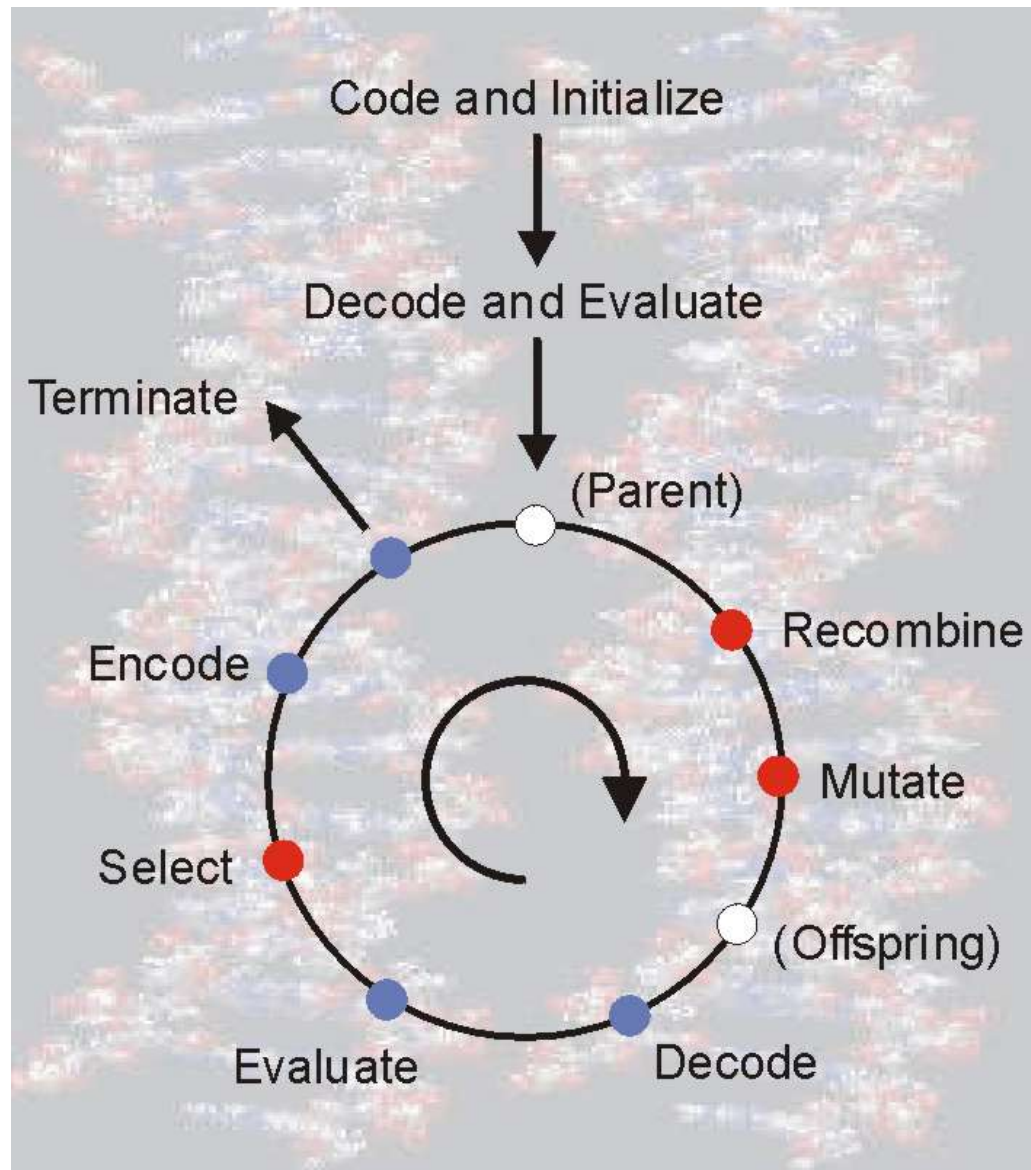


Evolutionäre Technologie

- Historie, GA und ES -

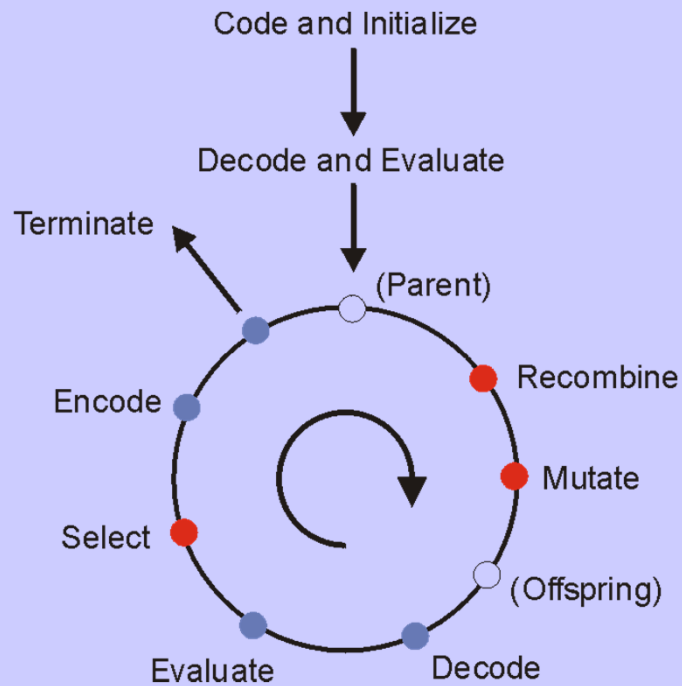
Evolutionäre Algorithmen (EA) - Grundprinzipien



- Übertragung von Aspekten der “Synthese der Evolutionsbiologie” auf die Optimierung bzw. Anpassung
- *Unterschiedlich* starke Ausrichtung auf die Genetik
- Schwerpunkt auf Variation und Selektion
- Anwendung in unterschiedlichen Problembereichen:
 - Scheduling Probleme
 - Belegungspläne
 - Designoptimierung
 - Prädiktion
 - Systemstrukturierung
 - Chemisch/Pharmazeutisch, etc.

Evolutionäre Algorithmen - Formal

Evolutionärer Zyklus



Algorithmus

```

t:=0
initialisiere  $P_G(0) = \{ g_1(0), \dots, g_\mu(0) \}$ 
decodiere und bewerte
 $P_G(0): \Psi( c^{-1}( P_G(0) ) )$ 

do
    rekombiniere  $P'_G(t) := r( P_G(t) )$ 
    mutiere  $P''_G(t) := m( P'_G(t) )$ 
    decodiere  $P''_P(t) := c^{-1}( P''_G(t) )$ 
    evaluiere  $P''_P(t): \Psi( P''_P(t) )$ 
    selektiere  $P_P(t+1) := s( P''_P(t) \cup Q )$ 
    codiere  $P_G(t+1) := c(P_P(t+1) )$ 
     $t := t + 1$ 

until stop
    
```

Nomenklatur

$P_G(t)$ Population, Genotyp Level, Generation t
 $P_P(t)$ Population, Phänotyp Level, Generation t
 Q Zusätzliche Individuen, z.B. $P_G(t)$

$g_1(t)$ Genotyp des ersten Individuums
 $\Psi(\dots)$ Fitnessfunktion
 r, m Rekombinations- und Mutationsoperator
 c, s Codierungs- und Selektionsoperator

Historie I

■ Simulation der Evolution von genetischen Systemen (Fraser, Crosby, White 1957-1968)

- diploide Binärstrings mit einfacher Genotyp-Phänotyp Abbildung
- n-Punkt Crossover mit Wahrscheinlichkeiten für alle Loci
- deterministische Selektion: Mittelwerte und Extremwerte

➔ Phänomen des Linkage und Epistasie
Rolle von Reproduktionsraten und Selektionsdruck

■ Evolution von Produktionssystemen (Box et al. 1954-1958)

- Änderungen (Mutation) der Parameter einer chemischen(!) Fabrik
- Änderungen durch Expertengremium
- Selektion des besten Settings
- Experimentbasierte Evaluation



IBM 704 (Ende 1956)

4K-32K Speicher, 10^3 FLOPS

- Microsoft Xbox 360, $115 \cdot 10^9$ FLOPS
Faktor 10^8
- Standard PC, $5\text{-}20 \cdot 10^9$ FLOPS
Faktor 10^7
- IBM BlueGene/L
Livermore National Laboratory
 $4.78 \cdot 10^{14}$ FLOPS
Faktor 10^{11}

Historie II

■ Evolution von Computerprogrammen

(Friedberg et al.1958 -1970)

- Satz von Machinensprachebefehlen, als Binärstring codiert, die einfache Berechnungen durchführen, z.B. Addition
- Nutzen des Prinzips der *intrinsischen Parallelität*
- das *credit assignment problem* wurde dadurch gelöst, dass gute bzw. schlechte Bewertungen über das Programm definiert wurden in dem die Machinensprachebefehle genutzt wurden

➡ wenig Erfolg, oftmals schlechter als Zufallssuche, viel Kritik in der Wissenschaftsgemeinde

■ Artificial Life and Evolving Strategies

(Barricelli et al.1954 -1963)

- Welche Randbedingungen haben Gene zu erfüllen, um die Entwicklung höherer Lebensformen zu realisieren?
 - Reproduktion
 - Mutation (kleine Veränderungen)
 - Symbiose (Cooperatives “Verhalten” zwischen Individuen)
- Simulation numerischer Elemente auf einem Netz, Evolution von wiederkehrenden stabilen Mustern, jedes Update ist eine Generation – ähnlich wie in Conways “Game of Life”

➡ erster (und seltener) Versuch über das Prinzip der natürlichen Selektion hinaus zu gehen

Conway's "Game of Life" – 70's

Zellen auf einem rechteckigen Gitter mit zwei Zuständen und Transitionsregeln

- jede lebende Zelle mit weniger als zwei Nachbarn stirbt aus (Einsamkeit)
- jede lebende Zelle mit mehr als drei Nachbarn stirbt aus (Überpopulation)
- jede tote Zelle mit genau drei Nachbarn wird geboren
- jede Zelle mit zwei oder drei Nachbarn verändert ihren Zustand nicht

Bsp. für Emergenz und Selbstorganization

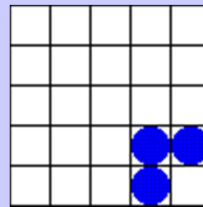
statische Strukturen *block*



dynamische Strukturen *blinker*



sich bewegende Strukturen *glider*



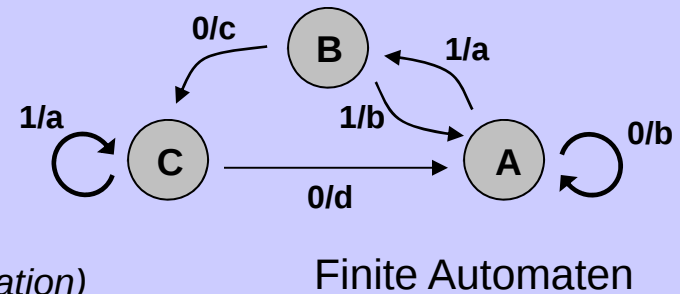
pusierende (*lebendige*) Strukturen
shooter plus glider

Historie: Evolutionary Programming (EP) (L. Fogel, 1962)

Künstliche Intelligenz durch simulierte Evolution

Grundsatz:

Intelligent behaviour is the composite ability to predict one's environment coupled with a translation of each prediction into a suitable response in light of a given goal (pay-off maximization)



- Gegenstand der Optimierung waren Finite Automaten, die solches “intelligentes Verhalten” hervorbringen sollten
- kleine Populationen (1-3), Mutation, Codierung variabler Länge

➔ Hauptkritik an der Zufälligkeit des Prozesses nicht an der natürlichen Selektion

Evolutionary Programming wird weiterhin genutzt und entwickelt

Anwendungsbereiche

- Optimierung neuronaler Strukturen (z.B. Damespiel)
- Zeitreihenprädiktion
- etc.



Genetische Algorithmen (GA): Hintergrund und Historie (Holland, 1967)

Mikroskopische Modell: Populationsgenetik

Fokussierung auf den Genotyp Level: Gene sind Gegenstand der Selektion

Schema Processing and the K-Armed Bandit

- Adaptation an eine gegebene Umwelt
- Adaptation ist eine Suchtrajektorie durch einen Zustandsraum mgl. Lösungen

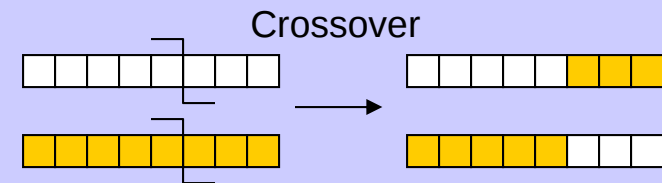
Aufgabe: Beste Strategie für das Ziehen von n-Zufallsvariablen, die einen definierten aber unbekannten Mittelwert und eine unbekannte Varianz haben



Trade-off zwischen Maximierung von Gewinn und Information

Kanonischer Genetischer Algorithmus

- genetische Codierung (String, Alphabet, z.B. binär)
- Veränderungen auf dem Genotyplevel
 - Crossover und Inversion
 - Punkt-Mutation (sekundär)  → 



- proportionale Selektion

die Wahrscheinlichkeit Nachkommen zu erzeugen ist proportional zur eigenen Fitness

Problem: Fitness muss immer positiv sein

$$p(x_i) = \frac{f(x_i)}{\sum_{j=1}^n f(x_j)}$$

Genetische Algorithmen

Wie funktioniert der kanonische Genetische Algorithmus?

- Information in der Population (String – Fitness) ist limitiert

[1 0 1 0 0]	0.25
[0 0 0 0 1]	0.1
[1 0 1 1 0]	0.4
[1 1 1 1 0]	0.75

Zentrale Idee der Schemata oder Templates

- Erweitere das Alphabet um ein *don't care* symbol { 0, 1, # }

[0 1 1 0 0] $\xrightarrow[\text{Schemata}]{\text{mögliche}}$ [0 # 1 0 0] [0 1 # # #]

Annahme:
kausale Verbindung
zwischen
Schemata und Fitness

- Bei einem Alphabet der Kardinalität k und einer Stringlänge l gibt es k^l Zustände, jedoch $(k+1)^l$ Schemata Vereinfachung?

- Jeder String (Länge l) in einer Population der Grösse n gehört zu 2^l Schemata, das heisst in der gesamten Population liegt Information über m schemata vor, wobei $2^l \leq m \leq n 2^l \rightarrow$ Prinzip der impliziten Parallelität

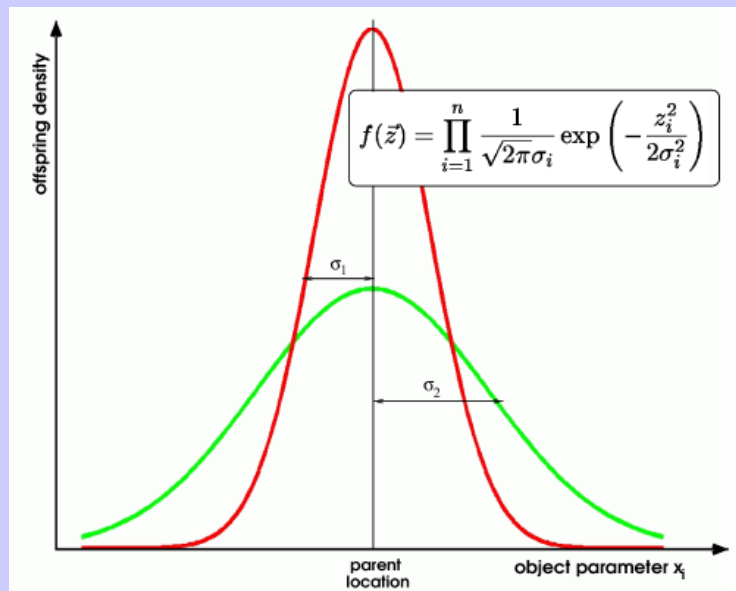
Das Prinzip der Schemata ist einer Suche auf Unterräumen nicht unähnlich
- Building Block Hypothese -

Evolutionsstrategie (ES): Hintergrund

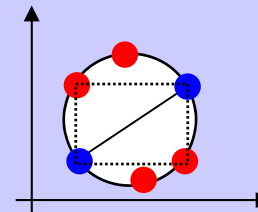
Makroskopische Modell

Fokussierung auf die Phänotypebene: Verhalten ist Gegenstand der Selektion

- Komponenten des Lösungsvektors sind phänotypische Verhaltensmerkmale
- Annahme: Wie auch immer der genetische Unterbau realisiert ist, die resultierende phänotypische Änderung ist Gaussverteilt
- Mutation ist der primäre Variationsoperator



- $c = \text{id}_{\mathbb{R}}$
- $g_1(t) \in \mathbb{R}^n$ - $x(t) := g_1(t)$ (Objektvariable)
- r = kontinuierliche Rekombination, z.B.

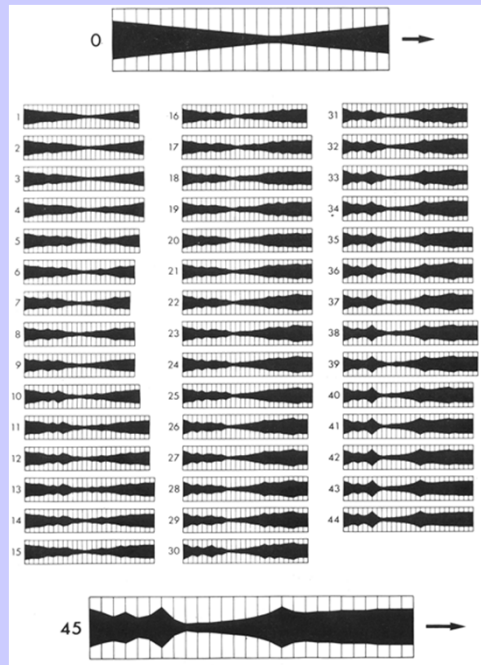


- s = deterministische Selektion
(μ, λ) und ($\mu + \lambda$) Selektion
- m = Mutation durch Addition normalverteilter Zufallszahlen
- Erweiterung mit multi-rekombination und begrenzte Lebenszeit: ($\mu, \lambda, \kappa, \delta$)
 $\mu = \delta \rightarrow$ center of mass Rekombination

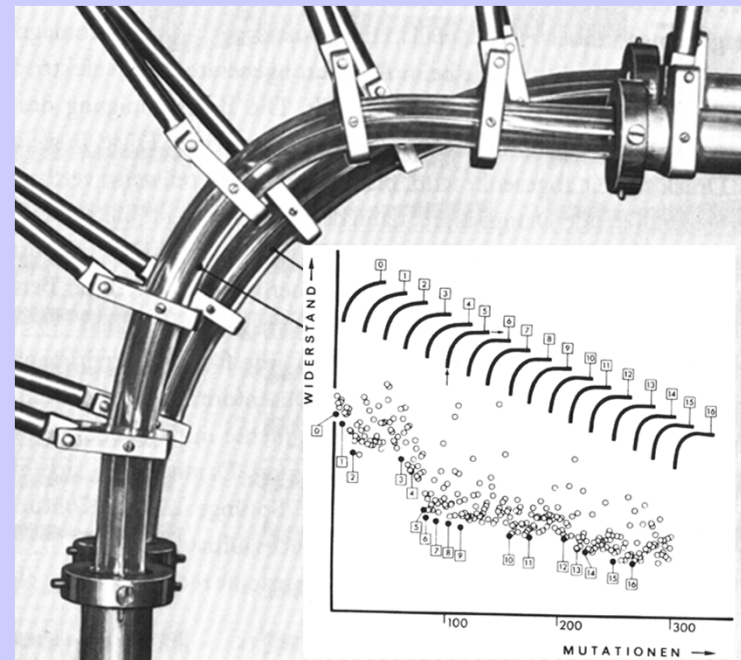
Evolutionsstrategie: Historie (Rechenberg, Bienert, Schwefel, 1965)

Evolutionäre Optimierung von fluid-mechanischem Design (2 und 3 dimensional)

- zunächst ein Elter und einen Nachkommen (1+1) Strategie
- nur Mutation als Veränderung, anfangs binomial verteilt später normal verteilte Zufallszahlen
- erste theoretische Analyse des Optimierungsprozesses für einfache Qualitätslandschaften
Korridormodell und Kugelmodell
Rechenbergs 1/5 Regel: der Quotient aus erfolgreichen und erfolglosen Mutanten sollte 1:5 sein

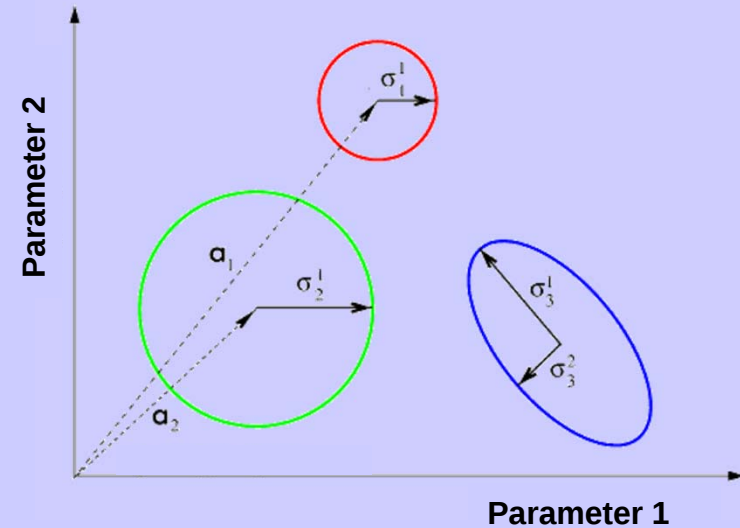
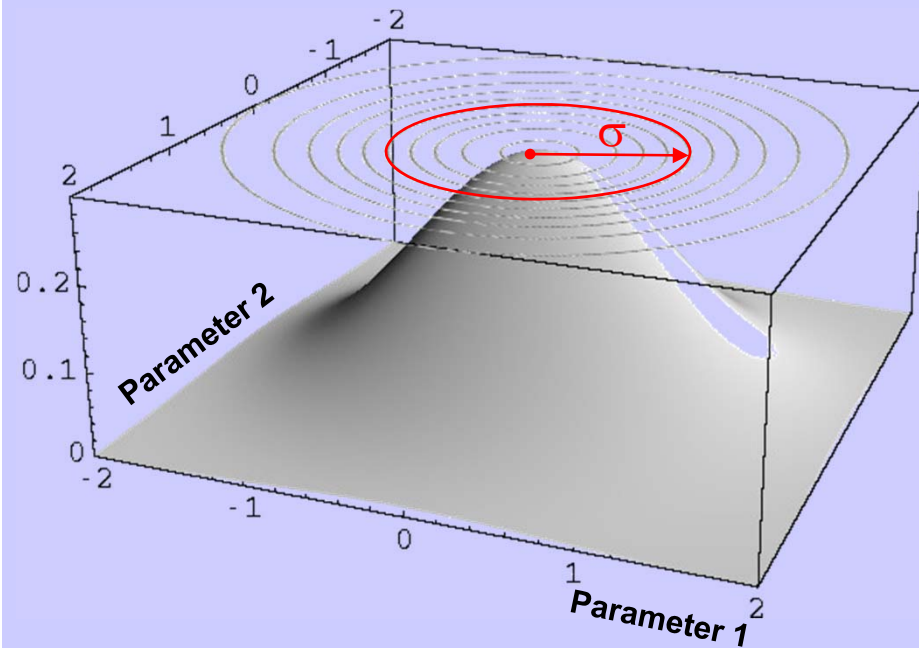


Evolutionary Optimierung einer 2-Phasen Überschalldüse (Schwefel et al. 1968)



Evolutionäre Optimierung einer 90° Grad Schlauchbiegung (Rechenberg et al. 1966)

Selbstadaptation 1



- die Varianzen σ^2 (Strategieparameter) beeinflussen die Mutationstärke (Schrittweite), d.h. die Veränderung die zu den Objektparametern addiert wird
- die optimale Grösse der Varianzen ist von der lokalen Topologie des Suchraumes abhängig und vom Zustand der Population

Selbstadaptation 2

➔ die Varianzen σ_i^2 werden selbst Gegenstand der evolutionären Optimierung
selbst-adaptation

- mutative self-adaptation, n : Suchraumdimension

$$z \sim \mathcal{N}(0, 1)$$

$$\tau = \frac{1}{\sqrt{2n}}$$

$$\tau' = \frac{1}{\sqrt{2\sqrt{n}}}$$

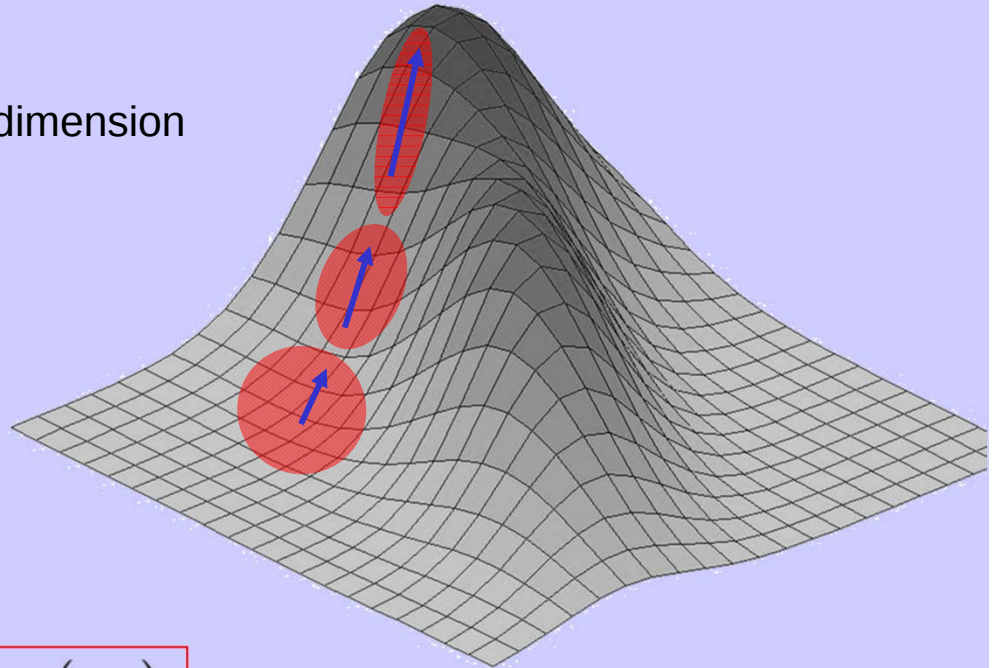
- für jede Suchraumdimension i

$$z_i \sim \mathcal{N}(0, 1)$$

$$\sigma_i(t) = \sigma_i(t-1) \exp(\tau' z) \exp(\tau z_i)$$

$$\tilde{z}_i(t) = \mathcal{N}(0, \sigma_i(t)^2)$$

$$x_i(t) = x_i(t-1) + \tilde{z}_i(t)$$

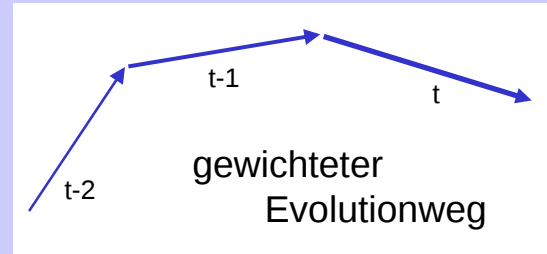


Selbstadaptation: Derandomisiert und Kovarianzmatrix

- eine (derandomisiert) Zufallszahl für die Anpassung der Objekt- und Strategieparameter; die *tatsächliche* Schrittweite im Parameterraum wird zur Anpassung der Strategieparameter genutzt

➡ wenn die Mutation stärker als erwartet $E(|N(0,1)|)$ aber erfolgreich war, wird die Varianz erhöht und umgekehrt

- *kumulative Schrittweitenadaptation*; der Evolutionsweg (Gedächtnis) wird bei der Adaptation genutzt
Analogie: Ensemble vs. Zeitmittel



- die Kovarianzmatrix der Gaußsschen Wahrscheinlichkeitsdichtefunktion wird an die lokale Topologie des Suchraumes angepasst

$$\delta \sim \frac{\sqrt{\det(\Sigma^{-1})}}{(2\pi)^{n/2}} \exp\left(-\frac{1}{2}(\vec{x} - \vec{\mu})^T \Sigma^{-1}(\vec{x} - \vec{\mu})\right)$$

da Σ^{-1} positiv definit sein muss mit $\det(\Sigma^{-1}) > 0$, sind die verschiedenen Matrixelemente nicht unabhängig und die detaillierte Vorgehensweise bei der Adaptation etwas komplizierter

Selektion I

Deterministische Selektion

Selektionsdruck $\sim \lambda / \mu$

- (μ, λ) Selektion: die Eltern werden nur aus den Nachkommen selektiert, lokale Minima können einfacher überwunden werden, gute Lösungen können vergessen werden
- $(\mu + \lambda)$ Selektion: Eltern der nächsten Generation werden aus der Nachkommen und der Elternpopulation (t-1) ausgewählt
- Elitist: der (die) besten Nachkommen werden behalten (mit bzw. ohne weitere Reproduktion)

Stochastische Selektion

Selektionsdruck ist über die *take-over time* definiert

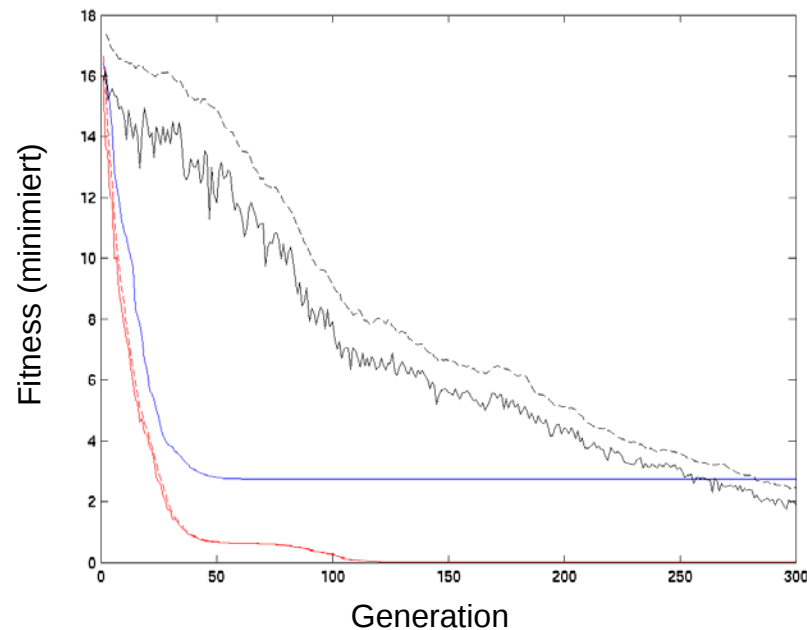
- **Rouletterad Selektion (RS):** Simulation eines Gewichteten Roulettrades, Gewichtung ist eine Funktion der Fitness der einzelnen Individuen
- **proportionale Selektion:**
Skalierung ist notwendig um einen hinreichenden Selektionsdruck aufrechtzuerhalten → RS
- **Ranking:** jedes Individuum erhält eine Selektionswahrscheinlichkeit die nur eine Funktion des Ranges des Individuums ist (Reihenfolge gemäss Fitnesswerte) → RS
- **q-tournament Selektion:** das beste Individuum aus einer zufällig gewählten Gruppe von q Individuen wird selektiert, der Prozess wird n -mal wiederholt (n – Populationsgrösse)
- **EP-style tournament:** alle Elter und Nachkommen werden mit einer zufälligen q -Gruppe verglichen. Alle gewonnenen Vergleiche werden aufsummiert, die Individuen mit dem höchsten Gewinn werden selektiert

Selektion II

Trade-off zwischen *Exploration* und *Exploitation*

- hoher Selektionsdruck (Exploitation):
 - + schnelle Qualitätssteigerung, fortschreiten entlang der *Gradientenlinie*
 - mangelnde Diversität in der Population → schwierig lokalen Minima zu entkommen
- niedriger Selektionsdruck (Exploration):
 - + hohe Diversität, globale Optimierung
 - Zerfließen der Information, Zufallssuche

Einfluss der Mutationstärke und der Selbstadaptation beachten



Evolutionsstrategie mit mutativer Selbstadaptation

- 30 dimensionale Ackley Funktion

$(\mu, \lambda) = (75, 100)$; Selektionsdruck ~ 1.3

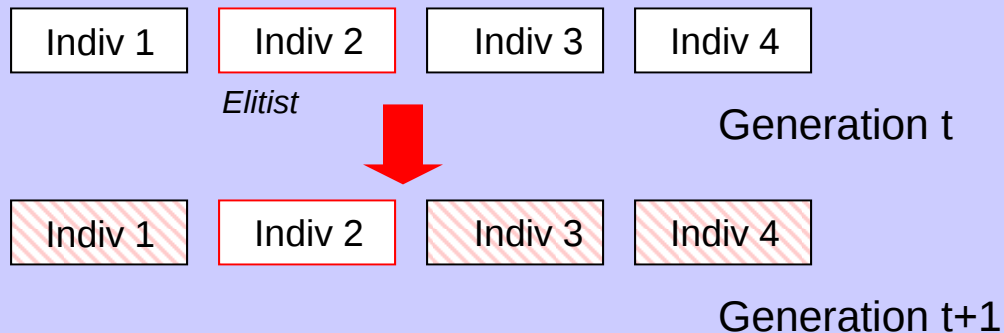
$(\mu, \lambda) = (15, 100)$; Selektionsdruck $\sim 6,7$

$(\mu, \lambda) = (1, 100)$; Selektionsdruck ~ 100

———— bestes Individuum
----- Populationsmittel

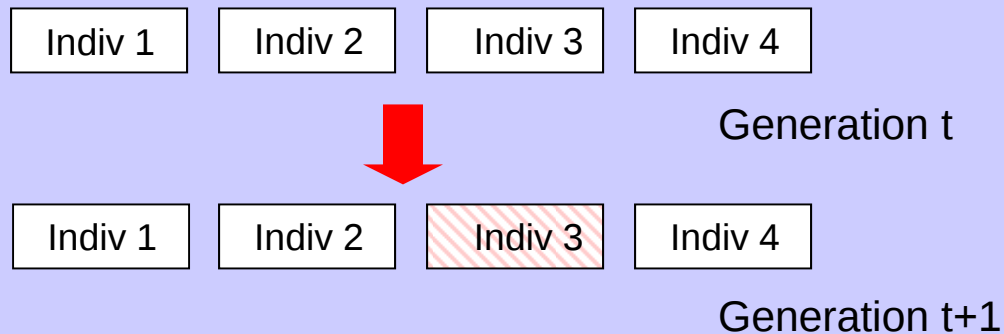
Generationsbasiert vs. *Steady-state*

Generationsbasiertes *Replacement*



- einfach zu realisieren und schnelle Konvergenz, keine Zusatzparameter
- robust gegenüber gestörten oder dynamischen Fitnessfunktionen
- gute Chromosomen können verloren werden

steady-state *Replacement*



- nur eine bestimmte Anzahl von Individuum wird ersetzt
- gute Chromosome bleiben lange erhalten
- gut geeignet für Parallelisierung
- mit Parallelisierung algorithmisch aufwendiger

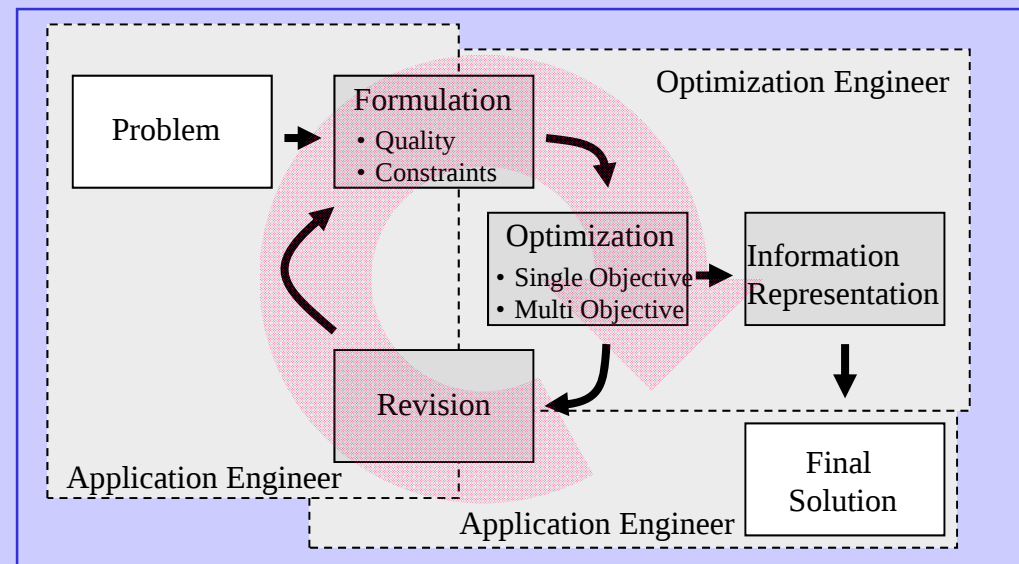
Evolutionäre Algorithmen in der Praxis

- Anwendung evolutionärer Algorithmen in der Praxis – Optimierung von Turbinenschaufeln
- Anwendung evolutionärer Algorithmen in der Praxis – Optimierung von Formel 1 Fahrzeugkomponenten

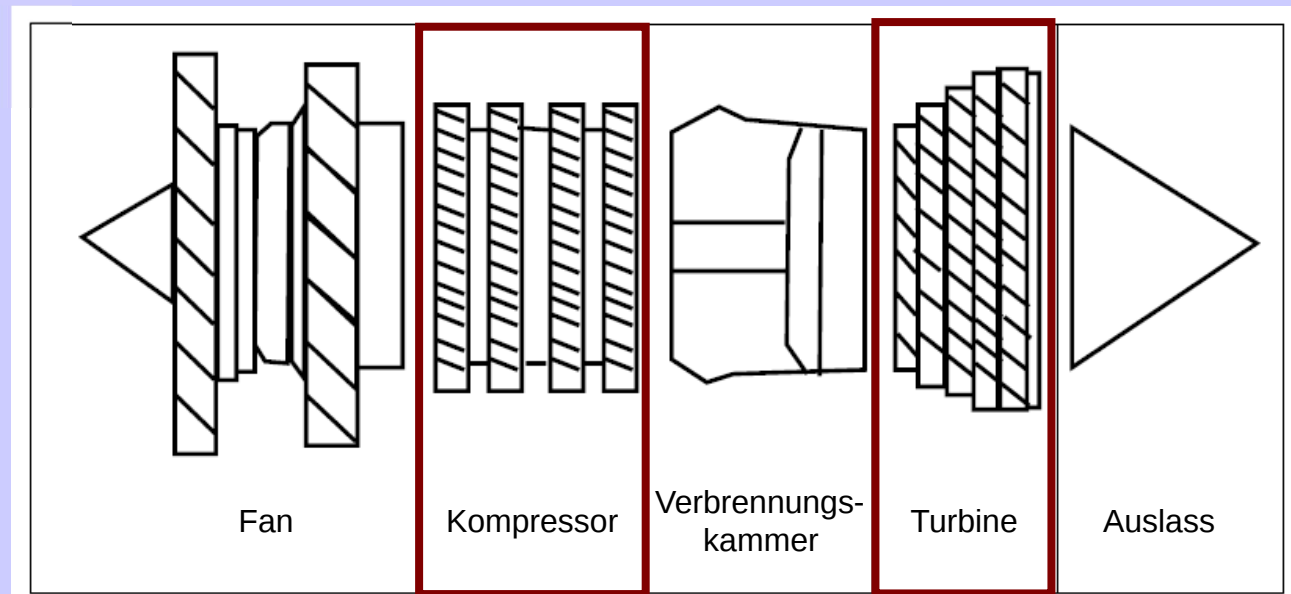
- Repräsentation,
- Evaluation
- Problemformulierung

➔ **patchwork optimization:**

„finding the problem is part of the problem“



Anwendungsbeispiel – *Strukturoptimierung von turbinen Blättern*

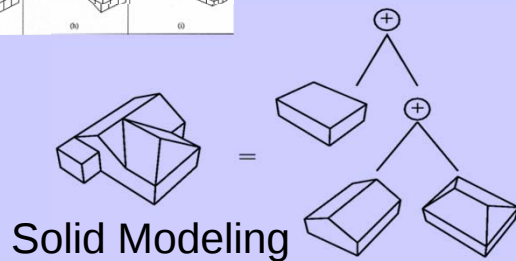
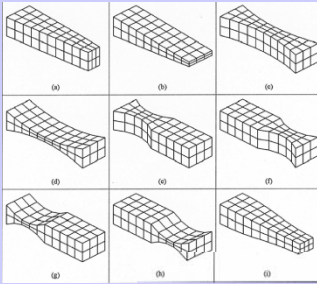


Optimierung von
Kompressorblättern

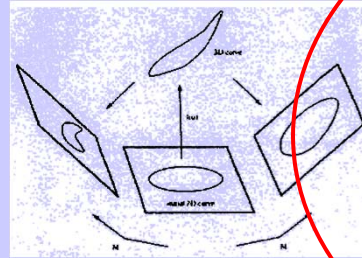
Optimierung von
Turbineblättern

Repräsentation Oberfläche und Struktur

Basis Vektoren

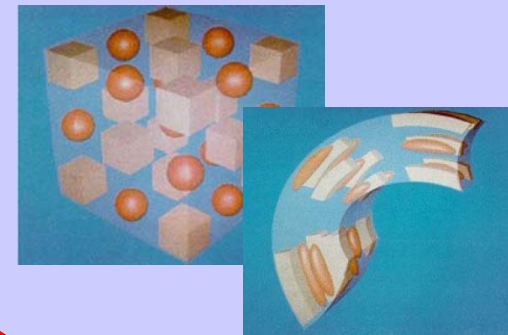


Constrained Deformation

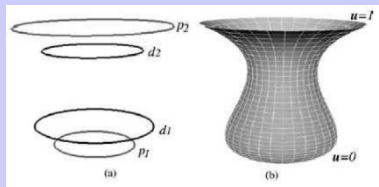


**Repräsentation
der Veränderung**

Free Form Deformation

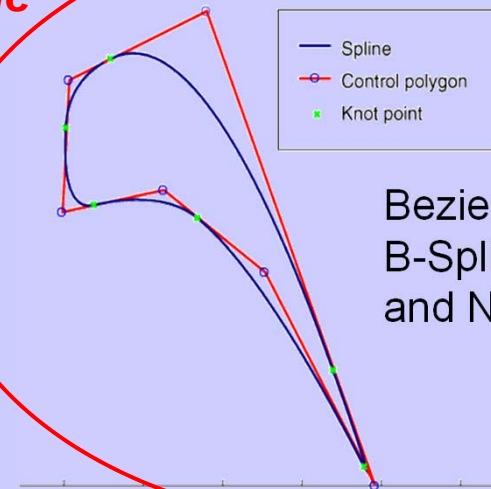
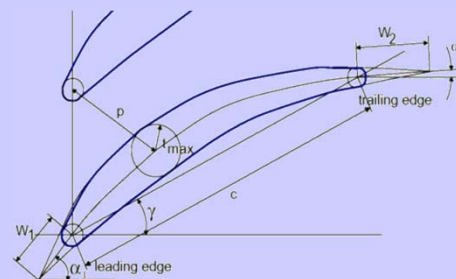


**Repräsentation
der Fläche**



Partielle Differential
Gleichungen

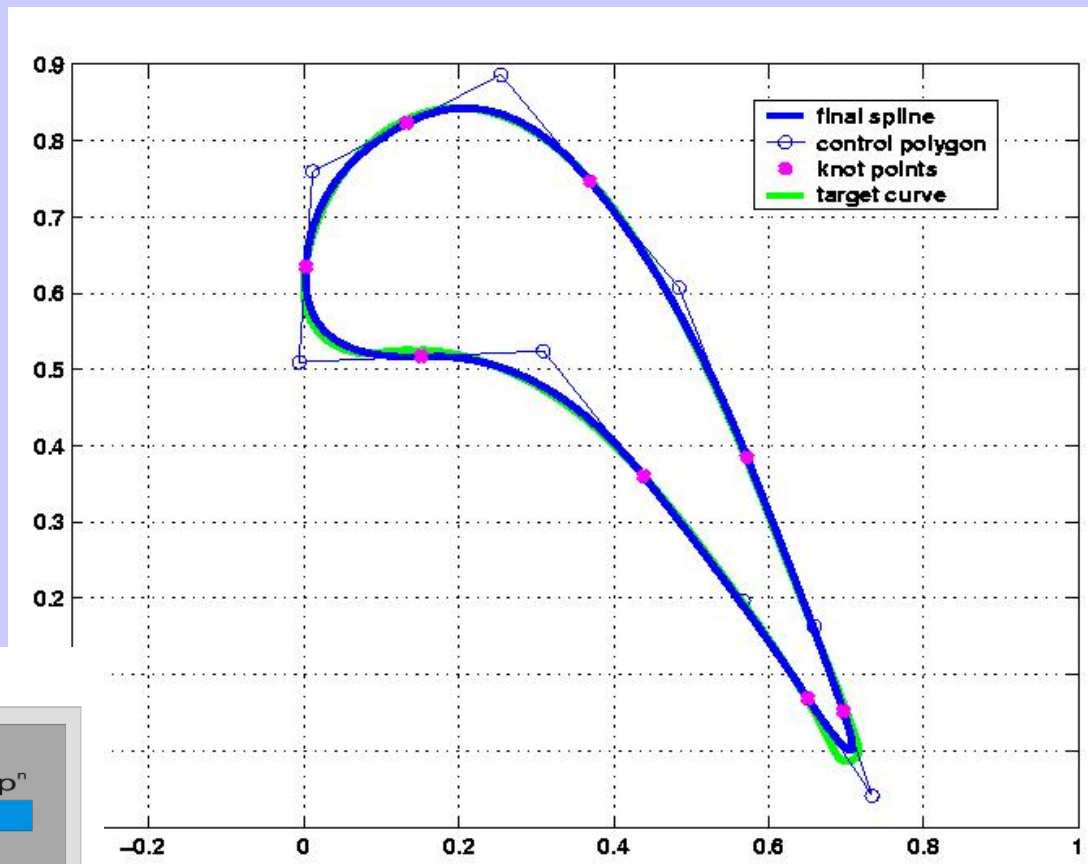
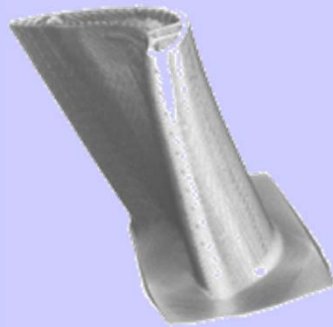
Expertenwissen



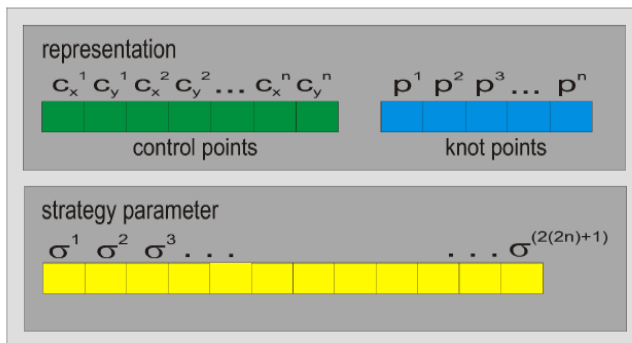
Bezier Curves,
B-Splines
and NURBS

Anwendungsbeispiel – *Strukturoptimierung von turbinen Blättern*

Repräsentation eines Turbinenblattes



individual

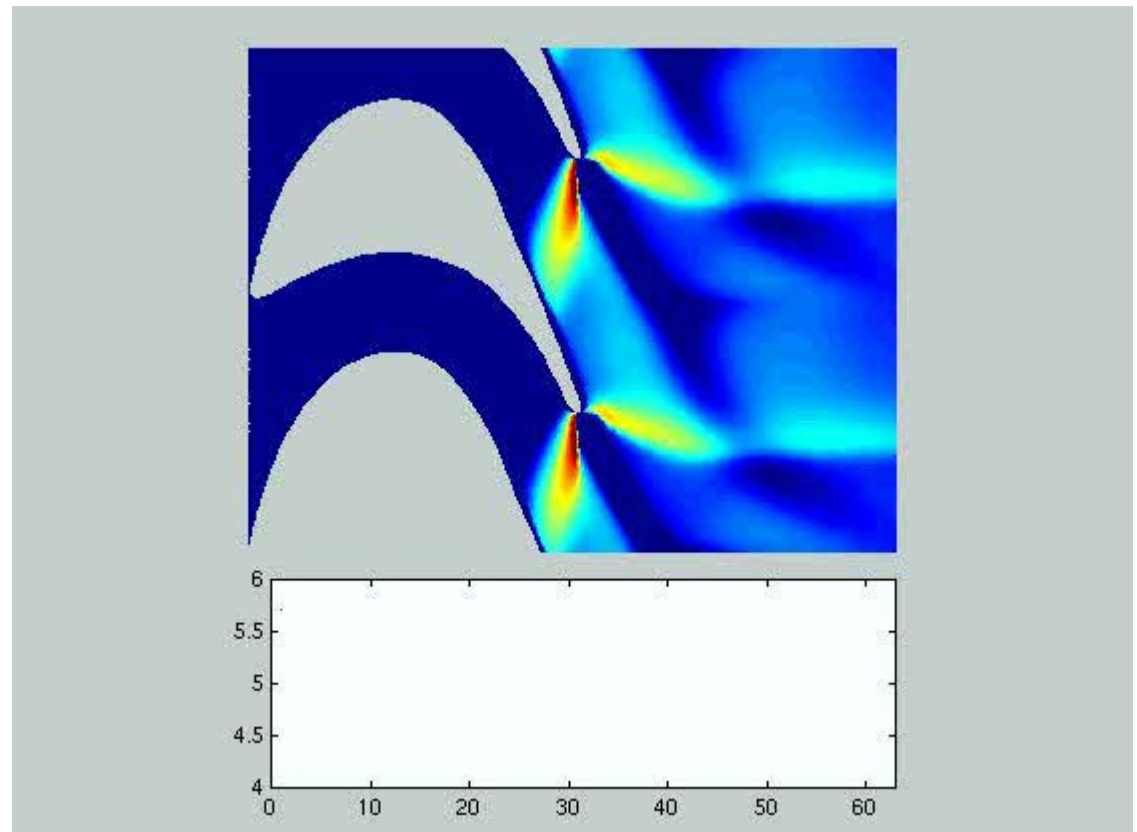


- jedes Individuum besteht aus drei Chromosomen, den Kontrollpunkten, den Knotenpunkten und den Strategieparametern

Anwendungsbeispiel – *Strukturoptimierung von turbinen Blättern*

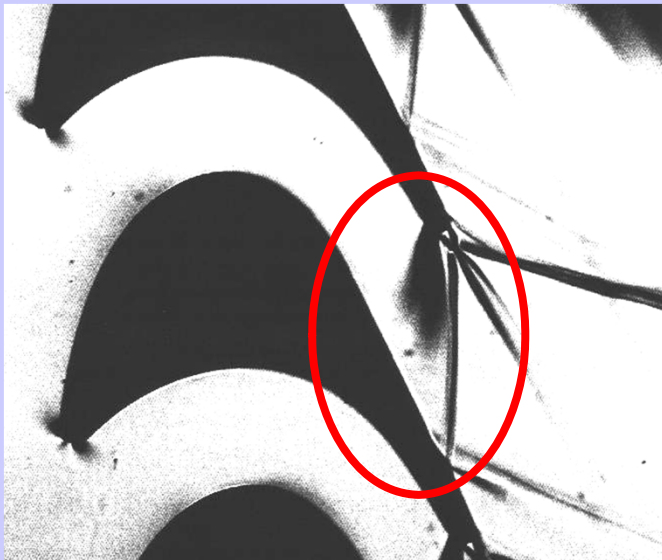
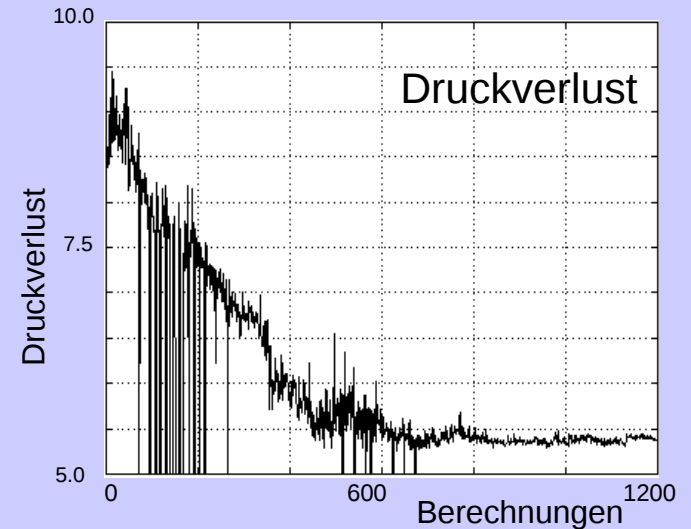
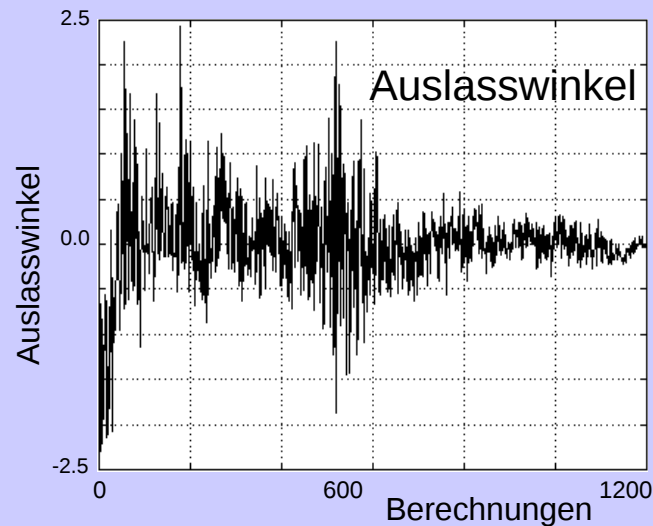
- Simulation der Fluidodynamik
- Visualisierung des Druckverlustes
- Fitnesswert ist eine lineare Kombination aus
 - Geometrischen Randbedingungen
 - Druckverlust
 - Auslasswinkel
- Die Gewichtung der einzelnen Qualitätswerte ist so gewählt, dass
 - geometrische Abweichungen einen hohen Penaltyfaktor haben
 - der Auslasswinkel zunächst angepasst wird (< 20. Generation)
 - zuletzt der Druckverlust vermindert wird (20. – 100. Generation)

Optimierung von Stator und Rotorblättern



Patentiertes Resultat

Typischer
Fitnessverlauf
Statorblätter

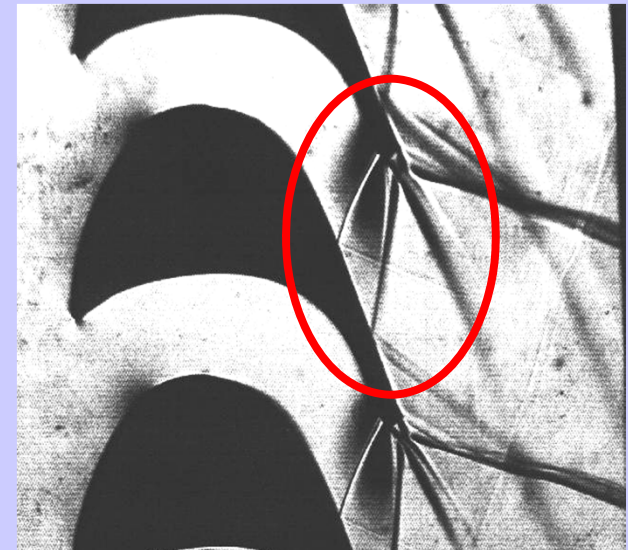


Original

Rotorblatt



bis zu 70%
Reduzierung des
Druckverlustes

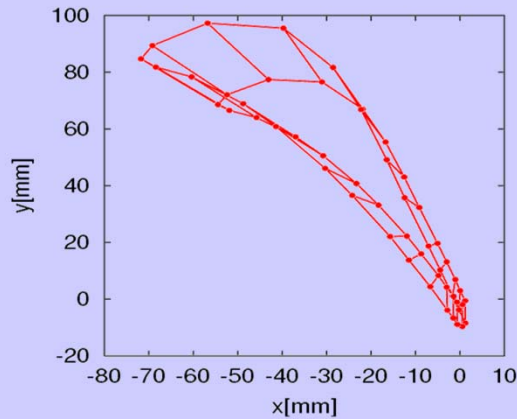


evolutionär optimiert

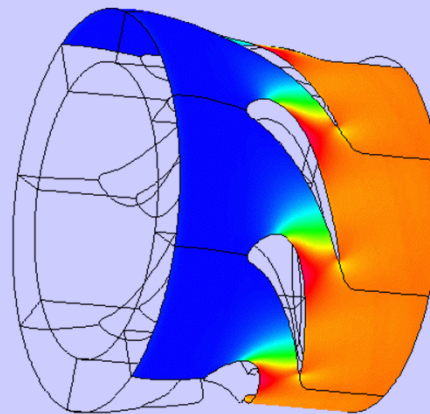
(Schlierenbilder aufgenommen im transonischen Windtunnel der DLR Göttingen)

3D Turbinenblatt Optimierung

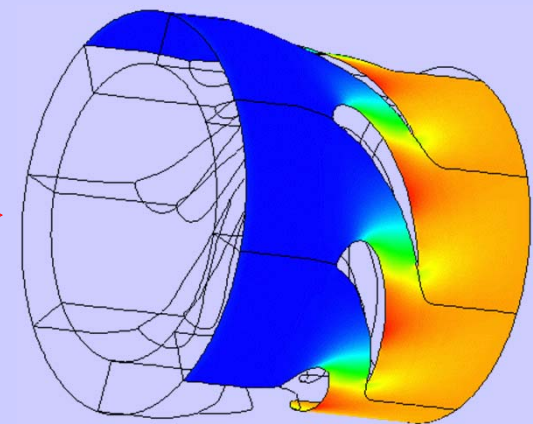
Erweiterung auf eine drei dimensionale Repräsentation (Oberfläche) prinzipiell einfach



Ausgangskonfiguration

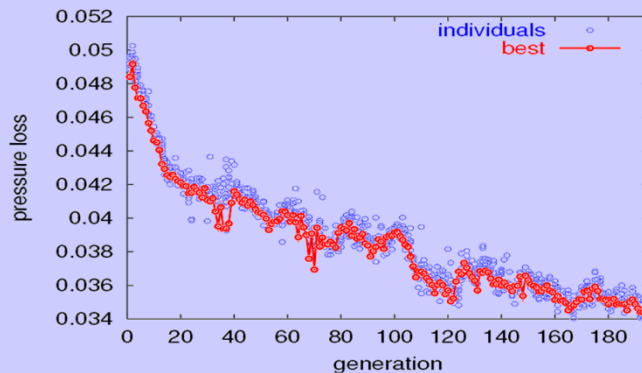


Optimiertes Design



VRLM

- Entwicklung des "loss" während der Optimierung

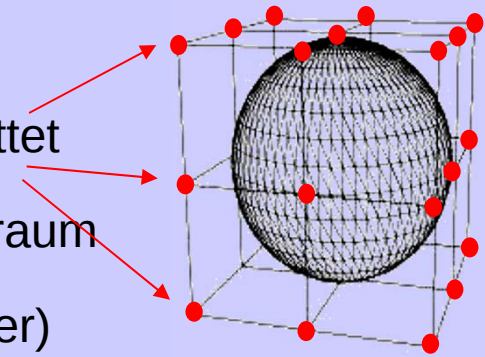


Problem: Zunehmende Komplexität der Objekte bedingt eine sehr hochdimensionale Parameterisierung

Free-Form Deformation

Free Form Deformation

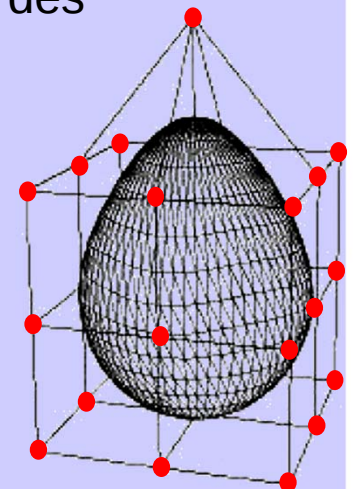
- die Form (Oberfläche) ist in ein Kontrollgitter eingebettet
- die Kontrollpunkte definieren einen Spline-Parameteraum
- alle Objektpunkte (Oberflächenpunkte oder CFD Gitter) werden in diesem Splineraum repräsentiert; dieser Prozess heisst **freezing**
- nach dem *freezing*, ist das Objekt mit dem Kontrollvolumen verbunden
- Veränderungen des Kontrollvolumens induzieren Veränderungen des Objektes, z.B. der Oberfläche



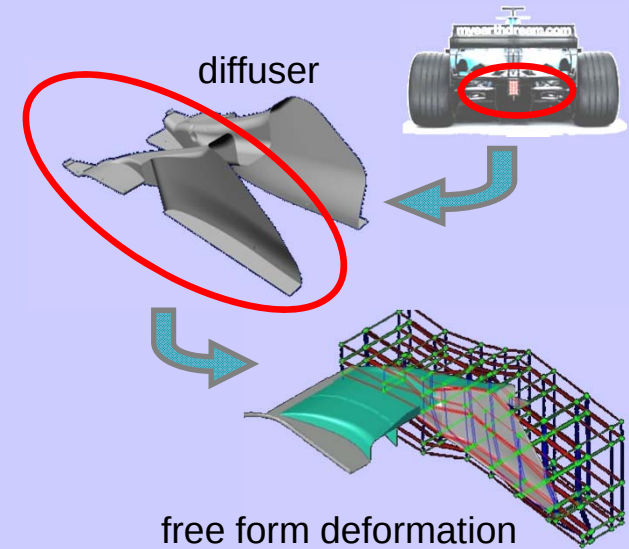
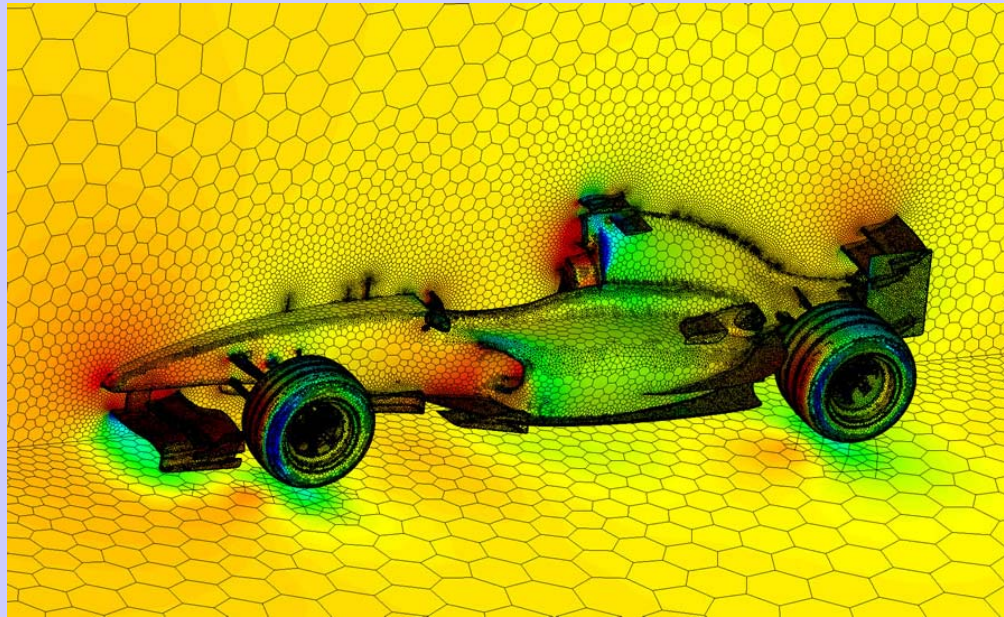
- Repräsentation ist unabhängig von der Objektkomplexität
- Veränderungen und nicht das Objekt werden repräsentiert
- das CFD Gitter wird **mit dem Objekt** deformiert



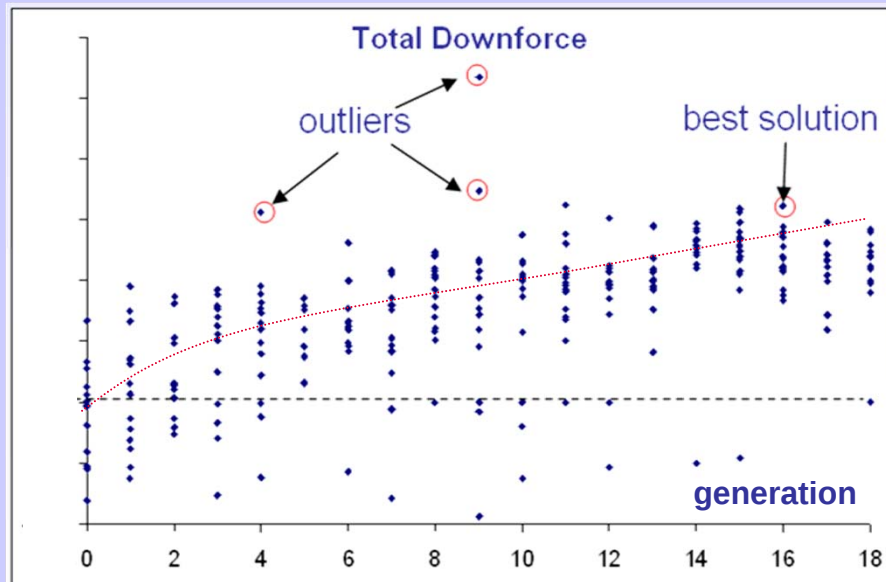
- Einfluss der Kontrollpunkte auf Objektveränderungen ist nicht mehr intuitiv
- der Detailgrad der realisierbaren Veränderungen hängt von dem Freiheitsgrad des Splineraumes ab



Optimierung von Komponenten eines Formel 1 Wagens



- Optimierung unter verschiedenen Anströmwinkeln und *ride-heights*
- Optimizing mehrerer Komponenten gleichzeitig



optimizer (2,15) ES-CMA - FFD
duration ca. 25 days
resource 512 CPU
target max. downforce / min. drag

video

Validierung



- die experimentelle Validierung
– ***Evolution im Windtunnel*** –
ist von entscheidender Bedeutung
- kein Ingenieur wird einen *black-box* Effekt als Resultat der Optimierung verwenden
- das Resultat der Optimierung ist nicht nur die Lösung sondern insbesondere der ***Denkanstoss***

die enge Zusammenarbeit zwischen ***Optimierer***, ***Simulierer*** und ***Realisierer*** ist Voraussetzung für eine erfolgreiche Optimierung

Anwendungsbeispiel

Randbedingungen

- Rechenzeit

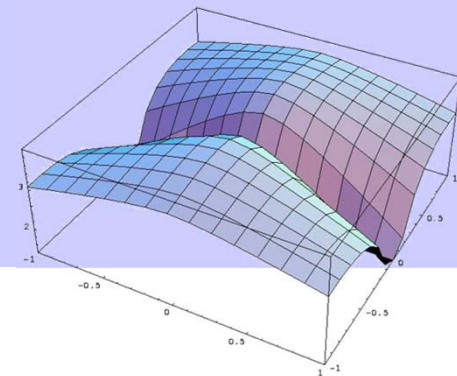
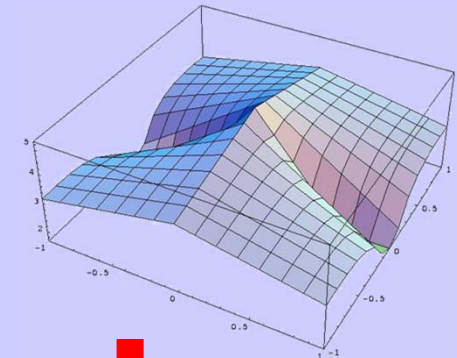
ohne Parallelität:

Anzahl Generationen x Populationsgrösse

n Prozessoren ($n \leq \#Pop$): *$1/n$ Anzahl Generationen x Populationsgrösse*

Annahme: fast identische Rechenzeit pro Individuum, d.h. keine Metamodelle

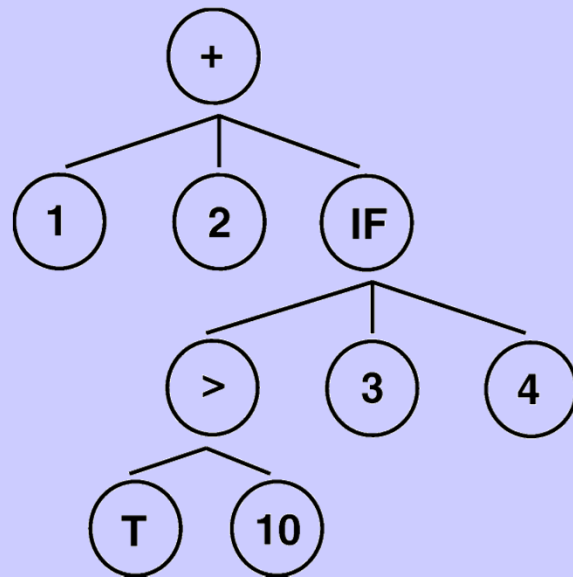
- Lange Laufzeiten – Ausfallsicherheit und Ausfallrobustheit
- Modelle zur Reduktion der Rechenzeit
- Verrauschte Evaluierungsfunktion
- Robustheit bzw. Arbeitsbereich der Lösungen
- Multi-kriterielle Optimierung
- Zielsetzung: Konzeptionell oder Detailoptimierung



Evolutionäre Technologie

*- Genetic Programming,
Artificial Life, Evolvable Hardware
and DNA Computing -*

Genetic Programming (Koza, 1992)



- Genotypen sind Baumstrukturen; allgemeiner hierarchische Repräsentation variabler Dimension
- Koza (1992): evolvieren von LISP Programmen
(+ 1 2 (IF (> T 10) 3 4))
- Algorithmus
 - wähle passenden Satz von Funktion- und Terminalsymbolen
 - initialisiere syntaktisch korrekte Population zufällig
 - berechne die mittlere Fitness jedes Programms für unterschiedlichen Input
 - Crossover (und Mutation) und Selektion: 10% werden (fitness proportional) unverändert kopiert, die restlichen 90% (proportionale Selektion) rekombiniert.
- *Crossover*
Austausch von Unterbäumen
- *Mutation*
Löschen und Zufallsinitialisierung von Unterbäumen

Genetic Programming (Koza, 1992)

- Codierungslänge ist variable, Änderung implizit durch Crossover bzw. Mutation
- Mutation werden mit geringer Wahrscheinlichkeit oder gar nicht genutzt stattdessen sehr grosse Population ~100.000
- grosse nicht-codierende Teilbäume können leicht auftreten (Bsp.: Multiplikation zweier Teilbäume, wobei der eine Teilbaum Null ist) *code bloating*
- komplexe Relation zwischen Genotyp und Phänotypraum
- Anwendungsbereiche:
 - Funktionenapproximation (insbesondere bei Vorwissen)
 - Prozessmodellierung
 - Evolvable Hardware - Schaltkreisoptimierung

Artificial Life

C. Langton on Artificial Life

*A synthetic approach to studying Life-as-it-could-be
and viewing Life-as-it-is within a larger context.*

Problem: Was ist Leben?

- obwohl sich eine ganze Wissenschaft mit den “lebenden Dingen” beschäftigt und
- obwohl jeder eine (scheinbar) klar Vorstellung hat

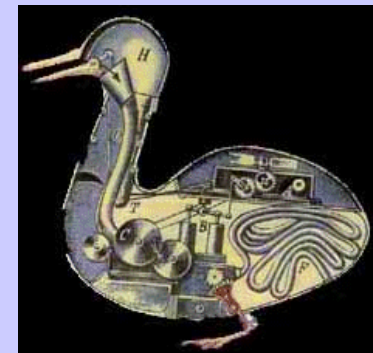
gibt es keine befriedigende Definition

Fast alle Versuche sind listen-basiert: physikalische vs. abstrakte Definitionen

- Selbstreproduktion (*Gegenargument: einfache Moleküle*)
- halb-geschlossenes physikalisches System welches seine innere Ordnung und den Energiefluss nutzt, um die innere Ordnung aufrechtzuerhalten bzw. zu erhöhen (Churchland) (*Gegenargument: Kerze, Städte*)
- etc.

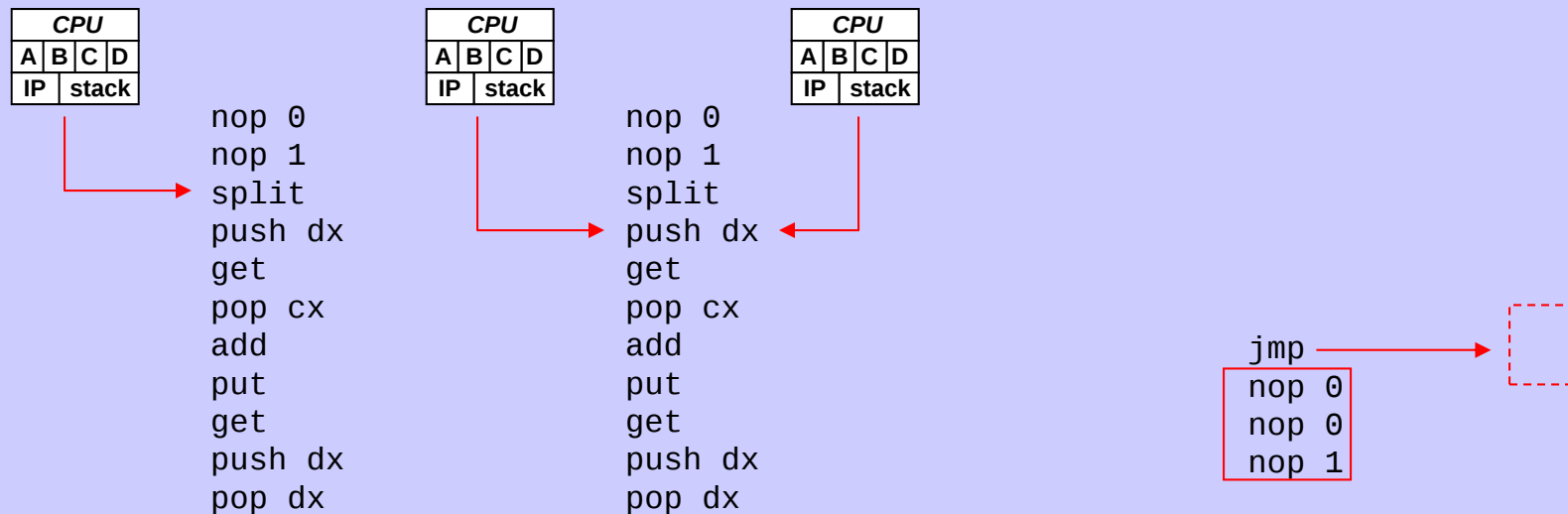
Zielsetzung von Artificial Life:

➔ *Erreichen von emergentem Verhalten durch
Selbstorganisation*



Tierra (T.S. Ray, 1992) und Avida (C. Adami, 1994)

- computer simuliert eine virtuelle Umgebung für selbst-reproduzierende Computerprogramme
- jedes Programm hat eine eigene virtuelle CPU, semi-permeabler Speicherplatz, d.h. jedes Prg. hat Lese- aber nicht Schreibzugriff
- Programme kopieren sich selbst an eine bestimmte Stelle des Speichers, und reservieren sich eine neue CPU



- Set aus 32 Befehlen (5 bit codiert), nachbarschaftlich repräsentiert
- Template Adressierung anstelle von absoluter Adressierung, wenn kein template gefunden, dann error flag

Tierra (T.S. Ray, 1992) und Avida (C. Adami, 1994)

- Mutation: Poisson-verteilt auf dem Binärcode der Instruktionen
Fehler beim Kopierprozess
- Fitness (the *slicer* or the *reaper*) ~ Anzahl der Nachkommen

Qualität des
Kopierprozesses

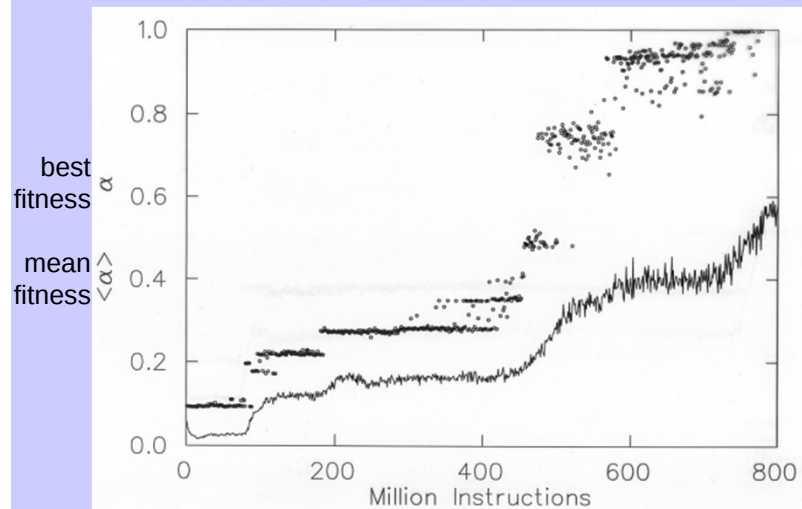
Allokierte CPU Zeit

Erlernte Aufgabe

Fehler

Programmgröße

- Reaper: wenn Speicher 80% voll, werden die schlechtesten Programme gelöscht nach Alter und Fitness



- Avida hat zusätzliche lokale topologische Eigenschaften
- Komplexität ist trade-off zwischen Entropie und Information

Erstaunliche Resultate (Parasiten) aber trotzdem limitierte Komplexität

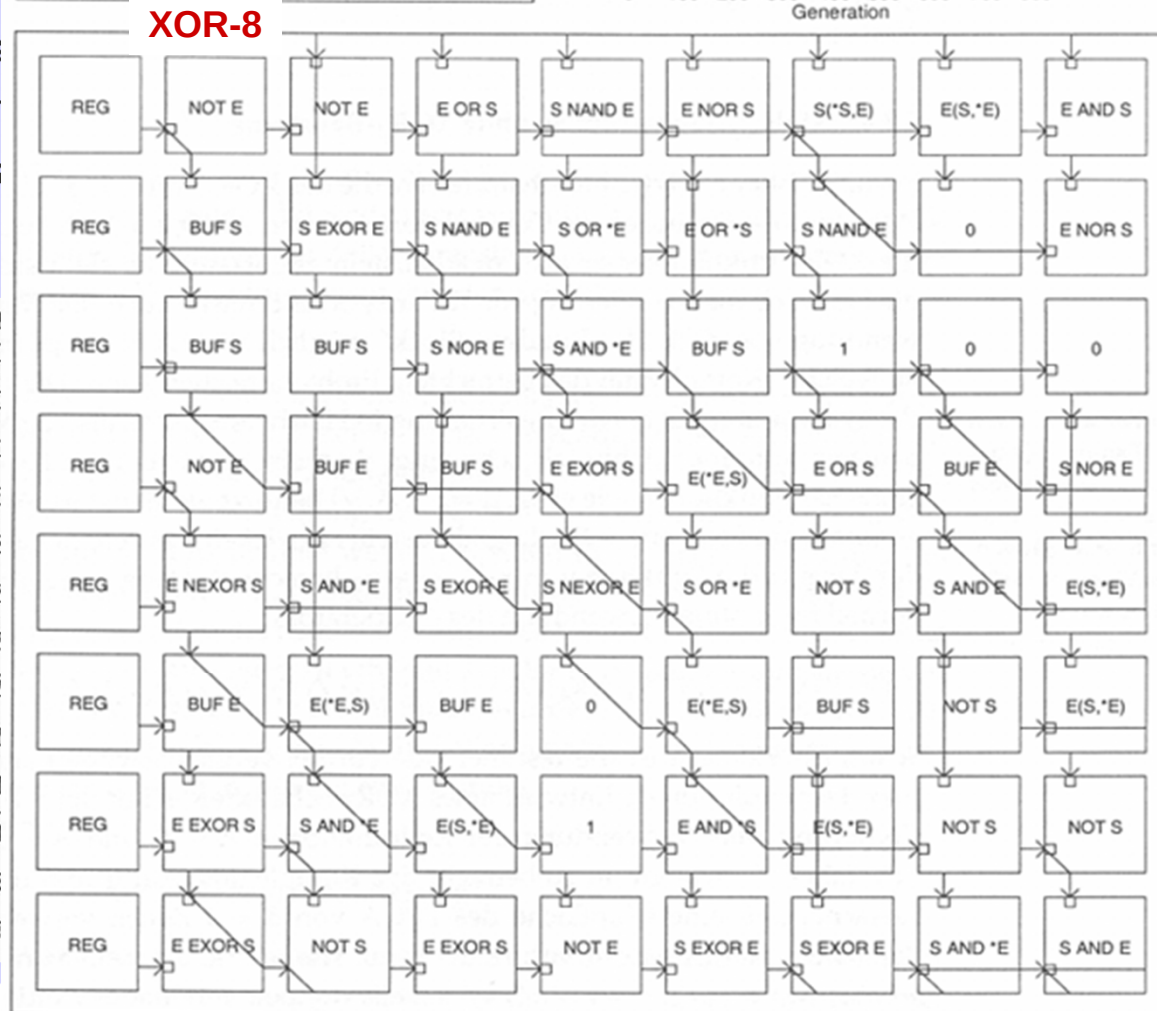
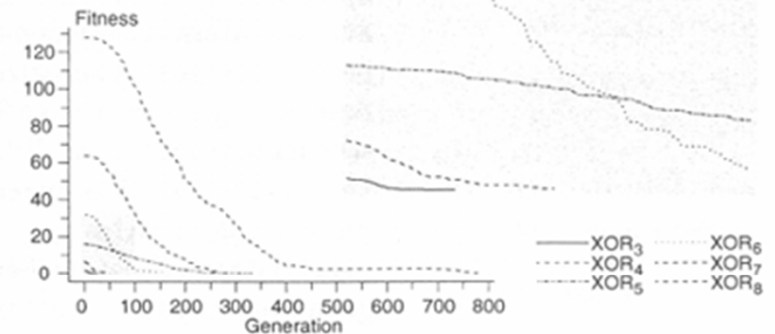
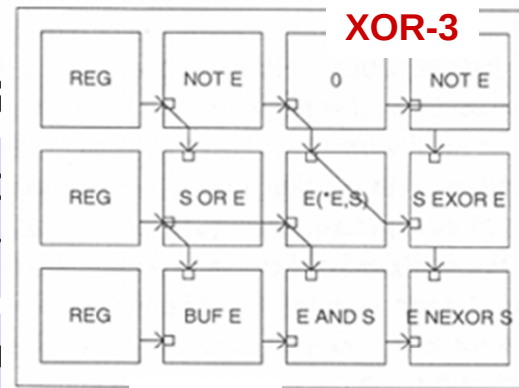
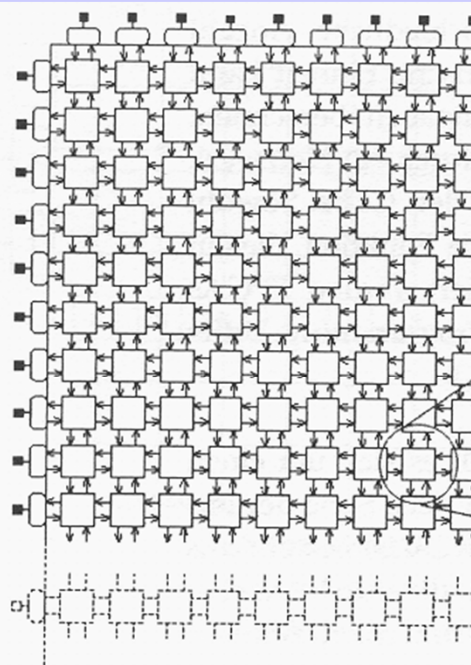
➔ ALife heisst noch nicht *alive*



Evolvable Hardware

intrinsische (komplett)
Trennung des evolutionären

- die Mehrzahl rekonfigurierbarer
- die bekanntesten PLDs (Programmable Logic Devices) logische Funktion jeder Verbindung flexibel
- Beispiel: Teil der 64x64

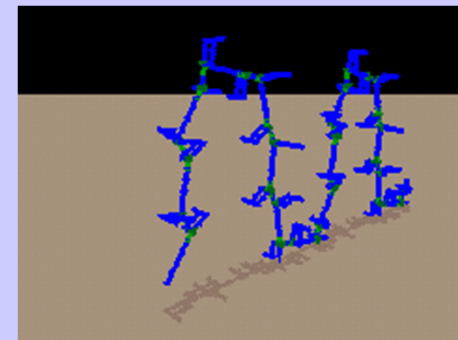


Evolvable Hardware

- A. Thompson (1995/96)
 - Evolutionäres Design von Schaltkreisen ohne Randbedingungen: Emergentes Verhalten und robustes Design
 - Physikalische Sekundär- (Schmutz)-effekte werden genutzt; Interne nichtlineare Dynamiken können evolviert werden
- Problemspektrum war bisher von begrenzter Komplexität, Beispiele XOR-8, Frequenzdiskrimination, Einfache Prothesen
- Sims (1995), Hornby and Pollack (2000): Evolvierbare Objekte zur Fortbewegung
- Lipson (2002): 3D Drucker und Rapid Prototyping Devices
intrinsische Evolution v. Morphologien



(Lipson)



(Hornby & Pollack)

➡ *Evolution von nano-technologischen Systemen*

DNA Computing

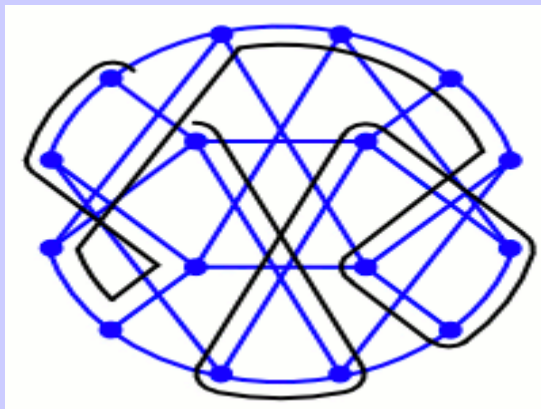
Prinzip der Komplementarität und Parallelität

Das Lösen schwieriger (z.B. NP-vollständiger) Probleme durch eine Abfolge bio-chemischer zumeist enzymbasierter Operationen auf DNA Segmenten

Operationen auf DNA Molekülen

- Trennen von DNA Strängen durch Erhitzen (die kovalenten Wasserstoffbindungen sind schwächer als die kovalenten Bindungen zwischen den Nukleotiden 85°-95°)
- Verbinden von DNA Strängen durch Abkühlen
- Auffüllen von Teilstücken durch Polymerase (z.B. unter Ausnutzung einer *primer* Sequenz)
- Synthetisieren einzelner Stränge durch iteratives Hinzufügen einzelner Nukleotide (synthesizing robots)
- Verkürzen (Exonuklease) oder Aufschneiden (Endonuklease) von DNA Strängen durch bestimmte Enzyme (an zufälliger oder bestimmter Stelle (durch "Erkenne" bestimmter Molekülmuster))
- Einfügen oder Entfernen kurzer Teilstücke
- Filtern von Molekülen (Musterbasiert oder Längenbasiert). DNA Stränge sind negativ geladen, d.h. durch Anlegen eines elektrischen Feldes können leichte (kurze) Moleküle von schweren (langen) Molekülen getrennt werden.

DNA Computing - Adleman Experiment (1994)



Hamiltonsches Pfadproblem - Entscheidungsproblem

- gibt es einen Weg von v_0 nach v_6 , der alle Vertices genau einmal durchläuft?

mus

DNA Operationen

Input gerichteter Graph G mit n Vertices
darunter v_{in} und v_{out}

1. generiere zufällig viele Pfade in G

ca. 4Std.
Massiv Parallel

1. codiere (zufällig) jeden Vertex durch einen Strang bestimmter Nukleotidanzahl (20);
erzeuge eine grosse Anzahl von Kopien

- erzeuge Kanten von vertex i zu vertex j :

code i : $s_i = u_i w_i$; code j : $s_j = u_j w_j$

code $e(ij) = \overline{w_i u_j}$

- die Codes der Kanten wirken als Brücken zwischen den Vertices

— : Komplementärcode

DNA Computing - Adleman Experiment (1994)

Algorithmus	DNA Operationen
2. entferne alle Pfade ohne v_{in} und v_{out} 3. entferne alle Pfade die nicht genau n Vertices enthalten 4. für jeden der n Vertices v: entferne alle Pfade die v nicht enthalten	2. Polymerasekettenreaktion mit primer s_{in} und s_{out} 3. entferne alle DNA Stränge, die nicht die Länge 140 haben (7 mal 20) 4. für jeden der n Vertices v: • trennen aller Stränge; Hinzufügen von $\overline{s_n}$ • verbinden der Stränge durch Abkühlen • entfernen aller nicht-verbundenen Moleküle
<i>Output</i> Ja, wenn ein Pfad übrig bleibt sonst nein	<i>Output</i> Ja, wenn "ein" Molekül übrig bleibt

ca. 7
Tage



Geschickte Codierung,
massiv parallele Zufallsuche (Polymerasekettenreaktion)
Identifikation der korrekten Lösung

Evolutionäre Technologie

- Theorie Evolutionärer Algorithmen -

Genetische Algorithmen: Schematheorem

Wie entwickelt sich die Anzahl von Schemata H in einer Population bei proportionaler Selektion, Crossover und Mutation?

- Erwartungswert der Anzahl von Schemata ohne Crossover und Mutation

$$E[m(H, t + 1)] = m(H, t) \frac{1}{f} \frac{1}{|S(H)|} \sum_{x \in S(H)} f(x) = \bar{f}_H$$

- Ordnung von Schema H : $o(H)$, ist die Anzahl von definierten Positionen
- Definierende Länge von Schema H : $\delta(H)$, ist der Abstand der äussersten definierten Positionen
- p_c und p_m sind die Crossover bzw. Mutationswahrscheinlichkeiten

 Schematheorem bzw. Fundamentales Theorem der GA

$$E[m(H, t + 1)] \geq m(H, t) \frac{\bar{f}_H}{f} \left[1 - p_c \frac{\delta(H)}{l - 1} - p_m o(H) \right]$$

Schemata kleiner Länge und hoher Fitness vermehren sich exponentiell

 *Building Block Hypothese*

Konvergenz Evolutionärer Algorithmen

Die globale Konvergenz evolutionärer Algorithmen kann nur über eine Wahrscheinlichkeit definiert werden

- kontinuierliches Optimierungsproblem
- diskretes Optimierungsproblem

$$\lim_{g \rightarrow \infty} \Pr \left[|F(y^{(g)}) - F(\hat{y})| \leq \varepsilon \right] = 1$$

$$\lim_{g \rightarrow \infty} \Pr \left[y^{(g)} = \hat{y} \right] = 1$$

$$\lim_{g \rightarrow \infty} \Pr \left[F(y^{(g)}) = F(\hat{y}) \right] = 1$$

- Konvergenzbeweise sind für EAs mit Elitisten intuitiv (jedoch mathematisch aufwendig);
- Für (μ, λ) - ES gibt es keinen Konvergenzbeweis, ein Ansatz würde über ein geschicktes Annealingschema (d.h. Adaptation) der Schrittweiten führen

$g \rightarrow \infty$ ist praktisch irrelevant  **lokales Verhalten evolutionärer Algorithmen**

- Mutationsraten müssen hinreichend klein sein, um den Wildtypen (siehe Quasi-spezies) stabilisieren zu können

 **Idee des Evolutionsfensters (window of opportunity)**

Markovkettenanalyse

Markovketten sind stochastische Prozesse bei denen die Wahrscheinlichkeit dass sich das System im Zustand j befindet nur vom Zustand i zur Zeit t abhängt

Die Wahrscheinlichkeit von Zustand j in den Zustand i überzugehen, ist durch den Eintrag $Q_{ij} = Pr[i | j]$ der Übergangsmatrix (transition matrix) $\mathbf{Q}$ gegeben

sei $\mathbf{p}(t)$ der Zustandsvektor zur Zeit t , das heisst $p_i(t)$ gibt die Wahrscheinlichkeit an, dass sich das System zur Zeit t im Zustand i befindet, dann gilt:

$$\mathbf{p}(t) = \mathbf{Q}^t \mathbf{p}(0)$$

entscheidend ist das Verhalten der Markovkette für $t \rightarrow \infty$:
gemäss des *Perron-Frobenius Theorems* existiert der Grenzwert $\mathbf{Q}^\infty = \lim_{t \rightarrow \infty} \mathbf{Q}^t$ und ist von $\mathbf{p}(0)$ unabhängig!

Markovkettenanalyse GA - Ansatz

Zur Analyse des kanonischen GA mit Markovketten, muss der Zustandsraum und die Transitionsmatrix gefunden werden
und *Ist der cGA eine Markovkette?*

Der Zustandsraum eines GAs ist durch die Menge aller mgl. Populationen gegeben, d.h. der Zustand i der Markovkette ist am besten durch eine bestimmte Population i beschrieben

- gesucht ist nun die $N \times N$ Transitionsmatrix Q
 Q_{ij} gibt die Wahrscheinlichkeit an, dass die Population P_j aus P_i nach Anwendung der Operatoren des kanonischen Genetischen Algorithmus (cGA) hervorgeht
- die Transitionsmatrix Q setzt sich aus drei Operationen zusammen: Selektion, Mutation und Crossover; prinzipiell ist die Herleitung nicht schwierig, benötigt jedoch eine Menge Kombinatorik

Markovkettenanalyse II

Der Zustandsraum eines GAs ist durch die Menge aller mgl. Populationen gegeben, d.h. der Zustand i der Markovkette ist am besten durch eine bestimmte Population i beschrieben

- sei ϕ ein Vektor der für jeden möglichen Zustand eines genetischen Algorithmus eine Komponente hat, d.h. $\dim(\phi) = k^l$ (k -Kardinalität des Alphabets, sei $k=2$) und l Länge der Strings)



jede Population der Grösse n lässt sich durch einen Vektor ϕ mit $SUM(\phi) = n$ darstellen

$\phi(i)$ gibt die Anzahl der Strings in der Population an, die im Zustand i sind, $0 \leq i \leq 2^l - 1$

- schreibt man all möglichen ϕ Vektoren hintereinander, so erhält man eine Matrix Z , die alle möglichen Belegungen aller Zustände angibt bei gegebenen n , und l , die Dimension der Matrix Z ist $2^l \times N$

$$N = \binom{2^l + n - 1}{2^l - 1}$$

- gesucht ist nun die $N \times N$ Transitionsmatrix Q
 $Q_{i,j}$ gibt die Wahrscheinlichkeit an, dass die Population P_j aus P_i nach Anwendung der Operatoren des kanonischen Genetischen Algorithmus (cGA) hervorgeht

Markovkettenanalyse GA - Beispiel

- Q und damit Q^∞ beschreibt nun das Konvergenzverhalten des kanonischen GA:
Beispiel:

Suchraum: $S = \{ 00, 01, 10, 11 \}$, Pop.-grösse $n = 4$ und Fitness $f(S) = \{1, 2, 3, 4\}$
Mutationprob: 0.1

die wahrscheinlichste Population ist (0,0,0,4) mit 14,3%, es gibt aber auch 15 weitere Populationen mit Wahrscheinlichkeit >0 , die das Optimum nicht enthalten

- ➔ der kanonische genetische Algorithmus ist nicht global konvergent
die Wahrscheinlichkeit den Besten wieder zu verlieren ist in jedem Schritt grösser als null: *Elitist ist notwendig*

Evolutionstrategie: Erfolgswahrscheinlichkeit

Theorie der Evolutionstrategie für bestimmte Qualitätsfunktionen

- Lokale Veränderungen stehen im Vordergrund



- Entwicklung in Potenzreihen
- Abruch nach dem 2ten Glied
- Vereinfachung
- y nahe null

$$Q = Q_0 + \sum_{k=1}^n c_k y_k - \sum_{k=1}^n d_k y_k^2; \quad d_k > 0$$

Erfolgswahrscheinlichkeit

- Wahrscheinlichkeit, dass ein Nachkomme seine Qualität gegenüber den Eltern verbessert

$$\Delta Q = \sum_{k=1}^n c_k z_k - \sum_{k=1}^n d_k z_k^2$$

$$w(z_k) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{z_k^2}{2\sigma^2}\right)$$

↓
Normalverteilte
Zufallszahlen
ergeben addiert wieder
eine Normalverteilung

$$\Delta Q \approx z^* - \sigma^2 \sum_{k=1}^n d_k$$

↓
für grosse n ergeben n quadrierte (0,1) normalverteilte
Zufallszahlen wieder eine Normalverteilung mit dem
Mittelwert n und der
Streuung $\sqrt{2n}$
- Chi-Quadrat Verteilung

$\Delta Q = \bar{Q}_N - Q_E$	Qualitätsunterschied
y	Objektvariablen
z	Zufallszahl mit $z \sim \mathcal{N}(0, \sigma^2)$
z^*	Zufallszahl mit $z^* \sim \mathcal{N}(0, \sigma^{*2})$
c_k, d_k	Parameter der Qualitätsfunktion
$\Pr[\Delta Q > 0]$	Erfolgswahrscheinlichkeit
$\phi', \phi'_{1,\lambda}$	Fortschrittsgeschwindigkeit (mit/ohne Selektion)
$\phi_{1,\lambda}$	mittlere Fortschrittsgeschwindigkeit
λ	Anzahl der Nachkommen
u	Zufallsvariable
$c_{1,\lambda}$	Fortschrittsbeiwerte

Evolutionstrategie: Erfolgswahrscheinlichkeit

Erfolgswahrscheinlichkeit

$$\Pr [\Delta Q > 0] = \frac{1}{2} \left(1 - \operatorname{erf} \left(\frac{\sigma \sum_{k=1}^n d_k}{\sqrt{2 \sum_{k=1}^n c_k^2}} \right) \right)$$

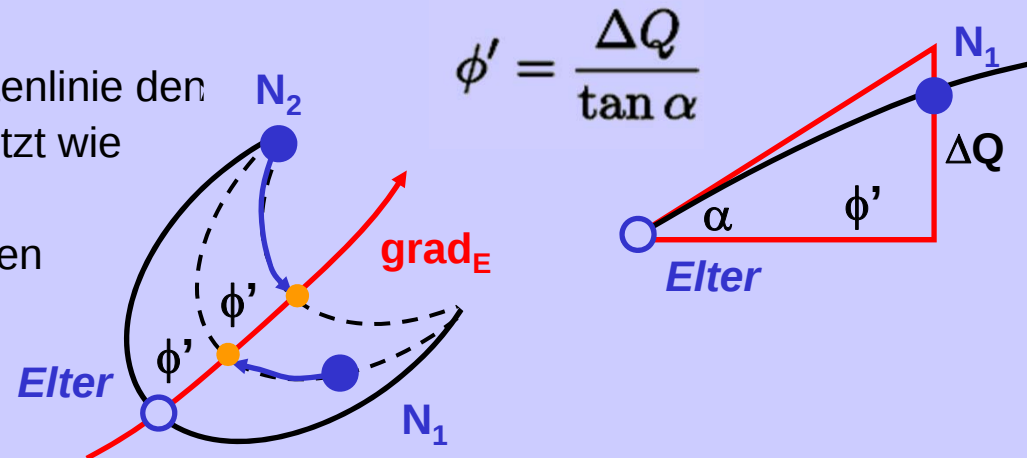
- $\operatorname{erf}(\dots)$ ist das Gaussche Fehlerintegral

Evolutionsstrategie: Fortschrittsgeschwindigkeit

Fortschrittsgeschwindigkeit

- Suche auf der elterlichen Gradientenlinie den Punkt der die gleiche Qualität besitzt wie der Nachkomme, die Länge des Gradientenstückes kennzeichne den individuellen Fortschritt

$$\phi' = z - \sigma^2 \frac{\sum_{k=1}^n d_k}{[\sum_{k=1}^n c_k^2]^{\frac{1}{2}}}$$



- Selektion (1,lambda): Wie ist die grösste Zufallszahl (u) unter lambda (0, sigma) normalverteilten Zahlen verteilt?

$$w_\lambda = \frac{\lambda}{\sqrt{2\pi}\sigma} \exp\left(-\frac{u^2}{2\sigma^2}\right) \left\{ \frac{1}{2} \left[1 + \operatorname{erf}\left(\frac{u}{\sqrt{2}\sigma}\right) \right] \right\}^{\lambda-1}$$

$$\rightarrow \phi_{1,\lambda} = \sigma c_{1,\lambda} - \sigma^2 \frac{\sum_{k=1}^n d_k}{[\sum_{k=1}^n c_k^2]^{\frac{1}{2}}}$$

$\Delta Q = Q_N - Q_E$	Qualitätsunterschied
y	Objektvariablen
z	Zufallszahl mit $z \sim \mathcal{N}(0, \sigma^2)$
z^*	Zufallszahl mit $z^* \sim \mathcal{N}(0, \sigma^{*2})$
c_k, d_k	Parameter der Qualitätsfunktion
$\Pr[\Delta Q > 0]$	Erfolgswahrscheinlichkeit
$\phi', \phi'_{1,\lambda}$	Fortschrittsgeschwindigkeit (mit/ohne Selektion)
$\phi_{1,\lambda}$	mittlere Fortschrittsgeschwindigkeit
λ	Anzahl der Nachkommen
u	Zufallsvariable
$c_{1,\lambda}$	Fortschrittsbeiwerte

Evolutionsstrategie: Fortschrittsgeschwindigkeit

Fortschrittsgeschwindigkeit

$$\phi_{1,\lambda} = \sigma c_{1,\lambda} - \sigma^2 \frac{\sum_{k=1}^n d_k}{[\sum_{k=1}^n c_k^2]^{\frac{1}{2}}}$$

- Fortschrittsbeiwerte

$$c_{1,\lambda} = \frac{\sqrt{2}\lambda}{\sqrt{\pi}2^{\lambda-1}} \int_{-\infty}^{\infty} z e^{-z^2} [1 + \operatorname{erf}(z)]^{\lambda-1} dz$$

- Selektion (μ, λ) : die Formel für die Fortschrittsgeschwindigkeit bleibt erhalten, die Fortschrittsbeiwerte ändern sich!
- Maximierung der Fortschrittsgeschwindigkeit $\Rightarrow$ Anpassung der Schrittweite

Evolutionsstrategie: Schrittweitenkontrolle

- bei der experimentellen Optimierung kann die Schrittweite oftmals konstant gehalten werden
- aus der Maximierung der Fortschrittsgeschwindigkeit ergibt sich die optimale Schrittweite, d.h. Standardabweichung der Normalverteilung

$$\frac{\partial \phi_{1,\lambda}}{\partial \sigma} = 0 \Rightarrow \sigma_{opt} = \frac{c_{1,\lambda}}{2} \frac{[\sum_{k=1}^n c_k^2]^{\frac{1}{2}}}{\sum_{k=1}^n d_k}$$

- Erfolgswahrscheinlichkeit:

$$\Pr[\Delta Q > 0] = \frac{1}{2} \left(1 - \operatorname{erf} \left(\frac{c_{1,\lambda}}{2\sqrt{2}} \right) \right)$$

$$c_{1,\lambda} \approx 1.5 \quad (\lambda \approx 10) \Rightarrow \Pr \approx 0.2$$

$\Delta Q = Q_N - Q_E$	Qualitätsunterschied
y	Objektvariablen
z	Zufallszahl mit $z \sim \mathcal{N}(0, \sigma^2)$
z^*	Zufallszahl mit $z^* \sim \mathcal{N}(0, \sigma^{*2})$
c_k, d_k	Parameter der Qualitätsfunktion
$\Pr[\Delta Q > 0]$	Erfolgswahrscheinlichkeit
$\phi', \phi'_{1,\lambda}$	Fortschrittsgeschwindigkeit (mit/ohne Selektion)
$\phi_{1,\lambda}$	mittlere Fortschrittsgeschwindigkeit
λ	Anzahl der Nachkommen
u	Zufallsvariable
$c_{1,\lambda}$	Fortschrittsbeiwerte

- 1/5 Erfolgsregel (für grosse n, Kugelmodell und $\lambda \approx 10$): Ist das Verhältnis zwischen erfolgreichen Nachkommen zu allen Nachkommen grösser als 1/5 dann erhöhe die Varianz, ist es kleiner als 1/5 dann verkleinere die Varianz

- in der Praxis sollte für die Schrittweite immer auch eine untere Grenze festgelegt werden

Evolvierbarkeit

Evolvierbarkeit - die Fähigkeit sich **weiter** zu entwickeln

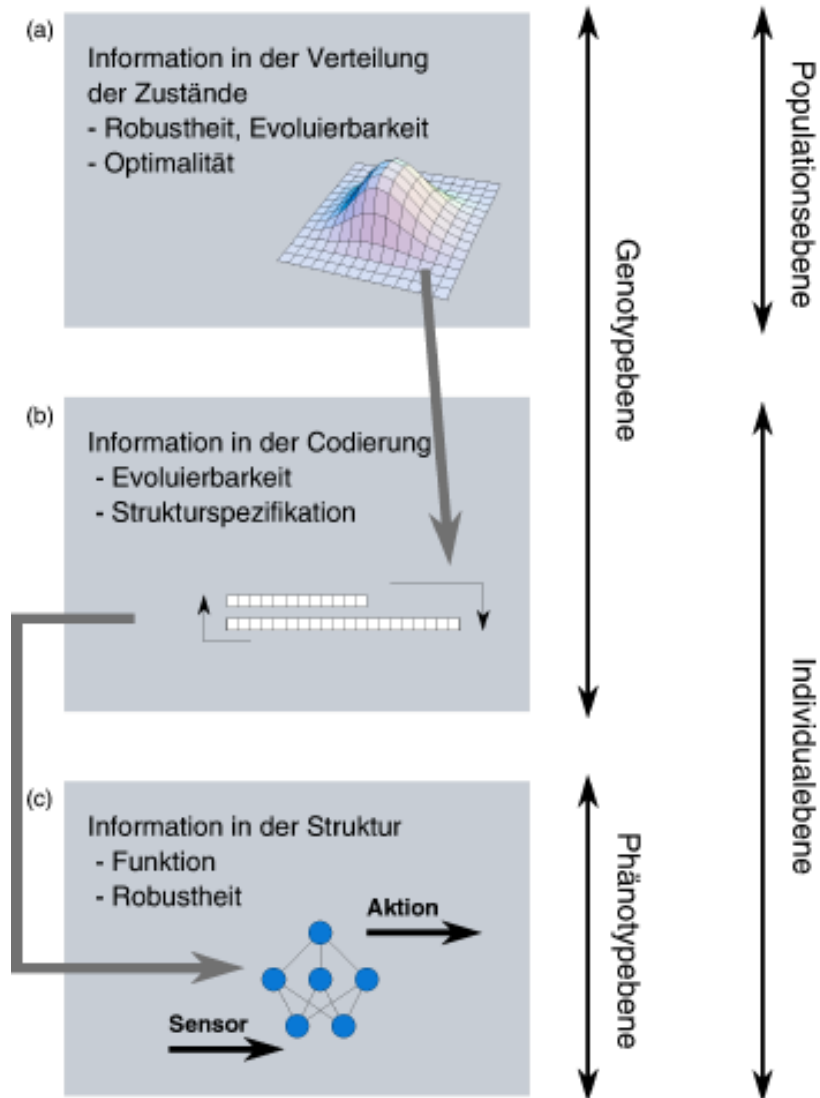
Evolvierbarkeit auf der Fitnesslandschaft

- kleine Schritte können sich akkumulieren - starke Kauslität
- lokale Optima können überwunden werden
- Reparaturmechanismen (Rekombination)
- Fähigkeit in dynamischen Fitnessräumen zu optimieren

Evolvierbarkeit auf dem Genotyp- und Phänotypraum

- Strukturen müssen aufeinander aufbauen können
Skalierbarkeit der Strukturen
- Stabilität muss mit Variabilität koppelbar sein
- Modulare Strukturen - Prinzip der Kompartimentierung - unterstützen Evolvierbarkeit
- Suche eine verallgemeinerungsfähige Speziallösung

Information, Entropie, Masse für Unordnung und Ordnung



- Verteilung der Mikrozustände (einzelne Individuen) im Zustandsraum enthält Information über die Fitnesslandschaft und den Zustand des Prozesses, d.h. der Evolution
- Informationsdefinition auf der Codierung verbindet die Ebene der Individuen (Codierung des Phänotypes) mit der Ebene der Population (Evolvierbarkeit) – *die Codierung definiert den Rahmen für evolutionäre Veränderungen*
- Diversität der Codierung
- Redundanz
- minimale Beschreibungslänge (Kolmogorov Komplexität, Kompressionsrate, Modularität)
- evolutionäre Strukturierung für Problemklassen anstelle von einzelnen Problemen

Zusammenfassung

- Evolutionäre Algorithmen (EA) sind stochastische Suchverfahren, die Prinzipien der “Synthese in der Evolutionsbiologie” algorithmisch umsetzen. Die frühen Arbeiten zu EAs waren zunächst Modelle zur biologischen Evolution
- Hauptkomponenten eines evolutionären Algorithmus sind die Repräsentation der Lösungen, die Variationsoperatoren (Mutation, Rekombination) und die Selektionsmethode
- Hauptrichtungen der EAs sind die Evolutionsstrategien, die genetischen Algorithmen, das evolutionäre Programmieren und das genetische Programmieren. Historisch lagen die Hauptunterschiede in der mikroskopischen bzw. eher makroskopischen Sichtweise der Evolution
- Zielsetzung von Artificial Life ist das Erreichen von emergentem Verhalten durch Selbstorganisation; DNA Computing nutzt die massive Parallelität biologischer Systeme geschickt, um schwierige kombinatorische Probleme zu lösen
- Die Theorie der ES ist zielfunktionsbezogen und baut auf der analytischen Behandlung der Erfolgswahrscheinlichkeit und der Fortschrittsgeschwindigkeit auf. Basis der Theorie zu GAs ist das Schematheorem und die daraus ableitbare Building Block Hypothese.



Weiterführende Literatur

- [1] H-G. Beyer, Theory of Evolution Strategies, Springer Verlag.
- [2] D.B. Fogel, Evolutionary Computation, IEEE Press.
- [3] M. Mitchell, An Introduction to Genetic Algorithms.
- [5] D. Goldberg, Genetic Algorithms.
- [4] I. Rechenberg, Evolutionsstrategie '94.
- [6] H.-P. Schwefel, Evolution and optimum seeking.
- [7] D.B. Fogel, Evolutionary Computation - The Fossile Record.
- [8] T. Bäck, Evolutionary Algorithms.
- [9] T. Bäck, D.B. Fogel and Z. Michaelwicz,
Evolutionary Computation I and II. (*Cook Book*)
- [10] C.S. Calude und G.Paun, Computing with Cells and Atoms.
- [11] C.R. Reeves and J.E. Rowe, Genetic Algorithms – Principles and Perspectives – A guide to GA Theory, Kluwer Academic Publisher