



Projekt für
Algorithmische Anwendungen

Genetische Algorithmen

Von
Mike Müller 11037091
Dennis Freese 11038517

Gruppe: B-ROT

Inhaltsverzeichnis

Allgemeines.....	3
Geschichtliche Entwicklung der genetischen Algorithmen.....	3
Was sind Genetische Algorithmen?.....	4
Wo werden Genetische Algorithmen angewendet?.....	5
Funktionsweise.....	6
Selektion.....	6
Crossover.....	6
Mutation.....	8
Codierung von Eigenschaften in Bit-Ketten:.....	8
Unser Beispiel mit einem Genetischen Algorithmus.....	9
Fazit.....	10
Literatur.....	11

Allgemeines

Als erstes möchten wir etwas über die geschichtliche Entwicklung der Genetischen Algorithmen erzählen, wonach wir darauf eingehen werden was Genetische Algorithmen sind und vor allem wo Sie angewendet werden.

Nachdem wir diese Punkte vorgestellt haben, können wir uns dann im nächsten Abschnitt der Funktionsweise im allgemeinen, sowie mit den einzelnen Funktionen von genetischen Algorithmen beschäftigen.

Geschichtliche Entwicklung der genetischen Algorithmen

Der Begriff Genetischer Algorithmus wurde in der Arbeit von Bagley [6] ursprünglich aufgegriffen. In der heute aber bekannten Form, wurden die genetischen Algorithmen aber von John H. Holland an der University of Michigan, USA, Anfang der 60er Jahre entwickelt.

Dabei ist John H. Holland der Pionier der Erforschung der genetischen Algorithmen.

Im gleichen Zeitraum wurden unter anderem auch Evolutionsstrategien und die evolutionäre Programmierung entwickelt, wobei diese einen größeren Wert auf Mutation und Selektion legen.

Im Jahre 1975 erschien die Primärliteratur (das Hauptwerk zu genetischen Algorithmen) Hollands: „Adaptation in Natural and Artificial Systems“.

Wonach John H. Holland dann mit Hilfe des genetischen Algorithmus die „broadcast language“ zur Beschreibung der Instruktionen und Grammatiken von Programmiersprachen entwickelte.

Wenig später schrieb Goldberg (ein Student von Holland) ebenfalls ein Buch zu den Genetischen Algorithmen mit dem Titel „*Genetic Algorithms in Search, Optimization, and Machine Learning*“ [1]. In seiner Dissertation von 1983 [7] beschrieb Goldberg die erste erfolgreiche Anwendung Genetischer Algorithmen. Dabei beschäftigte er sich mit der Steuerung von Gasleitungssystemen, wobei sein genetischer Algorithmus im Ergebnis sehr nah am Optimum lag.

Seit Mitte der 80er Jahre wurden die genetischen Algorithmen stets weiterentwickelt und zudem in vielen Bereichen optimiert, wobei die Grundfunktionen „Selektion, Crossover(Paarung) und Mutation“ aber unverändert in der Funktion und der Bedeutung geblieben sind.

Obwohl die genetischen Algorithmen sich von der Klasse der Evolutionsstrategien zwar am weitesten verbreitet haben, ist die Diskussion über genetische Algorithmen in den letzten Jahren abgeflacht. Trotz der recht großen Verbreitung in vielen Bereichen heutzutage.

Was sind Genetische Algorithmen?

Genetische Algorithmen sind Such Algorithmen die auf die Funktionsweise von natürlicher Selektion und Genetik basieren. Sie kombinieren das Prinzip „Überleben des Stärkeren“ und des zufälligen Informationsaustausches. Da Sie das Prinzip der Natur imitieren, wird eine künstliche Population erzeugt die eine Evolution unterläuft, um für das Problem eine Lösung zu finden. Der zufällige Teil der genetischen Algorithmen sorgt nicht dafür, dass er einen zufälligen Weg geht, sondern sie entnehmen effizient historische Daten von den Vorgängern, um mit erwarteter verbesserte Laufzeit auf neue Suchpunkte zu spekulieren.

Genetische Algorithmen sind heuristische Optimierungsverfahren. Das bedeutet, dass normalerweise für optimale Lösungen ein zu großer Aufwand benötigt wird und deshalb Annahmen (Heuristiken) getroffen werden die helfen sollen eine Lösung zu finden. Außerdem gehören Genetische Algorithmen zu den Evolutionären Algorithmen. Sie sind ein Optimierungsverfahren das die natürliche Evolution als Vorbild hat. Da die Natur durch bestimmte Verfahren in der Genetik imstande ist, das Leben den gegebenen Umständen anzupassen. Man kann diese Steuerungsmechanismen der Natur auf drei Funktionen zusammenfassen: Mutation, Selektion und Rekombination.

Genetische Algorithmen benötigen keine problemspezifischen Informationen über die zu lösende Aufgabe. Im Normalfall werden lediglich Parameter einer Gleichung, Formel oder in einer anderen vorgegebenen Form eines strukturierten Lösungsansatzes optimiert. Deswegen werden sie dort eingesetzt wo das Problem nicht gut verstanden werden kann oder deren Lösung wegen rechnerischen und mathematischen Gründen nicht lösbar sind. Dabei muss aber berücksichtigt werden, dass GA's nicht immer die beste Lösung finden. Vor allem die eingestellten Parameter beeinflussen die GA's sehr. So kann es durchaus passieren, dass wenn der Parameter für die Mutation zu hoch eingestellt wird, nie ein besseres Ergebnis (als schon vorhanden) erreicht werden kann. Bei einem zu klein gewählten Wert für diesen Parameter hingegen, kann es durchaus geschehen, dass eine einmal gefundene Lösung zu stark berücksichtigt wird. Dies führt dann zu einer schlechteren Endlösung, da keine besseren Lösungen gefunden werden können.

Wo werden Genetische Algorithmen angewendet?

Genetische Algorithmen werden hauptsächlich bei sehr komplexen und bei größeren Datenmengen angewendet. Also genau da, wo herkömmliche Algorithmen keine brauchbaren Ergebnisse erzeugen oder diese Ergebnisse eine viel zu lang Zeit beanspruchen würden.

Einige praktische Beispiele wären die Berechnung eines Stundenplans an einer sehr großen Schule. Die Platzierung von Containern auf einem Containerschiff um z.B. eine größtmögliche Stabilität des Schiffes zu gewährleisten sowie schnelles Entladen der Fracht in den Häfen. Es kann auch beim Entwurf von Kommunikationsnetzwerken eingesetzt werden, um viele Daten über möglichst wenige Leitungen und Knotenpunkte zu transportieren.

David E. Goldberg hat ein System entwickelt, um ein Pipeline System zu steuern. In diesem Pipeline System wird Erdgas vom Südwesten der USA in den Nordosten transportiert. Dabei kann das Erdgas nur über Ventile und Kompressoren geregelt werden. Da Änderungen sehr lange benötigen um eine Änderung des Gasdrucks im System zu bewirken, gibt es keine geschlossenen Lösungen des Steuerungsproblems. Die menschlichen Überwacher lernten aus Erfahrung genauso wie die Genetischen Algorithmen. Wobei die Algorithmen eine sehr gute Lösung erreichten und sogar für bestimmte Vorgänge ein Verhaltenskatalog aufgestellt werden konnte, wie z.B. welches Verhalten beim Rohrbruch das annähernd beste ist.

Genetische Algorithmen wurden auch schon für den Entwurf eines Mantelstrom-Düsentriebwerks eingesetzt. Diesen Algorithmus hatten Forscher vom Konzern General Electric und vom Rensselaer Polytechnic Institute gemeinsam verwendet. Beim Entwurf eines solchen Triebwerks gibt es mindestens 100 Variablen die sich ändern können und jeder Entwurf muss genauestens geprüft werden um festzustellen was die Änderungen bestimmter Variablen bewirken. Genetische Algorithmen zeigen schneller in welche Wege möglich sind und zudem zeigen Sie auch neue Wege die von den Entwicklern noch gar nicht bedacht worden sind.

Die oben genannten Einsatzgebiete zeigen aber nur ein kleines Einsatzgebiet der genetischen Algorithmen, da Sie für jedes Problem eingesetzt werden können, vorausgesetzt die Daten lassen sich codieren und bewerten. Dabei kann eine Codierung und Bewertung durchaus mehrere Wochen in Anspruch nehmen.

Funktionsweise

Selektion

Eine Generation von Genetischen Algorithmen beginnt mit der Selektion. Als erstes werden aus der Anfangspopulation Elemente ausgewählt. Dabei wird der aus der Fitness Funktionen berechnete Fitnesswert bei der Auswahl berücksichtigt. Das bedeutet das Elemente mit einem größeren Fitness Wert eine größere Chance besitzen für die nächste Generation ausgewählt zu werden. Dabei können Elemente auch mehrfach ausgewählt werden. Diese sind dann die „stärksten“ Elemente und die „schwächsten“ Elemente haben eine geringe Chance und werden nicht für die Reproduktion zugelassen. So werden schlechte Lösungen nicht weiter verfolgt. Bis jetzt haben wir nur die guten und schlechten Fitness Werte behandelt. Die mittleren Lösungen werden nicht für die Reproduktion ausgewählt, sondern werden ohne Paarung in die nächste Generation übernommen. Dort erhalten Sie noch eine Chance sich fortzupflanzen und ihren Teil bei der Lösung beizutragen.

In der Tabelle 1 sieht man dass aus der Anfangspopulation nur zwei für die nächste Generation ausgewählt wurden. Das Element mit der Nummer 2 wurde, da ihr Fitness Wert sehr gut ist, ausgewählt um sich mit dem Element 4 zu paaren um so für die nächste Generation bessere Elemente zu bekommen.

<i>Elementnr.</i>	<i>Anfangspopulation</i>	<i>Fitness Wert</i>	<i>Wahrscheinlichkeit</i>	<i>Anzahl für die Reproduktion</i>
1	0 1 1 0 1	169	14,44%	0
2	1 1 0 0 0	576	49,23%	1
3	0 1 0 0 0	64	5,47%	0
4	1 0 0 1 1	361	30,85%	1

Tabelle 1: Auswahl der ersten Generation

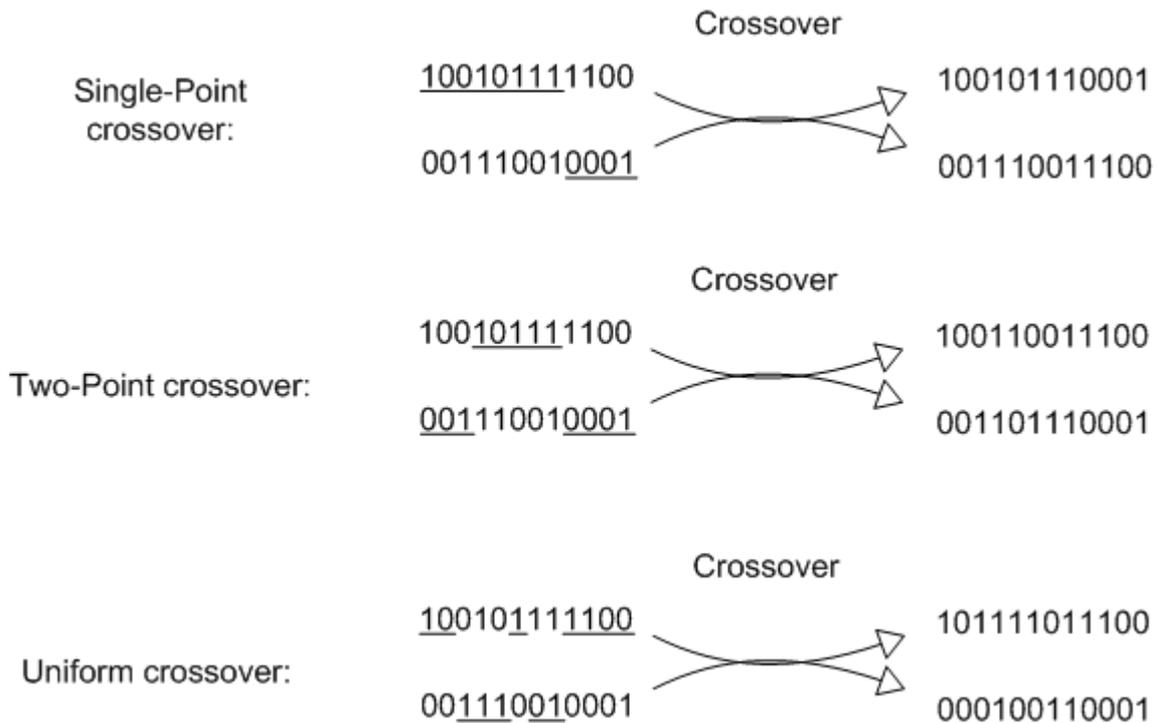
Crossover

Wenn wir nun einen aktiven Pool von paarungsfähigen Elementen zusammen haben können wir mit dem Crossover Verfahren beginnen. Das Verfahren besteht aus zwei Stufen. Als erstes werden aus dem Pool zufällig Elemente ausgewählt die sich paaren sollen.

Der zweite Schritt besteht daraus zufällig die Stelle auszuwählen wo sich die Elemente teilen und wieder zusammensetzen.

Algorithmische Anwendungen – Genetische Algorithmen

Es gibt für diesen 2ten Schritt folgende Möglichkeiten:



In folgender Tabelle 2 sieht man, wie die Individuen sich durch einen Single-Point crossover verändern.

<i>Elementnr.</i>	<i>Pool von paarungsfähigen Elementen</i>	<i>Paaren sich mit:</i>	<i>Crossover Stelle</i>	<i>Neue Population</i>
1	0 1 1 0 1	2	4	0 1 1 0 0
2	1 1 0 0 0	1	4	1 1 0 0 1
3	1 1 0 0 0	4	2	1 1 0 1 1
4	1 0 0 1 1	3	2	1 0 0 0 0

Tabelle 2: Neue Generation nach dem Crossover

Mutation

Die Mutation basiert auf einer bit-by-bit Basis. Da man bei der Mutation das Ergebnis sehr schnell verfälschen kann muss man die Wahrscheinlichkeit, das die Mutation auftritt, sehr gering halten. Dabei wird bei der Mutation nur ein Bit eines Elementes verändert. Die Stelle für die Mutation wird zufällig ausgewählt, so wird sichergestellt dass es eine zufällige Veränderung der Daten gibt und die Daten in einem veränderten Fluss bleiben und nicht stagnieren. Die Daten könnten sonst konvergieren da die Datenmenge durch den Selektionsdruck immer homogener werden.

Die Mutation kann auch als ein zusätzlicher Suchoperator eingesetzt werden, um den Suchvorgang zu beschleunigen. Mutationen können richtig eingesetzt zufällig und doch zielgerichtet Veränderungen bewirken. Da man immer nach Wegen sucht, um die Suche zu beschleunigen und zu verbessern sollte man die Mutation auch nutzen.

Codierung von Eigenschaften in Bit-Ketten:

Um einen evolutionsfähigen Algorithmus zu bekommen muss er so codiert sein das jede Änderungen seines Genotyps (seine Bits) Änderungen seines Phänotyps (seine Eigenschaften) bewirken und ihn nicht aktionsunfähig macht. Ein Individuum besteht bei den genetischen Algorithmen aus einer Reihe / Kette von Bits und jedes Bit steht für eine Eigenschaft, Aktion oder einem Programm-Impuls. Wenn man nun ein Bit ändert sollte sich auch das Verhalten des Individuums ändern.

Wenn man nun das Verhalten des Individuums binär codiert hat muss man auch darauf achten das die genetischen Operationen optimal abgestimmt sind. Sind die Codierung und die darauf operierenden genetischen Operationen nicht aufeinander abgestimmt kann der Algorithmus nicht funktionieren. Er würde im schlechtesten Fall versuchen sich ständig in eine schlechtere Richtung zu verbessern.

Ein großer Vorteil der binären Codierung ist es das die Informationen kompakt gespeichert werden und der Zugriff auf diese Informationen schneller von statten geht, als wenn man auf reellen oder natürlichen Zahlen zugreifen müsste. Durch die kompakte Speicherung können größere Populationen im Speicher untergebracht werden, wodurch die Verarbeitung beschleunigt wird.

Es gibt auch Nachteile bei der binären Codierung. Ein Nachteil ist die Positionsabhängigkeit der natürlichen Zahlen bei der Codierung. Dabei sollte folgendes Beispiel dieses Problem helfen zu verdeutlichen:

Die Binärcodierung für 1023 = 0111111111
 1029 = 1000000101.

Wenn jetzt der Wert 1025 das gesuchte Optimum wäre, bräuchte der genetische Algorithmus viele Durchgänge bis er dieses Ergebnis finden würde, da sich bei der Paarung viele Stellen ändern würden.

Um dieses Problem zu entschärfen, versucht man einschränkende Binärcodes wie den Gray-Code zu verwenden. Der Gray-Code ist eine Darstellungsform des Binärcodes und besteht darin, dass sich zwei benachbarte Gray-Zahlen nur in einem Bit unterscheiden dürfen. Durch diesen Umstand ist die Wahrscheinlichkeit, dass eine bessere Lösung schneller gefunden wird sehr viel höher, da bei einer Paarung sich nur wenige Einsen ändern würden.

Wie man an obigem Beispiel schon erkennen kann, gibt es eine Vielzahl von möglichen Darstellungsformen. Dies zeigt zwar zum einen die Flexibilität und die möglichen Anwendungsgebiete der genetischen Algorithmen, nur sollte dabei nicht vergessen werden, dass bei einer unbrauchbaren Darstellung und schlecht gewählten Funktionen (Selektion, Crossover, Mutation bzw. Fitnessfunktion) diese keine brauchbaren Ergebnisse liefern. Es kann durch eine sehr gute Auswahl oben genannter Elemente zudem sehr viel Rechenzeit verkürzt werden. Darum ist eine gut ausgewählte Codierung und der dazugehörigen Funktionen elementar wichtig.

Unser Beispiel mit einem Genetischen Algorithmus

Um die Vorgehensweise eines Genetischen Algorithmus an einem Beispiel zu zeigen haben wir uns überlegt ein Programm zu schreiben das das Maxima einer Funktion ermittelt. Für diese Problemstellung benötigt man natürlich keinen Genetischen Algorithmus, sondern man kann Sie auch mit einfachen mathematischen Verfahren lösen. Aber man kann sehr gut die Vorgehensweise der Genetischen Algorithmen darstellen.

Unsere erste Funktion ist die Gauß-Verteilung. Sie hat nur ein Maximum und wird nach folgender Formel erstellt.

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (\sigma > 0)$$

$$\mu = 0.5$$

$$\sigma = 0.1$$

Die zweite Funktion ist eine Funktion die 5 Maxima hat und diese auch noch gleich hoch sind. Dabei wird die Funktion mit folgender Formel erstellt.

$$f(x) = \sin^6(5\pi x)$$

Die dritte Funktion hat 4 Maxima und die unterschiedlichen Funktionswerte haben. Die Funktion wird mit folgender Formel erstellt.

$$f(x) = \sin^6(5\pi x) e^{-2\ln(2)\left(\frac{x-0.1}{0.8}\right)^2}$$

In unserem Beispiel werden die Kurven in dem Wertebereich von 0.0 – 1.0 dargestellt. Wir haben die Daten so codiert das man nur den Wert auf der X-Achse speichert und die Fitness durch den Y – Wert der jeweiligen Funktion dargestellt wird.

Algorithmische Anwendungen – Genetische Algorithmen

Bei unserem Programm hat man folgende Einstellmöglichkeiten. Zuerst kann die Funktion ausgewählt werden deren Maxima gesucht werden sollen. Bei der Population kann man angeben wie viele Individuen es geben soll die das Maximum suchen. Mit der Generation wird gesagt wie lange nach dem Maximum gesucht werden soll. Die Parameter Crossover und Mutation geben an wie oft die Individuen umgewandelt werden. In unserer Crossover Methode wird das beste Individuum immer in die nächste Generation übernommen.

Nach der Berechnung und nach jedem Schritt wird der Fitness Wert und die Position auf der X-Achse des besten Individuums und des schlechtesten Individuums angezeigt. Dazu wird noch der Mittelwert aus dem Fitness Wert angezeigt.

Durch den Knopf „INIT“ wird eine Population mit den angegebenen Parametern erzeugt. Durch „STEP“ wird die nächste Generation berechnet. Mit dem Knopf „START“ kann man auch alles berechnen lassen, so dass nur das Ergebnis angezeigt wird.

Die Kurve der ausgewählten Funktion wird im Plot dargestellt. Die Position der Individuen wird durch blaue Striche auf der Kurve sichtbar gemacht.

Im unteren Textfeld unter dem Plot werden bei jedem Step sowohl die Werte der einzelnen Individuen als auch deren Fitnesswert ausgegeben.

Fazit

Das Thema Genetische Algorithmen ist zugleich ein sehr interessantes Thema, welches zudem sehr viel Potenzial hat. Die vielfältigen Einsatzmöglichkeiten, also die Anwendung in nahezu jeglichen Bereichen hat uns sehr überrascht. Wobei auch die Flexibilität der Anpassung (Codierung) an fast jedes Problem möglich ist und im Bereich der Suchalgorithmen seines gleichen sucht. Dabei müssen bei der Implementierung noch nicht einmal alle Besonderheiten die evtl. auftreten könnten berücksichtigt werden, da der Evolutionsgedanke diese Probleme fast von alleine löst.

Wobei der Einstieg in dieses Thema nicht gerade einfach war und die sehr vielen Informationsquellen von unzähligen Büchern und Artikeln den Durchblick bzw. den Überblick sehr stark verzerren.

Literatur

- [1] *Genetic Algorithms in Search, Optimization, and Machine Learning*
David E. Goldberg
Addison-Wesley Publishing Company, INC. 1989
- [2] *Genetische Algorithmen*
John H. Holland
Spektrum der Wissenschaft, September 1992
- [3] *Mathematische Unterhaltungen*
Rick L. Riolo
Spektrum der Wissenschaft, Januar 1994
- [4] *Genetische Algorithmen*
http://www.evocomp.de/themen/genetische_algorithmen/genalg.html, 23.01.2006
- [5] *Entwicklung und systemtechnische Anwendung Evolutionärer Algorithmen*
Dissertation von Hartmut Pohlheim
http://www.pohlheim.com/diss/text/diss_pohlheim_ea-05.html#P140_34101, 24.01.2006
- [6] *The behavior of adaptive systems which employ genetic and correlation algorithms*
Bagley, J. D.
Doctoral dissertation, University of Michigan, 28(12), 5106B, University Microfilms No. 68-7556, 1967.
- [7] *Computer-aided gas pipeline operation using genetic algorithms and rule learning*
Goldberg, D. E.
Doctoral dissertation, University of Michigan, 44(10), 3174B, University Microfilms No. 8402282, 1983.
- [8] *Überblick über weitere Literatur*
<http://www.fachpublikationen.de/dokumente/01/19/01014.html>, 24.01.2006
- [9] *Überblick „Genetischer Algorithmus“ in Wikipedia*
http://de.wikipedia.org/wiki/Genetischer_Algorithmus, 24.01.2006