

1 Einleitung

Lebewesen sind vollendete Problemlöser. In der Vielzahl der Aufgaben, die sie bewältigen, übertreffen sie die besten Computerprogramme bei weitem – zur besonderen Frustration der Programmierer, die Monate oder gar Jahre harter geistiger Arbeit für einen Algorithmus aufwenden, während Organismen ihre Fähigkeiten durch den scheinbar ziellosen Mechanismus der Evolution erwerben.

J.H. Holland (Spektrum der Wissenschaft 9/1992)

Dieses Zitat von John Holland zielt weniger auf die biologische Evolution selbst, sondern viel mehr auf die Idee, Prinzipien der Evolution zu nutzen, um Probleme mit dem Computer zu lösen. Die klassische Strategie zur Lösung von Problemen mit dem Computer besteht darin, auf der Grundlage möglichst detaillierten Wissens über die Struktur und die Eigenschaften des Problems ein effizientes Verfahren zu entwickeln, das die gewünschte Lösung konstruiert. Voraussetzung für diese Vorgehensweise ist zum einen, dass die benötigten strukturellen Informationen über das Problem genau bekannt sind, und zum anderen, dass Experten und Programmierer die Zeit und den Aufwand investieren, an einem entsprechenden Modell und seiner Implementierung zu arbeiten. Die erste Voraussetzung ist nicht immer gegeben, die zweite bedeutet meist erhebliche Kosten. Experten müssen zunächst ein Modell erstellen, auf der Basis dieses Modells geeignete Problemlösungsstrategien entwickeln und diese Verfahren schließlich implementieren.

Mit der zunehmenden Leistungsfähigkeit von Rechnern hat in den letzten Dekaden eine Entwicklung begonnen, die einen alternativen Weg einschlägt. Man verwendet zur Problemlösung relativ universelle Strategien, in die sich strukturelles Wissen über das Problem ohne großen Formalisierungs- oder Modellierungsaufwand integrieren lässt. Der Preis besteht häufig in einem deutlich größeren Rechenaufwand, der aber durch die Geschwindigkeit heutiger Computer vielfach kompensiert werden kann und zudem nahezu kostenlos im Vergleich zur Bezahlung von Experten ist. Techniken, die auf dieser Philosophie basieren, werden zum Bereich der *Computational Intelligence* gezählt. Der Begriff Computational Intelligence wurde Anfang der neunziger Jahre geprägt [Bez92, Bez94, ZMIR94].

*Evolutionäre Algorithmen*¹ – neben Fuzzy-Systemen und neuronalen Netzen eines der wichtigsten Gebiete im Bereich Computational Intelligence – bilden eine Klasse von Verfahren, die sich die Prinzipien der biologischen Evolution zum Vorbild nehmen, um primär Optimierungsprobleme zu lösen. Wie in vielen Bereichen der Technik, sei es im Flugzeugbau oder bei den (künstlichen) neuronalen Netzen, geht es nicht darum, das natürliche Vorbild möglichst exakt nachzubilden. Die Flügelform von Flugzeugen ist zwar denen von Vögeln ähnlich, die Antriebstechniken sind jedoch gänzlich verschieden.

¹ Im Englischen verwendet man neben dem Begriff „evolutionary algorithms“ auch „evolutionary computation“.

Auch die neuronalen Netze, die in der Informatik verwendet werden, haben sich sehr weit von realistischen Modellen natürlicher Neuronennetzwerke entfernt. Genauso verhält es sich bei den evolutionären Algorithmen. Die Mechanismen der biologischen Evolution sind enorm komplex und bis heute noch nicht vollständig verstanden. Selbst wenn ein sehr exaktes Modell der biologischen Evolution bekannt wäre, sind die Populationsgrößen und die Zeiträume, die dabei eine Rolle spielen, für technische Anwendungen und Realisierungen auf dem Computer indiskutabel.

Die ersten Ansätze zur Ausnutzung von Konzepten der biologischen Evolution wurde bereits vor fast fünfzig Jahren publiziert [Box57, Fra57, Fri58, And53, Bre62]. Später entwickelten sich neben anderen Ansätzen die folgenden Hauptrichtungen, die auch heute noch von großer Bedeutung sind:

Evolutionsstrategien wurden in den sechziger Jahren von Rechenberg und Schwefel für Optimierungsprobleme aus dem technisch-physikalischen Bereich entwickelt und wurden zuerst in [Rec73, Sch75] publiziert. Evolutionsstrategien verwenden kontinuierliche Parameter und Mechanismen zur Selbstadaption ihrer Steuerungsparameter. Sie sind vor allem auf Optimierungsprobleme mit kontinuierlichen Parametern zugeschnitten.

Genetische Algorithmen gehen zum einen auf Holland, der sie eher aus der Sicht lernender oder adaptiver Systeme betrachtet hat [Hol75], und zum anderen auf De Jong zurück, der sie erstmals für Optimierungsprobleme eingesetzt hat [DJ75]. Genetische Algorithmen verwenden in ihrer ursprünglichen Form eine binäre Kodierung, so dass sie sich vor allem für kombinatorische Optimierung eignen.

Genetische Programmierung wurde – wie sie in der heutigen Form verwendet wird – vor allem durch Koza entwickelt [Koz90] und durch seine ersten beiden Bücher [Koz92, Koz94] populär. Im Gegensatz zu den Evolutionsstrategien und den genetischen Algorithmen, die vor allem auf Parameteroptimierung ausgerichtet sind, hat die genetische Programmierung die Entwicklung von optimalen Strategien in Form von einfachen Computerprogrammen zum Ziel.

Von historischer Bedeutung ist auch die *evolutionäre Programmierung*, die auf L.J. Fogel [FOW66] und D.B. Fogel [Fog92] zurückgeht, aufgrund einiger einschränkender Annahmen heute aber eine geringere Rolle als die anderen Richtungen spielt.

Lange Zeit haben vor allem Evolutionsstrategien, genetische Algorithmen und evolutionäre Programmierung als konkurrierende Modelle nebeneinander existiert. Einzelne Vertreter einer Richtung haben versucht, die Überlegenheit ihres favorisierten Ansatzes gegenüber den anderen Ansätzen theoretisch oder empirisch zu belegen. Die Oberbegriffe „evolutionäre Algorithmen“ und international vor allem „evolutionary computation“ haben sich erst später etabliert, als bereits alle Einzelrichtungen Erfolge in den Anwendungen vorweisen konnten.

Das gemeinsame Grundgerüst aller evolutionären Algorithmen sieht folgendermaßen aus. Zunächst muss für ein Problem definiert werden, wie eine Lösung – egal, ob sie gut oder schlecht ist – prinzipiell strukturiert ist. Dies definiert den Suchraum. Des weiteren ist eine Zielfunktion erforderlich, die die Elemente des Suchraums bewertet und