

# Eine Analyse des Effektes von Lernen auf Populationsfitness und Diversität in einer NK-Fitnesslandschaft

Seminararbeit von:  
LARS MELCHIOR  
lars.melchior@gmail.com

23. August 2012

Prüfer: PROF. DR. WINFRIED KURTH

## Zusammenfassung

Diese Seminararbeit untersucht, wie *Populations-Lernen* durch die Einführung von *Lebzeit-Lernen* beeinflusst werden kann. Sie Betrachtet dazu die Arbeit *An Analysis of the Effects of Lifetime Learning on Population Fitness and Diversity in an NK Fitness Landscape* von Dara Curran, Colm O'Riordan und Humphrey Sorensen [2]. Diese ergänzt die Erkenntnisse von Geoffrey E. Hinton und Steve J. Nowlan, die in dem Paper *How learning can guide Evolution* [3] bereits zeigten, dass Lebzeit-Lernen bei gradientlosen Fitnesslandschaften die Genetische Suche erheblich verbessert. Es wird gezeigt, dass Lebzeit Lernen die Genetische Suche auch bei NK Fitnesslandschaften verbessert und zusätzlich die Diversität der Population steigert.

Diese Arbeit geht zunächst auf die theoretischen Hintergründe des Papers ein und präsentiert anschließend die gewonnen Erkenntnisse.

# Inhaltsverzeichnis

<b>1</b>	<b>Theoretische Grundlagen</b>	<b>1</b>
1.1	Genetik . . . . .	1
1.1.1	Genom . . . . .	1
1.2	Genetische Algorithmen . . . . .	2
1.2.1	Einführung in Genetische Algorithmen . . . . .	2
1.2.2	Ablauf eines genetischen Algorithmus . . . . .	3
1.2.3	Selektion . . . . .	3
1.2.4	Fitnessfunktion . . . . .	4
1.3	NK-Modell . . . . .	4
1.3.1	Beschreibung . . . . .	4
1.3.2	Definition . . . . .	4
1.3.3	Beispiel 1 . . . . .	5
1.3.4	Beispiel 2 . . . . .	5
1.4	Baldwin-Effekt . . . . .	6
<b>2</b>	<b>Methoden</b>	<b>8</b>
2.1	Modell . . . . .	8
2.2	Genom . . . . .	8
2.3	Fitnessfunktion . . . . .	8
2.4	Bestimmung der Diversität . . . . .	10
2.5	Übersicht der Parameter . . . . .	10
<b>3</b>	<b>Ergebnisse</b>	<b>11</b>
<b>4</b>	<b>Zusammenfassung und Ausblick</b>	<b>13</b>
	<b>Literatur</b>	<b>14</b>

# 1 Theoretische Grundlagen

## 1.1 Genetik

### 1.1.1 Genom

Als Genom bezeichnet man die Gesamtheit der Vererbbaeren Informationen eines Organismus. Diese sind bei Lebewesen grotenteils in der Abfolge der Basen Adenin, Guanin, Cytosin und Thymin in der DNA verschlsselt. Dabei werden zu Lebzeit erlangte Eigenschaften des Organismus nicht auf die DNA bertragen. Durch, im Alterungsprozess auftretende Einflsse, verndert sich die DNA dennoch. Dabei knnen Basen hinzugefgt, entfernt oder ausgetauscht werden. Diese Mutationen knnen vorteilhafte oder auch nachteilhafte Auswirkung haben, in vielen Fllen bewirken die Mutationen jedoch keine Auswirkungen auf den Organismus, beispielsweise wenn die Betroffene Sequenz nicht genetisch relevant ist oder von anderen Merkmalen berdeckt wird (Rezessive Wirkung).

Das Konzept des Genoms ist jedoch nicht an Lebewesen gebunden und lsst sich leicht auf Systeme wie Schaltkreise oder Computerprogramme bertragen [4].

Bei der asexuellen Reproduktion wird das Genom eines Organismus direkt an seine Nachkommen vererbt. Dabei erhalten die Nachkommen bis auf zufllige Mutationen eine exakte Kopie des Vorgngergenoms.

Bei der Sexuellen Reproduktion werden die Genome der Nachkommen durch intra- und interchromosomale Rekombination von Mutter- und Vatergenomen erzeugt. Durch interchromosomale Rekombination erhalten die Nachkommen Kombinationen der Chromosomen von Vater und Mutter. Bei intrachromosomaler Rekombination wird durch Neukombination von Allelen<sup>1</sup> aus zwei Chromosomen ein neues Chromosom gebildet.

Die interchromosomale Rekombination wird oft auch als Cross-Over bezeichnet. Eine schematische Darstellung von One-Point Cross-Over, also Rekombination mit einer Schnittstelle, ist auf Abbildung 1 dargestellt. Es sind auch mehrere Schnittstellen mglich.

---

<sup>1</sup>Als Allel wird die Ausprgung eines Gens in einem Chromosom bezeichnet

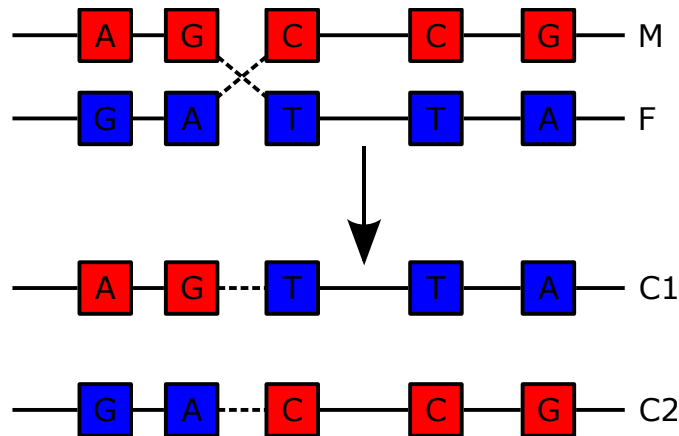


Abbildung 1: Schematische Darstellung der genetischen Rekombination durch One-Point Crossover. Der Rote und Blaue Strang symbolisieren die Basenstränge der Vorfahren. Der Crossover-Punkt entspricht der gestrichelten Linie. Die unteren beiden Stränge stellen die daraus resultierenden möglichen Nachfolgerstränge dar. [Quelle: <http://de.wikipedia.org/wiki/Crossing-over>, stand 13.08.2012 um 10:00]

## 1.2 Genetische Algorithmen

### 1.2.1 Einführung in Genetische Algorithmen

Genetische Algorithmen sind stochastische Suchalgorithmen, die sich an die natürlichen Prozesse der Genetik und Selektion anlehnen. Diese werden auf eine oder mehrere Populationen möglicher Lösungen angewendet, wobei die Informationen, die eine Lösung eindeutig charakterisieren, als Genom verschlüsselt werden. Durch genetische Operationen wie Rekombination und Mutation, die auf solche Genome wirken, werden iterativ neue Lösungen erzeugt. Durch Selektion werden bessere Lösungen dabei behalten, während schlechtere verworfen werden. Es gibt eine Vielzahl von Gründen, warum es sinnvoll ist Genetische Algorithmen bei Maschinellen Lernen anzuwenden. Da sie auf einen diskreten Lösungsraum wirken, werden sie dort angewendet, wo Gradient-basierte Suchalgorithmen nicht möglich sind [6]. Die gefundenen Lösungen werden durch eine einzige Zahl, der Fitness bewertet. Dies erlaubt es, genetische Algorithmen in Situationen zu verwenden, wo potentielle Lösungen nur anhand ihrer Leistung (Fitness) bewertet werden können. Genetische Algorithmen können auch in Problemstellungen angewendet werden, wo keine einzelne, sondern eine Gruppe von Lösungen gesucht wird, da diese selbst auf Populationen von Lösungen wirken. Aktuelle Anwendungsgebiete Genetischer Algorithmen sind unter anderem die Suche nach Quantenalgorithmen, Erzeugung von elektrischen Schaltkreisen, Regelsuche bei Zellulären Automaten und Erzeugung von Sortieralgorithmen [4].

## 1.2.2 Ablauf eines genetischen Algorithmus

Der genetische Algorithmus erzeugt aus einer oder mehreren zufällig erzeugten Anfangspopulationen iterativ neue Populationen, bis eine Abbruchbedingung erfüllt ist. Die Struktur des vorgestellten Genetischen Algorithmus ist eine Variation des in Kapitel 2.2 des Buches *Genetic Programming IV* von John Koza [4] vorgestellten Algorithmus:

1. Erzeuge eine Anfangspopulation (Generation 0) aus zufällig erzeugten Genomen.
2. Folgende Operationen werden nun iterativ auf die Population angewendet, bis die Abbruchbedingung erfüllt ist (Dieser Schritt wird auch als Generation bezeichnet):
  - (a) Bestimme für jedes Genom die entsprechende Fitness.
  - (b) Ersetze die Population durch eine neue Population mit gleicher Anzahl an Genomen. Diese wird durch sukzessive Anwendung jeweils mit Wahrscheinlichkeit  $p_i$  ausgewählter genetischen Operationen gebildet:
    - i.  $p_R$  : Reproduktion: Selektiere ein Genom und kopiere es in die neue Population.
    - ii.  $p_C$  : Crossover: Selektiere zwei Genome und erzeuge durch Rekombination zwei neue Genome.
    - iii.  $p_M$  : Mutation: Selektiere ein Genom und erzeuge ein neues Genom, welches sich durch zufällige Mutationen von dem ursprünglichen Genom unterscheidet.
3. Die zu dem Genom mit der höchsten Fitness gehörende Lösung wird als Ergebnis der Simulation ausgegeben.

Üblicherweise wird die Crossover-Wahrscheinlichkeit sehr hoch gewählt, während die Wahrscheinlichkeiten für Mutation und Reproduktion eher klein gehalten werden [4]. Die Geschwindigkeit, mit der sich die Population verändert, bezeichnet man als Evolutionsrate.

## 1.2.3 Selektion

Entscheidend für die Evolutionsrate ist die Methode mit der Genome aus einer Population selektiert werden. Die einfachste Variante ist die Wahrscheinlichkeit  $p_i$  ein bestimmtes Genom  $i$  zu selektieren, proportional zu der Fitness  $f_i$  des jeweiligen Genoms zu setzen. Dies führt jedoch bei Populationen mit hoher durchschnittlichen

Fitness dazu, dass alle Genome mit näherungsweise gleicher Wahrscheinlichkeit selektiert werden und dadurch die Evolutionsrate mit zunehmender Fitness sinkt. Eine mögliche Methode dies zu umgehen wäre die Sigma-Skalierung, bei der  $p_i$  proportional zur Differenz von  $f_i$  zu der durchschnittlichen Fitness  $\langle f \rangle$  der Population gesetzt wird [5]:

$$p_i = \begin{cases} 1 + \frac{f_i - \langle f \rangle}{2\sigma}, & \text{falls } \sigma \neq 0 \\ 1, & \text{sonst.} \end{cases}$$

$\sigma$  ist hierbei die Varianz der Fitness der Population.

### 1.2.4 Fitnessfunktion

Die Fitnessfunktion ordnet jedem Genom einen Fitnesswert zu. Dabei entscheidet sie wesentlich den Ausgang des Genetischen Algorithmus, da der Wert dieser Funktion maximiert werden soll. Wird auf den Raum der Genome eine Metrik definiert, so ergibt sich eine Fitnesslandschaft, welche im allgemeinen eine hohe Rauheit besitzt, auf der also viele lokale Hügel und Täler vorkommen.

## 1.3 NK-Modell

### 1.3.1 Beschreibung

Das NK-Modell beschreibt eine Menge von Abbildungen, die Zeichenketten der Länge  $N$  skalare Werte zuordnen. Diese Werte werden auch als Fitness der Zeichenkette bezeichnet. Wird ein Metrik zwischen den Zeichenketten gewählt, beispielsweise der Hamming-Abstand, erzeugt ein NK-Modell eine Fitnesslandschaft. Die Rauheit einer solcher Landschaft wird durch den Parameter  $K$  bestimmt. Je größer  $K$  gewählt wird, desto rauer ist die sich ergebende Landschaft.

Dieses Modell hat wurde in der Optimierungstheorie viel Aufmerksamkeit geschenkt, da es sich hierbei um ein verhältnismäßig einfaches Beispiel für ein NP-Vollständiges Problem handelt [7].

### 1.3.2 Definition

Ein NK Modell definiert eine Menge von skalaren Abbildungen  $F_{NK}$ , die auf Zeichenketten aus  $N$  Zeichen eines Alphabetes  $\mathbb{A}$  wirken.

$$F_{NK} : \mathbb{A}^N \rightarrow \mathbb{R}$$

Dabei setzt sich der Wert einer Zeichenkette  $S = \{S_1, S_2, \dots, S_N\}$  aus der Summe von Beiträgen der einzelnen Zeichen  $f(S_i)$  zusammen:

$$F_{NK}(S) = \sum_{i=1}^N f(S_i).$$

Der Einzelbeitrag eines Zeichens  $S_i$  hängt selbst von  $K$  weiteren Zeichen ab:

$$f(S_i) = f(S_i, S_1^i, S_2^i, \dots, S_K^i).$$

Die Form dieser Funktion  $f(S_i, S_1^i, S_2^i, \dots, S_K^i)$  und von welchen weiteren Zeichen  $S_j^i$  sie abhängt, bleibt offen.

### 1.3.3 Beispiel 1

Wir wählen ein NK-Modell mit  $N = 5$  und  $K = 1$ . Als Alphabet wählen wir das binäre Alphabet  $\mathbb{A} = \{0, 1\}$ . Den Einzelbeitrag eines Zeichens definieren wir als:

$$f(S_i) = \begin{cases} f(S_5, S_1), & \text{falls } i = 5 \\ f(S_i, S_{i+1}), & \text{sonst.} \end{cases}$$

und

$$f(0, 0) = 0, f(0, 1) = 1, f(1, 0) = 2, f(1, 1) = 0.$$

Die Zeichenketten 01100 sowie 01010 erhalten somit die Fitness:

$$\begin{aligned} F_{NK}(01100) &= f(0, 1) + f(1, 1) + f(1, 0) + f(0, 0) + f(0, 0) \\ &= 1 + 0 + 2 + 0 + 0 = 3 \\ F_{NK}(01010) &= f(0, 1) + f(1, 0) + f(0, 1) + f(1, 0) + f(0, 0) \\ &= 1 + 2 + 1 + 2 + 0 = 5 \end{aligned}$$

### 1.3.4 Beispiel 2

Auf Abbildung 2 ist eine  $NK$ -Fitnesslandschaft als Graph dargestellt.  $N$  wurde dabei auf 5 gesetzt, während  $K$  abwechselnd auf 0, 1 und 2 gesetzt wurde. Dabei ist zu erkennen, wie die Rauheit mit steigendem  $K$  ansteigt.



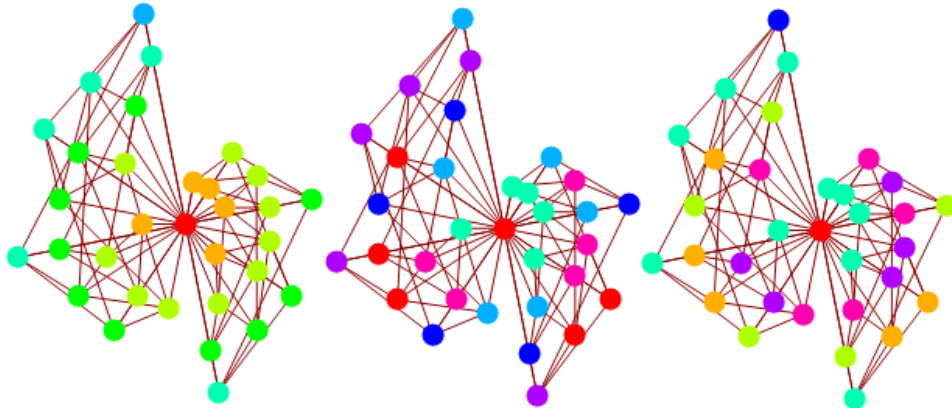


Abbildung 2: Darstellung einer NK Landschaft als Graph mit  $N = 5$  und binärem Alphabet. Die Knoten stellen alle 32 möglichen Zeichenketten dar, wobei Knoten mit einem Hamming-Anstand von 1 über Kanten miteinander verbunden sind.  $K$  ist im linken Bild 0, im mittleren 1 und im rechten Bild als 2 gewählt. Die Farbe der Knoten stellt die Fitness der entsprechenden Zeichenkette dar. Rote Werte bedeuten dabei höhere Fitness. [Quelle: [http://en.wikipedia.org/wiki/NK\\_model](http://en.wikipedia.org/wiki/NK_model), stand 13.08.2012 um 10:00]

## 1.4 Baldwin-Effekt

Der Baldwin Effekt bezeichnet einen evolutionären Mechanismus, der 1896 von Jack Mark Baldwin beschrieben wurde [1]. Dieser besagt, dass Lernen einen starken Einfluss auf den Lauf der Evolution hat, selbst wenn erlernte Eigenschaften nicht vererbbar sind. Geoffrey E. Hinton und Steve J. Nowlan haben diesen in ihrem Paper *How Learning can guide Evolution* [3] an dem folgenden einfachen (jedoch unrealistischen) Beispiel demonstriert.

Betrachtet man eine Fitnesslandschaft, die bis auf bei einer ganz bestimmten Allelkombination immer einen minimalen Wert annimmt, ist es für einen Genetischen Algorithmus nur durch Zufall möglich diese optimale Allelkombination zu finden, da kein Gradient existiert, der eine Richtung für die Evolution vorgibt (Nadel im Heuhaufen). Die evolutionäre Suche wird deutlich verbessert, wenn das Genom nur einige Eigenschaften vorgibt, andere jedoch dem lernen überlassen werden. Dadurch entsteht ein Bereich „erhöhter Fitness“ um den Peak (rot), da nun Genome in der Nähe des Peaks die richtigen Eigenschaften durch lernen erlangen können. Dadurch entsteht ein künstlicher Gradient, der dem Genetischen Algorithmus erlaubt, die richtige Allelkombination gezielt zu finden.

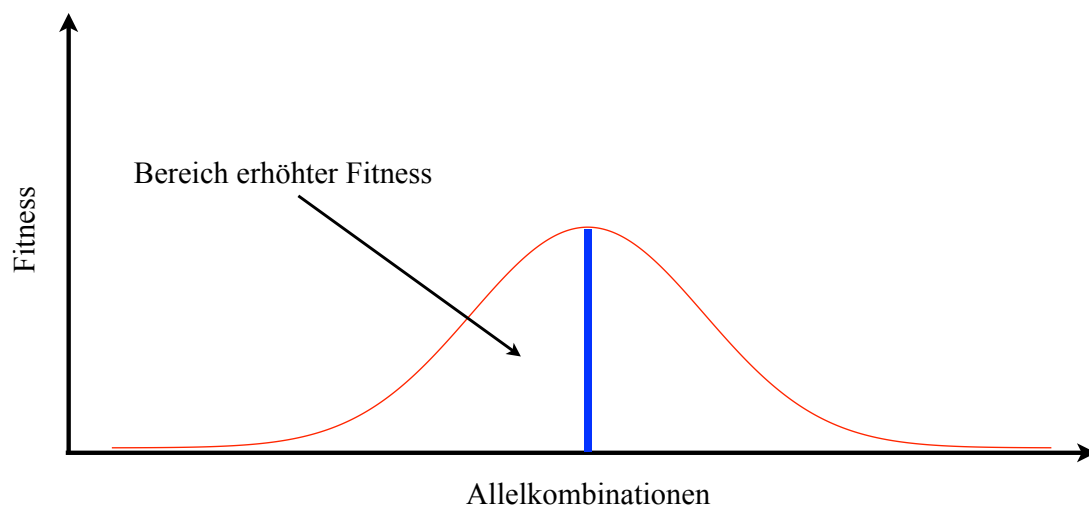


Abbildung 3: Veranschaulichung des Beispiels zum Baldwin-Effekt aus Kapitel 1.4. Die Abszisse stellt Allelkombinationen dar, ist also nicht wirklich eindimensional. Ohne lernen wäre die Fitnesslandschaft ein einzelner Peak (blau) bei der optimalen Allelkombination. Mit lernen entsteht um diesen Peak ein Bereich erhöhter Fitness.

## 2 Methoden

### 2.1 Modell

Es werden nun Populationen mit lernenden oder nicht-lernenden Agenten betrachtet. Dabei besitzt jeder Agent ein Genom aus einer Zeichenkette und einen Fitnesswert, der diesem über eine NK-Fitnesslandschaft zugeordnet wird. Die Populationen werden über genetische Algorithmen entwickelt und dabei die durchschnittliche Fitness und Diversität aufgezeichnet.

### 2.2 Genom

Das Genom eines einzelnen Agenten wird als Zeichenkette mit 20 Buchstaben aus einem Alphabet  $\mathbb{A}_{\text{Lernend}}$  für lernende bzw.  $\mathbb{A}_{\text{Nicht-lernend}}$  für nicht-lernende Agenten realisiert. Dabei ist  $\mathbb{A}_{\text{Nicht-lernend}} = \{0, 1\}$ , wobei bei der Anfangspopulation beide Zeichen mit gleicher Wahrscheinlichkeit gewählt werden.  $\mathbb{A}_{\text{Lernend}} = \{0, 1, ?\}$ , wobei bei der Anfangspopulation die Wahrscheinlichkeit für „?“ als 0,5 und die Wahrscheinlichkeit für 0 und 1 entsprechend als 0,25 festgelegt wird.

### 2.3 Fitnessfunktion

Als Fitnessfunktion wird eine normierte NK-Fitnesslandschaft erzeugt, wobei  $N = 20$ ,  $K = 15$  und  $\mathbb{A} = \{0, 1\}$  gewählt wird. Normiert bedeutet dabei, dass der maximal erreichbare Fitnesswert 1 beträgt. Die Fitness eines Agenten  $i$  mit Genom  $G_i$ ,  $F(G_i)$ , wird nun über die NK Fitnessfunktion  $F_{\text{NK}}$  bestimmt:

Für einen nicht-lernenden Agenten:  $F(G_i) = F_{\text{NK}}(G_i)$ .

Für einen lernenden Agenten:  $F(G_i) = \max(F_{\text{NK}}(G_i^1), F_{\text{NK}}(G_i^2), \dots, F_{\text{NK}}(G_i^L))$ , wobei  $G_i^j$  erzeugt wird, indem die „?“ im Genom  $G_i$  zu gleicher Wahrscheinlichkeit durch 0 oder 1 ersetzt werden.

Es wird nun für lernende sowie nicht lernende Agenten ein Phaenotyp<sup>2</sup> definiert. Bei nicht-lernenden Agenten ist dieses mit dem Genom gleichzusetzen, bei lernenden Agenten ist es die aus dem Genom resultierende Zeichenkette, die zur höchsten Fitness beigetragen hat. Der Prozess zur Fitnessbestimmung ist auf den Abbildungen 4 und 5 dargestellt.

---

<sup>2</sup>Der Phaenotyp bezeichnet die Menge aller Merkmale eines Organismus.

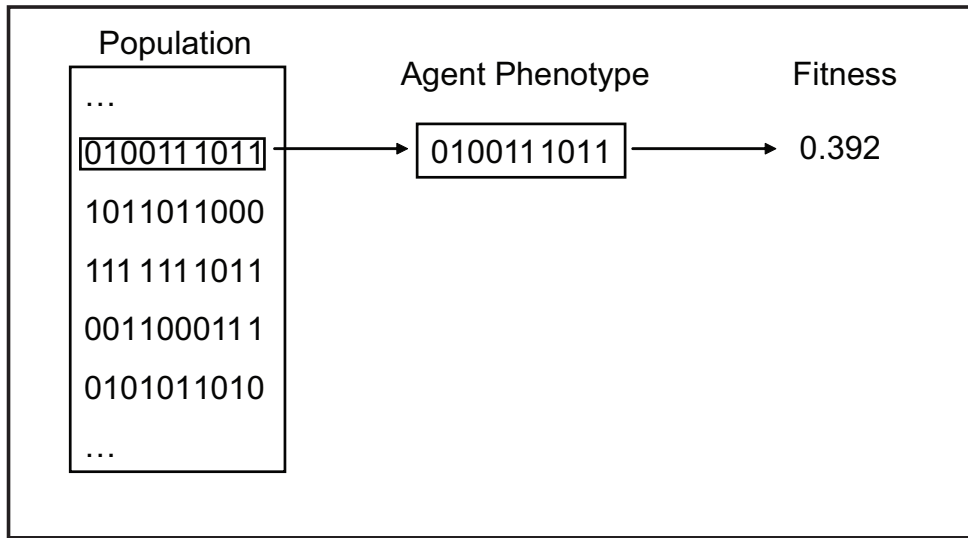


Abbildung 4: Veranschaulichung der Fitness Bestimmung bei nicht-lernenden Agenten. Aus: [2].

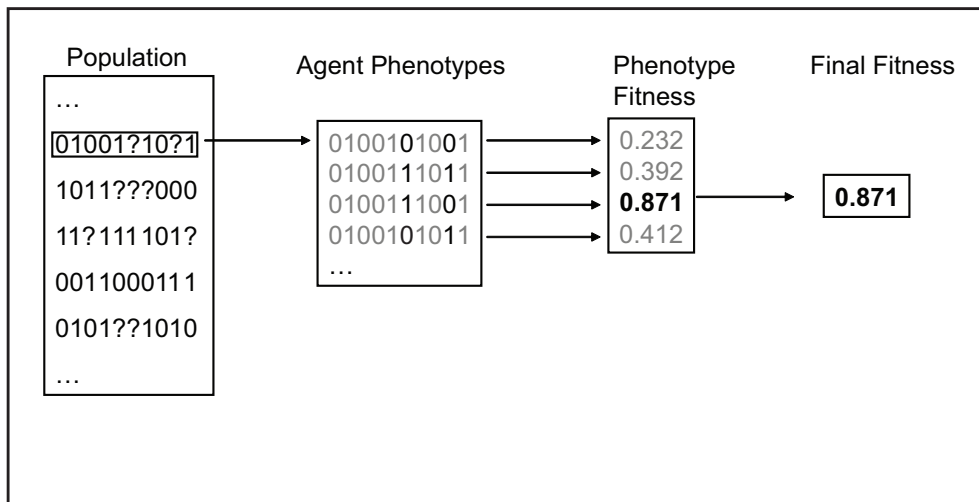


Abbildung 5: Veranschaulichung der Fitness Bestimmung bei lernenden Agenten. Aus: [2].

## 2.4 Bestimmung der Diversität

Die Diversität einer Population wird bestimmt über den mittleren Hamming Abstand  $h(x, y)$  zwischen zwei einzigartigen Phenotypen  $x$  und  $y$ , gewichtet mit dem Verhältnis einzigartiger Phenotypen  $m$  mit der Gesamtzahl an Phenotypen  $n$  der Population:

$$\text{Diversität} := \frac{2}{n(m-1)} \sum_{x=1}^m \sum_{y=x+1}^m h(x, y).$$

## 2.5 Übersicht der Parameter

Name	Wert
Populationen	3
$L$	0, 10, 30
Generationen	400
$p_C$	0,6
$p_M$	0,01
$p_R$	0,39
$N$	20
$K$	15

Tabelle 1: Übersicht der Parameter

Auf Tabelle 1 ist eine Übersicht der für die Simulation verwendeten Parameter dargestellt. Die folgenden Resultate sind der Mittelwert aus 20 unabhängigen Läufen. Über die Selektionswahrscheinlichkeit werden in dem Paper keine weiteren Angaben gemacht, vermutlich wird  $p_i = f_i$  gewählt<sup>3</sup>.

---

<sup>3</sup>siehe hierzu auch Kapitel 1.2.3

### 3 Ergebnisse

Die Mittleren Fitnessverläufe der drei Populationen sind auf Abbildung 6 abgebildet. Die mittlere Diversität ist auf Abbildung 7 abgebildet. Hierbei schneiden Populationen mit Lernen sowohl in ihrer Fitness als auch in ihrer Diversität besser ab, als Populationen ohne lernen.

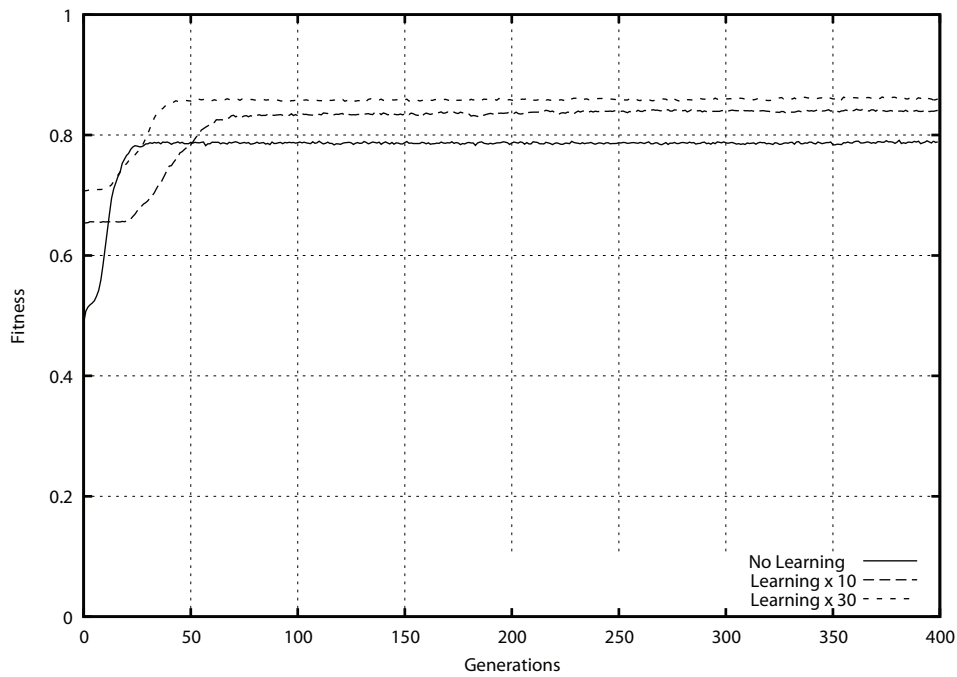


Abbildung 6: Mittlerer Fitnessverlauf aus 20 Läufen. Aus: [2].

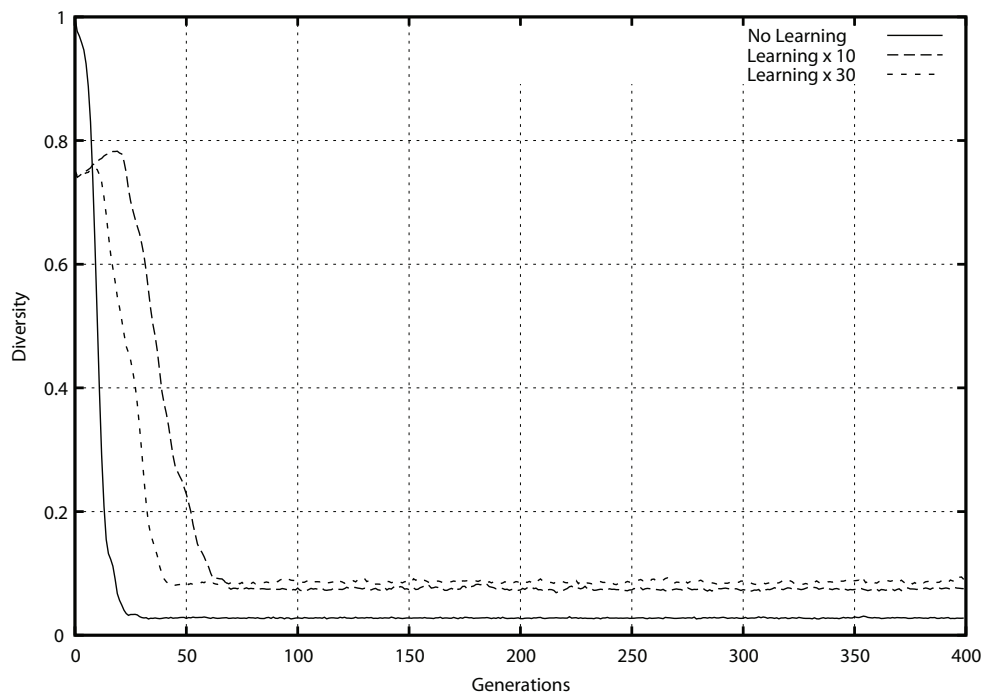


Abbildung 7: Mittlere Diversität aus 20 Läufen. Aus: [2].

## 4 Zusammenfassung und Ausblick

Die hier vorgestellte Arbeit: *An Analysis of the Effects of Lifetime Learning on Population Fitness and Diversity in an NK Fitness Landscape* von Dara Curran, Colm O'Riordan und Humphrey Sorensen zeigt nach eigenen Angaben einerseits, dass Lernen zu Lebzeit einen Anstieg der durchschnittlichen Fitness und der Diversität bewirkt. Andererseits wird gezeigt, dass das eingeführte Diversitätsmaß (siehe Kapitel 2.4) eine einfache Methode ist, diese Diversität zu bestimmen.

Die Erkenntnisse haben im Bereich der Anwendungsgebiete von genetischen Algorithmen weitreichende Auswirkungen. So sollte es motivieren, bei Anwendungsgebieten von Genetischen Algorithmen neben dem Populationslernen auch Lebzeit-Lernen einzuführen, da dies in bessere Lösungen resultieren könnte. Ein Beispiel ist eine Arbeit des Autors dieser Seminararbeit, bei der über genetische Algorithmen Funktionale für nichtlineare Optimierungsprobleme gesucht werden. Durch die Einführung von Lebzeit-lernen (in diesem Fall ein Levenberg-Marquardt Algorithmus) war es möglich die Qualität der gefundenen Lösungen drastisch zu erhöhen.



## Literatur

- [1] BALDWIN, James M.: A New Factor in Evolution. In: *The American Naturalist* 30 (1896), S. 441–451, 536–553
- [2] CURRAN, Dara ; O’RIODAN, Colm ; SORENSEN, Humphrey: An Analysis of the Effects of Lifetime Learning on Population Fitness and Diversity in an NK Fitness Landscape. In: *ECAL 2007, LNAI 4648 (Springer, Berlin)* (2007), S. 273–283
- [3] HINTON, Geoffrey E. ; NOWLAN, Steve J.: How Learning can guide Evolution. In: *Complex Systems, 1* (1987), S. 495–502
- [4] KOZA, John: *Genetic Programming IV*. Springer, 2005
- [5] MITCHELL, Melanie: *An Introduction to Genetic Algorithms*. The MIT Press, 1999
- [6] SHAPIRO, Jonathan: Genetic Algorithms in Machine Learning. In: *Machine Learning and its Applications, Springer, ISBN 3-540-42490-3* (2001), S. 146–168
- [7] WEINBERGER, Edward: NP-completeness of Kauffman’s N-K model, a Tuneably Rugged Fitness Landscape. In: *Santa Fe Institute Working Paper, 96-02-003* (1996)