Genetic Programming

Norbert Bartos

Seit kurzer Zeit sind auf Tagungen des Fachbereiches der „Artificial Intelligence" vermehrt Beiträge aus dem Gebiet des „Genetic Programmings" zu finden. Dabei geht es um das automatische, evolutionäre Erstellen von Programmen durch Computer. Es ist dies eine Teildisziplin des „Machine Learning" und reicht in seinen Wurzeln bis 1958 zurück. Damals hat R.Friedberg einige Assemblerprogramme einer 1-Bit-Maschine erfolgreich evolutionär durch einen Computer entwickeln lassen. Die Ideen sind aber dann wieder fallengelassen worden, da für wirklich sinnvolle und damit aber komplexe Problemstellungen, die damaligen Computer viel zu langsam waren. Erst Mitte der 80er-Jahre, bedingt durch das vermehrte Aufkommen von (massiv) parallelen Supercomputern, begannen wieder ernstzunehmende Publikationen in wissenschaftlichen Zeitschriften, bis schließlich 1992 durch das Buch von J.R.Koza ein starker Interessensanstieg hervorgerufen wurde. Er zeigte nämlich, dass man auf diese Weise Programme generieren kann, die in manchen Fällen sogar besser, als die durch den Menschen erstellten Programme sind.

"Genetic Programming" basiert auf genetischen Algorithmen. Die dazu notwendigen Basiselemente sind

- Functions (+ | - | * | ...) und Terminals (Inputs, Outputs, Konstante),
- ein Programmstrukturraum variabler Länge (Codiervorschrift),
- genetische Operatoren (Mutation, Crossover),
- Fitness Function und Selection Function.

Programme können hierbei repräsentiert werden in

- linearer (textueller) Form: ältestes und heute kaum mehr verwendetes Prinzip;
- Baumform bzw. intern durch Listen: heute meist verwendet;
- Graphenform: allgemeinste Struktur; erlaubt Schleifen, Rekursionen, Speicher; komplex, daher selten verwendet und derzeit noch kaum erforscht.

Die folgenden einfachen Beispiele mögen das Evolutionsprinzip für LISP-Programme erläutern:

a) Crossover:

Gegeben seien folgende Eltern:

Elter 1: (* (+ a b) c)

Elter 2: (* (- d e) (+ f g))

Durch Crossover der kompatiblen Teillisten (+ a b) und (- d e) in den Eltern entstehen dann die folgenden Kinder:

Kind 1: (* (- d e) c)

Kind 2: (* (+ a b) (+ f g))

b) Mutation:

Gegeben sei der Elter (* (+ a b) c). Durch Mutation beispielsweise eines Operators entsteht dann das Kind (* (- a b) c).

Das Verhältnis von Crossover zu Mutation beträgt in der Praxis meist 9:1.

Probleme für die praktische Anwendung:

- Es entstehen hohe Rechenzeiten, da mit großen Populationen (typisch 500 bis 5000 und mehr Programmexemplare) gearbeitet wird, welche über viele Generationen (typisch 100 bis 1000) verändert werden.
- Es ist ein hoher Speicherbedarf vorhanden, da alle Exemplare der aktuellen Population (Eltern und Kinder) gespeichert werden müssen.
- Die Evaluation der Fitness-Funktion gestaltet sich ebenfalls recht aufwendig und daher langwierig.

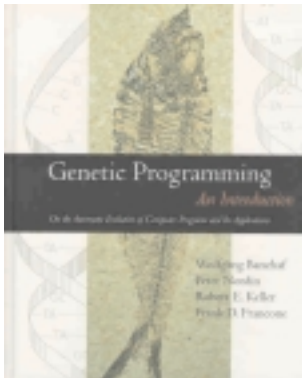
Es sind daher Parallelrechner zwingend erforderlich.

Die Anwendungen heute basieren meist auf der Sprache LISP, seltener auf anderen High-Level-Languages. Maschinencode-Evolution ist bei den modernen komplexen Prozessoren zu aufwendig. Die wichtigsten Anwendungsgebiete sind

- Autonome Roboter,
- Pattern Recognition,
- Image Processing.

Für interessierte Leserinnen und Leser ist das folgende Buch empfehlenswert:

Es ist dies ein umfassendes und sehr übersichtliches Werk, welches kein wesentliches Vorwissen im Bereich der evolutionären Techniken voraussetzt. Es ist auch für Anfänger in diesem Bereich leicht verständlich und zum Selbststudium geeignet.



Genetic Programming - An Introduction (On the Automatic Evolution of Computer Programs and its Applications) Wolfgang Banzhaf, Peter Nordin, Robert E. Keller, Frank D. Francone, 1998, Morgan Kaufmann Publishers / dpunkt.verlag, 470 Seiten; 715,-ATS, ISBN 3-920993-58-6

➢ Anfang stellen sollte. Zu beachten ist auch, dass die DOS-WIN3.1-Partition jedenfalls am Anfang (gleich hinter dem Boot-Manager) liegen sollte; weiter hinten kann DOS nicht daraus geladen werden. Für WIN95 trifft diese Einschränkung nicht zu und für OS/2 sowieso nicht, da OS/2 auf einem logischen Laufwerk in einer erweiterten Partition installierbar ist. Nur dadurch ist es überhaupt möglich, so viele Betriebssysteme auf einer Platte zu haben: die anderen müssen in Primärpartitions liegen, ebenso der Boot-Manager, und bei Vorhandensein einer erweiterten Partition können nur mehr drei Primärpartitions angelegt werden. In der erweiterten Partition kann man beliebig viele logische Laufwerke unterbringen, wenn's sein muss auch mit verschiedenen Versionen von OS/2 und einem Warp Server.

Nach Abschluss der Arbeit funktionierte der Computer tatsächlich in jeder Hinsicht so wie vorher, nur (dank der schnelleren Systemplatte) etwas schneller.

Ganz an Schluß habe ich das Image noch auf eine CD geschrieben, um den Wechseldatenträger wieder freizubekommen, das Image aber doch auch außerhalb der alten Festplatte aufzubewahren.

bartos@email.tgm.ac.at Norbert Bartos