Eckhart Arnold

Neuheit in evolutionären Algorithmen

Einleitung

Eines der Merkmale, welches die Evolution in der Natur so verblüffend erscheinen lässt, ist die Tatsache, dass sie Pflanzen und Tiere hervorbringt, die durch ihren außerordentlich hohen Grad von Zweckangepasstheit den Eindruck erwecken, als wären sie von einem intelligenten Wesen erfunden worden. Dass die Philosophen, bevor in Form der Evolutionstheorie eine wissenschaftliche Erklärung für dieses verblüffende Phänomen zur Verfügung stand, darin gerne einen Hinweis auf die Existenz eines Schöpfergottes sehen wollten, ist daher nicht einmal unverständlich. Zugleich verweist die aus heutiger Sicht naive Vorstellung vom intelligenten Schöpfergott auf eine der zentralen philosophischen Fragestellungen im Zusammenhang mit der Evolutionstheorie: Wie ist es möglich, dass ein evolutionärer Prozess, dessen Fortgang durch die Wirkung dreier simpler Komponenten, nämlich die *Reproduktion*, *Variation* und *Selektion* von Genen, vorangetrieben wird, genuin kreative Leistungen hervorbringt? Oder kurz: Wie können evolutionäre Prozesse Neuheit schaffen?

Denkt man ein wenig über diese Frage nach, dann stellt man rasch fest, dass mit dem bloßen Hinweis auf die drei „Darwinschen Module“: Reproduktion, Variation und Selektion bestenfalls der grobe Rahmen für eine Antwort gegeben wird. Denn sofort stellen sich weitere Fragen: Entsteht überall dort, wo die Bedingungen von Reproduktion, Variation und Selektion gegeben sind ein „schöpferischer“ evolutionärer Prozess? Führen evolutionäre Prozesse notwendig zur Entstehung von Neuheit, zu einer *Höher*entwicklung (was immer man im Einzelfall darunter verstehen mag)? Wieviel Zeit benötigen evolutionäre Prozesse? Welchen Grenzen unterliegt die Schöpferkraft der Evolution?

Es gibt unterschiedliche Herangehensweisen, um auf diese Fragen eine Antwort zu geben. Zum Beispiel könnte man die theoretische und empirische Literatur zur Evolution in den damit befassten Sachwissenschaften untersuchen, also vor allem der Biologie, wo man das reichhaltigste Material finden wird, aber auch der Kulturwissenschaften, denn einige Autoren glau-

ben, dass ganz analoge Entwicklungsmuster auch in der „kulturellen Evolution“ wieder zu finden sind.[1]

Eine ganz andere Herangehensweise, die in diesem Kapitel näher beleuchtet werden soll, besteht darin, evolutionäre Prozesse nicht aus einer empirisch-wissenschaftlichen, sondern aus einer technischen Perspektive heraus zu verstehen und evolutionäre Prozesse als Algorithmen aufzufassen, die bestimmte Probleme lösen sollen, besonders solche Probleme, bei denen wir noch gar keine genaue Vorstellung davon haben, wie die Lösung auszusehen hat. In der Tat setzt man im Ingenieurswesen schon seit einer Reihe von Jahren evolutionäre Algorithmen erfolgreich ein, etwa bei der Flugroutenoptimierung[2] oder im Maschinenbau[3]. Parallel dazu untersucht die theoretische Informatik die Bedingungen, nach denen evolutionäre Algorithmen funktionieren und die Gründe, weshalb sie oft erstaunlich gute Ergebnisse liefern[4]. Es liegt also nahe, den Ansatz der evolutionären Algorithmen einmal genauer daraufhin zu untersuchen, ob er Aufschluss über die Frage geben kann, wie in evolutionären Prozessen Neuheit entstehen kann.

Evolutionäre Algorithmen

Grundlagen evolutionärer Algorithmen

Fasst man den Prozess der Evolution algorithmisch auf, dann stellt man sich nicht mehr unbedingt Lebewesen oder Gene vor, die dem Prozess der Evolution unterliegen, sondern man spricht abstrakt von Punkten in einem *Suchraum*. Die Punkte im Suchraum entsprechen den Genen, genauer den Genotypen von Lebewesen in der Natur. Gesucht werden nun diejenigen

[1] Vgl. Gerhard Schurz: Natürliche und kulturelle Evolution: Skizze einer verallgemeinerten Evolutionstheorie, in: Wolfgang Wickler / Lucie Salwiczek (Hrsg.): Wie wir die Welt erkennen. Erkenntnisweisen im interdisziplinären Diskurs. Freiburg / München 2001, S. 329-376. – Kritisch zu diesem Unterfangen allerdings: John Dupré: Human Nature and the Limits of Science, Oxford 2001.

[2] Vgl. Ingrid Gerdes / Frank Klawonn / Rudolf Kruse: Evolutionäre Algorithmen, Wiesbaden 2004, S. 110ff.

[3] Vgl. Jürgen Willms: Evolutionäre Algorithmen: Chancen für die praxisorientierte Optimierung, Meschede, URL: www.fh-meschede.de/public/willms/ea/index.html (Zugriff: 15.12.2007).

[4] Vgl. Gerdes et al., a.a.O., S. 47-62. – Vgl. Thomas Jansen: Evolutionäre Algorithmen, Vorlesungssrkipt WS 2004/05, URL: http://ls2-www.cs.uni-dortmund.de/lehre/winter200405/evo-alg/#Skript (Zugriff: 15.12.2007).

Punkte des Suchraums, die eine bestimmte, über dem Suchraum definierte *Bewertungsfunktion* maximieren. Die Bewertungsfunktion gibt gewissermaßen den Grad der Angepasstheit bzw. der Fitness wieder. Man darf die Angepasstheit nicht mit dem Selektionserfolg verwechseln. Es ist gerade eine der zentralen empirischen Aussagen der Evolutionstheorie, dass die Fortpflanzungschancen von Lebewesen mit besserer Anpassung an die Umwelt steigen. Setzt man die Fitness schon definitorisch mit dem Selektionserfolg bzw. der Fortpflanzungswahrscheinlichkeit gleich, so würde man die Evolutionstheorie in eine Tautologie verwandeln.[5]

Auch bei der Technik der evolutionären Algorithmen trennt man die Bewertungsfunktion vom *Selektionsoperator*, wobei das Ergebnis der Bewertungsfunktion lediglich als „input" dient, mit dem der Selektionsoperator gefüttert wird. Natürlich sollte die Selektionswahrscheinlichkeit für einen bestimmten, als Punkt im Suchraum repräsentierten Typus um so höher ausfallen, je besser dessen Bewertung ist. Aber diese Bedingung wird von sehr vielen verschiedenen Selektionsoperatoren erfüllt. Die Wahl geeigneter Selektionsoperatoren gehört daher zur hohen Kunst der Konstruktion von evolutionären Algorithmen, denn vom Selektionsoperator hängt es ganz entscheidend ab, wie viel Zeit bzw. wie viele Generationen der evolutionäre Algorithmus benötigt, um eine möglichste gute Lösung zu finden. Die Möglichkeiten reichen hier von der sehr simplen „Rouletterad-Selektion", bei der die relativen Bewertungen proportional auf die Selektionswahrscheinlichkeiten abgebildet werden, bis hin zu raffinierten Verfahren wie der sogenannten „Boltzmann-Selektion"[6], bei der der Selektionsdruck kontinuierlich zunimmt. Letztere verdankt ihren Namen übrigens dem Physiker Ludwig Boltzmann, dem Erfinder der modernen Wärmelehre. Die Idee zu dieser Methode beruht nämlich auf der Erkenntnis, dass physikalische Substanzen bei der Abkühlung dem energieärmsten Zustand zustreben. Ein nach diesem Modell gebildeter Selektionsoperator sollte daher den Algorithmus nach anfänglicher breiter Streuung der (Geno-)Typen ebenfalls rasch in ein Extremum konvergieren lassen.

Die für die natürliche Evolution so wichtige Unterscheidung zwischen Genotyp und Phänotyp wird bei der Definition evolutionärer Algorithmen überlicherweise übergangen. Während in der Natur die Selektion stets am Phänotyp, also an den Merkmalsausprägungen der Gene im Lebewesen angreift und gerade nicht unmittelbar an den Genen selbst, beziehen sich die Bewertung und Selektion bei evolutionären Algorithmen meist unmittelbar

[5]Die Behauptung, dass die Evolutionstheorie eine Tautologie wäre, ist tatsächlich gelegentlich zu hören, beruht aber vermutlich genau auf diesem Missverständnis.

[6]Vgl. Gerdes et al., S. 83.

auf Punkte des Suchraums. Allerdings kann man diese Unterscheidung gegebenenfalls auch bei evolutionären Algorithmen einführen. Aus Gründen der Einfachheit empfiehlt es sich jedoch darauf zu verzichten.

Ähnlich wie die natürliche Evolution auf den drei „Modulen" Reproduktion, Variation und Selektion aufbaut, ist der Ablauf eines evolutionären Algorithmus in die vier Phasen der *Bewertung*, *Selektion*, *Reproduktion* und *Variation* unterteilt.[7] Die letzte Phase der Variation unterteilt sich dabei – wiederum analog zur Natur – noch einmal in *Mutation* und *Rekombination*. Mutationen sind dabei kleine und typischerweise seltene zufällige Veränderungen des „Genotyps". Wird der Genotyp beispielsweise durch einen Bitstring, also eine Folge von Nullen und Einsen, repräsentiert, dann entspricht eine Mutation der zufälligen Vertauschung eines Bits innerhalb dieser Folge von 0 nach 1 oder umgekehrt. Bei der Rekombination (auch „Crossover" genannt) wird dagegen aus einem oder mehreren „Eltern-Genotypen" durch Austausch von Teilfolgen ein neuer Genotyp gebildet. Die Analogie zur geschlechtlichen Vermehrung in der Natur liegt auf der Hand. Durch den Einsatz eines Rekombinationsoperators erhofft man sich, dass erfolgreiche Teillösungen sich rascher in einer Population verbreiten und sich eventuell mit anderen Teillösungen zu besseren Gesamtlösungen kombinieren. Es lässt sich nachweisen, dass evolutionäre Algorithmen durch den Einsatz von „Crossover" deutlich beschleunigt werden können.[8]

Nachdem nun in aller Kürze die Ingredienzien evolutionärer Algorithmen vorgestellt worden sind, mag sich manchem Leser die Frage stellen: Kann das funktionieren? Dies ist ja auch die Frage, die sich angesichts des Evolutionsprinzips als Erklärung für die Vielfalt der Arten in der Natur stellt. Ob es im Bereich der evolutionären Algorithmen funktionieren kann, soll nun anhand eines praktischen Beispiels untersucht werden.

Ein praktisches Beispiel[9]

Um ein möglichst anschauliches Beispiel zu liefern, soll an dieser Stelle demonstriert werden, wie ein evolutionärer Algorithmus eingesetzt werden kann, um das sogenannte „Problem des Handlungsreisenden" zu lösen. Bei

[7] Andere Einteilungen sind möglich. So werden in manchen Darstellungen beispielsweise zwei unterschiedliche Selektionsphasen eingeführt, nämlich die „Selektion zur Reproduktion" und die „Selektion zur Ersetzung" (Jansen, a.a.O., S. 10), was sich auch auf den Programaufbau auswirkt.

[8] Vgl. Jansen, a.a.O., S. 154ff.

[9] Den Quellcode des Beispielprogramm können Interessierte unter www.eckhartarnold.de/german/computers.html#evoldemo herunterladen.

dem Problem des Handlungsreisenden geht es darum, einen möglichst kurzen Weg zu finden, der eine Anzahl vorgegebener Orte verbindet und – in der hier untersuchten Variante – wieder zum Ausgangspunkt zurück führt. Dieses Problem gehört zur Klasse der sogenannten „NP[10]-harten“ Probleme, d.h. es handelt sich um ein Problem für das keine einfache Lösung existiert, und bei dem die Bestimmung der optimalen Lösung – einer zentralen Vermutung der Komplexitätstheorie zufolge[11] – schlimmstenfalls eine Anzahl von Rechenoperationen in Anspruch nimmt, die exponentiell mit der Anzahl der Orte steigt. Heuristische Suchverfahren konzentrieren sich daher nicht darauf, die optimale Lösung zu finden, sondern innerhalb möglichst kurzer Zeit eine möglichst gute Annäherung an das Optimum zu liefern. Der im Folgenden vorgestellte evolutionäre Algorithmus stellt eine solches heuristisches Verfahren dar. Das Ziel ist dabei nicht eine Heuristik zu finden, die alle anderen übertrifft, was bei einem wohlbekannten und gut untersuchten Problem wie dem des Handlungsreisenden ein kühnes Unterfangen wäre, sondern vor Augen zu führen, dass und wie evolutionäre Algorithmen funktionieren.

Der Algorithmus ist folgendermaßen aufgebaut: Als *Genom* wird naheliegenderweise die Folge der Orte verwendet, die nacheinander angesteuert werden. Die *Bewertungsfunktion* berechnet dann ganz einfach die Länge der Wegstrecke, die sich daraus ergibt. Da die *Fitness* eines Genoms umso größer sein muss, je kleiner die Strecke ist, wird zur Bestimmung der Fitness einfach der Kehrwert der Wegstrecke gebildet. Als *Selektionsoperator* wird eine Form der eben erwähnten Boltzmann-Selektion eingesetzt. In jeder Generation werden 40% der Genome durch *Rekombination* neu kombiniert. Dazu wird ein Teilstück von zwei Genomen, sprich von zwei Folgen von Orten ausgetauscht („2-Punkt Crossover“), wobei ein einfacher Reparaturmechanismus dafür sorgt, dass in den resultierenden Folgen wieder alle Orte enthalten sind. Um *Mutation* nachzubilden werden bei 20% der Genome (bei einer Basispopulation von 1000) zwei Ziele in der Folge der angesteuerten Orte zufällig vertauscht. Damit nicht durch Mutation oder Rekombination die beste bisher gefundene Lösung wieder verloren geht, sorgt zusätzlich ein *Eliteprinzip* dafür, dass wenigstens eine Kopie des besten Genoms der letzten Generation unverändert in die nächste Generation übernommen wird.

---

[10] „NP“ steht dabei als Abkürzung für „*N*icht *P*olynomial“, weil bei dieser Klasse von Problemen die Rechenzeit zur Lösung des Problems *exponentiell* und eben nicht mehr bloß *polynomial* ansteigt.

[11] Bisher ist diese Vermutung noch nicht bewiesen. Aber es existiert auch kein Gegenbeispiel und ihre Richtigkeit wird allgemein angenommen.

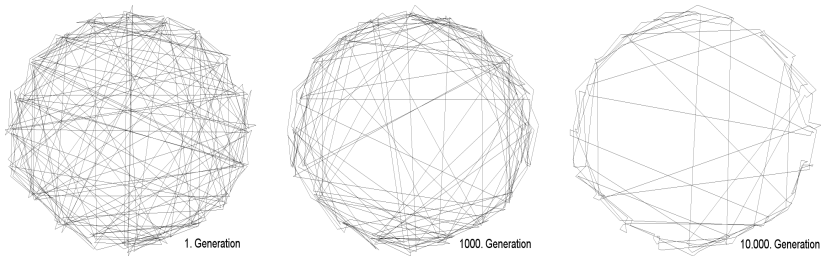

Abbildung 1: Das "Handlungsreisendenproblem". Lösungssuche mit einem evolutionären Algorithmus nach einer, 1.000 und 10.000 Generationen.

Es wäre plausibel den Algorithmus auf einer zufälligen Menge von Orten auf einer cartesianischen Ebene arbeiten zu lassen. Aber aus Gründen der Anschaulichkeit wird hier eine Ortsmenge verwendet, bei der die Orte in 20 kleinen zufälligen Anhäufungen konzentriert sind, die ihrerseits um einen größeren Kreis herum angeordnet sind, etwa so wie kleine weit voneinander entfernte Gehöfte von wenigen Häusern in einer ländlichen Gegend. Wählt man einen zufälligen Weg um die Zielpunkte dieser Menge zu verbinden, dann ergibt sich das auf *Abbildung 1* ganz links dargestellte Bild. Für den menschlichen Betrachter ist sofort ersichtlich, dass es sinnvoll wäre von Anhäufung zu Anhäufung zu reisen, und dabei zunächst die einzelnen Orte innerhalb einer Anhäufung abzuarbeiten, bevor man sich der nächsten Anhäufung zuwendet. Insbesondere würde man die Diagonalen, die quer über die Mitte des Kreises führen, tunlichst vermeiden. Evolutionäre Algorithmen arbeiten als „Black Box"-Algorithmen jedoch blind. Die einzige Information, die ihnen zu dem zu bearbeitenden Problem zur Verfügung steht, liefert ihnen die Bewertungsfunktion, die stets nur die Gesamtlänge einer gewählten Ortsfolge mitteilt. An diesem bewusst anschaulich gewählten Fallbeispiel lässt sich daher leicht erkennen, ob der evolutionäre Algorithmus trotzdem in der Lage ist zu „erkennen", dass quer über den Kreis führende Diagonalen besser zu vermeiden sind.

Das Ergebnis der Anwendung des beschriebenen evolutionären Algorithmus auf das Handlungsreisendenproblem kann auf *Abbildung 1* verfolgt werden. Von links nach rechts ist darauf die jeweils beste gefundene Lösung nach 1, nach 1.000 und nach 10.000 Generationen dargestellt. Wie man sieht, kommt es im Laufe der Entwicklung zu einer deutlichen Entzerrung der gewählten Verbindung. Besonders klar ist dies an dem Verschwinden eines großen Teils der Diagonalen zu erkennen. Die Wegstrecke des besten gefundenen Weges reduziert sich dabei von 10543 Einheiten nach der ersten Generation auf 7282 Einheiten nach der tausendsten Generation

und 2800 Einheiten nach der zehntausendsten Generation. Danach konvergiert der Algorithmus allerdings nur noch sehr langsam: Nach der hundertausendsten Generation hat sich die Wegstrecke immerhin auf 2047 Einheiten reduziert, also etwa auf ein Fünftel der besten der zufällig gewählten Anfangswegstrecken. Danach kommt es kaum noch zu Veränderungen, was darauf hindeuten könnte, dass der Algorithmus sich in der Nähe eines lokalen Optimums verfangen hat – ein bekanntes Problem bei evolutionären Algorithmen wie auch anderen Suchheuristiken –, denn die gefundene Lösung ist zwar erheblich besser als jede Wegstrecke in der Ausgangspopulation, aber sie ist offensichtlich noch bei Weitem nicht optimal. Zu erwähnen ist auch, dass der Algorithmus bei einer anderen Wahl der Steuerparameter (Mutationshäufigkeit, Selektionsoperator etc.) ein zum Teil wesentlich schlechteres Konvergenzverhalten zeigt. Dies verweist darauf, dass die Leistungsfähigkeit evolutionärer Algorithmen im Einzelfall stark vom „Feintuning“ abhängen kann.

Insgesamt sollte das Beispiel aber gezeigt haben, dass das Prinzip evolutionärer Algorithmen funktionieren kann.

Warum funktionieren evolutionäre Algorithmen?

Eine der wesentlichen Eigenschaften evolutionärer Algorithmen besteht darin, dass sie Probleme lösen können, ohne dass die Natur des Problems anders als durch die Bewertungsfunktion bekannt sein muss. Dies ist zugleich eine philosophisch erstaunliche Eigenschaft von evolutionären Algorithmen, denn man könnte sagen, dass evolutionäre Algorithmen „kreative Lösungssucher“ sind. Bei dem Beispiel des Handlungsreisenden, der eine Menge von Orten, die um bestimmte Häufungspunkte herum gruppiert sind, bereisen muss, springt dies unmittelbar ins Auge. Während für den menschlichen Betrachter (bei der im Beispiel gewählten Menge von Orten) nahezu optimale Lösungen sofort ersichtlich sind, nützt der verwendete evolutionäre Algorithmus die für uns visuell erfassbare Information über das Vorhandensein von Häufungspunkten nicht aus, noch enthält er einen expliziten Erkennungsmechanismus, um derartige Gegebenheiten festzustellen. Dennoch „lernt“ der Algorithmus, sich diese Gegebenheit zu Nutze zu machen.

Zugleich hat sich aber auch gezeigt, dass der Erfolg evolutionärer Algorithmen von Details seiner Implementierung abhängen kann. Erlaubt man sich diesen Befund einmal auf den Prozess der Evolution des Lebens in der Natur zu übertragen, so verdeutlicht dies, dass sowohl der Prozess selbst von sehr spezifischen Gegebenheiten abhängen kann, als auch welche Entwicklungsstufen der Prozess überhaupt erreicht. Immerhin dauerte es etwa

3 Milliarden Jahre, bis die Evolution über das Stadium von Einzellern hinaus gelangte.[12]

Kann man erklären, warum evolutionäre Algorithmen oft so erfolgreich sind? Und wenn sie schon erfolgreich sind, warum sind sie nicht immer erfolgreich? Um diese Fragen zu beantworten, sollen im Folgenden kurz zwei zentrale Befunde der Theorie der evolutionären Algorithmen vorgestellt werden.

Die „Building Block"-Hypothese

Einer der frühesten Ansätze zur theoretischen Beschreibung von evolutionären Algorithmen ist die „Schema-Theorie"[13]. Sie gipfelt in dem zentralen Theorem der „Building Block"-Hypothese. Der Grundgedanke der Schema-Theorie ist folgender: Evolutionäre Prozesse laufen so ab, dass zunächst „Teillösungen" gefunden werden, aus denen dann wie aus vorgefertigten Bausteinen Gesamtlösungen entstehen.

Formal werden diese Teillösungen durch „Schemata" beschrieben. Ein Schema ist dabei eine Art Schablone für ein Genom, bei dem bestimmte Gene schon festgelegt sind, während die Gene an anderen Stellen zunächst durch Platzhalter vertreten werden. Wenn also z.B. das Genom in einem Algorithmus als Folge der Buchstaben A,B oder C dargestellt wird und das *-Zeichen der Platzhalter ist, dann wäre „A*CB**" ein Schema, dass unter anderem auf die Genome „ACCBAB" und „AACBCC" passen würde. Je mehr Gene in einem Schema schon festgelegt sind und dementsprechend je geringer die Anzahl der Platzhalter im Schema ist, um so höher ist die „Ordnung" des Schemas. Neben der Ordnung des Schemas spielt auch dessen „definierende Länge" eine wichtige Rolle in der Schema-Theorie. Die definierende Länge ist dabei die Differenz zwischen der Position des letzten festgelegten Gens und der des ersten festgelegten Gens. Die beiden Schemata „**CA***" und „*B***C*" sind demnach beide vor der Ordnung 2, verfügen aber über eine unterschiedliche definierende Länge, nämlich 1 für das erste Schema und 4 für das zweite Schema. Unter bestimmten Voraussetzungen lässt sich nun zeigen („Schema-Theorem"), dass ein Schema, dessen Ordnung und definierende Länge hinreichend klein sind, sich bei

[12]Vgl. Ernst Mayr: What Evolution Is, New York 2001, S. 20.

[13]Vgl. Gerdes et al., a.a.O., S. 47ff. – Die Schema-Theorie geht auf J. Holland zurück. Vgl. John H. Holland: Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, Cambridge (Massachusetts) 1992 (1. Aufl.: 1975), S. 66ff.

überdurchschnittlicher Fitness[14] exponentiell verbreitet. Darauf stützt sich nun die „Building Block"-Hypothese, die sinngemäß besagt, dass evolutionäre Algorithmen sich bei der Suche nach einer möglichst optimalen Gesamtlösung zunächst auf solche kurzen und besonders „fitten" Schemata konzentrieren, aus denen sie die Gesamtlösung sukzessive aufbauen.[15]

Intuitiv erscheint der der „Building Block"-Hypothese zugrunde liegende Gedanke, dass die Evolution zunächst geeignete einfache Bausteine hervorbringt, um dann daraus ein komplexeres Gesamtwerk zu erschaffen, geradezu verlockend einfach. Allerdings wird die „Building Block"-Hypothese und die Schema-Theorie, auf die sie sich stützt, inzwischen häufig sehr kritisch beurteilt.[16] Im Detail weist sie (noch) viele Schwächen auf, denn es ist z.B. keineswegs gesichert, dass eine optimale Gesamtlösung aus für sich genommen besonders „fitten" Schemata zusammen gesetzt sein muss. Es kann sehr wohl vorkommen, dass mehrere für sich genommen sehr „fitte" Schemata in Kombination nur eine suboptimale Lösung ergeben, etwa wie Bausteine, die nicht zueinander passen.[17] Die „Building Block"-Hypothese repräsentiert daher eher einen frühen Ansatz zum Verständnis evolutionärer Algorithmen, keinesfalls aber ein allseits geteiltes wissenschaftliches Resultat .

Das „No Free Lunch"-Theorem

Während die „Building Block"-Hypothese zu erklären versucht, weshalb evolutionäre Algorithmen erfolgreich sind (in dem Sinne, dass sie in absehbarer Zeit brauchbare Lösungen produzieren) kann das sogenannte „No Free Lunch"-Theorem[18] genau umgekehrt dazu dienen, Grenzen der Möglichkeiten evolutionärer Algorithmen aufzuzeigen. Genaugenommen handelt es sich dabei nicht um ein Theorem, das nur evolutionäre Algorithmen betrifft, sondern das generell für Optimierungsalgorithmen gilt. Das „No Free Lunch"-Theorem gibt nämlich eine knappe und klare Antwort auf die Frage, ob es irgendwelche Optimierungsalgorithmen gibt, die generell,

[14]Die Fitness eines Schemas – wohl zu unterscheiden von der Fitness eines Genoms – wird dabei als die mittlere Fitness aller Genome definiert, auf die das Schema passt.

[15]Vgl. David E. Goldberg: Genetic Algorithms in Search, Optimization, and Machine Learning, Reading (Massachusetts) 1989, S. 41.

[16]Vgl. Jansen, a.a.O., S. 47-49. – Vgl. Gerdes et al., a.a.O., S. 55-56.

[17]Vgl. Gerdes et al., a.a.O., S. 55.

[18]Erstmals bewiesen von Macready und Wolpert in: W.G.Macready / D.H. Wolpert: No free lunch theorems for optimization, in: IEEE Transactions on Evolutionary Computing, Vol. 1, No. 1, April 1997, S. 67-82.

d.h. bezüglich *aller* denkbaren Probleme, besser sind als andere. Und diese Antwortet lautet erstaunlicherweise „Nein".

Diese Antwort ist deshalb erstaunlich, weil sie hochgradig kontraintuitiv ausfällt und sich nur schwer mit unserer alltäglichen Lebenserfahrung vereinbaren lässt. Denn unserer Lebenserfahrung entspricht es sehr wohl, dass manche Herangehensweisen an Probleme (welcher Art auch immer) generell besser sind als andere. So würde man systematische Herangehensweisen sicherlich für vielversprechender halten als unsystematische und zufällige. Warum gibt es dann aber keine Algorithmen, die über alle denkbaren Probleme eindeutig besser sind als andere?

Weiter oben wurde bereits festgestellt, dass evolutionäre Algorithmen ebenso wie alle anderen Suchalgorithmen auf einem *Suchraum* operieren und mit Hilfe einer *Bewertungsfunktion* Punkte in diesem Suchraum auswerten. Die Punkte im Suchraum entsprechen dabei möglichen Lösungen eines gegebenen Problems, während die Bewertungsfunktion gewissermaßen die Natur des Problems selbst beschreibt, denn von der Natur des Problems hängt es schließlich ab, ob eine bestimmte Lösung eine gute Lösung oder eine schlechte Lösung ist.

Schon aus dem täglichen Leben sind wir damit vertraut, dass wir mit sehr unterschiedlichen Arten von Problemen zu tun haben. Es gibt solche Probleme, für die nur eine einzige richtige Lösung existiert, während alle anderen Lösungen gleichermaßen schlecht sind. Und es gibt andere Probleme, zu denen man eine kontinuierliche Abstufung von besseren und schlechteren Lösungen finden kann. Schließlich gibt es Probleme mit einer eindeutig besten Lösung und wiederum andere Probleme mit mehreren besten Lösungen usw. Alle diese unterschiedlichen Arten von Problemen kann man durch unterschiedliche Lösungsfunktionen beschreiben. Wenn es z.B. für ein Problem nur eine eindeutig beste Lösung gibt und alle anderen Lösungen gleich schlecht sind, dann wird diesem Problem eine Bewertungsfunktion entsprechen, die dieser Lösung einen hohen Wert zuordnet und allen anderen Lösungen den gleichen niedrigen Wert. Es ist offensichtlich, dass ein Suchalgorithmus in diesem Fall vor großen Schwierigkeiten steht, da keine Anhaltspunkte vorhanden sind, die es erlauben würden, sich schrittweise an die Lösung heran zu tasten. Man nennt diese Funktion daher auch „Needle" Funktion,[19] einmal weil der Graph dieser Funktion an der Stelle des Maximums wie eine spitze Nadel aussieht, aber auch, weil die Lösungssuche in diesem Fall der sprichwörtlichen Suche nach der Nadel im Heuhaufen gleicht. Bei kontinuierlichen Funktionen dagegen kann man sich sehr wohl schrittweise an die Maxima heran tasten und die Qualität eines

[19]Vgl. Jansen, S. 38.

Suchalgorithmus zeichnet sich dadurch aus, wie schnell, d.h. mit wie wenig Auswertungen der Bewertungsfunktion, sich der Algorithmus dem Optimum annähert. Das besondere an Evolutionären Algorithmen besteht nun nicht so sehr darin, dass sie bei ganz bestimmten Problemen die optimale Lösung am schnellsten finden – für bestimmte, gut untersuchte Probleme, wie z.B. das Sortieren von Zahlen, die Suche von Zeichenketten in einem Text etc., gibt es fast immer sehr viel leistungsfähigere Spezialalgorithmen – sondern vielmehr darin, dass sie die Lösungen einer großen Klasse unterschiedlicher Probleme vergleichsweise rasch finden.

Nun besagt aber das „No Free Lunch"-Theorem gerade, dass über die Menge aller Bewertungsfunktionen, die die Punkte eines gegebenen Suchraums auf die Bewertungen einer bestimmten Wertmenge abbilden, die durchschnittliche Anzahl der Abfragen, die benötigt werden, um das Optimum zu erreichen, für alle Algorithmen gleich ist. Der Grund dafür ist folgender[20]: Jeder (deterministische) Algorithmus erreicht das Optimum auf einem bestimmten Lösungspfad, auf dem er nacheinander die Funktionswerte der Bewertungsfunktion für die Punkte des Suchraums abfragt. Für jeden Algorithmus gilt aber, dass die Anzahl aller Lösungspfade bezüglich der Menge der Bewertungsfunktionen genau gleich der Anzahl der Bewertungsfunktionen ist, denn größer kann sie nicht sein, sonst müssten irgendwann einmal ein und derselben Bewertungsfunktion zwei unterschiedliche Lösungspfade zugeordnet sein, was nicht möglich ist, da der gewählte Lösungspfad nur von der Bewertungsfunktion und dem (wie wir zunächst annehmen wollen: deterministischen) Algorithmus selbst abhängt. Und kleiner als die Anzahl der Bewertungsfunktionen kann die Anzahl der Lösungspfade auch nicht sein, denn aus demselben Grund müssen zwei unterschiedlichen Bewertungsfunktionen auch unterschiedliche Lösungspfade zugeordnet sein. Da die Menge aller Bewertungsfunktionen, der Menge aller möglichen Folgen von Funktionswerten entspricht, und da Lösungspfade auch Folgen von Funktionswerten entsprechen, folgt aus der gleichen Anzahl von Bewertungsfunktionen und Lösungspfaden, dass die Menge der Lösungspfade eines Algorithmus der Menge *aller* Folgen von Funktionswerten entspricht. Da dies für jeden Algorithmus gilt, haben alle Algorithmen die gleiche Menge von Lösungspfaden. (Man könnte sagen, sie unterscheiden sich nur dadurch, dass die Lösungspfade auf unterschiedliche Weise den Bewertungsfunktionen zugeordnet sind.) Das heißt aber, dass im *Durchschnitt* die Anzahl von Abfragen der Bewertungsfunktion bis zum Auffinden des Optimums für alle Algorithmen gleich sein muss. Dieses Resultat, das zunächst nur für deterministische Algorithmen gilt, lässt sich

[20]Für den ausführlichen mathematischen Beweis, dem die folgenden skizzenhaften Ausführungen zu Grunde liegen, siehe Jansen, a.a.O., S. 64-65.

leicht auf *randomisierte* Algorithmen übertragen, wenn man berücksichtigt, dass randomisierte Algorithmen auf Wahrscheinlichkeitsverteilungen über deterministische Algorithmen zurückgeführt werden können. Da die Leistung der deterministischen Algorithmen im Schnitt über alle Bewertungsfunktionen gleich ist, gilt dies dann auch für randomisierte Algorithmen.

Sehr stark vereinfacht kann man sich den Sachverhalt, den das „No Free Lunch"-Theorem beschreibt, auch folgendermaßen klar machen: Jeder Optimierungsalgorithmus arbeitet zur Lösung eines Problems eine Reihe von Lösungsschritten ab. Ein guter Algorithmus zeichnet sich dadurch aus, dass er schon nach wenigen Schritten am Ziel ist. Nun kann man aber auch zu dem besten Algorithmus immer ein Problem definieren, für das die vom Algorithmus gewählten Lösungsschritte leider genau die falschen sind. Und umgekehrt kann man für schlechte Algorithmen immer ein Problem definieren, bei dem er „zufällig" sehr schnell ans Ziel kommt. Das Theorem besagt nun, dass sich in der Menge aller denkbaren Problem die für einen Algorithmus günstigen und ungünstigen Fälle immer die Waage halten, so dass auf dieser – zugegebenermaßen unrealistisch großen Menge – gute und schlechte Algorithmen gar nicht wirklich unterschieden werden können.

Das „No Free Lunch"-Theorem zeigt, dass es keine Erfolgsgarantie, ja noch nicht einmal irgend eine Art komparativer Überlegenheitsgarantie für evolutionäre Algorithmen gibt – zumindest solange man die Menge der möglichen Probleme, die ein solcher Algorithmus lösen soll, nicht von vornherein einschränkt. Dieses Resultat mag beinahe wie eine Binsenweisheit klingen, aber es ist dennoch von nicht zu unterschätzender Bedeutung, zeigt es doch, dass die Leistungsfähigkeit eines evolutionären Algorithmus (und genau genommen auch eines jeden anderen Optimierungsalgorithmus') nicht *a priori* garantiert werden kann.

Besonders in philosophischer Hinsicht ist das „No Free Lunch"-Theorem von einiger Wichtigkeit, denn es beweist, dass eine gerade jüngst wieder diskutierte Lösungsstrategie für eines der klassischen Probleme der philosophischen Erkenntnistheorie vermutlich von vornherein zum Scheitern verurteilt ist. Weil dies noch zu wenig bekannt ist, sei der Zusammenhang in einem kurzen Exkurs hier ausgeführt. Das philosophische Problem, um das es sich handelt, ist das Induktionsproblem. Dabei geht es um das Problem, wie wir dazu berechtigt sein können, von der Vergangenheit auf die Zukunft zu schließen. Derartige Schlüsse ziehen wir im täglichen Leben ebenso wie in der Wissenschaft andauernd, aber wie schon David Hume festgestellt hat, berechtigt uns die bloße Tatsache, dass in der Vergangenheit jeden Tag die Sonne aufgegangen ist, in keiner Weise zu dem Schluss, dass dies in Zukunft auch geschehen *müsste*. Das *Induktionsprinzip* besagt, dass solche Schlüsse möglich sind. Alle bisherigen Versuche dieses Prinzip

zu beweisen, haben sich jedoch als irrtümlich herausgestellt. Die schwächere Annahme, dass wir wenigstens mit einer gewissen Wahrscheinlichkeit von der Vergangenheit auf die Zukunft schließen dürfen, enthält implizit ebenso unbegründete induktive Voraussetzungen, wie das bekannte Falsifikationsprinzip Karl Poppers. Karl Popper hat immer sehr entschieden die Ansicht vertreten, dass er das Induktionsprobproblem gelöst hätte.[21] Aber unter den Fachleuten, die sich heute mit dem Problem beschäftigen, gilt es als ausgemacht, dass ihm das nicht gelungen ist.[22] Poppers Falsifikationismus löst das Induktionsproblem deshalb nicht, weil eine Hypothese, die falsifiziert worden ist, in Zukunft trotzdem immer wahr sein könnte, es sei denn wir setzten eben doch irgend eine Art von Induktionsprinzip voraus.

Ein in jüngerer Zeit wieder aufgekommener Ansatz, der ursprünglich auf Hans Reichenbach zurückgeht, versucht das induktive Vorgehen auf eine vergleichsweise bescheidenere Weise, nicht als *notwendigerweise* erfolgreiches Verfahren zu verteidigen, aber als die unter allen Bedingungen „beste Alternative".[23] Der Grundgedanke diese Ansatzes ist der dieser: Nur in einer Welt die durch gewisse Regelmäßigkeiten geordnet ist, kann man mit Hilfe induktiver Schlussweisen gültige Erkenntnisse über die Welt gewinnen. *A priori* wissen wir aber nicht, ob unsere Welt eine solche Welt ist. (Und aus der Tatsache, dass sie es bisher gewesen ist, dürfen wir ohne Induktion nicht schließen, dass sie es auch in Zukunft sein wird.) Es wären auch *mögliche Welten* denkbar, die so unstrukturiert und chaotisch sind, dass Induktionsschlüsse darin nicht funktionieren. Andererseits kann man aber vermuten, dass in solchen „chaotischen" Welten auch andere Verfahren der Erkenntnisgewinnung als das induktive Vorgehen der Wissenschaft vor Schwierigkeiten stünden, so dass ein „Induktivist" sich zumindest in keiner ungünstigeren Lage befände als jeder andere. Insofern wäre es auf alle Fälle vernünftig, am Induktionsprinzip festzuhalten, denn wenn wir uns in einer entsprechend geordneten Welt befinden, dann funktioniert das Induktionsprinzip, und wenn nicht, dann hilft auch kein anderes Verfahren, so dass wir mit unserem induktiven Vorgehen jedenfalls nichts falsch machen können. Diese Rechtfertigungsstrategie des Induktionsprinzip bezeichnet man deshalb auch als „Best Alternative Justification".

---

[21]Vgl. Karl R. Popper: Objektive Erkenntnis. Ein evolutionärer Entwurf, Hamburg 1998, S. 1ff.

[22]Für den heutigen Stand der Diskussion vgl. Colin Howson: Hume's Problem. Induction and the Justification of Belief, Oxford 2000.

[23]Schurz, Gerhard: Der Metainduktivist: Ein spieltheoretischer Zugang zum Induktionsproblem, in: Proceedings der GAP.5, Bielefeld 22.-26.09.2003, S. 243-257, URL: www.gap5.de/proceedings/pdf/243-257_schurz.pdf

Um nun aber zu zeigen, dass die Rechtfertigung des Induktionsprinzips im Sinne einer „Best Alternative Justification“ tatsächlich gelingt, müsste man ein bestimmtes induktives und ggf. evolutionäres Lernverfahren spezifizieren, von dem man zeigen kann, dass es 1) in *allen* möglichen Welten höchstens marginal schlechter ist als beliebige Alternativen, und dass es 2) in bestimmten, nämlich den geordneten Welten aber besser abschneidet.[24] Genau dies ist der Punkt, an dem das „No Free Lunch“-Theorem jedoch erhebliche Zweifel wecken muss. Denn wenn man ein solches induktives Lernverfahren als einen Optimierungsalgorithmus versteht, der die Gültigkeit von Voraussagen maximieren soll, dann besagt das „No Free Lunch“-Theorem, dass es keinen solchen Algorithmus geben kann, der über alle möglichen Welten nur marginal schlechter ist als alle denkbaren Alternativen, in einigen aber eindeutig besser. In diesem Falle wäre der Algorithmus nämlich im *Durchschnitt* über alle möglichen Welten besser als andere Algorithmen, was das „No Free Lunch“-Theorem ausdrücklich verbietet. Ob das „No Free Lunch“-Theorem sich wirklich auf die „Best Alternative Justification“ anwenden lässt, hängt freilich von einigen subtilen philosophischen Interpretationsfragen ab. Insbesondere müsste geklärt werden, ob sich der formale Beweis für das „No Free Lunch“-Theorem präzise auf das Induktionsproblem beziehen lässt.[25] Aber nachdem sich der Ansatz – bisher ohne durchschlagenden Erfolg – fast ausnahmslos mit positiven Beweisversuchen der Möglichkeit einer „Best Alternative Justification“ beschäftigt hat, erscheint es lohnend auch einmal als Gegenprobe zu versuchen, ob sich nicht vielleicht die Unmöglichkeit einer „Best Alternative Justification“ beweisen lässt.

Bei den evolutionären Algorithmen befinden wir uns dabei noch in einer vergleichsweise günstigeren Lage, da wir in diesem Falle keine metaphysischen Fragen beantworten müssen und uns deshalb auch nicht genötigt zu fühlen brauchen, alle möglichen Welten in Betracht zu ziehen. Daher könnte man auf den Gedanken kommen, dass ein Theorem, das besagt, dass bezüglich aller denkbaren Optimierungsprobleme einer beliebigen aber bestimmten Art (die durch den Suchraum und die Bewertungsmenge festgelegt ist) kein Optimierungsalgorithmus in besonderer Weise gegenüber anderen ausgezeichnet werden kann, noch nicht viel bedeuten muss, denn sicherlich bilden die in der Realität vorkommenden Probleme nur eine kleine

[24] Vgl. Schurz, Der Metainduktivist, a.a.O., S. 245. – Schurz hat seinen Ansatz gegenüber dem hier zitierten Artikel inzwischen sehr stark ausgebaut. Allerdings sind die entsprechenden Arbeiten bisher leider noch nicht veröffentlicht.

[25] Dabei wäre vor allem die Frage zu klären, ob die von Schurz favorisierten „Metainduktivsten“ (Vgl. Schurz, a.a.0., S. 246ff.) als Optimierungsalgorithmen im Sinne des „No Free Lunch“-Theorems aufgefasst werden müssen.

Teilmenge der logisch möglichen Probleme. Um evolutionäre Algorithmen als besonders leistungsfähige Suchalgorithmen auszuzeichnen, würde es dann ausreichen, wenn man zeigen kann, dass evolutionäre Algorithmen auf „sinnvollen“ Teilmengen möglicher Bewertungsfunktionen überdurchschnittlich gut abschneiden. In diese Richtung gehen denn auch die Überlegungen in dem Zweig der theoretischen Informatik, der sich mit evolutionären Algorithmen beschäftigt.[26] Zwar gilt das „No Free Lunch“-Theorem selbst noch für solche Teilmengen der Menge aller Bewertungsfunktionen, die unter Permutation abgeschlossen sind. Aber es lässt sich auch zeigen, dass für bestimmte Teilmengen von Bewertungsfunktionen, z.B. solchen in denen keine Funktion vorkommt, bei der ein Minimum in der unmittelbaren Nachbarschaft eines Maximums liegt, die mit anderen Worten also ein gewisses Maß an „Glattheit“ aufweisen, kein „No Free Lunch“-Theorem mehr gelten kann.[27]

Insgesamt bleibt jedoch festzuhalten, dass man von evolutionären Algorithmen eben keine Wunder erwarten kann. In einigen Fällen, wie dem eingangs vorgestellten Beispiel funktionieren sie erstaunlich gut, besonders wenn man berücksichtigt, dass sie als „Black Box“-Problemlösungsstrategien so gut wie keinerlei problemspezifische Informationen ausnutzen. Warum sie so gut funktionieren, wenn sie funktionieren, kann man sich mit Hilfe der „Building Block“-Hypothese und verwandten theoretischen Ansätzen klar machen. Aber auch wenn sie gewöhnlichen Suchheuristiken oft weit überlegen sind, gibt es, wie das „No Free Lunch“-Theorem und verwandte Theoreme verdeutlichen, eben nicht von vornherein eine Erfolgsgarantie.

Evolutionäre Algorithmen und Neuheit

Wie die eben geführte Diskussion um das „No Free Lunch“-Theorem gezeigt hat, funktionieren auch evolutionäre Algorithmen nur unter bestimmten Bedingungen. Sind die Bedingungen einmal gegeben, dann können sie erstaunlich erfolgreich sein. Die Frage beleibt aber immer noch, inwiefern evolutionäre Algorithmen genuin Neues hervorbringen können. Dagegen könnte man einwenden, dass evolutionäre Algorithmen ja nur einen vorgegebenen Suchraum abarbeiten, so dass alles, was sie denkbarer Weise hervorbringen können, schon durch den Suchraum vorgegeben ist. Zudem könnte wohl manch einer an dem technischen Charakter von Computerpro-

[26] Vgl. Jansen, S. 68ff.
[27] Vgl. Jansen, S. 70.

grammen Anstoß nehmen, denen man grundsätzlich keine „echte“ Kreativität zutrauen könne.

Was zunächst die vermeintliche Vorgegebenheit der Lösung durch den Suchraum betrifft, so gilt auch, dass die Menge der Töne, die ein Orchester in 90 Minuten spielen kann eine endliche Menge ist. Bis man aber in dieser Menge durch zufälliges Ausprobieren die neunte Symphonie gefunden hat, würde mehr Zeit vergehen als der gesamten Menschheit jemals zur Verfügung stehen wird. Insofern muss Kreativität nicht unbedingt durch Unendlichkeit, Unbegrenztheit, absolute Unvorhersagbarkeit oder der dergleichen definiert werden, sondern sie kann sehr wohl bloß darin bestehen, aus einer unüberschaubaren endlichen Menge von Möglichkeiten innerhalb eines für uns fassbaren Zeitrahmens (etwa der Lebenszeit eines Komponisten) die für uns wertvollen Lösungen heraus zu suchen. In gewisser Weise leisten genau das – für vergleichsweise sehr viel einfachere Aufgaben – auch evolutionäre Algorithmen.

Andererseits gibt es in der Tat gewisse Aspekte von Kreativität und Neuheit, die man mit evolutionären Algorithmen nicht erfassen kann. Evolutionäre Algorithmen können nur dort überhaupt ansetzen, wo eine Bewertungsfunktion vorgegeben ist. Dies ist aber gerade für kreative Prozesse im ästhetischen Bereich in der Regel nicht gegeben. Dort ändern sich die Bewertungsmaßstäbe ja gerade auch mit den Werken, die geschaffen werden. Ähnliches gilt auch für die Evolution in der Natur. Die „Bewertung“ der evolvierten Lebenwesen hängt in der Natur von der Umwelt ab, an die sie besser oder schlechter angepasst sind. Aber diese Umwelt besteht zum wichtigsten Teil ihrerseits wiederum aus evolvierten Lebewesen. Insofern ist der Evolutionsprozess in der Natur ein hochgradig selbstbezüglicher Prozess. Für diesen Zusammenhang gibt es zumindest bei den hier vorgestellten sehr einfachen evolutionären Algorithmen keine Entsprechung. Abgesehen davon aber sollte uns der „technische Charakter“ evolutionärer Algorithmen nicht daran hindern zuzugestehen, dass zumindest bestimmte Formen von „Neuheit“ auch durch einen im Grunde einfachen mechanischen Vorgang, ohne eigentliches Verstehen der Sache hervorgebracht werden können.

Literatur

Dupré, John: Human Nature and the Limits of Science, Oxford 2001.

Gerdes, Ingrid / Klawonn, Frank / Kruse, Rudolf: Evolutionäre Algorithmen, Wiesbaden 2004.

Goldberg, David E.: Genetic Algorithms in Search, Optimization, and Machine Learning, Reading (Massachusetts) 1989.

Holland, John H.: Adaptation in Natural and Artificial Systems. An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence, Cambridge (Massachusetts) 1992 (1.Aufl.: 1975).

Howson, Colin: Hume's Problem. Induction and the Justification of Belief, Oxford 2000.

Jansen, Thomas: Evolutionäre Algorithmen, Vorlesungssrkipt WS 2004/05, URL:ls2-www.cs.uni-dortmund.de/lehre/winter200405/evo-alg/#Skript (Zugriff: 15.12.2007)

Mayr, Ernst: What Evolution Is, New York 2001.

Macready, W.G. / Wolpert, D.H.: No free lunch theorems for optimization, in: IEEE Transactions on Evolutionary Computing, Vol. 1, No. 1, April 1997, S. 67-82.

Popper, Karl R.: Objektive Erkenntnis. Ein evolutionärer Entwurf, Hamburg 1998.

Schurz, Gerhard: Natürliche und kulturelle Evolution: Skizze einer verallgemeinerten Evolutionstheorie, in: Wolfgang Wickler / Lucie Salwiczek (Hrsg.): Wie wir die Welt erkennen. Erkenntnisweisen im interdisziplinären Diskurs. Freiburg / München 2001, S. 329-376.

Schurz, Gerhard: Der Metainduktivist: Ein spieltheoretischer Zugang zum Induktionsproblem, in: Proceedings der GAP.5, Bielefeld 22.-26.09.2003, S. 243-257, URL: www.gap5.de/proceedings/pdf/243-257_schurz.pdf

Willms, Jürgen: Evolutionäre Algorithmen: Chancen für die praxisorientierte Optimierung, Meschede,URL: www.fh-meschede.de/public/willms/ea/index.html (Zugriff: 15.12.2007)