Universität Stuttgart
Institut für Informatik
Abteilung Formale Konzepte

Seminar
"Optimierung mit evolutionären Algorithmen"

Seminararbeit zu genetischen Algorithmen

Betreuer: Dipl. Math. Nicole Weicker

Autor: Michael Wenig, 184 22 38

11. Juli 2000

Inhaltsverzeichnis

Einleitung

Die Genetischen Algorithmen (GA) zählen zu den bekanntesten evolutionären Algorithmen.
Sie gehen zurück auf die Arbeiten des Biologen Fraser, der die biologische Evolution am Rechner simulieren wollte, um sie – aus biologischer Sicht – besser zu verstehen.

Die genetischen Algorithmen im heutigen, informationstechnischen Verständnis gehen auf die Arbeiten von Holland, Computerwissenschaftlers und Psychologe an der University of Michigan, zurück. Holland untersuchte, inwieweit die biologische natürliche Auswahl auf praktische Problemen anwendbar ist. Er untersuchte insbesondere adaptive Systeme, die über binäre Detektoren mit ihrer Umwelt kommunizieren.

DeJong erarbeitete aus den Ergebnissen von Holland schließlich die Grundlagen heutiger genetischer Algorithmen. Viele seiner Ergebnisse sind noch heute die grundlegenden Prinzipien genetischer Algorithmen zur Lösung praktischer Probleme.[1]

Der Genetische Algorithmus

Grundprinzip

Der genetische Algorithmus (GA) startet mit einer Population zufällig erzeugter Chromosomen. Diese werden nach ihrer Qualität in Bezug auf die Problemstellung bewertet. Anschließend werden die Bits von jeweils 2 Chromosomen nach einem bestimmten Crossover-Verfahren getauscht, wodurch ein neues Chromosom, Kind genannt, entsteht.
Durch Mutationen werden einzelne Bits zufällig gesetzt, um die Erschließung bisher nicht beachteter Lösungsmöglichkeiten zu ermöglichen.

Die spezielle Implementierung genetischer Algorithmen ist sehr vielfältig. Diese Ausarbeitung bezieht sich hauptsächlich auf die Standard-Implementierung. Zunächst wird der Algorithmus im Gesamten dargestellt. Anschließend werden alle Schritte untersucht, die für eine Anwendung dieses Algorithmus durchgeführt werden müssen.

Der Standard-GA

Der Ablauf eines Standard-GAs wird im folgenden kurz vorgestellt:

0. Auswahl einer geeigneten Codierung
1. Erzeugen der Anfangs-Population durch zufällige Initialisierung
2. Bewertung der einzelnen Individuen

Die folgenden Schritte werden sooft ausgeführt, bis ein zufriedenstellendes Ergebnis erreicht wurde, oder ein Abbruchkriterium (z.B. Generationsindex > 1000) erfüllt ist:

3. Erzeugen von Kindern aus den Individuen der aktuellen Generation
4. Mutieren der erzeugten Kinder
5. Bewertung der einzelnen Individuen durch Fitnessfunktion
6. Selektion der Individuen der neuen Eltern-Population

Auswahl einer geeigneten Codierung

Genetische Algorithmen arbeiten im Gegensatz zu den Evolutionsstrategien ausschließlich auf binären Vektoren fester Länge, auch BitStrings oder Chromosomen genannt.

Diese BitStrings der Länge werden in der Regel in n gleichlange Segmente der Länge l aufgeteilt, so dass jedes Segment genau eine Variable des Problems codiert.

[1] vgl. Baeck S. 97 f

Es ist nun eine Decodierungsfunktion notwendig, um aus der Binärdarstellung eines Segments (Genotyp) den zugehörigen Problemwert der Variable (Phänotyp) ermittelt. Diese lässt sich formal darstellen durch:

$$Y^i : \{0,1\}^{l_i} \rightarrow W_i$$

Diese Funktion bildet Teile des BitStrings der Länge l_i auf konkrete Problemwerte W_i ab. Der Index i steht hier für eine einzelnen Problemvariablen.

Bei der Auswahl der Codierung ist zu beachten, dass kleine Änderungen im Genotyp keine großen Änderungen im Phänotyp hervorrufen. Dies ist insbesondere bei Reihenabhängigkeiten der Fall.

Beispiel Reihenabhängigkeit:
Die Anwendung erfordert die Darstellung von Permutationen der Zahlen 1 bis 8.
Codiert werden hier die Abstandsmaße zwischen den einzelnen Zahlen. Für die Codierung wird ein Gray-Code verwendet.

Abstand	Binär-Codierung		Abstand	Binär-Codierung
-7	0000		1	1100
-6	0001		2	1101
-5	0011		3	1111
-4	0010		4	1110
-3	0110		5	1010
-2	0111		6	1011
-1	0101		7	1001
0	0100		8	1000

Die Permutation	1	2	3	7	5	6	4	8
führt auf die Abstandsmaße	+1	+1	+1	+4	-2	+1	-2	+4
und wird codiert als	1100	1100	1100	11**1**0	0111	1100	0111	1110
nun wird das 15te Bit mutiert zu	1100	1100	1100	11**0**0	0111	1100	0111	1110
es ergeben sich die Abstandsmaße	+1	+1	+1	+1	-2	+1	-2	+4
daraus ergibt sich	1	2	3	4	2	3	1	5

Man erkennt hier, dass die Änderung lediglich eines Bits ein Ergebnis erzeugt, welches keine Permutation mehr darstellt. Dieser Fall, die Erstellung ungültiger Lösungen, tritt bei dieser Codierung mit sehr hoher Wahrscheinlichkeit ein.

Eine geeignete Codierung minimiert diese Wahrscheinlichkeit.

Die Auswahl der geeigneten Codierung ist zusammen mit der Fitnessfunktion maßgeblich für den Erfolg eines Genetischen Algorithmus!

Die Fitnessfunktion Φ

Die Fitnessfunktion dient der Abbildung der Qualität eines Individuums auf einen reellen positiven Zahlenwert. Der Standard-GA erfordert hierbei zusätzlich, dass dieser Zahlenwert mit der Qualität steigt.

Die Fitnessfunktion lässt sich somit mathematisch ausdrücken durch

$$\Phi : B^l \rightarrow \Re^+$$

B^l steht hier für einen Binärvektor der Länge l (=Chromosom=Individuum).

Die Fitnessfunktion ist einer der ausschlaggebenden Bestandteile für den Erfolg eines GA, da sie zur Lenkung der Optimierung beiträgt, indem die ermittelte Fitness zur Steuerung der Crossover- und Mutationsoperatoren verwendet wird.

Der Crossover

Der Crossover erzeugt durch Kombination aus 2 Eltern-Individuen u und v 1 Kind-Individuum w.

Der Crossover wird für zwei Individuum a nur mit einer bestimmten Wahrscheinlichkeit $p_c \in [0, 1]$ durchgeführt. Typische Werte[2] für p_c liegen im Bereich

$$p_c \in [0.6,0.95]$$

Im Standard-GA kommen one-point-crossover und uniform-crossover zum Einsatz:

one-point-crossover

Beim one-point-crossover wird zunächst eine Bitposition x zufällig ausgewählt:

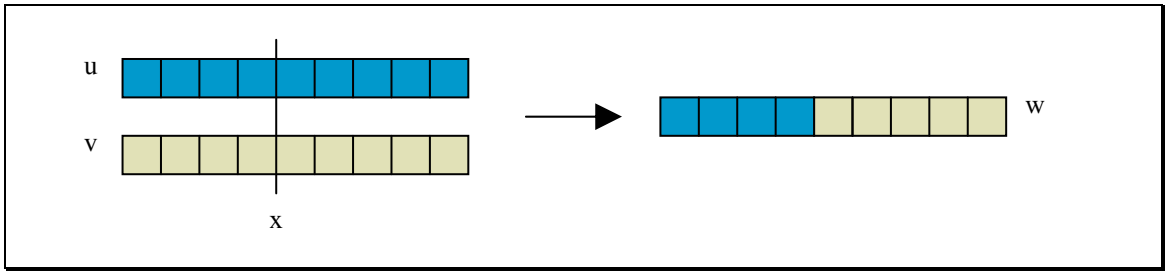
$$x \in \{1,...,l-1\}$$

Der angegebene Wertebereich schließt das erste und letzte Bit als mögliche Position aus!

Das Kind-Individuum w setzt sich nun zusammen aus

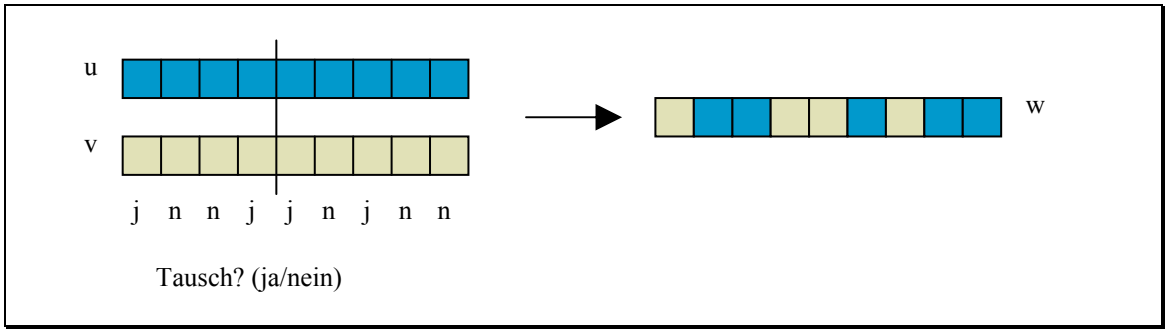
$$w = (u_1,...,u_{x-1},u_x,v_{x+1},...,v_l)$$

Graphisch lässt sich dies anschaulich darstellen durch



uniform-crossover

Beim uniform-crossover wird für jede Bitposition eine Wahrscheinlichkeit berechnet, ob die entsprechenden Bits vertauscht werden oder nicht.



Die Mutation

Die Mutation ändert einzelne Bits durch Invertierung. Dies hat den Zweck, zufällige Fluktuationen in den Individuen einzubringen, um z.B. das Hängenbleiben an lokalen Optima zu verhindern.

[2] vgl. Baeck S. 103

Wie der crossover wird auch die Mutation nur mit einer bestimmten gleichverteilten Mutationswahrscheinlichkeit $p_m \in [0, 1]$ angewendet, die in der Regel relativ gering ist. Typische Werte[3] für p_m sind

$$p_m \in [0.001, 0.01]$$

Untersuchungen haben hierbei ergeben, dass die Performance der GA sowohl bei hohen Mutationswahrscheinlichkeiten in Verbindung mit großen Populationsgrößen, sowie niedrige Mutationswahrscheinlichkeiten in Verbindung mit kleinen Populationen, abnimmt.[4]

Die Auswirkungen der Mutation werden beschrieben durch:

$$a_i' = \begin{cases} a_i & x_i > p_m \\ 1 - a_i & x_i \leq p_m \end{cases}$$

Der Index von x_i verdeutlicht, dass diese Zufallszahl für jede Bitposition neu berechnet wird.

Holland war der Ansicht, dass geringe Mutationswahrscheinlichkeiten garantieren, dass nur geringe Änderungen im Individuum auftreten. Dies ist korrekt wenn man nur den Genotyp betrachtet. Diese Aussage ist aber in der Regel nicht haltbar, sobald eine nicht-lineare Decodierfunktion verwendet wird.[5]

Die Selektion

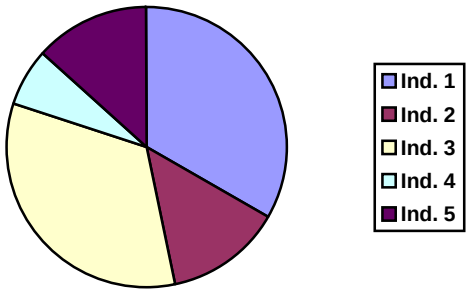
Hier wird bestimmt, welche Individuen in der nächsten Population vertreten sein sollen. Im Standard-GA wird hierbei ein fitnessproportionales Verfahren, das als "Roulette-Wheel" bezeichnet wird, verwendet. Dies soll sicherstellen, dass qualitativ hochwertige Individuen eine größere Überlebenschance erhalten als Individuen geringerer Qualität.

Hierzu wird die Idee eines Glücksrades verwendet, welches pro Individuum ein Abschnitt enthält, dessen Breite proportional zu dessen Fitness ist.

Dies soll an einem Beispiel verdeutlicht werden:
Für die Anschaulichkeit soll die aktuelle Population lediglich aus 5 Individuen bestehen;

Individuum	Fitness
1	5
2	2
3	5
4	1
5	2

das zugehörige Glücksrad:



Die Wahrscheinlichkeit mit der ein Individuum ausgewählt wird, lässt sich mathematisch wie folgt errechnen:

$$p_s(a_k) = \frac{\Phi(a_k)}{\sum_{n=1}^{\mu} \Phi(a_n)}$$

für das Individuum 1 ergibt sich somit:

$$p_s(a_1) = \frac{\Phi(a_1)}{\sum_{n=1}^{\mu} \Phi(a_n)} = \frac{5}{15} = \frac{1}{3} = 33\%$$

Hierbei ist Φ(a) die errechnete Fitness des Individuums a.

An diesem Rad wird nun μ mal gedreht und das jeweils ausgewählte Individuum der neuen Population hinzugefügt.

---

[3] vgl. Baeck S. 102
[4] vgl. Baeck S. 102
[5] vgl. Baeck S. 103

Da die Auswahl mit Zurücklegen erfolgt, kommen einige Individuen mehrfach in der neuen Population vor. Die wahrscheinliche Anzahl gleicher Individuen lässt sich wie folgt berechnen:

$$\eta(a_k) = \mu \bullet p_s(a_k)$$

Dies birgt die Gefahr, dass Super-Individuen, d.h. Individuen mit im Vergleich zur restlichen Population überdurchschnittlicher Qualität, in frühen Phasen überdurchschnittlich hoch, in späterem Verlauf aber unterdurchschnittliche Chancen erhalten. Dies soll an folgendem Beispiel erläutert werden:

Generation t=1

Individuum	Fitness
1	1
2	1
3	1
4	1
5	5
Summe	**9**

Generation t=1000

Individuum	Fitness
1	101
2	101
3	101
4	101
5	105
Summe	**509**

Die Wahrscheinlichkeit, dass Individuum 5 ausgewählt wird, errechnet sich nun als

$$p_s(a_1(1)) = \frac{\Phi(a_1(1))}{\sum_{n=1}^{\mu} \Phi(a_n(1))} = \frac{1}{9} \approx 11\% \qquad p_s(a_1(1000)) = \frac{\Phi(a_1(1000))}{\sum_{n=1}^{\mu} \Phi(a_n(1000))} = \frac{101}{509} \approx 20\%$$

$$p_s(a_5(1)) = \frac{\Phi(a_5(1))}{\sum_{n=1}^{\mu} \Phi(a_n(1))} = \frac{5}{9} \approx 55\% \qquad p_s(a_5(1000)) = \frac{\Phi(a_5(1000))}{\sum_{n=1}^{\mu} \Phi(a_n(1000))} = \frac{105}{509} \approx 21\%$$

Man erkennt, dass in der Generation 1 das Super-Individuum 5 mit sehr großer Wahrscheinlichkeit oft gewählt wird, wohingegen in Generation 1000 alle Individuen nahezu gleichberechtigt sind.

Um diesem Problem Abhilfe zu schaffen, wurden als Erweiterung verschiedene Skalierungsfunktionen eingeführt.

Erweiterungen des Standard-GA

Skalierungsfunktionen δ

Um dem Problem der Bevorzugung von Super-Individuen, wie sie bei der Selektion dargestellt wurden, entgegenzuwirken, wurden verschiedene Skalierungsfunktionen eingeführt. Diese habe die Aufgabe, die Relationen zwischen den verschiedenen Fitnesswerte so zu verändern, dass eine zielgerichtere Arbeitsweise des GA ereicht wird.

Inwieweit die Kombination aus Selektion und verschiedener Skalierungsfunktionen Auswirkungen auf die Performance eines GA hat, wurde bisher weder theoretisch noch empirisch untersucht.

Aus den verschiedenen Skalierungsfunktionen soll hier nur eine kleine Auswahl kurz vorgestellt werden:

linear static scaling

Hier werden alle Fitnesswerte auf eine Basis, die ungleich 0 sein kann, bezogen.

$$\delta\left(f\left(\mathrm{Y}(a)\right), \{c_0, c_1\}\right) = c_0 \bullet f\left(\mathrm{Y}(a)\right) + c_1$$

$c_0 \in \Re - \{0\}$ $\quad c_1 \in \Re$

Der Faktor c_0 ermöglicht, sowohl die Beeinflussung der Abstände zwischen den verschiedenen Fitnesswerten, sowie die Invertierung der Fitnesswerte, für den Fall, dass nicht ein Maximum sondern ein Minimum gesucht wird. Der Faktor c_1 verschiebt die Basis. Ist c_0 negativ, so ist c_1 so zu belegen, dass der kleinste Fitnesswert > 0 ist.

linear dynamic scaling

Diese Methode arbeitet ähnlich wie linear static scaling, mit dem Unterschied dass hier die Basis dynamisch auf den kleinsten Fitnesswert gesetzt wird.

$$\delta\left(f\left(\mathrm{Y}(a)\right),\{c_0,c_1\}\right)=c_0 \bullet f\left(\mathrm{Y}(a)\right)-\min\{f\left(\mathrm{Y}\left(a_j\right)\right)\ \left|a_j \in P(t)\right.\}$$

Hierbei ist P(t) die aktuelle Population und $c_0 \in \Re - \{0\}$

Selektionsmethoden

generational replacement [6]

Bei dieser Methode wird die komplette Population durch ihre Nachkommen ersetzt.

Hauptvorteil dieser Methode ist, dass die Dominanz einiger weniger guten Individuen unterbrochen werden kann, wodurch eine verfrühte Einschränkung des Suchraums verhindert wird.

Diese Methode birgt allerdings ähnliche Gefahren wie die Komma-Variante der ES in sich. Es kann vorkommen, dass sowohl die Fitness des besten Individuums als auch die durchschnittliche Fitness schlechter als die der Elternpopulation ist.

Um diesem Problem Abhilfe zu schaffen wurden Methoden wie elitism entwickelt, die die besten Individuen konservieren.

elitism[7]

Bei dieser Methode werden die n besten Individuen der aktuellen Population unverändert in die nächste Generation übernommen. Typische Werte für n sind $n \in \{1, 2\}$

weak-elitism[8]

Der weak-elistism arbeitet ähnlich wie die elitism-Methode, mit dem Unterschied dass die n besten Individuen vor der Übernahme zunächst mutiert werden.

delete-n-last[9]

Hier werden die n schlechtesten Individuen der aktuellen Population durch Individuen der Vorgänger-Population ersetzt. Diese Individuen können z.B. durch eine Roulette-Wheel-Selektion ermittelt werden.

Zusatzoption Altersheim[10]

Rechenberg war der Meinung, dass das maximale bzw. durchschnittliche Alter einer biologischen Generation genetisch bedingt ist. Er folgerte daraus, dass eine Simulation des Alterungsprozesses vorteilhaft für die Optimierung sein könnte.

Dabei werden 2 natürliche Zahlen g, h gewählt:

g	≤ Anzahl vorgesehener Generationen
h	≤ Populationsgröße μ

Nun wird zunächst eine gewählte Selektionsmethode ausgeführt. Aus den ausgewählten Elementen werden h Elemente zusätzlich in einen Altersheim-Pool eingefügt. Die Individuen in diesem Pool werden in den nächsten Generationen beim crossover mitberücksichtigt und nach g Generationen aus dem Altersheim entfernt.

---

[6] vgl. Schöneburg S. 205
[7] vgl. Schöneburg S. 206
[8] vgl. Schöneburg S. 207
[9] vgl. Schöneburg S. 207
[10] vgl. Schöneburg S. 209

Die Grundidee hierbei ist, dass die Erzeugung eines Individuums "zu früh" erfolgen kann, nämlich zu einem Zeitpunkt an dem der crossover mit anderen Individuen keine Verbesserung der Fitness bewirkt. Nach dem Schema-Theorem, auf das hier nicht näher eingegangen werden soll, sind diese Individuen mit großer Wahrscheinlichkeit Kinder von überdurchschnittlich guten Eltern, wodurch es Sinn macht, diese aufzubewahren und an den crossover-Prozessen zu beteiligen, bis die Voraussetzungen für crossover mit besserem Ergebnis gegeben sind.

Zusatzoption Kindergarten[11]

Aus ähnlichen Gründen wie die Altersheim-Option wurde auch die Kindergarten-Option entwickelt. Ihr liegt die Beobachtung zugrunde, dass höherentwickelte Lebewesen ihre Nachkommen meist zunächst in einer geschützten Umgebung aufwachsen lassen.

Hierfür werden wiederum 2 natürliche Zahlen g, h gewählt:

g ≤ Anzahl vorgesehener Generationen
h ≤ Populationsgröße μ

Nun wird wiederum eine gewählte Selektionsmethode ausgeführt. Pro Generation werden h Nachkommen für g Generationen in einen Kindergarten-Pool eingeführt. Während ihrer Verweildauer in diesem Pool werden die Individuen lediglich der Mutation unterworfen.

Nach jeweils g Generationen werden die Individuen, die sich während ihrer "Kindheit" bezogen auf die Fitness verbessert haben, aus dem Kindergarten entfernt und in die aktuelle Population eingefügt. Hat sich ihre Fitness nicht verbessert, so werden sie eliminiert.

Mutations-Schemata

Wie auch bei der Selektion sind bei der Mutation mittlerweile mehrere Varianten entstanden.

gleichverteilte Mutation/2

Hier wird ein durch eine gleichverteilte Zufallszahl ausgewählte Gen zufällig auf ein Allel aus {0, 1} gesetzt.

normalverteilte Mutation

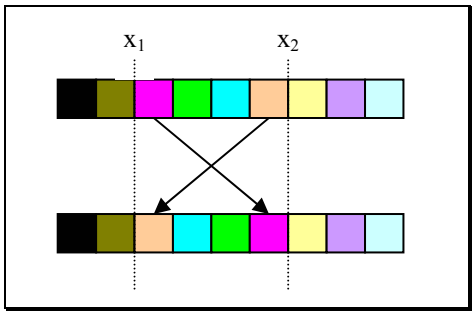
hier wird das zu mutierende Gen durch eine normalverteilte Zufallszahl ermittelt.

Positions-Mutation

Hier wird für jede Gen-Position eine eigene Mutationswahrscheinlichkeit festgelegt. Dies hat den Vorteil, dass in Bezug auf den Phänotyp höherwertige Gene eine geringere Mutationswahrscheinlichkeit erhalten können, als in Bezug auf den Phänotyp niederwertigere Gene.

Chromosomen-Inversion[12]

Hier wird innerhalb eines Chromosoms ein Teilstring umgedreht. Hierfür werden 2 Zahlen x_1, x_2 mit $1 \le x_1 \le x_2 \le n$ gewählt.



[11] vgl. Schöneburg S. 210
[12] vgl. Holland S. 106ff

selbst-optimierende Strategien [13]

Da die Abstimmung der einzelnen Parameter eines GA, insbesondere die verschiedenen Wahrscheinlichkeiten, großen Einfluss auf die Qualität eines GA hat, wurde die Idee entwickelt, die Wahrscheinlichkeiten ebenfalls durch den GA zu optimieren. Dies geschieht durch Erweiterung der Chromosomen um Codierungen der einzelnen Wahrscheinlichkeiten, so dass diese ebenfalls allen Operatoren und Methoden des GA unterworfen werden.

Implementierung eines GA

Für eine konkrete Implementierung eines GA sind folgende Schritte durchzuführen:

- Auswahl einer geeigneten Codierung
- Erstellung einer zugehörigen Decodier-Funktion
- Auswahl einer Populations-Größe μ
- Auswahl eines geeigneten Abbruchkriteriums
- Auswahl eines Crossover-Verfahrens oder Auswahl mehrerer Verfahren und einer Auswahlstrategie
- Auswahl einer Crossoverwahrscheinlichkeit
- Auswahl einer Mutations-Methode oder Auswahl mehrerer Methoden und einer Auswahlstrategie
- Auswahl einer Mutationswahrscheinlichkeit
- je nach verwendeten Methoden Auswahl geeigneter Werte für deren Parameter

Kombination mit anderen Verfahren

GA und Neuronale Netze

Als Diskussionsgrundlage wurde die Idee verwendet, wie man Genetische Algorithmen zur Konzeption von Neuronalen Netzen einsetzen könnte.

Hauptsächliche Diskussionspunkte sind hier, welche Bestandteile fest vorgegeben werden und welche von einem GA ermittelt werden.

Weiterer Diskussionspunkt war die Realisierung einer passenden Fitnessfunktion.
Für diese müsste für jedes erstellte NN eine Simulation des Einsatzes im Anwendungsgebiet durchgeführt werden, und daraus eine Bewertung der Güte des Netzes ermittelt werden.

Will man ein neuronales Netz für eine Strategie für das Spiel "Reversi"[14] entwerfen, so wären die Eingaben für jedes Feld 2 Neuronen (1x schwarz gesetzt, 1x rot gesetzt) und die Ausgaben ebenfalls für jedes Feld 2 Neuronen (setze schwarz, setze rot).

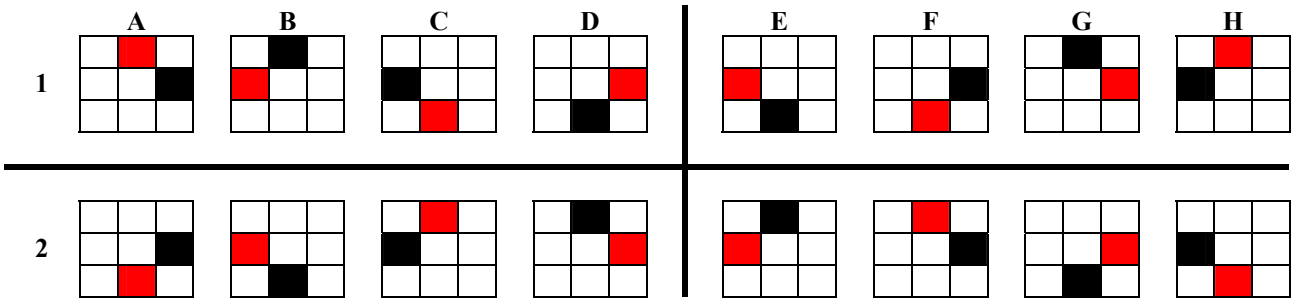
Da ein Reversi 8 x 8 Felder hat und jedes dieser Felder entweder mit rot, schwarz oder nichts belegt ist, ergibt sich somit als Obergrenze für die von Netz zu unterscheidenden Möglichkeiten:

$$8 \bullet 8 \bullet 3 = 192 \text{ Situationen}$$

Das hierfür erforderliche Netz würde aber sehr groß werden, und einen großen Anteil an Redundanz besitzen, da viele der implementierten Spielsituationen zwar nicht identisch aber dennoch äquivalent sind.

[13] vgl. Mitchell S. 177
[14] Die Spielregeln sind z.B. http://www.planet-interkom.de/grummelbaer/reversi.html

Dies soll an einem Beispiel verdeutlicht werden, wobei das Spielfeld aus Darstellungsgründen hier auf von 8 x 8 Feldern auf 3 x 3 Felder verkleinert wurde:



Die Situation A1 wurde durch Drehungen in die Situationen B1 – D1 überführt. Die Situationen E1 – H1 ergaben sich dann durch Spiegelung an der eingezeichneten vertikalen Achse. Durch eine weitere Spiegelung an der eingezeichneten horizontalen Achse ergaben sich die Situationen A2 – H2.

Entgegen der Vermutung, dass nun zu jeder Situation je 16 äquivalente Situationen existieren, stellt sich aber heraus, dass o.B.d.A. jeweils 2 Situationen identisch sind. Zum Beispiel ist die Situation A1 identisch mit der Situation F2. Der Versuch der Vergrößerung des Spielfeldes ändert diesen Sachverhalt ebenfalls nicht.

Da jede dieser Situationen aber mindestens 8 äquivalente Situationen besitzt, wäre bei einer entsprechenden Implementierung diese Anzahl auf nur

$$\frac{192}{8} = 24$$ unterschiedliche Situationen reduzierbar.

Weiteres Problem ist die Entscheidung "Was ist ein guter Zug", da sich die Güte nicht direkt nach dem Zug ermitteln lässt, und am Spielende auch nicht mehr entschieden werden kann, ob der Erfolg an diesem speziellen Zug lag, oder der Zug eigentlich sogar kontraproduktiv war und durch einen anderen Zug wieder ausgeglichen wurde.

Abhilfe zu diesem Problem wäre, sehr viele unterschiedliche neuronale Netze gegeneinander spielen zu lassen, und dann die Anzahl der Erfolge als Qualitätskriterium zu verwenden.

Anhang A: Übersicht der verwendeten Symbolik

Symbol	Bedeutung
B	Menge der Binärwerte: {0,1}
P	Population
W_i	Wertebereich der Problemvariablen i
Y^i	Decodierfunktion für Segment i
a	Individuum
$f(\mathrm{Y})$	Fitnessfunktion
l_x	Länge des Segmentes x
$p_c(a)$	Crossoverwahrscheinlichkeit von Individuum a
$p_m(a)$	Mutationswahrscheinlichkeit von Individuum a
$p_s(a)$	Selektionswahrscheinlichkeit von Individuum a
$\delta(f)$	Skalierungsfunktion
μ	Populationsgröße
Φ_i	Fitnessfunktion für Segment i

Anhang B: Abkürzungsverzeichnis

Abkürzung	Bedeutung
GA	Genetischer Algorithmus
o.B.d.A.	ohne Beweis der Allgemeinheit
NN	neuronales Netz
vgl.	vergleiche
z.B.	zum Beispiel

Anhang C: Literaturverzeichnis

Abkürzung	Volltitel
Schöneburg	E. Schöneburg, F. Heinzmann, S. Feddersen Genetische Algorithmen und Evolutionsstrategien Addison-Wesley, 1. Auflage 1994
Baeck	T. Bäck Doktorarbeit Evolutionary algorithms in theory and practice Universität Dortmund, 1994
Holland	J. H. Holland Adaptation in Natural and Artificial Systems MIT Press, 2. Auflage 1992
Mitchell	Melanie Mitchell An Introduction to Genetic Algorithms MIT Press, 1996

Anhang D: Glossar

Allel	konkrete Ausbildung eines Gens (0, 1)
Chromosom	= Individuum
Crossover	Konstruktion eines Kindes aus zwei Eltern
Decodierfunktion	Funktion, die ein Chromosom oder ein Segment in eine Problemlösung abbildet
Elter	Ein beim Crossover verwendetes Individuum
Fitness	Qualität eines Individuums im Sinne der Problemlösung
Gen	Bit eines Chromosoms
Generation	Die in einem Durchlauf des GA ermittelten Individuen
Genotyp	Darstellung als BitString
Individuum	BitString, der eine mögliche Problemlösung repräsentiert. Ein Individuum besitzt die codierten Werte aller Problemvariablen
Kind	Das beim Crossover entstehende Individuum
Mutation	eigenständige Änderung im Genotyp
Phänotyp	Zu einem Genotyp zugehörige Problemlösung
Population	Eine Menge von Individuen (i.d.R gleich der Generation)
Segment	Teil eines Individuums, welches genau eine Problemvariable codiert
Skalierungsfunktion	Nimmt Änderungen an den relativen Abständen zwischen Fitnesswerten vor.
Super-Individuum	Individuum mit im Vergleich zu den anderen Individuen der entsprechenden Generation ausserordentlich hoher Fitness