Selbstevolution in einem Binärstringsystem

PETER DITTRICH[1] und WOLFGANG BANZHAF[2]

30. Mai 1995

- Abstract -

Weitere Stichworte: Selbstorganisation, präbiotische Evolution, molekulare Computer Selbstprogrammierung

Selbstevolution steht für ein Arbeitskonzept, das Populationssysteme beschreiben soll, in denen der Entwicklungsprozeß ausschließlich durch die beteiligten Spezies selbst bewirkt wird, und nicht durch externe, künstliche Operatoren, wie sie in klassischen Evolutionären Algorithmen (EA) zu finden sind.

In einem EA werden die Individuen als passive Träger der genotypischen Information behandelt. Variationen erfolgen durch explizite Mutations- und Rekombinationsoperatoren. Aus der Sicht der Population erfolgt eine Fremdorganisation durch diese Operatoren. Für ein Konzept der Selbstevolution ist es jedoch zwingend erforderlich, daß die Spezies aktiv als evolutionäre Maschinen agieren können. Dies wird zum Beispiel in [FONTANA 1991] mit Hilfe von CHURCHs Lambda-Kalkül realisiert. Ferner sind diesem Beitrag Arbeiten mit Systemen von Binärstrings der Längen $N = 4$ und $N = 9$ bits vorausgegangen [BANZHAF 1993; BANZHAF 1995].

Hier werden als Spezies Binärstrings von konstanter Länge $N = 32$ bits verwendet. Diese können miteinander wechselwirken und nach der Reaktiongleichung

$$s + s' \longrightarrow s + s' + s''$$

ein Produkt s'' bilden. Dazu wird s zu einem deterministischen endlichen Automaten transformiert, der s' als Eingabe erhält. Der Automat ist so konstruiert, daß er nach endlich vielen Schritten hält. Seine Ausgabe s'' ist ebenfalls ein Binärstring der Länge 32. Für die durchgeführten Simulationen wird eine Population von M zufälligen Binärstrings erzeugt und folgender Algorithmus iteriert:

1. Wähle zufällig zwei Strings s, s' der Population.
2. Falte s zu einem Automaten A_s.
3. Starte den Automaten A_s mit der Eingabe s'.
4. Falls seine Ausgabe s'' ungleich s und ungleich s' ist, ersetze einen zufälligen String der Population durch s''.

Zufallsereignisse werden hier nur für die Simulation eines homogenen, gut durchmischten Reaktionsbehälters genutzt, in dem potentiell jeder String mit jedem anderen wechselwirken kann. Variationen jedoch werden ausschließlich durch die Spezies in ihrer Maschinenform, und damit deterministisch, vorgenommen.

[1] Fachbereich Informatik, Universität Dortmund,
e-mail: dittrich@LS11.informatik.uni-dortmund.de
[2] Fachbereich Informatik, Universität Dortmund und Informatik Centrum Dortmund (ICD),
e-mail: banzhaf@LS11.informatik.uni-dortmund.de

Bei Simulationen sind Phänomene der Innovation, Selbstorganisation und Evolution beobachtbar. Es entwickeln sich nicht-triviale Organisationsstrukturen kooperierender Spezies. Die Abbildung 1 zeigt für einen typischen Lauf einen Ausschnitt der evolutiven Phase. Man erkennt, daß laufend Spezies neu entstehen und aussterben. Auch ist eine Entwicklung in der Struktur der Spezies vorhanden. Der Entwicklungsprozess hat den Charakter einer präbiotischen Evolution, was vermutlich auf die räumlich homogene Struktur und gute Durchmischung der Population zurückzuführen ist.

Indem die Spezies sich selbst verändern, modifizieren sie auch die Art und Weise, wie diese Veränderung im weiteren Verlauf erfolgen wird. Besonders deutlich ist dies bei der spontanen Entstehung eines Crossover-Mechanismus geworden. Dabei haben sich Spezies gebildet, die in der Lage sind, einen Teil ihres Binärcodes in den Binärcode anderer Spezies einzupflanzen.

Der für die Wechselwirkung definierte Automat besteht aus Elementen gängiger Computer. Er läßt sich leicht als Gatterschaltung realisieren. Es liegt deshalb die Vermutung nahe, daß nicht nur der hier verwendete Automat, sondern auch reale Rechnersysteme eine "natürliche" intrinsische Fähigkeit zur Selbstorganisation bzw. "Selbstprogrammierung" besitzen.

Der obige Algorithmus realisiert ein Modell eines chemischen Reaktors. Kürzlich erschienene Arbeiten haben gezeigt, wie Reaktionssysteme zur Informationsverarbeitung und -speicherung eingesetzt werden können [Hjelmfelt, Weinberger, und Ross 1991; Adleman 1994; Dittrich und Banzhaf 1995]. Eine Integration der hier gezeigten Entwicklungsfähigkeit in derartige Ansätze erscheint vielversprechend.

Literatur

Adleman, L. M. (1994). Molecular Computation of Solutions to Combinatorial Problems. *Science 266*, 1021–1024.

Banzhaf, W. (1993). Self-replicating sequences of binary numbers – Foundations I and II: General and Strings of Length N = 4. *Biological Cybernetics 69*, 269–281.

Banzhaf, W. (1995). Self-Organizing Algorithms Derived from RNA Interaction. In M. Banzhaf und F. H. Eeckman (Eds.), *Evolution and Biocomputation: Computational Models of Evolution*, S. 9–102. Springer, Berlin-Heidelberg-New York.

Dittrich, P. und W. Banzhaf (1995). The Chemical Computation Metaphor and Robot Control. ECAL 95, Posterbeitrag.

Fontana, W. (1991). Algorithmic Chemistry. In C. G. Langton und et al. (Eds.), *Artificial Life II: Proceedings of the Second Artificial Life Workshop*, S. 159–209. Addison-Wesley, Reading MA.

Hjelmfelt, A., E. D. Weinberger, und J. Ross (1991). Chemical implementation of neural networks and Turing machines. *Proc. Natl. Acad. Sci. USA 88*, 10983–10987.

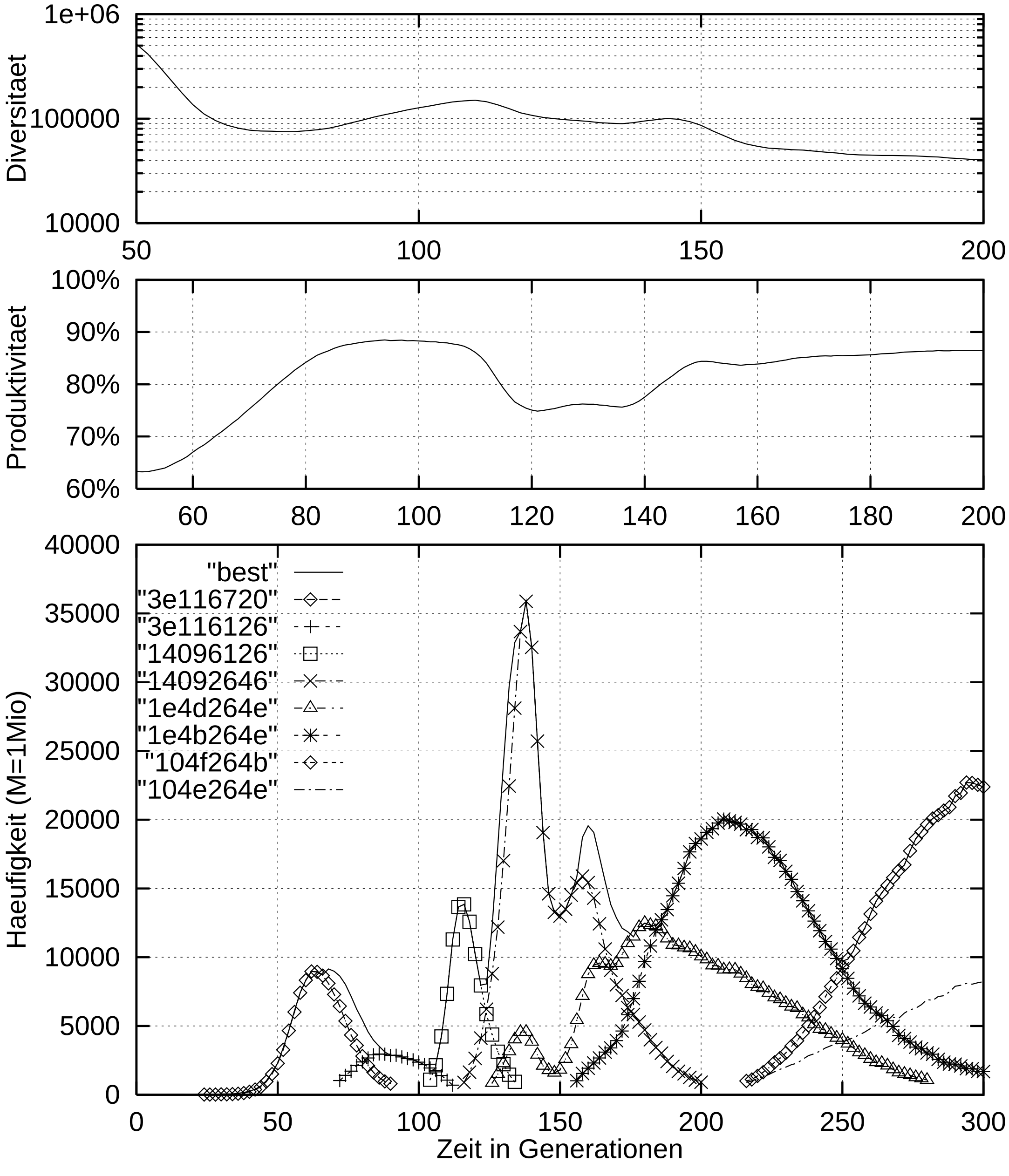

Abb. 1: *Evolutive Phase eines Laufs mit einer Populationsgröße von $M = 10^6$ Die Diversität gibt die absolute Anzahl unterschiedlicher Spezies an. Die Produktivität ist der Anteil der reaktiven Iterationen einer Generation. Die dargestellten Häufigkeitsverläufe einzelner Spezies stehen stellvertretend für ganze Gruppen von Binärstrings, die auftauchen und wieder verschwinden. Eine Generation entspricht $M = 10^6$ Iterationen.*