Modelloptimierung mit evolutionären Algorithmen

Thomas Bäck Ulrich Hammel Hans-Paul Schwefel

Universität Dortmund
Fachbereich Informatik
Lehrstuhl für Systemanalyse
D–44221 Dortmund

Zusammenfassung

Die in der modernen Systemanalyse unverzichtbare Kopplung von *Simulation* und *Optimierung* stellt eine Reihe von kritischen Anforderungen an verwendbare Optimierverfahren, denen klassische Methoden häufig nicht gerecht zu werden vermögen. *Evolutionäre Algorithmen*, Optimierverfahren nach dem Vorbild der biologischen Evolution, erfüllen hingegen viele dieser Forderungen und werden daher — auch wegen ihres Potentials zur Nutzung massiv paralleler Rechner — als Alternativen zur herkömmlichen Vorgehensweise vorgeschlagen. Als bedeutendste Repräsentanten dieser Klasse von Algorithmen werden *Evolutionsstrategien* und *genetische Algorithmen* im Hinblick auf ihre wesentlichen Merkmale vorgestellt und miteinander verglichen.

1. Parameteroptimierung

Optimierungsprobleme treten in einer Vielzahl von Bereichen der technischen und industriellen Anwendung sowie in der Forschung auf. Derartige Probleme sind dadurch charakterisiert, daß eine Einstellung $\vec{x} = (x_1, \ldots, x_n) \in M$ der freien Parameter des zu optimierenden Systems zu finden ist, so daß ein (meist reellwertiges) Gütekriterium $f(\vec{x})$ ($f : M \to I\!R$ wird üblicherweise auch *Zielfunktion* genannt) maximiert wird (Minimierung ist wegen $\max\{f(\vec{x}\} = -\min\{-f(\vec{x})\}$ äquivalent):

$$f(\vec{x}) \quad \to \quad \max \ . \tag{1}$$

Das Gütekriterium kann durch reale Systeme von beliebiger Komplexität definiert sein, z.B. durch einen Bioreaktor [FWBW92], einen Produktionsprozeß [Hau89], oder eine Zweiphasendüse [KS70], als Computermodell der Realität vorliegen oder, im einfachsten Fall, analytisch beschreibbar (und ggf. analytisch lösbar) sein.

Für einen gesuchten Lösungsvektor $\vec{x}^* \in M$ gilt das Kriterium

$$\forall \vec{x} \in M \quad : \quad f(\vec{x}) \leq f(\vec{x}^*) = f^* \ . \tag{2}$$

$\vec{x}^*$ heißt dann *globaler Optimalpunkt*, f^* *globales Optimum*. Im allgemeinen Fall eines nichtlinearen Gütekriteriums f kann das Problem sogenannte *lokale Optima* $\hat{\vec{x}}$ aufweisen, für die

$$\exists \varepsilon \in I\!R, \varepsilon > 0 \ : \ \forall \vec{x} \in M \ : \ \|\hat{\vec{x}} - \vec{x}\| < \varepsilon \quad \Rightarrow \quad f(\vec{x}) \leq f(\hat{\vec{x}}) = \hat{f} \tag{3}$$

gilt. Sogenannte *lokale Optimierverfahren* sind iterative Methoden, welche ausgehend von einem gegebenen Startpunkt das nächstgelegene lokale Optimum auffinden und dann terminieren. Da die Güte eines lokalen Optimums aber weit unter der des globalen Optimums liegen kann, ist man an Verfahren interessiert, für welche *globale Konvergenz mit Wahrscheinlichkeit Eins* zugesichert werden kann, d.h.

$$\mathcal{P}\{\lim_{t\to\infty} \vec{x}_t = \vec{x}^*\} \quad = \quad 1 \,, \tag{4}$$

wobei $\vec{x}_t$ eine Folge von durch den Algorithmus generierten Testpunkten bezeichnet. Diese Aussage ist von lediglich theoretischem Nutzen, da eine allgemeine obere Schranke für t nicht existiert.

Neben der Existenz mehrerer lokaler Optima wird die Optimierungsaufgabe häufig durch die Struktur der Menge $M \subseteq M_1 \times \ldots \times M_n$ (einige M_i können kontinuierlich, andere diskret sein) und durch die Existenz von *Restriktionsfunktionen* $g_j : M_1 \times \ldots \times M_n \to I\!R$ erschwert, welche den zulässigen Lösungsraum einschränken:

$$M \quad = \quad \{\vec{x} \in M_1 \times \ldots \times M_n \mid g_j(\vec{x}) \geq 0 \; \forall j\} \,. \tag{5}$$

Reale Optimierungsaufgaben weisen häufig neben der Existenz mehrerer lokaler Optima eine Reihe von zusätzlichen Schwierigkeiten auf, z.B. Unstetigkeitsstellen der Gütefunktion, die Existenz von „Plateaus" bzw. „flachen Bereichen" in der hochdimensionalen Topologie oder auch unzusammenhängende zulässige Bereiche. In praktischen Anwendungen genügt aber meist die Bestimmung einer guten — nicht notwendigerweise global optimalen — Lösung, welche eine Verbesserung des Gütekriteriums im Vergleich zur bisher bekannten Lösung ermöglicht, so daß der Einsatz von Optimierverfahren trotz der allgemeinen Unlösbarkeit des Problems der globalen Optimierung (z.B. [TŽ89], S. 6) sinnvoll und nutzbringend ist.

2. Simulation und Optimierung

Optimierungsprobleme treten im Zusammenhang mit der Modellbildung (z.B. [Mey84]) typischerweise bei der Identifikation als Parameterschätzproblem im Sinne einer Minimierung für eine Summe von quadratischen Abweichungen auf. Im allgemeinen Fall einer nichtlinearen Abhängigkeit von den zu schätzenden Parametern stellt die Identifikation ein hochgradig komplexes multimodales Optimierungsproblem dar.

Im Zusammenhang mit der Simulation steht die Optimierung von aus dem Modellbildungsprozeß gewonnenen Simulationsmodellen im Vordergrund. Dabei sind die freien Parameter des Modells derart zu bestimmen, daß ein bestimmtes Gütekriterium an die Ausgabe des Modells in möglichst optimaler Weise erfüllt wird. Die Modellausgabe ist dabei nur numerisch experimentell (durch eine Simulation) ermittelbar und — ganz im Sinne einer „black box" — lediglich Information über die Modelleingabe $\vec{x}$ und den zugehörigen Gütewert $f(\vec{x})$ vorhanden.

Faßt man in diesem Sinne das Simulationsmodell als Zielfunktion f im Rahmen eines Parameteroptimierungsproblems (1) auf, so lassen sich eine Reihe von Charakteristika für f identifizieren, welche die Optimierung zu einer komplexen Aufgabenstellung werden lassen:

(1) Die Zielfunktion ist nicht (stetig) differenzierbar bzw. „glatt"; analytische Information über Ableitungen ist nicht vorhanden.

(2) Die Zielfunktion ist multimodal und hochdimensional.

(3) Optimalstellen der Zielfunktion sind möglicherweise sensibel gegenüber kleinen Parameterveränderungen, so daß die technische Umsetzung im Rahmen der vorgegebenen Toleranzen nicht möglich ist. Im Extremfall kann die Zielfunktion fraktale Charakteristika aufweisen.

(4) Die Optimalstelle (und das Optimum) kann zeitlich variieren.

(5) Üblicherweise sind — meist wiederum nichtlineare — Restriktionen zu berücksichtigen. Der zulässige Bereich kann sogar unzusammenhängend sein.

(6) Unter Umständen sind mehrere Gütekriterien gleichzeitig zu optimieren (*mehrkriterielle Optimierung*).

(7) Die Zielfunktion kann stochastischen Störungen unterworfen sein.

Auf der Basis dieser Charakteristika des Optimierungsproblems ergeben sich eine Reihe von Anforderungen für die verwendbaren Optimierverfahren. Zunächst sollten sie *direkte* Suchverfahren sein, also allein auf der Basis von Parametervektoren und zugehörigen Zielfunktionswerten arbeiten. Ferner sollten *globale* Verfahren zum Einsatz kommen, also wenigstens prinzipiell das Auffinden einer globalen Optimalstelle garantiert sein. Zeitliche Variationen sowie die Existenz von Störungen führen zu der Anforderung, gesammelte Information wieder „vergessen" zu können und ggf. Parametervektoren wiederholt auszuwerten. Schließlich sollte das Verfahren zumindest prinzipiell in Suchräumen beliebiger Dimensionalität anwendbar sein und die Berücksichtigung von Restriktionen ermöglichen. Aus Sicht des Benutzers ergibt sich die Forderung nach möglichst einfacher Anwendbarkeit des Optimierverfahrens, d.h. es sollte selbst möglichst wenige einzustellende Parameter besitzen und sich robust gegenüber kleinen Änderungen dieser Parameter verhalten.

Wie ein umfassender, von Schwefel durchgeführter, Vergleich von Optimierverfahren gezeigt hat, scheitern viele klassische Optimierverfahren (Mustersuche von Hooke und Jeeves [HJ61], Strategie der konjugierten Richtungen nach Powell [Pow62], Simplex Strategie von Nelder und Mead [NM65], Complex Strategie von Box [Box65], Rosenbrock Strategie [Ros60], um nur einige zu nennen) bereits bei Aufgabenstellungen, für welche nur wenige der angegebenen erschwerenden Charakteristika zutreffen [Sch77, Sch79].

Die Untersuchungen von Schwefel lieferten deutliche Hinweise, daß Evolutionsstrategien als Vertreter der Klasse der sogenannten *evolutionären Algorithmen* für die Modelloptimierung eine hervorragende Alternative zu den klassichen Optimierverfahren bieten.

3. Evolutionäre Algorithmen

Evolutionäre Algorithmen sind direkte, stochastische Suchverfahren, welche nach dem Vorbild der biologischen Evolution (welche auch Pate stand für die im Bereich der evolutionären Algorithmen gebräuchliche Terminologie) auf dem kollektiven Lernprozeß innerhalb einer *Population* von *Individuen* basieren, wobei jedes Individuum einen Punkt im Suchraum repräsentiert. Die Anfangspopulation evolviert durch teilweise probabilistische *Selektions-*, *Mutations-* und *Rekombinationsprozesse* in der Art, daß die durchschnittliche Güte (in Anlehnung an die biologische Terminologie auch *Fitneß* genannt) der Individuen nach und nach zunimmt. Durch den Selektionsoperator werden Individuen höherer Fitneß bevorzugt in die Folgegeneration übernommen, so daß der Prozeß eine Richtung erhält. Der Rekombinationsoperator ermöglicht den Austausch von Teilinformation zwischen Individuen, und Mutation erzeugt genetische Innovationen, welche sich als vor- oder nachteilig herausstellen können und im letzteren Fall durch die Selektion wieder aus der Population entfernt werden.

Implementierungen derartiger Algorithmen können sowohl als (vereinfachendes) Modell der biologischen Evolution als auch als Parameteroptimierungsverfahren verstanden werden; im folgenden gehen wir von letzterer Auffassung aus.

Auf einer sehr hohen Abstraktionsstufe kann ein Basisalgorithmus (Algorithmus (1)) nach dem Vorbild biologischer Evolution formuliert werden, welcher die existierenden Ansätze subsumiert. Dazu bezeichen t einen Generationszähler und $P(t) \in I^{\mu}$ eine Multimenge von μ Individuen zur Generation t. $P'(t)$ und $P''(t)$ sind intermediär gebildete Zwischenstufen der Größe λ auf dem Wege zu $P(t+1)$. Ferner bezeichne $Q \in \{P(t), \emptyset\}$ eine ggf. bei der Selektion zusätzlich zu berücksichtigende Menge von Individuen. Der Algorithmus erzeugt durch Rekombination und anschließende Mutation neue Individuen, welche ausgewertet und danach — eventuell zusammen mit der Vorgängerpopulation $P(t)$ — der Selektion unterworfen werden. Diese Folge von Operationen wird iteriert, bis ein vorgegebenes Abbruchkriterium erfüllt ist.

ALGORITHMUS 1 (Evolutionärer Basisalgorithmus)

$t := 0$;
initialisiere $P(0) \in I^\mu$;
bewerte $P(0)$;
while not *terminiere*$(P(t))$ **do**
  rekombiniere: $P'(t) := r(P(t))$;
  mutiere: $P''(t) := m(P'(t))$;
  bewerte $P''(t) \in I^\lambda$;
  selektiere: $P(t+1) := s(P''(t) \cup Q)$;
  $t := t + 1$;
od

Als wichtigste Repräsentanten evolutionärer Algorithmen sind *Evolutionsstrategien* (ES) und *genetische Algorithmen* (GAs) zu nennen. GAs orientierten sich zunächst an allgemeinen Adaptationsprozessen [Hol75] und fanden später auch breite Anwendung im Bereich von Parameteroptimierungsaufgaben [DeJ75, Gol89]. ES wurden zunächst zur experimentellen Optimierung eingesetzt [Rec73] und später in verschiedenen Formen auch auf dem Computer simuliert [Sch77, Sch81].

In der algorithmischen Realisierung gehen Evolutionsstrategien und genetische Algorithmen recht unterschiedliche Wege. Dies zeigt sich schon in der Form der Repräsentation von Suchpunkten durch die Individuen, welche bei GAs als binäre Zeichenketten $\vec{a} \in \mathbb{B}^l$ ($\mathbb{B} = \{0, 1\}$) vorliegen (die Zielfunktion gemäß Abschnitt 1 ist dann von der Funktionalität $f : M \subseteq \mathbb{B}^l \rightarrow \mathbb{R}$) und demzufolge zur approximativen Anwendung auf kontinuierliche Optimierprobleme einem aufwendigen Dekodierungsprozeß unterworfen werden müssen. In ES wird demgegenüber direkt mit n-dimensionalen Vektoren $\vec{x} \in \mathbb{R}^n$ von Objektvariablen gearbeitet (für eine Zielfunktion der Form $f : M \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$), wobei die Individuen zusätzlich um einen n-dimensionalen Vektor von (positiven) Standardabweichungen $\vec{\sigma} \in \mathbb{R}^n_+$ erweitert werden, mit deren Hilfe im Rahmen der Mutation die Variation der Objektvariablen erfolgt. Dabei wird jede Objektvariable x_i additiv mit einer normalverteilten $N(0, \sigma_i^2)$-Zufallszahl variiert, deren Erwartungswert Null ist und deren Standardabweichung σ_i *Bestandteil des Individuums* ist. Die Standardabweichungen σ_i sind nun nicht konstant oder exogen vorgegeben, sondern unterliegen ebenfalls einem multiplikativen, logarithmisch-normalverteilten Variationsprozeß:

$$\begin{aligned} \sigma_i' &= \sigma_i \cdot \exp(\tau' \cdot N(0,1) + \tau \cdot N_i(0,1)) \\ x_i' &= x_i + \sigma_i' \cdot N(0,1) \; . \end{aligned} \tag{6}$$

Dabei setzt sich die Mutation der σ_i aus einem gemeinsamen Anteil $\tau' \cdot N(0,1)$ und einem individuellen Anteil $\tau \cdot N_i(0,1)$ (der Index i deutet an, daß für jeden Wert von i eine eigene $N(0,1)$-verteilte Zufallszahl generiert wird) zusammen. Schwefel gibt die Empfehlungen $\tau' \sim 1/\sqrt{2n}$ und $\tau \sim 1/\sqrt{2\sqrt{n}}$ [Sch77].

Durch eine implizite Kopplung von hohen Qualitätsgewinnen (d.h. nutzbringenden Veränderungen der Objektvariablen) und vorteilhaften Standardabweichungen (d.h. sinnvollen *internen Modellen* der Zielfunktionstopologie) wirkt so die Selektion auch im Hinblick auf die Einstellungen der Strategieparameter. Dieser Prozeß der sogenannten *Selbstadaptation* von (anderenfalls exogen einzustellenden) Strategieparametern erlaubt das selbständige Anpassen der Suche an die topologischen Erfordernisse des Problems [Sch87, HB92, Bäc92a]. Zusätzlich zu den Standardabweichungen können bei ES auch bis zu $n \cdot (n-1)/2$ Kovarianzen der verallgemeinerten n-dimensionalen Normalverteilung in den Selbstadaptationsprozeß einbezogen werden, was auch linear *korrelierte Mutationen* zuläßt und damit das Voranschreiten bei komplexer lokaler Topologie beschleunigen kann [Sch81, HB93].

Gegenüber der wichtigen Rolle normalverteilter Mutationen bei ES wird Mutation in GAs traditionell als untergeordneter Operator betrachtet. Mutation ist dort als eine zufällig gleichverteilte, aber wenig wahrscheinliche Bit-Inversion realisiert, wobei die Mutationswahrscheinlichkeit p_m pro Bit typischerweise im Bereich $p_m \approx 0.001$ liegt, meist unabhängig von der Zahl der Variablen. Neuere Analysen zeigen, daß die Bedeutung der Mutation in GAs bisher erheblich unterschätzt wurde [Bäc92b, Bäc93, Müh92].

Andererseits wird in GAs die Rolle der Rekombination (*Crossover*) als wesentlicher Suchoperator betont, wobei die einfachste Form aus zwei zufällig gewählten Individuen dadurch zwei neue erzeugt, daß die Informationsgehalte ab einer zufällig bestimmten Position zwischen den Individuen ausgetauscht werden [Hol75]. Dies kann in natürlicher Weise auf mehrere Bruchstellen und Austauschstücke erweitert werden [DeJ75, Gol89] bis hin zu *gleichförmigem* (*uniform*) Crossover, wobei für jedes Bit einzeln entschieden wird, ob ein Austausch erfolgt [Sys89]. Auch für Crossover muß bei GAs eine Anwendungswahrscheinlichkeit p_c pro Individuum angegeben werden (meist $p_c \approx 0.6$). Da die Crossover-Bruchstellen keinerlei Rücksicht nehmen auf ihre Bedeutung für die Bitkodierung der Variablen, also auch innerhalb des Teilbitstrings einer Objektvariablen liegen können, spielt die Rekombination bei GAs eine Rolle, welche über das Mischen phänotypischer Merkmale hinausgeht. Sie schließt im Grunde genommen die innovative Rolle der Mutationen bei ES mit ein.

Versuche mit ES haben eindeutig ergeben, daß Rekombination insbesondere für den Selbstadaptationprozeß der Strategieparameter erforderlich ist, wobei auf den Strategieparametern *intermediäre* Rekombination zu empfehlen ist. Dabei werden zur Nachkommengenerierung die Strategieparameter einem Mittelungsprozeß unterworfen, wobei im Standardfall jeweils der Mittelwert der Parameter beider Eltern den entsprechenden Parameter des Nachkommen bestimmt: $\sigma_i^{neu} = (\sigma_i^A + \sigma_i^B)/2$ (die hochgestellten Indices A und B bezeichnen die beiden beteiligten Eltern). Für die Objektvariablen hingegen ist die diversitätserhaltend wirkende *diskrete* Rekombination vorzuziehen, welche die Objektvariablen zufällig von einem der Eltern übernimmt (analog zu uniform Crossover auf reellwertigen Objektvariablen).

Schließlich verbleibt die Selektion, welche in ES nur auf der Grundlage einer rangbasierten Abstufung der Individuen nach ihren Zielfunktionswerten arbeitet und rein deterministisch ist, während GAs den Individuen Selektionswahrscheinlichkeiten $p(\vec{x}_i)$ zuordnen, welche sich aus dem Anteil der individuellen Fitneß an der Gesamtfitneß der Population ergeben (*proportionale Selektion*):

$$p(\vec{x}_i) \quad = \quad \frac{f(\vec{x}_i)}{\sum_{j=1}^{\mu} f(\vec{x}_j)} \; . \tag{7}$$

Dies impliziert stets positive Fitneßwerte und das Vorliegen einer Maximierungsaufgabe; anderenfalls muß mit Skalierungsfunktionen nachgeholfen werden [Gol89]. Proportionale Selektion hält die Populationsgröße konstant, d.h. $\mu = \lambda$.

Bei ES werden hingegen auf rein deterministische Weise μ Elternindividuen der Generation $t+1$ nur aus den $\lambda > \mu$ Nachkommen ((μ,λ)-Selektion) oder aus Nachkommen und Eltern ($(\mu+\lambda)$-Selektion, $Q = P(t)$) selektiert, wobei im letzteren Fall zwar einerseits keine Verschlechterungen möglich sind, andererseits aber sowohl die Selbstadaptation behindert als auch die Anwendung bei veränderlicher oder gestörter Zielfunktion unmöglich gemacht wird.

Für das Verhältnis der Anzahl der Nachkommen zu der der Eltern gibt Schwefel eine Empfehlung $\lambda/\mu \approx 7$ [Sch87].

Obwohl zunächst das Repräsentationsschema der Individuen den wesentlichen Unterschied zwischen ES und GAs auszumachen scheint, spielt der Selbstadaptationsprozeß eine wesentlich wichtigere Rolle, da auf diese Weise die Notwendigkeit zu langwieriger Feineinstellung strategischer Parameter „von Hand" fast vollständig entfällt. Drei Kriterien für eine erfolgreiche Selbstadaptation wurden von Schwefel empirisch identifiziert [Sch90]:

- Benutzung einer (μ,λ)-Strategie, um das Aussterben unangepaßter Individuen zu gewährleisten (eine Art *Vergessen*, wie sie beim Simulated Annealing [vLA87] und neueren Lernverfahren bei künstlichen neuronalen Netzen auch angewandt wird [vHIKV89]).
- Verwendung eines nicht zu starken Selektionsdruckes, d.h. μ sollte deutlich größer als Eins gewählt werden (häufig wird eine (15,100)-Strategie verwendet), wobei jedes der μ Elternindividuen — unabhängig von der Rangordnung gemäß ihrer Fitneß — im Mittel λ/μ Nachkommen erzeugt.
- Anwendung von (insbesondere intermediärer, s.o.) Rekombination auch auf die Strategieparameter.

GAs reagieren demgegenüber recht empfindlich auf Modifikationen kritischer Parameter wie p_c und p_m und müssen häufig je nach Anwendungsfall bezüglich dieser (und weiterer) Parameter „von Hand" optimiert werden [DeJ75, SCED89].

Für detailliertere Darstellungen von ES und GAs verweisen wir auf die Originalliteratur ([Rec73, Sch77, Sch81] zu ES, [Hol75, Gol89, Dav91] zu GAs) und auf neuere Übersichtsartikel [SB92, BS93, Bäc92a].

Eine Rekapitulation der in Abschnitt 2 spezifizierten Charakteristika von Simulationsmodellen als zu optimierende Zielfunktion macht den potentiellen Wert evolutionärer Algorithmen in diesem Anwendungsbereich deutlich. Evolutionäre Algorithmen arbeiten lediglich auf der Basis der Zielfunktionswerte der Individuen einer Population, sind also direkte Optimierverfahren. Der Problemdimensionalität sind durch die Algorithmen keine prinzipiellen Grenzen gesetzt [Sch77]. Für spezielle Varianten der Strategien existieren globale Konvergenzbeweise [Rud92, Rud93]. Die auf eine Generation beschränkte Lebensdauer der Individuen (bei der (μ,λ)-Strategie) garantiert Robustheit gegenüber stochastischen Störungen sowie Anwendbarkeit im Falle zeitlich variierender Zielfunktionen (eine noch nicht abgeschlossene Arbeit scheint allerdings darauf hinzudeuten, daß eine maximale Lebensdauer von etwa fünf Generationen noch günstigere Resultate liefert). Bei Aufgaben der mehrkriteriellen Optimierung kann, wie Kursawe gezeigt hat, eine erweiterte ES beliebig viele (entsprechend der Populationsstärke μ) Punkte der Pareto-Menge in einem Zuge generieren [Kur91, Kur92]. Restriktionen können — sofern sie das Erfolgsgebiet für Mutationen nicht zu sehr verkleinern — durch Verwerfen von restriktionsverletzenden Individuen (*Letalmutationen*) und entsprechende Neugenerierung von Individuen durch Mutation berücksichtigt werden [Sch77]. Schließlich ist — zumindest für ES — die Zahl der noch vom Benutzer einzustellenden Parameter minimal.

Erste Experimente zur Modelloptimierung mit evolutionären Algorithmen deuten darauf hin, daß die Algorithmen den gestellten Anforderungen gerecht werden [HS82, HB93].

4. Ausblick

Der Aspekt einer Kopplung von Simulation und Optimierung ist bisher bei den gängigen kommerziellen Simulationsumgebungen kaum berücksichtigt worden, sondern wenigen, vornehmlich in der Forschung eingesetzten, Tools vorbehalten [HS78]. Sowohl hinsichtlich der Leistungsfähigkeit moderner Rechner — wobei insbesondere auf die heute gegebene Möglichkeit der Auswertung der Simulationsmodelle durch parallele evolutionäre Algorithmen hingewiesen sei [Hof91, GS89, SM91] — als auch der Verfügbarkeit entsprechender Optimierverfahren auf der Basis der biologischen Evolution — der evolutionären Algorithmen — sollte sich eine solche Kopplung in näherer Zukunft durchsetzen können.

Die in der Literatur dokumentierten erfolgreichen Anwendungen evolutionärer Algorithmen auf schwierige Optimierungsprobleme in nahezu allen technischen Bereichen (eine am Lehrstuhl erstellte Literaturübersicht [BHS92] enthält zur Zeit mehr als 260 Einträge) geben eine eindeutige Demonstration des praktischen Nutzens dieser Verfahren, wenn auch das theoretische Fundament zum vollen Verständnis ihrer Grundprinzipien gerade erst gelegt wird.

Literatur

[Bäc92a] Thomas Bäck. Evolutionary algorithms. *ACM SIGBIO Newsletter*, 12(2):26–31, Juni 1992.

[Bäc92b] Thomas Bäck. The interaction of mutation rate, selection, and self-adaptation within a genetic algorithm. In Männer und Manderick [MM92], S. 85–94.

[Bäc93] Thomas Bäck. Optimal mutation rates in genetic search. In Stephanie Forrest, Hrsg., *Genetic Algorithms: Proceedings of the 5th International Conference*. Morgan Kaufmann Publishers, San Mateo, CA, 1993.

[BHS92] Thomas Bäck, Frank Hoffmeister, und Hans-Paul Schwefel. Applications of evolutionary algorithms. Interner Bericht des Lehrstuhls für Systemanalyse SYS–2/92, Universität Dortmund, Fachbereich Informatik, Februar 1992.

[BS93] Thomas Bäck und Hans-Paul Schwefel. An overview of evolutionary algorithms for parameter optimization. *Evolutionary Computation*, 1(1):1–23, 1993.

[Box65] M. J. Box. A new method of constrained optimization and a comparison with other methods. *The Computer Journal*, 8:42–52, 1965.

[Dav91] Lawrence Davis, Hrsg. *Handbook of Genetic Algorithms*. Van Nostrand Reinhold, New York, 1991.

[FWBW92] Stephan Freyer, Dirk Weuster-Botz, und Christian Wandrey. Medienoptimierung mit Genetischen Algorithmen. *BioEngineering*, 8(5+6):16–25, 1992.

[Gol89] David E. Goldberg. *Genetic algorithms in search, optimization and machine learning*. Addison Wesley, Reading, MA, 1989.

[GS89] Martina Gorges-Schleuter. ASPARAGOS: an asynchronous parallel genetic optimization strategy. In Schaffer [Sch89], S. 422–427.

[HB93] Ulrich Hammel und Thomas Bäck. Einsatz evolutionärer Algorithmen zur Optimierung von Simulationsmodellen. *Beitrag zum 6. Workshop Simulation & Künstliche Intelligenz*, 1993. (Im Druck).

[Hau89] Martin Haupt. *Betriebspunktoptimierung kunststoffverarbeitender Prozesse*. Dissertation, RWTH Aachen, 1989.

[HS78] Rainer Heckler und Hans-Paul Schwefel. Superimposing direct search methods for parameter optimization onto dynamic simulation models. In H.J. Highland, N.R. Nielsen, und L.G. Hull, Hrsg., *Proc. Winter Simulation Conf.*, Band 1, S. 173–181, Miami Beach, FL, Dezember 1978.

[vHIKV89] J. L. van Hemmen, L. B. Ioffe, R. Kühn, und M. Vaas. Increasing the efficiency of a neural network through unlearning. *Physica A*, 163(1):386–392, 1989.

[Hof91] Frank Hoffmeister. Scalable parallelism by evolutionary algorithms. In D. B. Grauer und M. Pressmar, Hrsg., *Parallel Computing and Mathematical Optimization*, Band 367 der Reihe *Lecture Notes in Economics and Mathematical Systems*, S. 177–198, Springer, Berlin, 1991.

[HB92] Frank Hoffmeister und Thomas Bäck. Genetic self–learning. In *Proceedings of the First European Conference on Artificial Life, December 11-13, 1991*, S. 227–235. The MIT Press, Cambridge, MA, 1992.

[Hol75] John H. Holland. *Adaptation in natural and artificial systems.* The University of Michigan Press, Ann Arbor, MI, 1975.

[HJ61] R. Hooke und T.A. Jeeves. Direct search solution of numerical and statistical problems. *Journal of the ACM*, 8:212–229, 1961.

[HS82] Eckhard Höpfinger und Hans-Paul Schwefel. Trade between industrialized and resource-rich countries — a game-theoretic approach. In W. Eichhorn, R. Henn, K. Neumann, und R. W. Shephard, Hrsg., *Economic Theory of Natural Resources*, S. 175–185. Physica-Verlag, Würzburg, 1982.

[DeJ75] Kenneth A. De Jong. *An analysis of the behaviour of a class of genetic adaptive systems.* Dissertation, University of Michigan, 1975.

[KS70] Jürgen Klockgether und Hans-Paul Schwefel. Two–phase nozzle and hollow core jet experiments. In D.G. Elliott, Hrsg., *Proc. 11th Symp. Engineering Aspects of Magnetohydrodynamics*, S. 141–148, California Institute of Technology, Pasadena CA, März 1970.

[Kur91] Frank Kursawe. A variant of Evolution Strategies for vector optimization. In Hans-Paul Schwefel und Reinhard Männer, Hrsg., *Parallel Problem Solving from Nature — Proceedings 1st Workshop PPSN I*, Band 496 der Reihe *Lecture Notes in Computer Science*, S. 193–197. Springer, Berlin, 1991.

[Kur92] Frank Kursawe. Evolution strategies for vector optimization. In Gwo-Hshiung Tzeng und Po Lung Yu, Hrsg., *Preliminary Proceedings of the Tenth International Conference on Multiple Criteria Decision Making, Taipei, 19.–24.7.1992*, S. 187–193. National Chiao Tung University, Taipei, 1992.

[vLA87] P. J. M. van Laarhoven und E. H. L. Aarts. *Simulated Annealing: Theory and Applications.* Mathematics and Its Applications. Kluwer, Dordrecht, 1987.

[MM92] Reinhard Männer und Bernard Manderick, Hrsg. *Parallel Problem Solving from Nature 2.* Elsevier, Amsterdam, 1992.

[Mey84] Walter J. Meyer. *Concepts of Mathematical Modeling.* McGraw Hill, New York, 1984.

[Müh92] Heinz Mühlenbein. How genetic algorithms really work: I. mutation and hillclimbing. In Männer und Manderick [MM92], S. 15–25.

[NM65] J.A. Nelder und R. Mead. A simplex method for function minimization. *The Computer Journal*, 7:308–313, 1965.

[Pow62] M. J. D. Powell. An iterative method for finding stationary values of a function of several variables. *The Computer Journal*, 5:147–151, 1962.

[Rec73] Ingo Rechenberg. *Evolutionsstrategie: Optimierung technischer Systeme nach Prinzipien der biologischen Evolution.* Frommann–Holzboog, Stuttgart, 1973.

[Ros60] H. H. Rosenbrock. An automatic method for finding the greatest or least value of a function. *The Computer Journal*, 3(3):175–184, Oktober 1960.

[Rud92] Günter Rudolph. Parallel approaches to stochastic global optimization. In W. Joosen und E. Milgrom, Hrsg., *Parallel Computing: From Theory to Sound Practice, Proceedings of the European Workshop on Parallel Computing*, S. 256–267. IOS Press, Amsterdam, 1992.

[Rud93] Günter Rudolph. Convergence analysis of canonical genetic algorithms. *IEEE Transactions on Neural Networks, Special Issue on Evolutionary Programming*, 1993. (Im Druck).

[Sch89] J. David Schaffer, Hrsg. *Proceedings of the 3rd International Conference on Genetic Algorithms and Their Applications.* Morgan Kaufmann Publishers, San Mateo, CA, Juni 1989.

[SCED89] J. David Schaffer, Richard A. Caruana, Larry J. Eshelman, und Rajarshi Das. A study of control parameters affecting online performance of genetic algorithms for function optimization. In Schaffer [Sch89], S. 51–60.

[Sch77] Hans-Paul Schwefel. *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*, Band 26 der Reihe *Interdisciplinary systems research.* Birkhäuser, Basel, 1977.

[Sch79] Hans-Paul Schwefel. Direct search for optimal parameters within simulation models. In *Proc. 12th Annual Simulation Symposium*, S. 91–102, Tampa, FL, März 1979.

[Sch81] Hans-Paul Schwefel. *Numerical Optimization of Computer Models.* Wiley, Chichester, 1981.

[Sch87] Hans-Paul Schwefel. Collective phenomena in evolutionary systems. In *Preprints of the 31st Annual Meeting of the International Society for General System Research, Budapest*, Band 2, S. 1025–1033, Juni 1987.

[Sch90] Hans-Paul Schwefel. Systems analysis, systems design, and evolutionary strategies. *Systems Analysis, Modeling, and Simulation*, 7(11/12):853–864, 1990.

[SB92] Hans-Paul Schwefel und Thomas Bäck. Künstliche Evolution — eine intelligente Problemlösungsstrategie ? *KI*, 6(2):20–27, Juni 1992.

[SM91] Piet Spiessens und Bernard Manderick. A massively parallel genetic algorithm: Implementation and first analysis. In Richard K. Belew und Lashon B. Booker, Hrsg., *Proceedings of the 4th International Conference on Genetic Algorithms and their Applications*, S. 279–286. Morgan Kaufmann Publishers, San Mateo, CA, 1991.

[Sys89] Gilbert Syswerda. Uniform crossover in genetic algorithms. In Schaffer [Sch89], S. 2–9.

[TŽ89] Aimo Törn und Antanas Žilinskas. *Global Optimization*, Band 350 der Reihe *Lecture Notes in Computer Science.* Springer, Berlin, 1989.