Durchlauf B nach diesen Ereignissen über denen von Durchlauf A, so sind die durch Rekombination entstandenen KI-Spieler in ihren Strategien wahrscheinlich auch stark und erreichen so in diesen Runden mehr Punkte als AlphaQwixx-Spieler (siehe [Abb. 9b](#), z. B. Generation 46 bis 52). Da immer noch ein großer Anteil der Population AlphaQwixx-Spieler sind, passiert es selten, dass die maximalen Punkte

vom Trainingszyklus B unter die vom Trainingszyklus A fallen, da AlphaQwixx-Spieler unter gleichen Bedingungen gleiche Entscheidungen treffen und dadurch mindestens die Punkte von Durchlauf A erreichen (es sei denn, die Spielrunden werden wegen „dummen“ Mitspielern in ihrer Dauer verkürzt).

Nichtsdestotrotz sind die durch Rekombination entstandenen KI-Spieler langfristig nicht in der Lage, ihre Spitzenposition zu halten. Der beste KI-Spieler am Ende der 500 Generationen hat immer noch die AlphaQwixx-Strategie.

Der AlphaQwixx-Spieler ist also sowohl gegen KI-Spieler, die Spielrunden schnell beenden als auch gegen KI-Spieler, die mit Würfelglück mehr Punkte erreichen können, durchsetzungsfähig.

5.5 Funktionsweise der „AlphaQwixx“-Strategie

Die Strategie des bisher besten KI-Spielers („AlphaQwixx“) ist eine Strategie ersten Grades, das heißt, sie besitzt nur zwei Koeffizienten-Arrays. Vereinfacht, also auf weniger Dezimalstellen reduziert, sieht sie wie folgt aus:

$$b : [1,5 ; -0,75 ; 0,75 ; -3,75]$$

$$c : [0,0 ; 0,0 ; -9,0 ; 0,0] \quad (8)$$

Die Strategie wird in einer fiktiven Situation betrachtet. Die Ausgangssituation sieht so aus, dass bereits fünf Kreuze gesetzt sind: Das sind die schwarzen Kreuze in [Abb. 10](#).

Diese Situation hat die Form $[1 ; 2 ; 1 ; 3 ; 1 ; 12 ; 2 ; 10 ; 0]$. Im nächsten Zug soll eine 5 angekreuzt werden.

Ein möglicher nächster Zug, also eine hypothetische Situation, kann z.B. so aussehen, dass die 5 in der gelben Reihe angekreuzt werden soll. Diese hypothetische Situation hat die Form $[1 ; 2 ; 2 ; 5 ; 1 ; 12 ; 2 ; 10 ; 0]$. Auf Grundlage der Entscheidungsfunktion (siehe Abschnitt 3.2), ergibt sich eine Qualität von 2,3 für die Wahl, Gelb anzukreuzen.

Analog wird die Entscheidung, Rot anzukreuzen mit 1,5 bewertet, die Wahl, Grün anzukreuzen erhält eine Bewertung von -1,5 und Blau hat eine Qualität von 0.

Da die Bewertung des Zuges, der Gelb ankreuzt, mit 2,3 am höchsten ist, entscheidet sich AlphaQwixx für diesen Zug. Logisch betrachtet ist dies auch die beste Wahl.

Hätte AlphaQwixx sich für das Kreuz in der roten Reihe entschieden, hätte es zwei Felder überspringen müssen, die er für den Rest des Spiels nicht mehr ankreuzen darf. In der gelben Reihe überspringt er nur ein Kreuz und verbaut sich somit mit diesem Zug weniger Möglichkeiten, als wenn er das rote Kreuz setzen würde. Der hypothetische Zug in der grünen Reihe wird am schlechtesten bewertet, da hier eine Lücke von sechs Feldern entstehen würde. Auch die hypothetische Situation in der blauen Reihe würde mit vier übersprungenen Feldern eine große Lücke verursachen, da sie kleiner ist als in der grünen Reihe, ist die Bewertung allerdings besser.

5.6 Vergleich der künstlichen Intelligenz mit menschlichen Spielern

Ziel dieses Projektes ist das Finden einer starken Strategie für Qwixx. Um zu zeigen, dass diese Strategie tatsächlich gut ist, sollte ein KI-Spieler gegen einen menschlichen Spieler gewinnen können.

Aus den Daten bisheriger Spiele, die ich mit meiner Familie und Freunden gespielt habe, hat sich eine Durchschnittspunktzahl von ca. 83,5 Punkten herausgestellt. Das absolute Maximum liegt dabei bei 139 Punkten und das absolute Minimum bei 10 Punkten. Über 50 Prozent der erreichten Punkte lagen im Bereich von 67 bis 105 Punkten.

Daraus ergibt sich, dass eine starke Strategie zu mindestens 140 erreichten Punkten führen sollte.

Nach bisherigem Forschungsstand kann nur die AlphaQwixx-KI, die bis zu 166 Punkte erreicht hat, Punktzahlen über 139 erreichen (siehe [Abb. 9a](#)). Die maximale Punktzahl, die theoretisch erreicht werden kann, sind 266 Punkte. Diese Punktzahl ist allerdings mit einem extrem unwahrscheinlichen Würfelglück verbunden und deshalb ist

es unrealistisch, dass ein KI-Spieler eine derart hohe Punktzahl im Spiel gegen einen Menschen erreicht.

Im direkten Spiel der AlphaQwixx-KI gegen einen Menschen (mich), hat die AlphaQwixx-KI bei 26 von 50 Spielen gewonnen (siehe [Abb. 11](#)). Somit spielt die AlphaQwixx-KI mindestens auf dem Niveau guter menschlicher Spieler.

6. Zusammenfassung und Ausblick

Ziel des Projekts war es, mithilfe von maschinellem Lernen eine KI zu entwickeln, die eine starke Strategie für Qwixx findet. Es sollten auch verschiedene Trainingsmethoden (Parametereinstellungen) in Hinblick auf die Strategiestärken und die Reproduzierbarkeit der Strategien verglichen werden.

Mit der Strategie des gefundenen „AlphaQwixx-Spielers“ stellt die Arbeit eine KI vor, die in der Lage ist, im Maximum mehr Punkte zu erreichen als menschliche Spieler und die auch gegen einen guten menschlichen Gegenspieler gewinnen kann. Auch im Spiel gegen andere KI-Spieler, die etwa das Spiel sehr schnell mit wenigen Punkten beenden oder teilweise sogar mehr

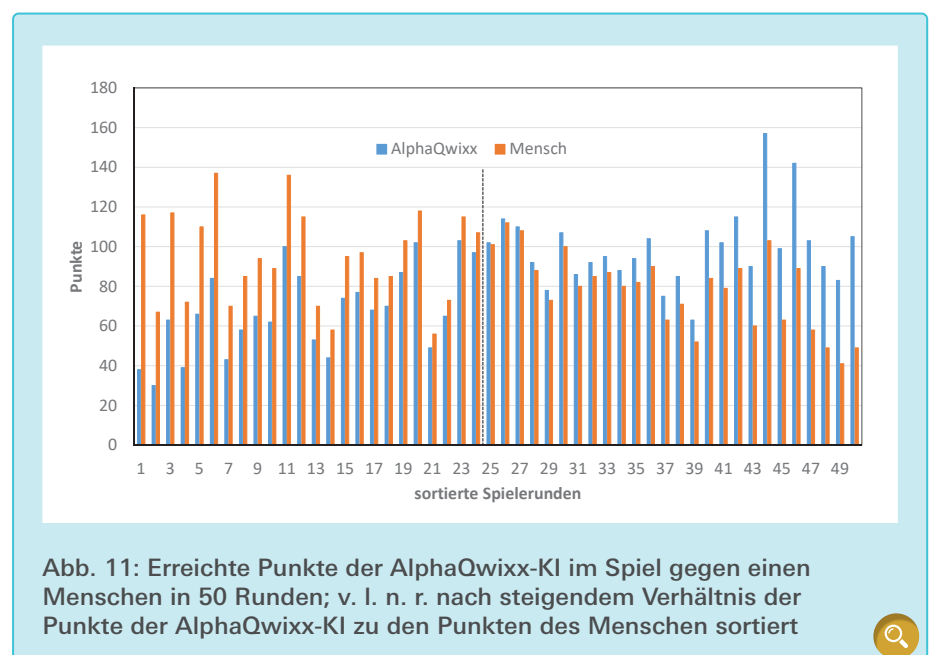


Abb. 11: Erreichte Punkte der AlphaQwixx-KI im Spiel gegen einen Menschen in 50 Runden; v. l. n. r. nach steigendem Verhältnis der Punkte der AlphaQwixx-KI zu den Punkten des Menschen sortiert

Punkte erreichen können, setzt sich die AlphaQwixx-KI evolutionär durch.

Aus der Arbeit geht auch hervor, dass bei größeren Gruppen die Punktzahlen tendenziell höher sind als in kleineren Gruppen. Eine rein elitäre Evolution ist nicht zielführend, und eine Strategie zweiten Grades ist effektiver als eine Strategie dritten Grades.

Außerdem geht aus der Evaluation hervor, dass der evolutionäre Algorithmus, der im Trainer Anwendung findet, verlässlich funktioniert und die dort erzielten Ergebnisse reproduzierbar sind.

Es bleibt zukünftig interessant, nach weiteren sehr starken Qwixx-KI mit dem Trainings-Algorithmus zu suchen. Dazu kann der evolutionäre Algorithmus weiter optimiert werden, indem z. B. alternative Herangehensweisen an die Fitness-Funktion getestet werden. Außerdem können z. B. auch Hyperparameter angelegt werden, also Parameter, die die Parameter des Trainings-Algorithmus optimieren. Es wäre auch sinnvoll, Experimente wiederholt durchzuführen, bei denen Zufall häufiger auftritt (hauptsächlich aufgrund einer relativ hohen Mutationsrate). Damit würde eine größere Datenbasis geschaffen werden, um neben den bereits getroffenen Vorkehrungen – wie den Seeds in den Würfelisten oder dem Erstellen von Ranglisten basierend auf den durchschnittlichen Punkten, die die KI-Spieler in mehreren Spielen erzielt haben – die Aussagekraft und Verlässlichkeit der Ergebnisse weiter zu erhöhen.

Auch wäre es spannend, einen KI-Spieler zu entwickeln, der seine Strategie in Abhängigkeit vom Spielverhalten der Mitspieler anpassen kann, um etwa sehr schnell Punkte zu maximieren oder um sich auf ein langes Spiel einzustellen.

Danksagung

Prof. Dr. -Ing. Joachim Denzler, Leiter des Lehrstuhls für Digitale Bildverarbeitung an der Friedrich-Schil-

ler-Universität Jena, ermöglichte mir im Sommer 2019, dort ein zweiwöchiges Praktikum zu absolvieren. Er unterstützte mich stets wohlwollend und konstruktiv mit zielführenden Kritiken und Anregungen. In Jena betreute mich auch M. Sc. Christian Reimers, der mich in die Grundlagen des maschinellen Lernens einführte und dabei half, meine Programmierkenntnisse zu erweitern. An meiner Schule wurde mein Praktikum (und eine daraus resultierende Besondere Lernleistung) von meinem Fachlehrer Thomas Dahnke betreut, den ich in regelmäßigen Konsultationen über den Stand meines Projektes informierte. Mein Bruder, Jonathan Janetzki, brachte mir vor Beginn meines Praktikums die Grundlagen des Programmierens mit Python bei. Allen Unterstützern danke ich recht herzlich.

Literaturverzeichnis

- [1] P. Stone, R. Brooks, E. Brynjolfsson, et al., Artificial Intelligence and Life in 2030 - One Hundred Year Study on Artificial Intelligence: Report of the 2015-2016 Study Panel, https://ai100.stanford.edu/sites/g/files/sbiybj9861/f/ai_100_report_0831fml.pdf, besucht am 27. 08. 2021.
- [2] K. Lincoln, The Ringer, Deep You, <https://www.theringer.com/tech/2018/11/8/18069092/chess-alphazero-alphago-go-stockfish-artificial-intelligence-future>, besucht am 27. 12. 2020.
- [3] BBC News, Google achieves AI 'breakthrough' by beating Go champion, <https://www.bbc.com/news/technology-35420579>, besucht am 27. 08. 2021.
- [4] S. Brenndorf, Qwixx, <https://www.brettspiele-report.de/images/q/qwixx/Spielanleitung-Qwixx.pdf>, besucht am 22. 07. 2019.
- [5] J. Fisher, Genetic Algorithms - Jeremy Fisher, <https://www.youtube.com/watch?v=7J-Df5S2bnI>, besucht am 07. 10. 2019.
- [6] J. Deepak, random.seed(), <https://www.geeksforgeeks.org/random-seed-in-python/>, besucht am 24. 08. 2021.
- [7] W. Gellert, H. Kästner, Dr. S. Neuber, Lexikon der Mathematik (S. 529), 3. Auflage, VEB Bibliographisches Institut Leipzig, Leipzig, 1981
- [8] O. Freudenreich, Qwixx-Spielblock und Bildlizenz, https://commons.wikimedia.org/wiki/File:Qwixx_scorecard_nofonts.svg#mw-jump-to-license, besucht am 22. 07. 2019.
- [9] About - wiki, PyGame, <https://www.pygame.org/wiki/about>, besucht am 14. 02. 2020.
- [10] Fullstack Academy, Genetic Algorithm Tutorial - How to Code a Genetic Algorithm, <https://www.youtube.com/watch?v=XP8R0yzAbdo>, besucht am 07. 10. 2019.
- [11] F. Shaikh, Simple Beginner's guide to Reinforcement Learning & its implementation, <https://www.analyticsvidhya.com/blog/2017/01/introduction-to-reinforcement-learning-implementation/>, besucht am 23. 07. 2019.
- [12] A. Thomas, Reinforcement learning tutorial using Python and Keras, <https://adventure-sinmachinelearning.com/reinforcement-learning-tutorial-python-keras/>, besucht am 05. 10. 2019.
- [13] SAP, SAP Insights, What is Machine learning, <https://www.sap.com/uk/insights/what-is-machine-learning.html>, besucht am 04. 01. 2021.
- [14] C. Brucato, The Travelling Salesman Problem, <https://www.mathematics.pitt.edu/sites/default/files/TSP.pdf>, besucht am 27. 08. 2021.
- [15] H. Bansal, Becoming Human, Best languages for Machine Learning Medium, <https://becominghuman.ai/best-languages-for-machine-learning-in-2020-6034732dd242>, besucht am 04. 01. 2021.
- [16] P. E. Brunskill, Stanford CS234: Reinforcement Learning | Winter 2019 | Lecture 1 - Introduction, <https://www.youtube.com/watch?v=FgzM3zpZ55o&t=3s>, besucht am 05. 10. 2019.
- [17] S. Kansal und B. Martin, Reinforcement Q-Learning from Scratch in Python with OpenAI Gym, <https://www.learndatasci.com/tutorials/reinforcement-q-learning-scratch-python-openai-gym/>, besucht am 23. 07. 2019.
- [18] T. Steinfeld, Monte-Carlo-Methode, <https://mathepedia.de/Monte-Carlo-Methode.html>, besucht am 14. 02. 2020.
- [19] E. Stoltz, Evolution of a salesman: A complete genetic algorithm tutorial for Python, <https://towardsdatascience.com/evolution-of-a-salesman-a-complete-genetic-algorithm-tutorial-for-python-6fe5d2b3ca35>, besucht am 07. 10. 2019.
- [20] G. Grosche, V. Ziegler, D. Ziegler, Taschenbuch der Mathematik (S. 680), 23. Auflage, BSB B. G. Teubner Verlagsgesellschaft, Leipzig, und Verlag Nauka, Moskau, 1987



Publiziere auch Du hier!

Forschungsarbeiten von
Schüler/Inne/n und Student/Inn/en

In der Jungen Wissenschaft werden Forschungsarbeiten von SchülerInnen, die selbstständig, z. B. in einer Schule oder einem Schülerforschungszentrum, durchgeführt wurden, veröffentlicht. Die Arbeiten können auf Deutsch oder Englisch geschrieben sein.

Wer kann einreichen?

SchülerInnen, AbiturientInnen und Studierende ohne Abschluss, die nicht älter als 23 Jahre sind.

Was musst Du beim Einreichen beachten?

Lies die [Richtlinien für Beiträge](#). Sie enthalten Hinweise, wie Deine Arbeit aufgebaut sein soll, wie lang sie sein darf, wie die Bilder einzureichen sind und welche weiteren Informationen wir benötigen. Solltest Du Fragen haben, dann wende Dich gern schon vor dem Einreichen an die Chefredakteurin Sabine Walter.

Lade die [Erstveröffentlichungserklärung](#) herunter, drucke und fülle sie aus und unterschreibe sie.

Dann sende Deine Arbeit und die Erstveröffentlichungserklärung per Post an:

Chefredaktion Junge Wissenschaft

Dr.-Ing. Sabine Walter
Paul-Ducros-Straße 7
30952 Ronnenberg
Tel: 05109 / 561508
Mail: sabine.walter@verlag-jungewissenschaft.de

Wie geht es nach dem Einreichen weiter?

Die Chefredakteurin sucht einen geeigneten Fachgutachter, der die inhaltliche Richtigkeit der eingereichten Arbeit überprüft und eine Empfehlung ausspricht, ob sie veröffentlicht werden kann (Peer-Review-Verfahren). Das Gutachten wird den Euch, den AutorInnen zugeschickt und Du erhältst gegebenenfalls die Möglichkeit, Hinweise des Fachgutachters einzuarbeiten.

Die Erfahrung zeigt, dass Arbeiten, die z. B. im Rahmen eines Wettbewerbs wie **Jugend forscht** die Endrunde erreicht haben, die besten Chancen haben, dieses Peer-Review-Verfahren zu bestehen.

Schließlich kommt die Arbeit in die Redaktion, wird für das Layout vorbereitet und als Open-Access-Beitrag veröffentlicht.

Was ist Dein Benefit?

Deine Forschungsarbeit ist nun in einer Gutachterzeitschrift (Peer-Review-Journal) veröffentlicht worden, d. h. Du kannst die Veröffentlichung in Deine wissenschaftliche Literaturliste aufnehmen. Deine Arbeit erhält als Open-Access-Veröffentlichung einen DOI (Data Object Identifier) und kann von entsprechenden Suchmaschinen (z. B. BASE) gefunden werden.

Die Junge Wissenschaft wird zusätzlich in wissenschaftlichen Datenbanken gelistet, d. h. Deine Arbeit kann von Experten gefunden und sogar zitiert werden. Die Junge Wissenschaft wird Dich durch den Gesamtprozess des Erstellens einer wissenschaftlichen Arbeit begleiten – als gute Vorbereitung auf das, was Du im Studium benötigst.



Richtlinien für Beiträge

Für die meisten Autor/Inn/en ist dies die erste wissenschaftliche Veröffentlichung. Die Einhaltung der folgenden Richtlinien hilft allen – den Autor/innen/en und dem Redaktionsteam

Die Junge Wissenschaft veröffentlicht Originalbeiträge junger AutorInnen bis zum Alter von 23 Jahren.

- Die Beiträge können auf Deutsch oder Englisch verfasst sein und sollten nicht länger als 15 Seiten mit je 35 Zeilen sein. Hierbei sind Bilder, Grafiken und Tabellen mitgezählt. Anhänge werden nicht veröffentlicht. Deckblatt und Inhaltsverzeichnis zählen nicht mit.
- Formulieren Sie eine eingängige Überschrift, um bei der Leserschaft Interesse für Ihre Arbeit zu wecken, sowie eine wissenschaftliche Überschrift.
- Formulieren Sie eine kurze, leicht verständliche Zusammenfassung (maximal 400 Zeichen).
- Die Beiträge sollen in der üblichen Form gegliedert sein, d. h. Einleitung, Erläuterungen zur Durchführung der Arbeit sowie evtl. Überwindung von Schwierigkeiten, Ergebnisse, Schlussfolgerungen, Diskussion, Liste der zitierten Literatur. In der Einleitung sollte die Idee zu der Arbeit beschrieben und die Aufgabenstellung definiert werden. Außerdem sollte sie eine kurze Darstellung schon bekannter, ähnlicher Lösungsversuche enthalten (Stand der Literatur). Am Schluss des Beitrages kann ein Dank an Förderer der Arbeit, z. B. Lehrer und Sponsoren, mit vollständigem Namen angefügt werden. Für die Leser kann ein Glossar mit den wichtigsten Fachausdrücken hilfreich sein.
- Bitte reichen Sie alle Bilder, Grafiken und Tabellen nummeriert und zusätzlich als eigene Dateien ein. Bitte geben Sie bei nicht selbst erstellten Bildern, Tabellen, Zeichnungen, Grafiken etc. die genauen und korrekten Quellenangaben an (siehe auch [Erstveröffentlichungserklärung](#)). Senden Sie Ihre Bilder als Originaldateien oder mit einer Auflösung von mindestens 300 dpi bei einer Größe von 10 · 15 cm! Bei Grafiken, die mit Excel erstellt wurden, reichen Sie bitte ebenfalls die Originaldatei mit ein.
- Vermeiden Sie aufwendige und lange Zahlentabellen.
- Formelzeichen nach DIN, ggf. IUPAC oder IUPAP verwenden. Gleichungen sind stets als Größengleichungen zu schreiben.
- Die Literaturliste steht am Ende der Arbeit. Alle Stellen erhalten eine Nummer und werden in eckigen Klammern zitiert (Beispiel: Wie in [12] dargestellt ...). Fußnoten sieht das Layout nicht vor.
- Reichen Sie Ihren Beitrag sowohl in ausgedruckter Form als auch als PDF

ein. Für die weitere Bearbeitung und die Umsetzung in das Layout der Jungen Wissenschaft ist ein Word-Dokument mit möglichst wenig Formatierung erforderlich. (Sollte dies Schwierigkeiten bereiten, setzen Sie sich bitte mit uns in Verbindung, damit wir gemeinsam eine Lösung finden können.)

- Senden Sie mit dem Beitrag die [Erstveröffentlichungserklärung](#) ein. Diese beinhaltet im Wesentlichen, dass der Beitrag von dem/der angegebenen AutorIn stammt, keine Rechte Dritter verletzt werden und noch nicht an anderer Stelle veröffentlicht wurde (außer im Zusammenhang mit **Jugend forscht** oder einem vergleichbaren Wettbewerb). Ebenfalls ist zu versichern, dass alle von Ihnen verwendeten Bilder, Tabellen, Zeichnungen, Grafiken etc. von Ihnen veröffentlicht werden dürfen, also keine Rechte Dritter durch die Verwendung und Veröffentlichung verletzt werden. Entsprechendes [Formular](#) ist von der Homepage www.junge-wissenschaft.ptb.de herunterzuladen, auszudrucken, auszufüllen und dem gedruckten Beitrag unterschrieben beizulegen.
- Schließlich sind die genauen Anschriften der AutorInnen mit Telefonnummer und E-Mail-Adresse sowie Geburtsdaten und Fotografien (Auflösung 300 dpi bei einer Bildgröße von mindestens 10 · 15 cm) erforderlich.
- Neulingen im Publizieren werden als Vorbilder andere Publikationen, z. B. hier in der Jungen Wissenschaft, empfohlen.

Impressum

[JUNGE]
wissenschaft



Junge Wissenschaft

c/o Physikalisch-Technische
Bundesanstalt (PTB)
www.junge-wissenschaft.ptb.de

Redaktion

Dr. Sabine Walter, Chefredaktion
Junge Wissenschaft
Paul-Ducros-Str. 7
30952 Ronnenberg
E-Mail: sabine.walter@verlag-jungewissenschaft.de
Tel.: 05109 / 561 508

Verlag

Dr. Dr. Jens Simon,
Pressesprecher der PTB
Bundesallee 100
38116 Braunschweig
E-Mail: jens.simon@ptb.de
Tel.: 0531 / 592 3006
(Sekretariat der PTB-Pressestelle)

Design & Satz

Sebastian Baumeister
STILSICHER - Grafik & Werbung
E-Mail: baumeister@stilsicher.design
Tel.: 05142 / 98 77 89

