

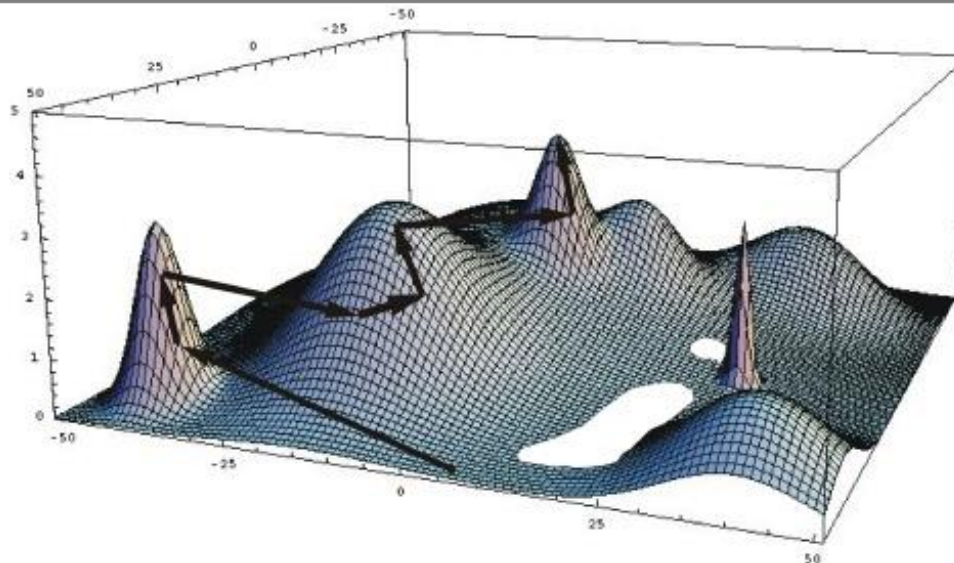
Vorlesung Computational Intelligence

Teil 4: Evolutionäre und Memetische Algorithmen

4.5 Klassische Evolutionäre Algorithmen

Ralf Mikut, Wilfried Jakob, Markus Reischl

Institut für Automation und angewandte Informatik (IAI) / Campus Nord



4.5 Klassische Evolutionäre Algorithmen

Übersicht:

- Übersicht über die klassischen EAs
- Evolutionsstrategie
- Genetische Algorithmen
- Strategieparameter Populationsgröße

Klassische EAs – Übersicht

Übersicht (1):

Parallele Entwicklung der EAs in den USA und Deutschland

■ **Evolutionstrategie (ES)**

Mitte der 60-iger, Rechenberg und Schwefel [Rec73, Schw95]

- Chromosom aus reellen Zahlen (Entscheidungsvariable + Strategieparameter)
- verwendet adaptive Mutation
- vergleichsweise weit fortgeschrittene Theorie
- Anwendungen: vor allem numerische aber auch kombinatorische Optimierung

■ **Genetische Algorithmen (GA)**

Anfang der 70-iger, Holland [Hol75]

- Chromosom besteht aus Bitstrings (später auch: real-coded GAs)
- verwendete anfangs fitnessproportionale Selektion
- Bit-Codierung bringt Vorteile bei der Theoriebildung. Ergebnisse aber umstritten
- Anwendungen: numerische und kombinatorische Optimierung, maschinelles Lernen (LCS learning classifier systems)

Klassische EAs – Übersicht

Übersicht (2):

■ Evolutionäre Programmierung (EP)

Mitte der 60-iger, Fogel, Owens und Walsh [Fog66]

- Chromosomenstruktur nicht festgelegt, anfangs Repräsentation endlicher Automaten, später Erweiterung auf reellwertige Variable
- basiert im Wesentlichen auf Mutation einschließlich selbstadaptierender Mutation
- in der reellwertigen Variante Ähnlichkeiten zur ES

■ Genetische Programmierung (GP)

Ende der 80-iger, Koza [Koza89]

- Chromosome repräsentieren Computerprogramme als Baum (z.B. LISP) oder Maschinencode, Ziel ist die Erstellung von Computerprogrammen
- Crossover und Mutation:
 - Crossover wegen der Baumstruktur mächtiger als bei GAs oder der ES.
 - Mutation mit Reparaturmechanismen
- Anwendungen: maschinelles Lernen, Elektronikentwicklung, Sortier- und Suchalgorithmen, ...

Klassische EAs – Evolutionsstrategie

Evolutionsstrategie:

Chromosom = reellwertiger Vektor bestehend aus

- n zu optimierenden **Entscheidungsvariablen** x_i , $1 \leq i \leq n$
- n' Mutationsschrittweiten σ_j , $1 \leq j \leq n' \leq n$

Meist entweder eine einheitliche Mutationsschrittweite σ oder für jede Entscheidungsvariable ein eigenes σ_j .

Beide Parametersätze unterliegen der Evolution

- Optimierung auf der **Problemebene** **UND** der **Ebene der Schrittweiten**
- Anpassung der Suche an den Suchraum (**Selbstadaption**)
Gefahr, an lokalen Optima hängen zu bleiben oder größere undefinierte Gebiete (**implizite Restriktionen**) nicht überspringen zu können.

Vorteil: schnellere Suche

Klassische EAs – Evolutionsstrategie

Ablaufschema:

(siehe auch Folie 4, Kapitel 4.4)

1. Initialisierung der Startpopulation

Zufällige Wahl der μ Individuen, (eher zu große) Vorbelegung der Schrittweiten

2. Partnerwahl

Zufällige Wahl der Eltern für λ Kinder. λ sollte deutlich größer als μ sein.

Empfehlung von Schwefel und Bäck: $\lambda = 7 \cdot \mu$ oder mehr bei starker Multimodalität.

Warum?



3. Erzeugung eines Nachkommens

Rekombination mit anschließender Mutation der Schrittweiten und danach der Entscheidungsvariablen:

1. **Intermediäre Rekombination** der Schrittweiten
2. **Mutation** der Schrittweiten, liefert σ'_j .
3. **Diskrete Rekombination** der Entscheidungsvariablen
4. **Mutation der Entscheidungsvariablen** mit den geänderten Schrittweiten σ'_j

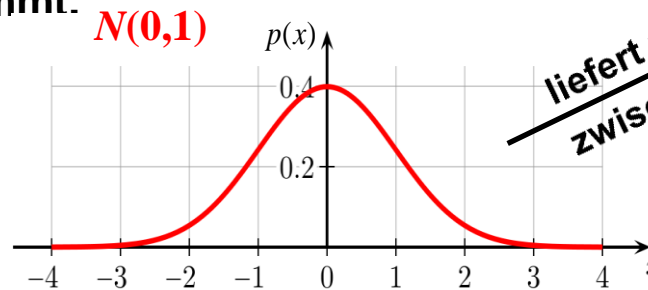
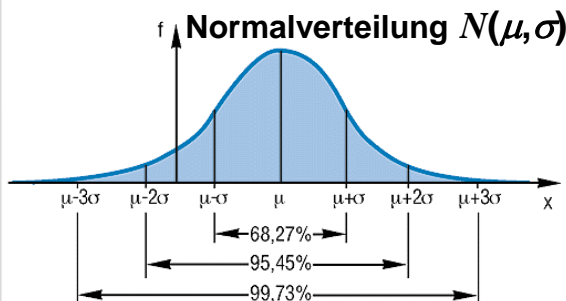
3. Erzeugung eines Nachkommens (2):

Schritt 2: **Mutation** der Schrittweiten

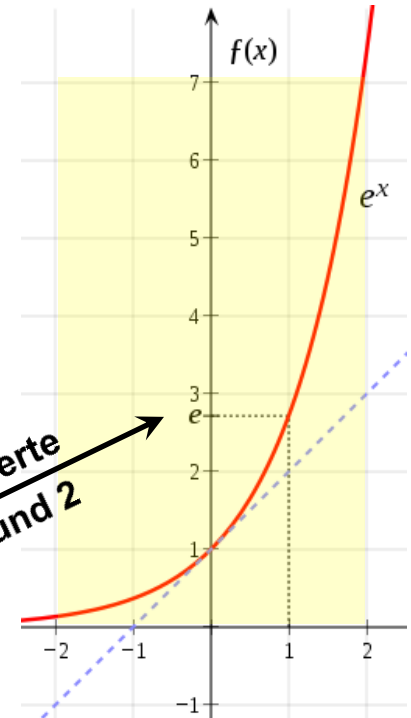
$$\sigma'_j = \sigma_j \cdot e^{(N(0,1) + N_j(0,1))}$$

mit: $N(0,1)$ normalverteilte Zufallsgröße mit Erwartungswert 0 und Standardabweichung 1

$N(0,1)$ gilt allgemein, $N_j(0,1)$ wird für jedes σ'_j neu bestimmt.



liefert meist Werte zwischen -2 und 2



Wirkung: kleine Änderungen wahrscheinlich, große weniger, sehr große unwahrscheinlich, aber möglich

Schritt 4: **Mutation** der Entscheidungsvariablen mit den geänderten Schrittweiten σ'_j :

$$x'_j = x_j + N_j(0, \sigma'_j)$$

Klassische EAs – Evolutionsstrategie

4. Akzeptanzregel

Zwei unterschiedliche Strategien:

(siehe auch Folie 18, Kap. 4.4)

➤ $(\mu+\lambda)$ –Strategie (Plus-Strategie):

Auswahl der μ besten aus Eltern und Nachkommen zur Bildung der Folgegeneration (**elitär**).

➤ (μ, λ) –Strategie (Komma-Strategie):

Die μ besten Nachkommen ersetzen die Elterngeneration vollständig.

5. Abbruchkriterium

Geringe Fitnessunterschiede in der Population gelten als Stagnationsindikator (Schwefel):

$$fitness_{\max} - fitness_{\min} < \varepsilon, \quad \varepsilon > 0$$

Weitere Abbruchkriterien, wie Erreichen einer **Zielfitness**, eines **Zeit-** oder **Generationenlimits**, ...

Klassische EAs – Evolutionsstrategie

Besonderheiten der Evolutionsstrategie:

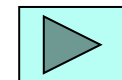
- **Adaptive Schrittweitensteuerung**
- **auslöschende Selektion** (schlechte Individuen haben keine Chance zur Reproduktion): vor allem durch die Koma-Strategie
- Die **Populationsgröße** μ darf wegen des starken Selektionsdrucks nicht zu klein sein. Geeignete Werte für μ sind anwendungsabhängig.

Weitere Eigenschaften:

- **Plus-Strategie**: Neigung zu vorzeitiger Konvergenz an einem Suboptimum
- **Explizite Beschränkungen**: Letalmutation. Eventuell durch geschickte Codierung vermeidbar.
- **Implizite Beschränkungen** können je nach Größe und aktuellen Schrittweiten nicht übersprungen werden.

Gegenmaßnahmen:

- μ deutlich größer als Anzahl der Beschränkungen
- Möglichst gleichmäßige Verteilung der Startpopulation im Suchraum
- ES sehr gut zur **Optimierung kontinuierlicher Probleme** geeignet.



Linsendemo
I.Rechenberg

Klassische EAs – Evolutionsstrategie

Strategieparameter:

- Populationsgröße μ
- Anzahl der Nachkommen pro Generation λ
- Komma- oder Plus-Strategie
- Anzahl der adaptierten Schrittweiten
- Parameter der Schrittweitensteuerung
- Parameter von alternativen Selektions- oder Akzeptanzverfahren

Klassische EAs – Evolutionsstrategie

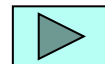
Evolutionsstrategie und kombinatorische Probleme:

- Ganzzahlige Entscheidungsvariable statt reellwertige
- angepasste Mutationsoperatoren
- Beibehaltung der adaptiven Schrittweitensteuerung

■ Einige erfolgreiche Anwendungen:

- Traveling Salesman Problem (TSP),
alle Arten von Tourenplanung
- Magisches Quadrat
- Rubik's Cube
- Hochwasserspeichersystem
(Kombination verschiedener Becken)

...



Quelle: I.Rechenberg

Albrecht Dürer, *Melancholie*, 1514



Klassische EAs – Evolutionsstrategie

Weiterentwicklungen der ES:

■ geschachtelte Evolutionsstrategie

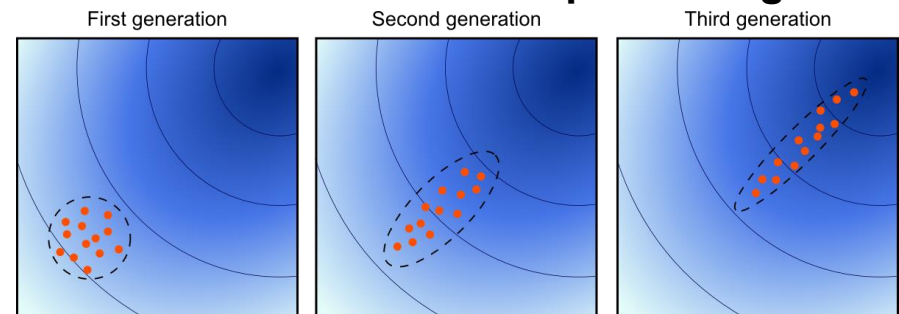
Spezielles Migrationsmodell basierend auf sich eine Zeit lang getrennt entwickelnden Unterpopulationen (Variante des Inselmodells)

■ CMA-ES (Covariance Matrix Adaptation Evolution Strategy)

- Die **Kovarianzmatrix** der mehrdimensionalen Normalverteilung der Schrittweiten beschreibt die paarweisen Abhängigkeiten zwischen den Variablen
- derandomisiertes Verfahren zur Adaption der Kovarianzmatrix
- Idee: Erhöhung der Wahrscheinlichkeit von vormals erfolgreichen Schritten, Verbesserte Anpassung der **Schrittweitensteuerung** gegenüber der ES
- gut geeignet für „schwierige“ Aufgaben der kontinuierlichen Optimierung

[Hansen]

Anpassung der Verteilung der Nachkommen im Verlauf der Evolution:



Quelle: Wikimedia

Klassische EAs – Genetische Algorithmen

Klassische Genetische Algorithmen:

Chromosom = Bitstring

- **Codierung:** Abbildung der Entscheidungsvariablen auf Bitabschnitte
Anzahl der Bits Entscheidungsvariable je nach Wertebereich oder gewünschter Präzision
- **Decodierung** zur Fitnessberechnung notwendig

Anordnung der Entscheidungsvariable wie bisher: Chromosom = (P_1, P_2, \dots, P_n)

- Jede Entscheidungsvariable wird binär codiert.
- Wertigkeit der Bits bei ganzen Zahlen: $2^n, \dots, 2^1, 2^0$ (umgekehrte Reihenfolge!)

Ermöglicht neutrale genetische Operatoren,
die ohne phänotypischen Zusammenhang wirken (müssen).

Vorteil: Vereinfacht die Implementierung

Nachteil: Anwendungsbezogene genetische Operatoren schwierig
zu implementieren

Klassische EAs – Genetische Algorithmen

Codierungsbeispiel:

Chromosom = (P_1, P_2, \dots, P_n)

Wertigkeit der Bits bei ganzen Zahlen: $2^n, \dots, 2^1, 2^0$

Beispiel für drei ganzzahlige und eine boolesche Variable P_4 :

$$0 \leq P_1 \leq 60, \quad 5 \leq P_2 \leq 30, \quad -12 \leq P_3 \leq 14$$

Anzahl der Bits pro Variable: $P_1: 6$ $P_2: 5$ $P_3: 5$


Also insgesamt 17 Bits

Beispiel eines Chromosoms: 0 1 0 1 1 0 1 1 0 0 1 1 1 1 0 0 1 $P_4=1$

$P_1 = 2^4 + 2^2 + 2^1 = 22$

$P_2 = 2^4 + 2^3 + 2^0 = 25$

**Negative Zahl:
2er-Komplement!**



1100	2er-Kompl.
-0001	1 abziehen
1011	invertieren
0100	Binärzahl

 $P_3 = -4$

Klassische EAs – Genetische Algorithmen

Ablaufschema:

(siehe auch Folie 4, Kapitel 4.4)

1. Initialisierung der Startpopulation

Zufällige Wahl der μ Individuen oder Vorbelegung einiger.

2. Partnerwahl

fitness-proportionale Selektion der Eltern

3. Erzeugung eines Nachkommens

1-Punkt-Crossover ($P_c > 0.6$) mit anschließender Mutation beider Offspring.

Mutation: Invertierung eines jeden Bits mit P_m (z.B. $P_m = 0.0001$).

4. Akzeptanz

Die so erzeugten μ Nachkommen ersetzen die Elterngeneration vollständig.
(**generational replacement**)



elitär?

5. Abbruchkriterium

Zeit, Evaluationen, Generationen. Besser: **Erreichte Fitness** oder **Stagnation**

Das Crossover ist der Hauptoperator.

Die zweitrangige Mutation soll der Fixierung von Allelwerten entgegenwirken.

Klassische EAs – Genetische Algorithmen

Viele Variationen:

■ Rangbasierte Selektion wie **Lineares Ranking**

(auch bekannt als **roulette wheel selection**).

■ **Wettkampfselektion**

- Zufällige Auswahl von n_m Individuen, die gegeneinander antreten.
- Das jeweils Beste kommt in die Elterngruppe.
- Zufällige Bestimmung der Eltern aus dieser Elterngruppe

■ 2-Punkt- oder **Uniform-Crossover** anstelle des 1-Punkt-Crossover

Liefern im Allgemeinen bessere Ergebnisse.

Uniform-Crossover: Zufällige Wahl des das Bit liefernde Elter pro Bit.

■ **Steady State GA**

Schrittweise Erzeugung von Nachkommen, die das jeweils schlechteste Mitglied der Population ersetzen, sofern sie sich von allen unterscheiden.

Es gibt keine Population von Nachkommen.



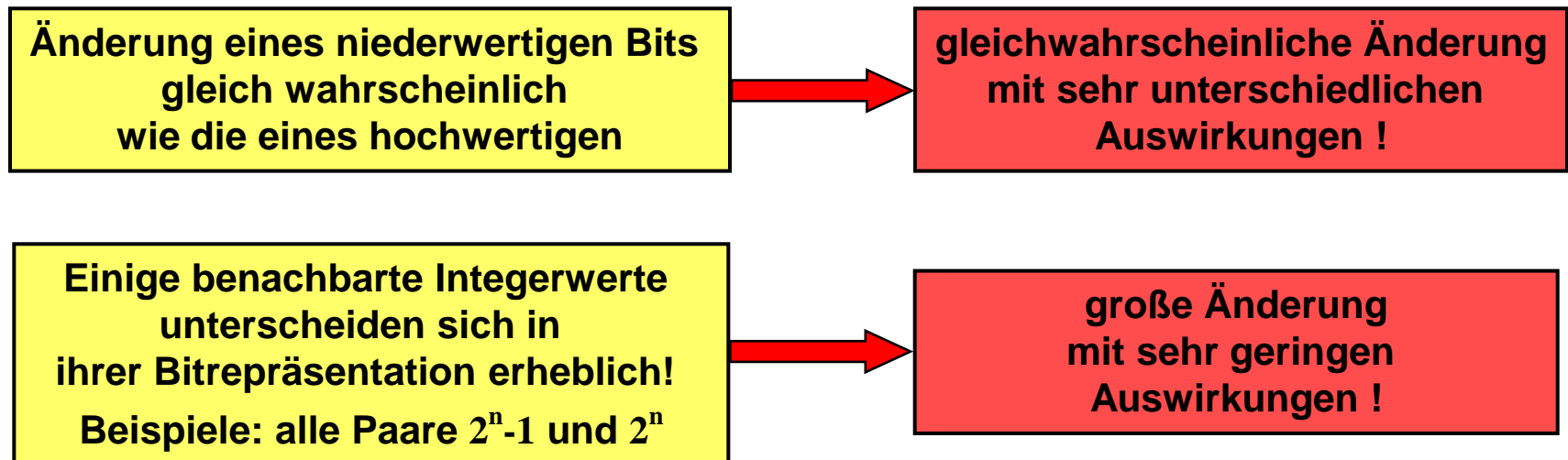
-
- Förderung des Erhalts der genotypischen Varianz

Klassische EAs – Genetische Algorithmen

Kritik der binären Codierung:

- Erschwert die Formulierung problembezogener genetischer Operatoren
- Keine **starke Kausalität** bei Änderungen zwischen Geno- und Phänotyp
Ziel: Kleine Änderung, kleine Wirkung; große Änderung, große Wirkung

Stattdessen:



■ Abhilfe: Graycodes

(Codes bei denen sich benachbarte Zahlen nur in einem Bit unterscheiden)

Klassische EAs – Genetische Algorithmen

Weitere Varianten:

■ Real-coded GAs

- Bei kontinuierlichen Problemen weitaus bessere Ergebnisse als klassische GAs
- Beibehaltung von Selektions- und Akzeptanzmechanismen
- Eigentlich ein eigenständiger EA

■ Der **Grouping Genetic Algorithm** arbeitet mit Gengruppen und Chromosomen variabler Länge.

Für kombinatorische Probleme wie *Bin Packing* oder *Clusteranalyse*.

■ ■ ■

Anwendungsgebiete:

■ **kombinatorische Probleme** wie Scheduling, TSP, Rucksackproblem, ...

■ ganzzahlige Optimierungsprobleme

■ **kontinuierliche Optimierung** (insbesondere **real-coded GAs**)

■ ■ ■

Klassische EAs – Genetische Algorithmen

Strategieparameter:

- **Populationsgröße**

häufig $30 \leq \mu \leq 200$

- **Mutationsrate**

- **Crossoverrate**

- **Anzahl der Nachkommen pro Paarung oder Generation**

- **Parameter von alternativen Selektions- oder Akzeptanzverfahren**

- **Wahl zwischen alternativen Operatoren**

Allgemeines Thema einer Suche basierend auf Metaheuristiken:

- Balance zwischen *exploration* und *exploitation*
 - Durch Wahl eines geeigneten **Selektionsdrucks** (u.a. **Partnerwahl**, **Akzeptanz**)
 - Auch die **Populationsgröße μ** beeinflusst den Selektionsdruck.
 - Wahl einer geeigneten Populationsgröße
 - zu klein: Gefahr vorzeitiger Konvergenz
 - zu groß: Verschwendung von Rechenzeit

