Genetic Algorithms, Genetic Programming

Roland Hasenberger

Allgemeines

Dieser Beitrag beschäftigt sich mit Genetischen Verfahren zur Erstellung von Programmen (Genetic Programming). Die Grundlage dafür bilden die Genetischen Verfahren zur Optimierung von Parametern (Genetic Algorithms), an Hand derer ich die Grundlagen der "informatischen" Genetik darlegen werde.

Evolutionäre Methoden

Evolutionäre Methoden verwenden Methoden der (natürlichen) Evolution, um Lösungen für gegebene Probleme zu finden ("Züchten von Lösungen").

Sie werden eingesetzt um

- große Suchräume mit minimalem Wissen zu durchsuchen und
- die natürliche Evolution zu erforschen

Während sie bei der Durchsuchung großer Suchräume nachgewiesenermaßen sehr effizient sind, bestehen (berechtigte) Zweifel, ob die Erforschung der natürlichen Evolution mit ihrer Hilfe möglich ist, da bei der Abbildung der natürlichen Evolution auf technische Methoden doch wesentliche Vereinfachungen durchgeführt wurden.

Die Evolution in diesem Sinne zeichnet sich dabei durch folgende Merkmale aus:

- Es existiert eine Population von Individuen mit unterschiedlichen Merkmalen.
- Die Unterschiede wirken sich auf die Fähigkeit, in der Umwelt zu überleben, aus.
- Individuen vermehren sich und geben ihre Merkmale an ihre Nachkommen weiter; es bilden sich aber auch zufällige Veränderungen (Mutationen).

Genetische Algorithmen (GA)

GA sind Verfahren zur Optimierung von Parametern bei Problemen mit bekannter Lösungsstruktur.

Das Äquivalent zum in der Natur vorkommenden Speicher der Erbinformation[1] sind hier Bitstreams[2], durch die auf Applikationsebene die Parameter der Problemlösung kodiert werden.

Population

Es wird hier von einer bestimmten Population[3] ausgegangen. Die Eigenschaften der ersten Generation werden üblicherweise per Zufallsgenerator erzeugt und danach werden die diversen genetischen Methoden angewendet. Übliche Populationen liegen in der Größenordnung von 100 Individuen[4], wobei einige 10 Generationen vergehen können, bis sich etwas sinnvolles gebildet hat.

survival of the fittest

Bei allen Evolutionären Methoden muß eine "fitness-Funktion" zur Verfügung stehen, welche die Qualität des Individuums bewertet[5]. Ausgehend von dieser "fitness-Funktion" werden die Individuen zur weiteren Verwendung herangezogen (als Elternteil oder direkt in die nächste Generation kopiert).

Grundsätzlich gibt es hier 2 Verfahren: Eines, bei dem in jeder Generation der/die schwächsten ausgeschieden werden und der Rest eben oft genug "gepaart" wird, bis sich die Größe der Population wieder eingestellt hat (steady state) oder aber die Auswahl jener, die sich paaren dürfen/kopiert werden.

2 mögliche Auswahlmethoden sind:

- Tournier
- Roulette

Turnier

Beim Turnier treten jeweils einzelne Exemplare[6] einer Population gegeneinander an und die Gewinner (d.h. jene mit der besten "fitness-Funktion") werden für die weitere Verwendung herangezogen.

Wesentlich hierbei ist, daß immer nur Teile der Population gegeneinander antreten, wodurch nicht der absolut beste gewinnt, sondern der beste der Teilpopulation. Dadurch ist gewährleistet, daß auch schwächere Individuen eine Überlebensschance haben. Das erscheint zwar auf den ersten Blick unsinnig, erhält aber Sinn, wenn beachtet wird, daß die jetzt schwächeren Individuen Erbinformationsteile besitzen, die später noch gebraucht werden können.

Roulette

Beim Roulette wird die gesamte Population herangezogen und es wird "unfaires" Roulette gespielt, d.h. die Wahrscheinlichkeit der Auswahl eines bestimmten Individuums ist in irgend einer Form proportional zu dessen fitness. Auch hier ist es wieder wesentlich, daß auch die schwächsten Mitglieder der Population eine Wahrscheinlichkeit ungleich 0 haben, daß sie überleben[7].

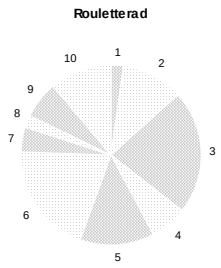

Bild 1 Rouletterad zu Population von Tabelle 2

In Bild 1 wird ein Beispiel für ein derartiges unfaires Rouletterad zur Population von Tabelle 2 angegeben.

Vermehrungs-/Veränderungsmechanismen

Mutation

Bei der biologischen Mutation ändern Basensequenzen einfach Ihren Platz oder Basen werden gegen andere Basen ausgetauscht.

Individuum	fitness
1	1
2	5
3	10
4	3
5	6
6	9
7	2
8	1
9	3
10	5

Tabelle 2 Population zum Rouletterad-Beispiel

Bei der informatischen Mutation werden entweder Bitsequenzen ausgetauscht oder einzelne Bits (zufällig) invertiert.

---

[1] In diesem Fall die DNA, kodiert mit 4 verschiedenen Basen (Adenin, Cytosin, Guanin, Thymin)

[2] Diese sind für die gesamte Vererbung wirklich als Bitstreams zu interpretieren, unabhängig davon, wie sie auf Applikationsebene interpretiert werden.

[3] Im Gegensatz zur Natur bleibt die Größe der Population, d.h. die Anzahl der Individuen konstant.

[4] Jedes mit unterschiedlichen Eigenschaften.

[5] Bei der natürlichen Evolution ist dies die Fähigkeit der Individuen zu überleben und einen Geschlechtspartner zu finden, um mit ihm Nachkommen zu zeugen.

[6] Die wiederum per Zufallsgenerator ausgewählt werden.

[7] Andernfalls würde das Verfahren gegen ein Hill-Climbing Verfahren (d.h. Gradienten-Verfahren) degenerieren, welches das Problem hat, daß es sehr stark dazu tendiert gegen ein lokales Optimum ("einen Maulwurfshügel") zu konvergieren anstatt gegen das globale Optimum ("den Mt. Everest").

Die Mutation ist wesentlich, da dadurch "zufällige" Sprünge im Parameterraum ausgeführt werden, die verhindern sollen und können, daß der Algorithmus gegen ein lokales Optimum konvergiert.

Rekombination

Die Rekombination ist das Konzept des geschlechtlichen Vermehrung; es werden hierbei Teile der Erbinformation von Mutter und Vater ausgetauscht. Dieser Vorgang wird in Bild 3 dargestellt, wobei hier eine Besonderheit der informatischen Vererbung zu Tage tritt: es werden in der Regel 2 Nachkommen gleichzeitig produziert.

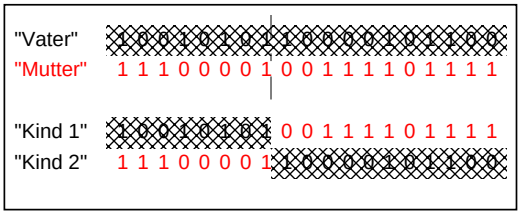

Bild 3: Vererbung

Durch den Vorgang der Vererbung werden Erbinformationen zwischen den "Partnern" ausgetauscht, womit die alten Eigenschaften kombiniert werden.

Funktionsweise der Genetischen Algorithmen

In Bild 3 ist der grundsätzliche Ablauf von Genetischen Algorithmen dargestellt. Die Schleife enthält als Abbruchbedingung auch eine Zeitbegrenzung, da im Gegensatz zur natürlichen Evolution informatische Populationen nie aussterben (es wird solange kopiert und rekombiniert, bis die ursprüngliche Populationsgröße wieder erreicht ist), es aber dennoch möglich ist, daß sich der Algorithmus "verrennt". Daher geht man davon aus, daß, wenn nach einer bestimmten Anzahl von Evolutionszyklen kein sinnvolles Ergebnis erzielt werden konnte, auch keines mehr gefunden wird und bricht die Suche ab.

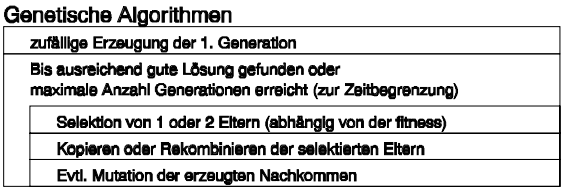

Bild 3: Ablauf von Genetischen Algorithmen

Anwendungen

GAs wurden bereits für verschiedenste Themen angewendet, darunter

- Optimierung der Düse für einen Raumgleiter[1].
- Konstruktion von Brücken, erdbebensicheren Gebäuden, billigen Betonschalen, geräuscharmen Ventilatoren
- Verbesserung von Ganzarmprothesen
- Ermittlung von Prüfabläufen für die automatische Software-Prüfung

Genetic Programming (GP)

Bei Genetischen Algorithmen werden Parameter optimiert. Daraus ergibt sich, daß die Lösungsstruktur bereits bekannt sein muß. Es liegt jetzt nahe, das Verfahren so zu erweitern, daß auch das Verfahren selbst auf genetische Weise gefunden wird.

In diesem Fall sind die einzelnen Individuen keine Bitstreams konstanter Länge mehr, welche die Parameter des (vorgegebenen) Lösungsalgorithmus kodieren sondern ganze Programme. Das Ergebnis ist in diesem Fall der Lösungsalgorithmus selbst.

Programme für Genetic Programming

Der erste Denkansatz, einfach ein z.B. C-Programm auf genetische Art und Weise erzeugen zu wollen, indem die Buchstaben des Sourcecodes mit evolutionären Methoden aneinandergereiht werden, wird recht bald Schiffbruch erleiden, da in diesem Fall der Großteil der so erzeugten Individuen nicht einmal compilierbar wäre.

Daraus ergibt sich, daß man für die Implementation von GP erst eine eigene Sprache erfinden mußte, die geeignet ist, der Rekombination und der Mutation unterworfen zu werden.

Dies ist erfolgt durch die Betrachtung von Programmen als Bäume, welche an den Knotenpunkten die Funktionen tragen und an den Endpunkten die Parameter bzw. Eingangsvariablen des Problems. Alle Funktionen erwarten dabei die selbe Type von Eingangsvariablen (z.B. Real) und liefern diese auch wieder. Derartige Programme sind jedenfalls (!) syntaktisch richtig und damit übersetzbar.

Die Rekombination und Mutation von derartigen Programmen ist damit der Austausch von beliebigen Teilbäumen, wie dies auch in Bild 3 dargestellt ist.

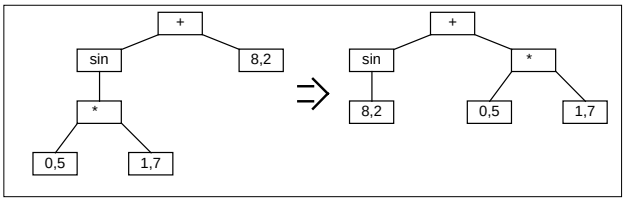

Bild 4: Mutation von GP-Programmen

Eigenschaften der Evolutionären Methoden

Mit diesen Methoden können sehr unübersichtlich wirkende Probleme mit vielen, vielleicht gegeneinander wirkenden Parametern/Eigenschaften gelöst werden. Je nachdem, ob eine Lösungsstruktur bekannt ist oder nicht kann GA oder GP eingesetzt werden.

Ein Problem stellt der Aufwand dieser Methoden dar: Bei einer Population von 500 und 50 Generationen (durchaus nicht so unüblich) muß die fitness von 25000 Individuen ermittelt werden. Wenn zur Ermittlung der fitness eines Individuums 1 Minute benötigt wird, bedeutet das einen Zeitbedarf von 17,36 Tagen allein dafür. Allerdings sind diese Verfahren auch sehr gut parallelisierbar. Die Ermittlung der fitness eines Individuums kann unabhängig von allen anderen erfolgen. Dadurch ist hier sogar eine Parallelisierung innerhalb eines LANs denkbar. Besonders bei GA erscheint hier die Parallelisierung sehr effizient durchführbar zu sein. Jeder der beteiligten Computer weiß die Lösungsstruktur, und es werden zur Bestimmung der fitness eines Individuums nur die neuen Parameter zu einem der Rechner geschickt. Dieser rechnet dann und liefert einen Wert (die fitness) an den Master (den für die Evolution zuständigen Rechner: "Gott") zurück. Die Kosten, die dabei für die Verteilung der Parameter entstehen, erscheinen denkbar gering im Vergleich zur Effizienzsteigerung durch die parallele Bestimmung der fitness. Bei GP müssen den Rechnern jeweils etwas mehr Daten übermittelt werden (immerhin das ganze zu untersuchende Programm); letztendlich sollte es sich aber trotzdem auszahlen.

Die evolutionären Verfahren sind auf dem Zufall basierende, d.h. es gibt eine jedenfalls nicht vernachlässigbare Wahrscheinlichkeit[2], daß keine oder eine nicht optimale Lösung (lokales Optimum) gefunden wird. Auch die oben angesprochene fitness-Funktion ist nicht immer ganz einfach zu konstruieren; vor allem kann man davon ausgehen daß alles, was man bei der fitness-Funktion nicht berücksichtigt hat, von den evolutionären Methoden (GP wie auch GA) in unverschämter Weise ausgenützt wird.

Genetic Programming

Aufgrund der Suche nach der Lösungsstruktur mit evolutionären Methoden bei Genetic Programming ist dafür nur sehr wenig Wissen über das Problem selbst erforderlich; es reicht wenn man eine fitness-Funktion zum Problem angeben kann.

Das Wissen über das Problem sammelt das Verfahren selbst während der Suche nach der Lösung. ➢

[1] In diesem Fall wurde mit GA übrigens eine Lösung gefunden, die ein Mensch nie angedacht hätte, die aber tatsächlich besser ist als die diversen konventionellen.

[2] Zumindestens für GP ist noch nicht bewiesen, daß damit immer die optimale Lösung gefunden wird; bei GA wurde dieser Beweis laut dem Vortrag von Fr. Dorothea Heiss bereits geführt. Außerdem besteht die Problematik daß auch dann, wenn eine gute Lösung gefunden wird, dies sehr (zu ?) lange dauern kann.

Evolutionsalgorithmus in C

Peter Speckmayer DSK-470: GENETIK.ZIP

Was ist ein Evolutionsalgorithmus?

Ein Evolutionsalgorithmus ist ein Computerprogramm, mit dem man Vorgänge, Variablen, usw. optimieren kann. Ein solcher Algorithmus hat den Vorteil, daß er im Vergleich zu anderen Optimierungsmethoden sehr schnell arbeitet und meistens sehr nahe an das Optimum herankommt. Der Nachteil ist, daß man mit ihm nicht feststellen kann, wie gut die gefundene Lösung wirklich ist.

Wie funktioniert ein (einfacher) Evolutionsalgorithmus?

Beim Programmieren eines Evolutionsalgorithmus nimmt man sich die Natur zum Vorbild. Die Optimierung erfolgt in Generationsschritten. Zu Beginn werden zufällige Wertegruppen (eine Generation) gebildet, von denen jede Wertegruppe ein ´Individuum´ charakterisiert (diese Wertegruppen werden im folgenden Text oft als GENe bezeichnet). Die Werte werden in eine sogenannte Fitness-Funktion (siehe auch Kapitel: Fitness) eingesetzt. Man erhält für jede Wertegruppe (für jedes GEN) einen Wert der die Qualität des jeweiligen GENs (Wertegruppe) angibt. Die Gruppen mit den schlechtesten Qualtiätswerten werden nun gelöscht und durch ´Kinder´ der mit guten Qualitätswerten ersetzt. Solche Kinder können auf verschiedenste Weise ermittelt werden, wobei man sich auch hier wieder die Natur als Vorbild nimmt. Die Kinder können beispielsweise durch Rekombination (crossing-over) ermittelt werden. Bei dieser Rekombination kreuzt man zwei ´Elternindividuen´ aus indem man einige Werte der Wertegruppen vom einen und die restlichen vom anderen Elternteil nimmt. Andere Möglichkeiten sind auch die Mutation, bei der die ´Kinder´ sich durch geringfügige zufällige Veränderungen von den Elternindividuen unterscheiden, oder eine Rekombination über mehr als zwei Eltern. Die neu ermittelte Generation wird nun wieder auf ihre Fitness getestet, sortiert und durch Auskreuzen oder/und Mutation verändert. Danach beginnt dieser Vorgang von vorne, bis eine gewünschte Fitness vorhanden ist. ➢

➢

Quellenverzeichnis

- Züchten von Computerprogrammen
  Dipl.-Ing. Dorothea Heiss
  Vortrag beim Jung-Elektrotechniker-Treffen des ÖVE vom 11.4.1994:
- Genetic Programming with C++
  Andy Singleton
  Byte Februar 1994
- Test and Evaluation by Genetic Algorithms
  Alan Cschultz, John J. Grefenstette, Kenneth A. De Jong
  IEEE Expert 8 (1993)

Zusätzliche Literatur zu GA und GP

- On the Programming of Computers by Means of Natural Selection
  John R. Koza (Stanford University)
  MIT Press, 1992
  ISBN 0-262-11170-5□

Beispiel

Als Beispiel werden hier geometrische Körper genommen die in eine vorgegebene Form passen sollen. Die geometrischen Formen werden durch die Parameter Form (Kreis, Quadrat, Dreieck) und Größe (klein, mittel, groß) bestimmt. Eine Anfangspopulation zu diesem Beispiel könnte folgendermaßen aussehen (siehe **Abbildung 1**).

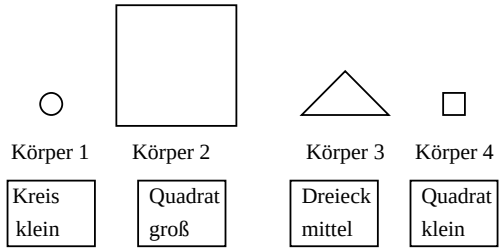

Abbildung 1: Beispiel einer Anfangspopulation

Diese Körper werden nun mittels einer Fitnessfunktion getestet und sortiert (siehe **Abbildung 2**). Die Fitness-Funktion des Beispieles gibt den Größenunterschied zwischen einem großen Kreis und den einzelnen Individuen (Körpern) an. Die Körper werden nach der Größe dieses Unterschiedes sortiert.

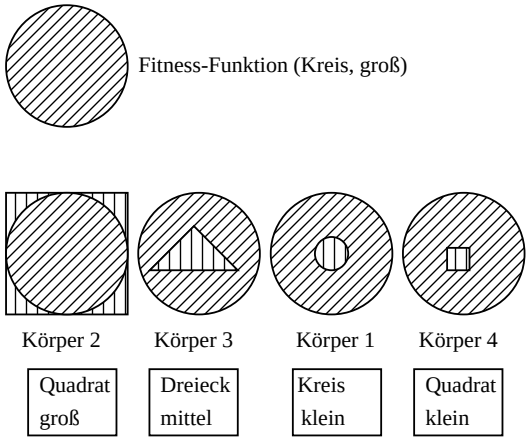

Abbildung 2: Auf Fitness testen

Die schlechtesten zwei Körper werden nun durch Kinder der zwei besten ersetzt. Wobei die Bildung der Kinder, wie schon erwähnt, durch Kreuzen erfolgt (**Abbildung 3** zeigt wie so ein Auskreuzen vor sich geht).

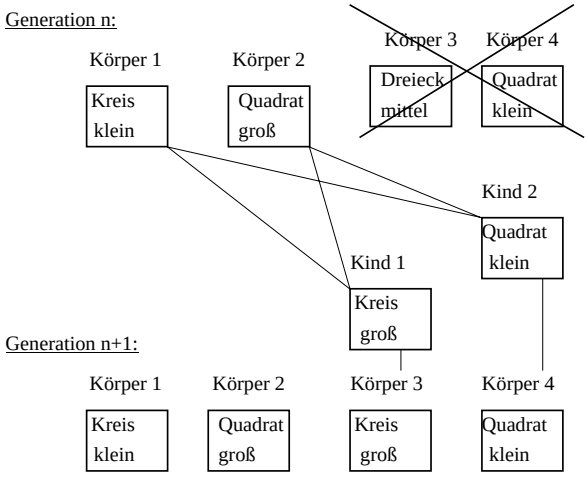

Abbildung 3: Bildung von Kindern durch Kreuzen zweier Eltern

Dieser Vorgang wird solange durchgeführt, bis ein gewünschtes Ergebnis erreicht ist oder eine gewisse Anzahl an Generationen berechnet worden sind (zur Zeitbegrenzung).