Evolutionäre Algorithmen

Karsten Weicker
Institut für Informatik, Abteilung Formale Konzepte
Universität Stuttgart, Breitwiesenstr. 20–22, 70565 Stuttgart
Karsten.Weicker@informatik.uni-stuttgart.de

1 Einführung

Erste Ansätze und Ideen der evolutionären Algorithmen (Abk.: EA, engl.: *evolutionary computation*) gehen bereits auf Arbeiten in den späten 50er und in den 60er Jahren zurück (z.B. Friedman, 1956; Box, 1957; Fraser, 1957; Friedberg, 1958; Friedberg, Dunham & North, 1959; Rechenberg, 1964; Fogel, Owens & Walsh, 1965, 1966; Bremermann, Rogson & Salaff, 1966; Crosby, 1967; Bossert, 1967; Kaufman, 1967; Fraser, 1968; Holland, 1969). Während in diesen frühen Jahren die Motivationen für die Betrachtungen ausgesprochen unterschiedlich waren, bildeten sich in den 70er und 80er Jahren unterschiedliche Schulen der evolutionären Algorithmen mit einem stärkeren Focus auf der Optimierung heraus. Hierbei handelt es sich um die genetischen Algorithmen (engl. *genetic algorithms*), die Evolutionsstrategien (engl. *evolution strategies*), das evolutionäre Programmieren (engl. *evolutionary programming*) und in der jüngeren Zeit das genetische Programmieren (engl. *genetic programming*). Im Jahre 1991 haben sich Vertreter aller verschiedenen Richtungen auf den Überbegriff *evolutionary computation* geeinigt. Seither kann auch eine starke Konvergenz und Zusammenarbeit der unterschiedlichen Richtungen beobachtet werden.

In den ersten Jahren der Forschung auf diesem Gebiet waren die gestellten Aufgaben meist eher von akademischem Interesse und haben erst in den vergangenen Jahren begonnen, großflächig ihren Weg in die praktische Anwendung in Industrie und Wirtschaft zu finden.

Dieser Artikel beschreibt im folgenden kurz die wesentlichen Prinzipien der evolutionären Algorithmen und vergleicht sie mit dem biologischen Vorbild (Abschnitt 2). In Abschnitt 3 werden die verschiedenen Instanzen der evolutionären Algorithmen vorgestellt und in Abschnitt 4 in Zusammenhang zum gesamten Bereich des Softcomputing gesetzt. Der Abschnitt 5 diskutiert theoretische Beschränkungen, die sich für die evolutionären Algorithmen ergeben, bevor auf die neuen Anforderungen eingegangen wird, die sich durch den Übergang auf industrielle Probleme ergeben (Abschnitt 6). Abschnitt 7 schließt diesen Artikel mit einer kurzen Diskussion der Arbeitsgebiete in unserer Arbeitsgruppe ab.

2 Von der Biologie zur Optimierung

Bei der biologischen Evolution findet ein Anpassungsprozess einzelner Arten an die Umweltbedingungen statt. Dabei verändert sich der genetische Code, die Erbsubstanz, der Individuen einer Population von einer Generation zur nächsten. Diese Erbsubstanz beinhaltet alle wesentlichen Informationen zu Aufbau, Organisation, Funktionalität und Erscheinungsbild des Individuums.

Die wichtigsten treibenden Faktoren in der biologischen Evolution sind hierbei

- die Mutation, durch welche bei der Teilung einer Zelle kleine Fehler bei der Replikation des genetischen Codes vorkommen, und

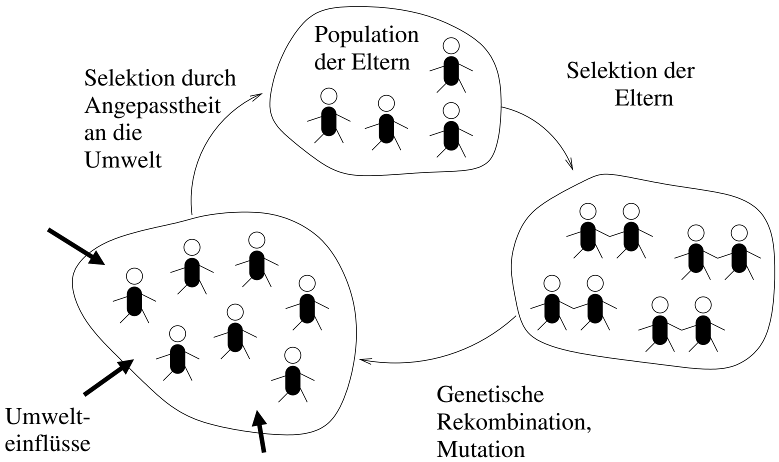

Abbildung 1: Evolution in der Biologie (vereinfachte Darstellung)

- die Selektion, bei welcher sich aufgrund einer unterschiedlichen Angepasstheit an die Umwelt die Genfrequenz ändert und sich tauglichere, besser angepasste Individuen gegenüber den anderen Individuen der Population durchsetzen.

Durch die Mutation werden Neuerungen in das vorhandene genetische Material einer Population eingeführt, während die Selektion zwischen sinnvollen und unnützen Neuerungen unterscheidet, wobei erstere sich durchsetzen können und zweitere wieder fallengelassen werden. Dieses Wechselspiel erzeugt eine hochgradige Angepasstheit an die Umweltbedingungen, welche als Fitness bezeichnet wird. Diese Fitness kann jedoch nicht direkt erfasst werden, sondern nur indirekt über die Anzahl der Nachkommen (unter der Annahme, dass sich die besser angepassten stärker vermehren werden).

Neben der Mutation und der Selektion geht auch noch die Rekombination in die Evolution mit ein, die an die sexuelle Fortpflanzung geknüpft ist. Hierbei handelt es sich um eine neue Kombination des väterlichen und mütterlichen Erbguts. Die Rekombination wird allgemein nicht als wesentlicher Evolutionsfaktor betrachtet, da es sich lediglich um ein erneutes Kombinieren bereits vorhandener Information – es kann, laut der Lehrmeinung, durch die Rekombination keine neue Information in den Genpool eingebracht werden, und so ist eine Evolution durch reine Rekombination auch nur bis zu einem gewissen Grad möglich, bis nämlich die beste Kombination der vorhandenen Informationen gefunden wurde.

Der resultierende biologische Zyklus der Evolution ist in Abbildung 1 veranschaulicht. Dabei wird deutlich, dass die Selektion an zwei verschiedenen Punkten greift:

- der Überlebenschance in der Umwelt durch unterschiedliche Lebensfähigkeit oder Behauptungsvermögen und

- der Fähigkeit, einen Geschlechtspartner zu finden.

Abschließend soll zur biologischen Evolution nochmals betont werden, dass es sich hierbei um keinen Optimierungsvorgang handelt, bei dem ein globales Optimum als Endziel angestrebt wird. Es kann vielmehr als Adaptationsprozess aufgefasst werden, der sich den ständig sich verändernden Umweltverhältnissen anpasst.

Dieser biologische Vorgang dient als Vorbild für die evolutionären Algorithmen, und so findet sich auch der evolutionäre Zyklus in den evolutionären Algorithmen wieder (Abbildung 2). Allerdings steht nun nicht mehr die Anpassung im Vordergrund, sondern ein konkretes Optimierungs-

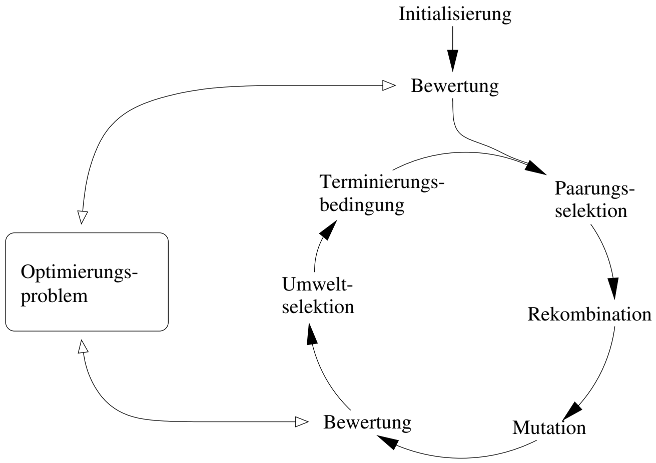

Abbildung 2: Optimierung mit evolutionären Algorithmen

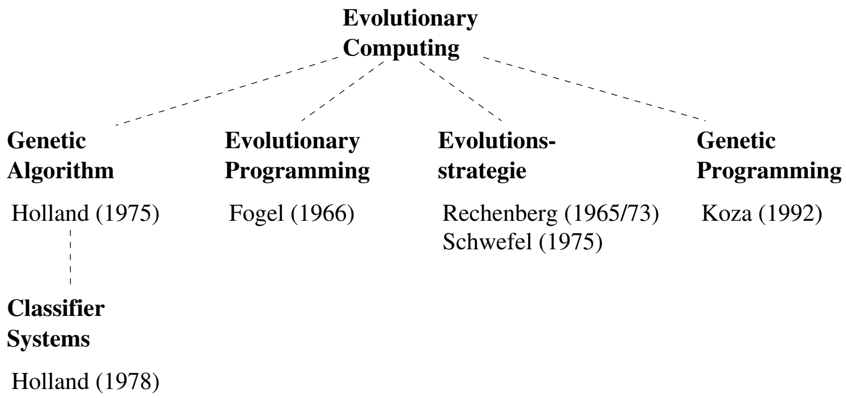

Abbildung 3: Instanzen der evolutionären Algorithmen

problem. Hierbei kann die Güte eines Individuums konkret gemessen werden und wird als Fitness bezeichnet. Ein solcher Fitnesswert wird dann bei der Selektion herangezogen, um die besseren Individuen mit größerer Wahrscheinlichkeit in die nächste Generation zu übernehmen oder häufiger als Eltern heranzuziehen. Bei den meisten evolutionären Algorithmen wird jedoch nur einer der beiden Selektionsmechanismen benutzt, um die Suche zu steuern.

3 Instanzen der evolutionären Algorithmen

In diesem Abschnitt werden kurz die verschiedenen Ausprägungen der evolutionären Algorithmen vorgestellt, die in Abbildung 3 schematisch aufgezeigt werden.

Genetische Algorithmen (Holland, 1969, 1973, 1992) im klassischen Sinn arbeiten auf einer binären Darstellung des Problemraums. Der Selektionsdruck wird hierbei über eine stochastische, fitnessproportionale Selektion der Eltern aus der Vorgängerpopulation erzeugt. Als Operatoren dienen ein Mutationsoperator, welcher einzelne Bits im Individuum invertiert, und ein Rekombinationsoperator, welcher die Chromosomen der Eltern vermischt (ein sog. Crossover-Operator). In der Theorie der genetischen Algorithmen wird dabei dem Crossover-Operator durch das Schema-Theorem besonders viel Bedeutung beigemessen, welches die starke Verbreitung und Kombination vorteilhafter Informationen in der Population postuliert. Eine Unterform der genetischen Algorithmen bilden die Classifier Systeme (Holland & Reitman, 1978), bei welchen in der binären Repräsentation Regeln zur Klassifikation oder Steuerung eines Systems erlernt werden.

Evolutionsstrategien (Rechenberg, 1973, 1994; Schwefel, 1981) arbeiten auf einer reellwertigen Darstellung des Problemraums. Die Eltern werden zufällig ausgewählt, und der Selektionsdruck entsteht ausschließlich bei der Umweltselektion, welche deterministisch nur die besten Individuen übernimmt. Bei der klassischen Evolutionsstrategie wurde lediglich die Mutation benutzt, wobei hier additiv gemäß einer Normalverteilung Änderungen an den reellwertigen Elementen eines Individuums vorgenommen werden. Sehr bald hat sich hier jedoch auch die Verwendung von Rekombinationsoperatoren sowie der Einsatz von selbstadaptiven Steuerungen der Parameter der Mutation abgezeichnet.

Evolutionäres Programmieren (Fogel et al., 1966) beschränken sich nicht auf eine bestimmte Repräsentation, sondern arbeiten auf der natürlichen Darstellung des Problems. Evolutionäres Programmieren wurde hierbei zunächst für die Entwicklung von endlichen Automaten zur Zeitserienvorhersage eingesetzt. Charakteristische Merkmale dieses Verfahrens sind ein Fehlen des Rekombinationsoperators sowie eine spezielle Turnier-Selektion zur Erzeugung der nächsten Generation. Unabhängig von den Evolutionsstrategien wurden auch beim evolutionären Programmieren selbstadaptive Steuerungen der Mutationsparameter entwickelt.

Genetisches Programmieren (Koza, 1989, 1992) arbeitet auf einer dynamischen Darstellung, wie z.B. Baumstrukturen, und wird daher vornehmlich für die evolutionäre Entwicklung von Computerprogrammen u.ä. eingesetzt.

Neben diesen strikt getrennten klassischen Formen haben sich inzwischen eine große Anzahl von Mischformen gebildet.

4 Platz im Softcomputing

Die traditionelle Datenverarbeitung arbeitet mit harten Rechenregeln und wird dementsprechend auch manchmal als Hardcomputing bezeichnet. Demgegenüber stehen die Softcomputing-Techniken, die auf der Verarbeitung von unscharfem Wissen, Unexaktheiten und nicht klar definierten Begriffen beruhen.

Während sowohl bei den Fuzzy-Methoden, die mit Ungenauigkeiten und Unschärfe arbeiten, als auch die neuronalen Netze, die Lern- und Anpassungsvorgänge simulieren, u.a. ihren Schwerpunkt auf einer Modellierung haben, dienen die evolutionären Algorithmen in erster Linie zur Suche und Optimierung.

Allerdings umfasst Softcomputing weit mehr als die einzelnen Teildisziplinen – so sind insbesondere die Verknüpfungen zwischen den Disziplinen von Interesse. Diese werden für die evolutionären Algorithmen im folgenden kurz dargestellt.

Neuronale Netze: Evolutionäre Algorithmen können bei den neuronalen Netzen zur Optimierung unterschiedlicher Aspekte eingesetzt werden, so z.B. als Alternative zu den klassischen Lernverfahren zur Gewichtsoptimierung, wie z.B. Backpropagation, aber auch zur Topologieoptimierung (Schaffer, Whitley & Eshelman, 1993; Yao, 1995; Balakrishnan & Honavar, 1995;

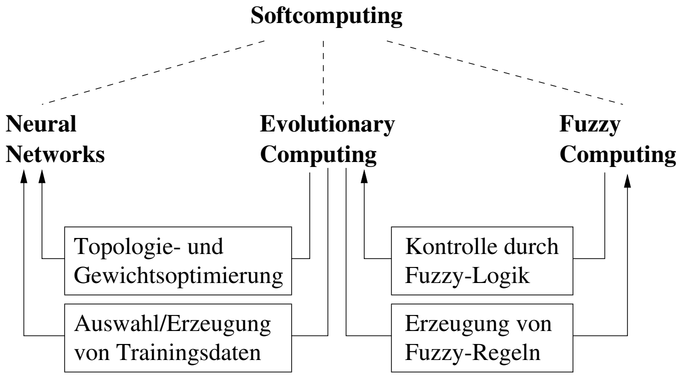

Abbildung 4: Platz der evolutionären Algorithmen im Softcomputing

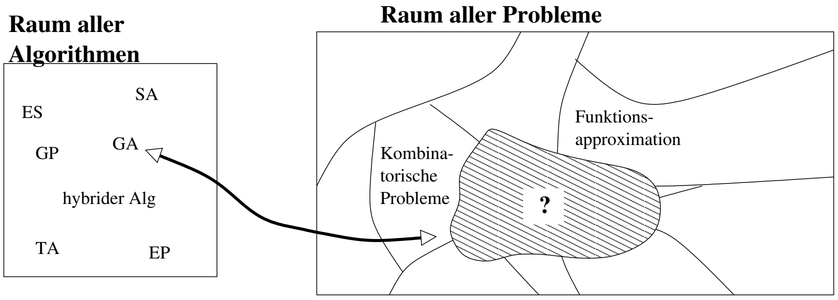

Abbildung 5: Spezielle Anwendungsgebiete eines Algorithmus

Porto, 1997) oder der Erzeugung oder Gewichtung von Trainingsdaten (Mayer, 1998; Reeves & Taylor, 1998)

Fuzzy-Methoden: Auch hier können Fuzzy-Regeln mit evolutionären Algorithmen erzeugt werden bzw. die Struktur und die Parameter eines Fuzzy-Modells angepaßt werden. Eine andere Möglichkeit der Zusammenarbeit stellt die Kontrolle der Parameter eines evolutionären Algorithmus durch Fuzzy-Regeln dar. (Karr, 1997; Bersini, 1998)

5 Universelle Optimierer

Wenn man die Ursprünge und Motivationen der evolutionären Algorithmen betrachtet, stellt man einen gewissen Glauben an die Erschaffung eines universellen Optimierers fest. Dies zeigt sich z.B. bei den genetischen Algorithmen durch die binäre Kodierung und damit einhergehend der Verneinung einer Übernahme jeglichen Problemwissens. Dies zeigt sich auch in den Versuchen vieler Forschungsarbeiten, durch experimentelle Analyse verschiedener Benchmark-Probleme die Überlegenheit eines Algorithmus über einen anderen zu beweisen.

Dieser Glaube wurde durch das „No Free Lunch“-Theorem (Wolpert & Macready, 1997) erschüttert, in welchem in einer wahrscheinlichkeitstheoretischen Analyse bewiesen wurde, dass gemittelt über alle möglichen Probleme alle Algorithmen gleich gut bzw. gleich schlecht sind. Dies gilt insbesondere auch für den Algorithmus, der lediglich jedesmal eine rein zufällig gezogene Lösung

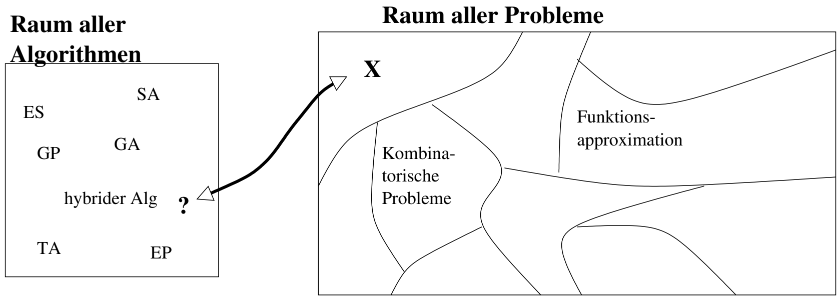

Abbildung 6: Anpassung eines Algorithmus auf ein Problem

aus dem Suchraum betrachtet. Aus diesem Resultat und der Überlegenheit eines Algorithmus über einen anderen bezüglich eines speziellen Problems folgt direkt, dass es Nischen im Raum der Probleme gibt, in denen diese verschiedenen Algorithmen jeweils überlegen sind.

Dies führt zu zwei praktischen Konsequenzen:

- Einerseits stellt sich die Frage, was genau die Nische für jeden einzelnen Algorithmus ist (vergleiche Abbildung 5). Allerdings fehlen trotz jahrelanger Forschungsaktivitäten noch eindeutige Charakteristika für die Anwendbarkeit eines speziellen Algorithmus.
- Andererseits ergibt sich daraus die Problematik, für ein festes Problem den passenden Algorithmus zu finden (Abbildung 6). Dies führt zur Anpassung des Algorithmus an das Problem und die Inkorporation von Problemwissen, womit sich nahezu alle Anwender beschäftigen. Dann spricht man auch von hybriden evolutionären Algorithmen, die lokale Suchalgorithmen oder Heuristiken im evolutionären Algorithmus mitbenutzen.

6 Anforderungen in industriellen Anwendungen

In den vergangenen Jahren haben evolutionäre Algorithmen Einzug in die Lösung zahlreicher praxisrelevanter Probleme gehalten. Allerdings gelten gerade für industrielle Anwendungen neben dem im vorigen Abschnitt diskutierten, ernüchternden Resultat, dass für jedes Problem der Algorithmus anzupassen ist, oft noch weitere Anforderungen, die eine Optimierung erschweren. Diese können weit über die in der Forschung berücksichtigten Benchmark-Probleme hinausgehen. Verschiedene solche Anforderungen werden im folgenden kurz skizziert.

Explosion in den Dimensionen: Eine starke Zunahme der Suchraumdimensionen führt nicht nur zu einer wesentlich vergrößerten Anzahl der lokalen Optima, sondern die Konvergenzgeschwindigkeit nimmt auch unverhältnismäßig stark ab. Dem kann z.T. mit einer Aufsplittung in Teilprobleme begegnet werden, solange keine starken gegenseitigen Abhängigkeiten in den Suchraumdimensionen vorliegen.(z.B. Potter, 1997; Valenzuela & Jones, 1994)

Randbedingungen: Eine große Anzahl an Randbedingungen kann den Effekt haben, dass die Suche komplizierter wird, da eine Abwägung zwischen der Erfüllung der Randbedingungen und der Optimierung der Fitnessfunktion während des Optimierungsprozesses sich schwierig gestaltet. Ein Standardlösungsansatz sind Strafterme in der Fitnessfunktion bei verletzten Randbedingungen. Diese Vorgehensweise liefert jedoch oft nur unzureichende Resultate und ist überhaupt nicht einsetzbar, wenn die Fitnessfunktion bei verletzten Randbedingungen nicht mehr berechenbar ist. (Michalewicz, 1995; Michalewicz & Schoenauer, 1996)

Mehrzieloptimierung: Bei industriellen Problemen wird meist nicht nur ein, sondern mehrere Zielattribute zur Bewertung einer Lösung herangezogen, die sich oft auch widersprechen können. Eine A-priori-Gewichtung der Attribute in der Fitnessfunktion führt nur bei gutmütigen Problemen zu einer sinnvollen Lösung, nämlich dem Kompromiß zwischen den verschiedenen Zielattributen. Andere Verfahren, die verschiedene Alternativlösungen berechnen, sind oft nur effizient für nicht mehr als zwei Zielattribute anwendbar. (Fonseca & Fleming, 1995, 1997; Horn, 1997)

Verrauschte Fitnessfunktionen: Aufgrund von Messfehlern oder stochastischen Simulationen kann es zu verrauschten Zielwerten kommen. Dem kann durch Mehrfachbewertung einer Lösung mit anschließender Mittelung begegnet werden. Dies führt allerdings zu deutlich höheren Kosten und ist insbesondere bei teuren experimentellen Bewertungen nicht machbar. In den vergangenen Jahren wurden erste Strategien entwickelt, die die notwendigen Mehrfachbewertungen minimieren sollen. Dasselbe Problem kann auch auftreten, wenn die Eingangsgrößen eines technischen Systems einzuregeln sind und nur über Messsensoren erfasst werden können. Dann sollte eine Lösung des Problems stabil gegen kleine Variationen in den Eingangsgrößen sein (vergleiche Abbildung 7).(Aizawa & Wah, 1994; Nissen & Propach, 1998; Branke, 1998)

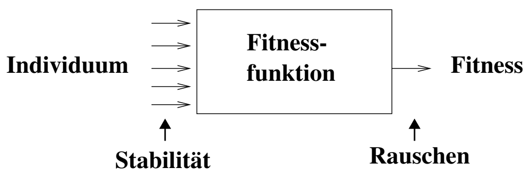

Abbildung 7: Rauschen bei der Bewertung einer Lösung

Dynamische Fitnessfunktionen: Probleme können sich während einer Optimierung dynamisch verändern. Das Auffinden eines Optimums in einer solchen Umgebung ist ungleich schwieriger als bei statischen Problemen. Ebenso müssen wandernde Optima durch den Suchalgorithmus nachvollzogen und neue bessere Optima entdeckt werden. Die Forschung hat sich bei diesen Problemen insbesondere auf Verfahren zum Erhalt der Diversität in den Populationen (und damit der Adaptivität), reaktive Verfahren, die auf Veränderungen in der Umwelt reagieren, und auf Verfahren mit einem Erinnerungsvermögen für oszillierende Probleme konzentriert. (Grefenstette, 1992; Lewis, Hart & Ritchie, 1998; Branke, 1999; Weicker & Weicker, 1999a)

Obwohl für alle Problemfelder bereits Forschungsergebnisse vorliegen, gibt es noch keine geeigneten Standardtechniken. Eine starke Zunahme an Arbeiten auf diesen Gebieten auf den Fachkonferenzen der letzten Jahre zeigt, mit welchem Druck an der Lösung dieser Aufgaben gearbeitet wird.

7 Arbeitsgebiete in der Abteilung Formale Konzepte

In der Abteilung „Formale Konzepte" arbeiten Nicole und Karsten Weicker auf dem Gebiet der evolutionären Algorithmen. Dabei konzentriert sich unsere Arbeit im wesentlichen auf drei Teilgebiete: die Erarbeitung theoretischer Grundlagen, die Optimierung zeit-abhängiger Probleme und die Untersuchung von koevolutionären Ideen.

Theoretische Grundlagen: Bei evolutionären Algorithmen handelt es sich um hochgradig komplexe stochastische Vorgänge. Hierfür sind nur in sehr geringem Umfang und Ausmaß theoretische Aussagen über die Wirkungsweise der Algorithmen und ihre Erfolgswahrscheinlichkeit möglich. Es gibt grundsätzlich zwei verschiedene Ansätze für solche theoretischen Untersuchungen: eine Modellierung des gesamten Suchprozesses (makroskopische Sicht) und eine Betrachtung der lokalen Vorgänge (mikroskopische Sicht). Teilweise kann von mikroskopischen Analysen auf den gesamten Suchprozess zurückgeschlossen werden. Dennoch sind in der Regel Vereinfachungen notwendig, so werden oft nur sehr einfache Beispiele untersucht und genau analysiert. In unserer Arbeit gehen wir einen anderen Weg, indem Voraussetzungen über das lokale Verhalten von Algorithmen oder Suchoperatoren formuliert werden, aus denen sich dann Eigenschaften für die gesamte Suche ableiten lassen. So konnten Resultate im Stil der „No Free Lunch"-Theoreme gezeigt werden, die eine theoretischen Rechtfertigung für adaptive Suchverfahren liefern. (Weicker & Weicker, 1998, 1999)

Dynamische Optimierung: Im Rahmen der dynamischen Fitnessfunktionen sind für uns insbesondere die Fragen von Interesse, wie sich die Qualität von Verfahren auf solchen Problemen messen läßt, wie sich Unterklassen der dynamischen Probleme bilden lassen, deren Optimierung ähnliche Eigenschaften aufweist, sowie die Frage, wie bestimmte Regelmäßigkeiten der Dynamik vom Optimierer genützt werden können. (Weicker & Weicker, 1999a)

Koevolutionäre Verfahren: Evolutionäre Algorithmen, die lediglich auf den biologischen Prinzipien der Mutation, Selektion und Rekombination basieren, geraten für viele Probleme an ihre Grenzen. Daher ist ein anderer Schwerpunkt unserer Arbeit die Frage, inwieweit sich die Betrachtung von anderen evolutionsrelevanten Vorgängen, wie z.B. der Koevolution in der Form von Parasit-Wirt-Beziehungen und symbiotischen Interaktionen, für eine Optimierung von komplexen oder dynamischen Problemen nutzen lassen.(Weicker & Weicker, 1999b)

Genom: In einem mittlerweile abgeschlossenen Projekt zweier Projektgruppen wurden die Konzepte für ein Prototyping-System für evolutionäre Algorithmen entwickelt, in welchem Kodierung und Algorithmus vom Benutzer völlig frei definiert werden können. Eine erste Implementation wurde in der funktionalen Programmiersprache SML abgeschlossen. Allerdings ist in der derzeitigen Implementation aus laufzeiteffizienten Gründen lediglich die Bearbeitung von sehr kleinen Problemen möglich. Eine effiziente Reimplementierung ist aus personellen Gründen derzeit nicht möglich. (Amos et al., 1995; Jung & Weicker, 1995; Großmann, Leonhardi & Schmidt, 1997; Leonhardi, 1997; Leonhardi, Reissenberger, Schmelmer, Weicker & Weicker, 1998; Großmann, 1999)

8 Standardliteratur

Einführung/Übersicht:

- Einführung in evolutionäre Algorithmen (Nissen, 1997)
- Genetic Algorithms + Data Structures = Evolution Programs (Michalewicz, 1992)
- Evolutionsstrategie '94 (Rechenberg, 1994)
- Genetische Algorithmen (Heistermann, 1994)
- An Introduction to Genetic Algorithms (Mitchell, 1996)
- Genetische Algorithmen und Evolutionsstrategien (Schöneburg, Heinzmann & Feddersen, 1996)
- Genetic Programming - An Introduction (Banzhaf, Nordin, Keller & Francone, 1998)
- Numerical optimization of Computer models (Schwefel, 1981)
- Evolutionary Algorithms in Theory and Practice (Bäck, 1996)
- Local Search in Combinatorial Optimization (Aarts & Lenstra, 1997)

- Wissensbasierte Genetische Algorithmen: Integration von Genetischen Algorithmen und Constraint-Programmierung zur Lösung kombinatorischer Optimierungsprobleme (Bruns, 1996)

Nachschlagwerke:

- Handbook of Genetic Algorithms (Davis, 1991)
- Handbook of Evolutionary Computation (Bäck, Fogel & Michalewicz, 1997)
- Genetic Algorithms and Engineering Design (Gen & Cheng, 1997)

Konferenzbände:

- Proc. of the Second Int. Conf. on Genetic Algorithms (Grefenstette, 1987)
- Proc. of the Third Int. Conf. on Genetic Algorithms (Schaffer, 1989)
- Proc. of the Fourth Int. Conf. on Genetic Algorithms (Belew & Booker, 1991)
- Proc. of the Fifth Int. Conf. on Genetic Algorithms (Forrest, 1993)
- Proc. of the Sixth Int. Conf. on Genetic Algorithms (Eshelman, 1995)
- Proc. of the Seventh Int. Conf. on Genetic Algorithms (Bäck, 1997)
- Foundations of Genetic Algorithms (Rawlins, 1991)
- Foundations of Genetic Algorithms 2 (Whitley, 1993)
- Foundations of Genetic Algorithms 3 (Whitley, Mathias & Pyeatt, 1995)
- Foundations of Genetic Algorithms 4 (Belew & Vose, 1997)
- Foundations of Genetic Algorithms (FOGA-5) Preliminary version of the Proc. (Banzhaf & Reeves, 1998)
- Parallel problem solving from nature: 1st Workshop; PPSN I (Schwefel & Männer, 1991)
- Parallel problem solving from nature 2 (Männer, 1992)
- Parallel Problem Solving from Nature - PPSN III (Davidor, Schwefel & Männer, 1994)
- Parallel problem solving from nature - PPSN IV (Voigt, 1996)
- Parallel Problem Solving from Nature - PPSN V (Eiben, Bäck, Schoenauer & Schwefel, 1998)
- Evolutionary Programming VI (Angeline, Reynolds, McDonnell & Eberhart, 1997)
- Evolutionary Programming VII (Porto, Saravanan, Waagen & Eiben, 1998)

Literatur

Aarts, E., & Lenstra, J. K. (Eds.). (1997). *Local search in combinatorial optimization.* Chichester, UK: John Wiley and Sons.

Aizawa, A. N., & Wah, B. W. (1994). Scheduling of genetic algorithms in a noisy environment. *Evolutionary Computation*, *2*(2), 97–122.

Amos, F., Jung, K., Kawetzki, B., Kuhn, W., Pertler, O., Reißing, R. & Schaal, M. (1995). *Abschlußbericht der Projektgruppe Genetische Algorithmen* [Final report of the project group genetic algorithms] (Tech. Rep. No. FK 1/95). Stuttgart, Germany: Dept. Formal Concepts, Faculty of Computer Science, University of Stuttgart. (german)

Angeline, P. J., Reynolds, R. G., McDonnell, J. R. & Eberhart, R. (Eds.). (1997). *Evolutionary Programming VI.* Berlin: Springer. (Lecture Notes in Computer Science 1213)

Bäck, T. (1996). *Evolutionary algorithms in theory and practice.* New York: Oxford University Press.

Bäck, T. (Ed.). (1997). *Proc. of the Seventh Int. Conf. on Genetic Algorithms.* San Francisco, CA: Morgan Kaufmann.

Bäck, T., Fogel, D. B. & Michalewicz, Z. (Eds.). (1997). *Handbook of Evolutionary Computation.* Bristol, New York: Institute of Physics Publishing and Oxford University Press.

Balakrishnan, K., & Honavar, V. (1995). *Evolutionary design of neural architectures* (Tech. Rep. No. CS TR 95-01). Iowa State University, Department of Computer Science.

Banzhaf, W., Nordin, P., Keller, R. & Francone, F. (Eds.). (1998). *Genetic programming - an introduction.* San Francisco, CA: Morgan Kaufmann.

Banzhaf, W., & Reeves, C. (Eds.). (1998). *Foundations of Genetic Algorithms (FOGA-5).* (Preliminary version of the proceedings)

Belew, R. K., & Booker, L. B. (Eds.). (1991). *Proc. of the Fourth Int. Conf. on Genetic Algorithms.* San Mateo, CA: Morgan Kaufmann.

Belew, R. K., & Vose, M. D. (Eds.). (1997). *Foundations of Genetic Algorithms 4.* San Francisco, CA: Morgan Kaufmann.

Bersini, H. (1998). Fuzzy-evolutionary systems. In E. H. Ruspini, P. P. Bonissone & W. Pedrycz (Eds.), *Handbook of Fuzzy Computation* (pp. D3.1:1–D3.6:2). Bristol: Institute of Physics Publishing.

Bossert, W. (1967). Mathematical optimization: Are there abstract limits on natural selection? In P. S. Moorehead & M. M. Kaplan (Eds.), *Mathematical challenges to the neo-darwinian interpretation of evolution* (pp. 35–46). Philadelphia, PA: The Wistar Institute Press.

Box, G. E. P. (1957). Evolutionary operation: A method for increasing industrial productivity. *Applied Statistics*, *6*(2), 81–101.

Branke, J. (1998). Creating robust solutions by means of evolutionary algorithms. In A. E. Eiben, T. Bäck, M. Schoenauer & H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature – PPSN V* (pp. 119–128). Berlin: Springer. (Lecture Notes in Computer Science 1498)

Branke, J. (1999). Memory enhanced evolutionary algorithms for changing optimization problems. In *1999 Congress on Evolutionary Computation* (pp. 1875–1882). Piscataway, NJ: IEEE Service Center.

Bremermann, H. J., Rogson, M. & Salaff, S. (1966). Global properties of evolution processes. In H. H. Pattee, E. A. Edlsack, L. Fein & A. B. Callahan (Eds.), *Natural Automata and Useful Simulations* (pp. 3–41). Washington D.C.: Spartan Books.

Bruns, R. (1996). *Wissensbasierte Genetische Algorithmen: Integration von Genetischen Algorithmen und Constraint-Programmierung zur Lösung kombinatorischer Optimierungsprobleme.* Sankt Augustin: Infix Verlag.

Crosby, J. L. (1967). Computers in the study of evolution. *Science Progress Oxford*, *55*, 279–292.

Davidor, Y., Schwefel, H.-P. & Männer, R. (Eds.). (1994). *Parallel Problem Solving from Nature – PPSN III.* Berlin: Springer. (Lecture Notes in Computer Science 866)

Davis, L. (Ed.). (1991). *Handbook of genetic algorithms.* New York: Van Nostrand Reinhold.

Eiben, A. E., Bäck, T., Schoenauer, M. & Schwefel, H.-P. (Eds.). (1998). *Parallel Problem Solving from Nature – PPSN V.* Berlin: Springer. (Lecture Notes in Computer Science 1498)

Eshelman, L. J. (Ed.). (1995). *Proc. of the Sixth Int. Conf. on Genetic Algorithms.* San Francisco, CA: Morgan Kaufmann.

Fogel, L. J., Owens, A. J. & Walsh, M. J. (1965). Artificial intelligence through a simulation of evolution. In M. Maxfield, A. Callahan & L. J. Fogel (Eds.), *Biophysics and cybernetic systems: Proc. of the 2nd cybernetic sciences symposium* (pp. 131–155). Washington, D.C.: Spartan Books.

Fogel, L. J., Owens, A. J. & Walsh, M. J. (1966). *Artificial intelligence through simulated evolution.* New York: John Wiley and Sons.

Fonseca, C. M., & Fleming, P. J. (1995). An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, *3*(1), 1–16.

Fonseca, C. M., & Fleming, P. J. (1997). Multiobjective optimization. In T. Bäck, D. B. Fogel & Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation* (pp. C4.5:1–9). Bristol, New York: Institute of Physics Publishing and Oxford University Press.

Forrest, S. (Ed.). (1993). *Proc. of the Fifth Int. Conf. on Genetic Algorithms.* San Mateo, CA: Morgan Kaufmann.

Fraser, A. S. (1957). Simulation of genetic systems by automtic digital computers i. introduction. *Australian Journal of Biological Sciences*, *10*, 484–491.

Fraser, A. S. (1968). The evolution of purposive behaviour. In H. von Foerster, J. D. White, L. J. Peterson & J. K. Russell (Eds.), *Purposive Systems* (pp. 15–23). Washington D.C.: Spartan Books.

Friedberg, R. M. (1958). A learning machine: Part I. *IBM Journal of Research and Development*, *2*(1), 2–13.

Friedberg, R. M., Dunham, B. & North, J. H. (1959). A learning machine: Part II. *IBM Journal of Research and Development*, *3*(3), 282–287.

Friedman, G. J. (1956). *Selective feedback computers for engineering synthesis and nervous system analogy.* Masters-/Diplomarbeit, University of California, Los Angeles.

Gen, M., & Cheng, R. (1997). *Genetic algorithms and engineering design.* New York: John Wiley and sons.

Grefenstette, J. J. (Ed.). (1987). *Genetic algorithms and their applications : Proc. of the second Int. Conf. on Genetic Algorithms.* Hillsdale, NJ: Lawrence Erlbaum Assoc.

Grefenstette, J. J. (1992). Genetic algorithms for changing environments. In R. Männer & B. Manderick (Eds.), *Parallel Problem Solving from Nature 2 (Proc. 2nd Int. Conf. on Parallel Problem Solving from Nature, Brussels 1992)* (pp. 137–144). Amsterdam: Elsevier.

Großmann, M. (1999). *Anwendung evolutionärer Verfahren auf multiobjektive Optimierungsprobleme.* Masters-/Diplomarbeit, Institut für Informatik, University of Stuttgart, Stuttgart, Germany. (German)

Großmann, M., Leonhardi, A. & Schmidt, T. (1997). *Abschlußbericht der projektgruppe evolutionäre algorithmen* [Final report of the project group evolutionary algorithms] (Tech. Rep. No. FK1997/02). Stuttgart, Germany: Dept. Formal Concepts, Faculty of Computer Science, University of Stuttgart. (german)

Heistermann, J. (1994). *Genetische Algorithmen* [Genetic algorithms]. Stuttgart: Teubner. (german)

Holland, J. H. (1969). A new kind of turnpike theorem. *Bulletin of the American Mathematical Society*, *75*(6), 1311–1317.

Holland, J. H. (1973). Genetic algorithms and the optimal allocation of trials. *SIAM Journal on Computing*, *2*(2), 88–105.

Holland, J. H. (1992). *Adaptation in natural and artifical systems.* Cambridge, MA: MIT Press.

Holland, J. H., & Reitman, J. S. (1978). Cognitive systems based on adaptive algorithms. In D. A. Waterman & F. Hayes-Roth (Eds.), *Pattern–Directed Inference Systems* (pp. 313–329). New York: Academic Press.

Horn, J. (1997). Multicriterion decision making. In T. Bäck, D. B. Fogel & Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation* (pp. F1.9:1–15). Bristol, New York: Institute of Physics Publishing and Oxford University Press.

Jung, K., & Weicker, N. (1995). *Funktionale Spezifikation des Software-Tools EAGLE* [Functional specification of the software tool EAGLE] (Tech. Rep. No. FK 1/95). Stuttgart, Germany: Dept. Formal Concepts, Faculty of Computer Science, University of Stuttgart. (german)

Karr, C. L. (1997). Fuzzy-evolutionary systems. In T. Bäck, D. B. Fogel & Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation* (pp. D2.1:1–2:9). Bristol, New York: Institute of Physics Publishing and Oxford University Press.

Kaufman, H. (1967). An expirmental investigation of process identification by competitive evolution. *IEEE Trans. on Systems Science and Cybernetics*, *SSC-3*(1), 11–16.

Koza, J. R. (1989). Hierarchical genetic algorithms operating on populations of computer programs. In N. S. Sridharan (Ed.), *Proc. of the 11th Joint Conf. on Genetic Algorithms* (pp. 786–774). San Francisco, CA: Morgan Kaufmann Publishers.

Koza, J. R. (1992). *Genetic programming: on the programming of computers by means of natural selection.* Cambridge, MA: MIT Press.

Leonhardi, A. (1997). *Eine Beschreibungssprache für Evolutionäre Algorithmen* [A description language for evolutionary algorithms]. Masters-/Diplomarbeit, Institute of Computer Science, University of Stuttgart, Stuttgart, Germany. (german)

Leonhardi, A., Reissenberger, W., Schmelmer, T., Weicker, K. & Weicker, N. (1998). Development of problem-specific evolutionary algorithms. In A. E. Eiben, T. Bäck, M. Schoenauer & H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature – PPSN V* (pp. 388–397). Berlin: Springer. (Lecture Notes in Computer Science 1498)

Lewis, J., Hart, E. & Ritchie, G. (1998). A comparison of dominance mechanisms and simple mutation on non-stationary problems. In A. E. Eiben, T. Bäck, M. Schoenauer & H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature – PPSN V* (pp. 139–148). Berlin: Springer. (Lecture Notes in Computer Science 1498)

Männer, R. (Ed.). (1992). *Parallel problem solving from nature 2.* Amsterdam: North-Holland.

Mayer, H. A. (1998). Symbiotic coevolution of artificial neural networks and training data sets. In A. E. Eiben, T. Bäck, M. Schoenauer & H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature – PPSN V* (pp. 511–520). Berlin: Springer. (Lecture Notes in Computer Science 1498)

Michalewicz, Z. (1992). *Genetic Algorithms + Data Structures = Evolution Programs.* Berlin: Springer.

Michalewicz, Z. (1995). Heuristic methods for evolutionary computation techniques. *Journal of Heuristics*, *1*(2), 177–206.

Michalewicz, Z., & Schoenauer, M. (1996). Evolutionary algorithms for constrained parameter optimization problems. *Evolutionary Computation*, *4*(1), 1–32.

Mitchell, M. (1996). *An Introduction to Genetic Algorithms.* Cambridge, MA: MIT Press.

Nissen, V. (1997). *Einführung in evolutionäre algorithmen.* Braunschweig: Vieweg.

Nissen, V., & Propach, J. (1998). On the robustness of population-based versus point-based optimization in the presence of noise. *IEEE Transactions on Evolutionary Computation*, *2*(3), 107–119.

Porto, V. W. (1997). Neural-evolutionary systems. In T. Bäck, D. B. Fogel & Z. Michalewicz (Eds.), *Handbook of Evolutionary Computation* (pp. D1.1:1–3:2). Bristol, New York: Institute of Physics Publishing and Oxford University Press.

Porto, V. W., Saravanan, N., Waagen, D. & Eiben, A. E. (Eds.). (1998). *Evolutionary Programming VII.* Berlin: Springer. (Lecture Notes in Computer Science 1447)

Potter, M. A. (1997). *The design and analysis of a computational model of cooperative coevolution.* Doktorarbeit, George Mason University.

Rawlins, G. J. (Ed.). (1991). *Foundations of genetic algorithms.* San Mateo, CA: Morgan Kaufmann.

Rechenberg, I. (1964). *Kybernetische Lösungsansteuerung einer experimentellen Forschungsaufgabe* [Cybernetic solution path of an experimental problem]. presented at the Annual Conference of the WGLR at Berlin in September 1964.

Rechenberg, I. (1973). *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Stuttgart: frommann-holzbog. (German)

Rechenberg, I. (1994). *Evolutionsstrategie '94.* Stuttgart: frommann-holzbog. (German)

Reeves, C. R., & Taylor, S. J. (1998). Selection of training data for neural networks by a genetic algorithm. In A. E. Eiben, T. Bäck, M. Schoenauer & H.-P. Schwefel (Eds.), *Parallel Problem Solving from Nature – PPSN V* (pp. 633–642). Berlin: Springer. (Lecture Notes in Computer Science 1498)

Schaffer, J. D. (Ed.). (1989). *Proc. of the Third Int. Conf. on Genetic Algorithms.* San Mateo, CA: Morgan Kaufmann.

Schaffer, J. D., Whitley, D. & Eshelman, L. J. (1993). Combinations of genetic algorithms and neural networks: a survey of the state of the art. In *COGANN-92, Int. Workshop on Combinations of Genetic Algorithms and Neural Networks* (pp. 1–37). ?

Schöneburg, E., Heinzmann, F. & Feddersen, S. (1996). *Genetische Algorithmen und Evolutionsstrategien* [Genetic algorithms and evolution strategies]. Bonn, Germany: Addison-Wesley. (german)

Schwefel, H.-P. (1981). *Numerical optimization of computer models.* Chichester: John Wiley & Sons, Ltd.

Schwefel, H.-P., & Männer, R. (Eds.). (1991). *Parallel problem solving from nature: 1st Workshop, PPSN I.* Berlin: Springer.

Valenzuela, C. L., & Jones, A. J. (1994). Evolutionary divide and conquer (I): A novel genetic approach to the TSP. *Evolutionary Computation*, *1*(4), 313–333.

Voigt, H.-M. (Ed.). (1996). *Parallel problem solving from nature - PPSN IV.* Berlin: Springer.

Weicker, K., & Weicker, N. (1998). Locality vs. randomness – dependence of operator quality on the search state. In *Foundations of Genetic Algorithms (FOGA-5). Preliminary Version of the Proceedings* (pp. 289–308). Leiden.

Weicker, K., & Weicker, N. (1999a). On evolution strategy optimization in dynamic environments. In *1999 Congress on Evolutionary Computation* (pp. 2039–2046). Piscataway, NJ: IEEE Service Center.

Weicker, K., & Weicker, N. (1999b). On the improvement of coevolutionary optimizers by learning variable interdependencies. In *1999 Congress on Evolutionary Computation* (pp. 1627–1632). Piscataway, NJ: IEEE Service Center.

Weicker, N., & Weicker, K. (1999). Algorithms' local potential – breackfast included? In *1999 Congress on Evolutionary Computation* (pp. 1392–1397). Piscataway, NJ: IEEE Service Center.

Whitley, D., Mathias, K. & Pyeatt, L. (1995). Hyperplane ranking in simple genetic algorithms. In L. J. Eshelman (Ed.), *Proc. of the Sixth Int. Conf. on Genetic Algorithms* (pp. 231–238). San Francisco, CA: Morgan Kaufmann.

Whitley, L. D. (Ed.). (1993). *Foundations of genetic algorithms 2.* San Mateo, CA: Morgan Kaufmann.

Wolpert, D. H., & Macready, W. G. (1997). No free lunch theorems for optimization. *IEEE Trans. on Evolutionary Computation*, *1*(1), 67–82.

Yao, X. (1995). Evolutionary artificial neural networks. In A. Kent & J. G. Williams (Eds.), *Encyclopedia of Computer Science and Technology* (Vol. 33, pp. 137–170). Marcel Dekker Inc.