

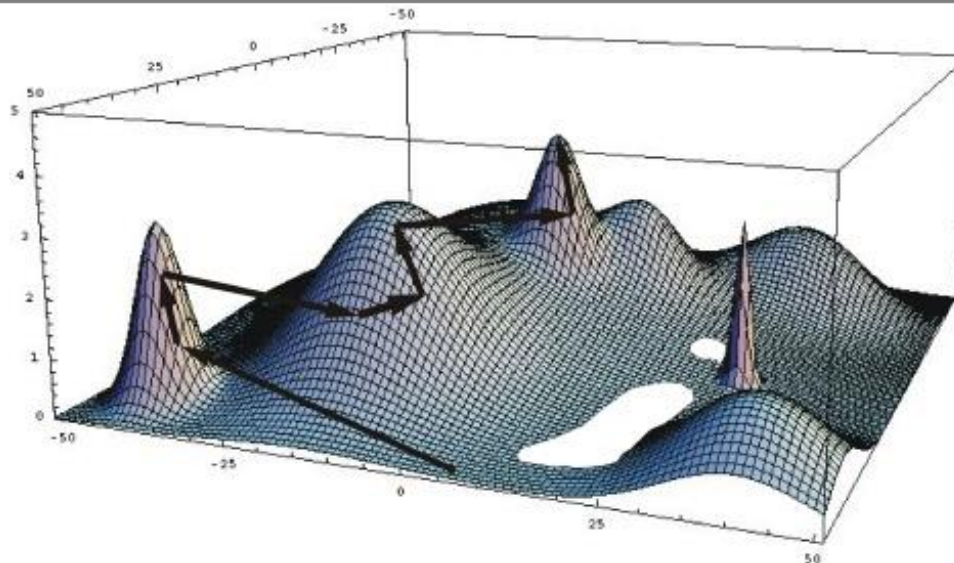
Vorlesung Computational Intelligence

Teil 4: Evolutionäre und Memetische Algorithmen

4.6 Der Evolutionäre Algorithmus GLEAM

Ralf Mikut, Wilfried Jakob, Markus Reischl

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



4.6 Der Evolutionäre Algorithmus GLEAM

Übersicht:

- Motivation
- Aufbau von GLEAM
 - Repräsentation
 - Genmodell
 - Segmentierung
 - Aktionsmodell
 - Roboterbeispiel
 - Ablaufschema und Pseudocode
 - Genetische Operatoren
 - Reparaturmechanismen
 - Bewertung
 - Akzeptanz
 - Abbruchkriterien

GLEAM – Motivation

General Learning Evolutionary Algorithm and Method

Ein eigenständiger Evolutionärer Algorithmus, der Grundelemente

- der **Evolutionsstrategie** und
- der **Genetischen Algorithmen**
- mit Konzepten der Informatik (**abstrakte Datentypen**)

verbindet.

Ideen:

- Breites Anwendungsspektrum durch **anwendungsnahe Codierung**, was anwendungsbezogene genetische Operatoren ermöglicht. (**Genmodell**)
- Nutzung von durch die Biologie inspirierten **Metastrukturen** der Chromosome: **Segmentierung** der Chromosome des EA
- Nutzung des Evolutionsprinzips zur Planung, Optimierung und Steuerung **dynamischer Prozessabläufe** (**Aktionsmodell**)

GLEAM – Aufbau – Repräsentation - Genmodell



Bedeutung eines Gens in der Biologie?



Wie viel Parameter benötigt eine phänotypische Eigenschaft?



Welcher Art (ganzzahlig, reell, Wertebereich) sind diese Parameter?

**Anwendungsabhängige Darstellung der Entscheidungsvariable
in den
Genen des Chromosoms**

Regeln des Genmodells (1):

1. Ein **Gen** enthält so viele Entscheidungsvariable oder Parameter von geeignetem Datentyp, wie es die Anwendung erfordert.
2. Ein **Gentyp** beschreibt den Aufbau eines Gens samt Wertebereichsgrenzen der Entscheidungsvariable oder Parameter.

Beispiel
eines Gentyps:

Genkennung (Typ)			
Kerbentiefe	double	[0.5, 20]	
Kerbenbreite	double	[4, 40]	
Kerbenposition	int	[0, 1]	// oben, unten
Kerbenabstand	double	[4, 100]	

} obligatorischer Teil
eines Gens

} Optionale Parameter
eines Gens
Sie unterliegen der
Evolution.

Designoptimierung eines Faltenbalgs: Beschreibung einer Kerbe

Beispiel eines Gens vom Typ *Kerbe*:

Kerbe
14.53
27.0
1
67.2

Regeln des Genmodells (2):

4. Konstruktionsregeln für **Chromosomen** je nach **Chromosomentyp**:

- **Typ 1:** Jeder Gentyp kommt genau einmal vor.
Gen-Reihenfolge ist nicht bedeutungstragend.
- **Typ 2:** Jeder Gentyp kommt genau einmal vor.
Gen-Reihenfolge ist bedeutungstragend.
- **Typ 3:** Jeder Gentyp kommt beliebig oft einschließlich gar nicht vor.
Gen-Reihenfolge ist bedeutungstragend
Dynamische Chromosomenlänge!

Parameteroptimierung

Kombinatorische (und
Parameter-) Optimierung

Optimierung dynamischer Abläufe,
Parameter- und kombinatorische
Optimierung

Eigenschaften des Genmodells:

1. Das **Genmodell** erlaubt die Formulierung allgemeingültiger

- Routinen zur Chromosomengenerierung,
- **genetischer Operatoren** einschließlich Reparaturmechanismen, sowie
- **Mutationen**, die die Wertebereichsgrenzen beachten.

2. Die anwendungsnahe Darstellung von Entscheidungsvariablen und Genen ermöglicht anwendungsbezogene genetische Operatoren.

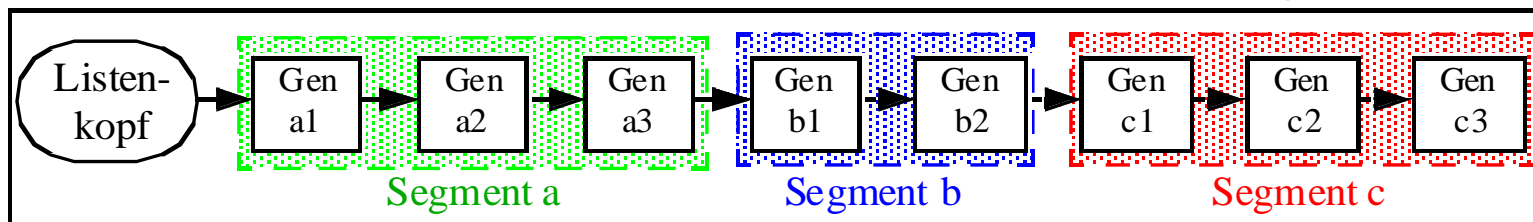
Segmentierung:

Der Evolution unterliegende Metastruktur zur Zusammenfassung und Vererbung von (guten) Teilstücken eines Chromosoms

- Organisation des Chromosoms als **lineare Liste** mit Listenkopf
- Zusammenhängende Gene bilden Einheiten, genannt **Segmente**.
- Anzahl und Größe der Segmente unterliegen der Evolution:
 - **Verschiebung** von Segmentgrenzen
 - **Teilung** von Segmenten
 - **Zusammenfassung** benachbarter Segmente
- **Mutationen** und **Crossover** greifen auf die Segmentgrenzen zurück.

Keine
phänotypischen
Auswirkungen

Beispiel eines Chromosoms als lineare Liste mit Listenkopf und drei Segmenten:



Die Erbinformation ist mehr als ein Bauplan:

- Steuerung von Wachstumsprozessen
- Steuerung der Geschlechtsreife bereits entwickelter Individuen
- Steuerung von Heilungsprozessen
- . . .

→ **Zeitbezug der Chromosomen-Interpretation**

Umsetzung in GLEAM:

- **Gene** sind **Aktionen**, die in der realen (technischen) Welt ausgeführt werden.
- Als Aktionen erhalten Gene einen Zeitbezug.
- Die Parameter der Aktionen legen die Details der Ausführung fest.
- Parameter, Aktionsanzahl und –reihenfolge unterliegen der Evolution.

Sprachliche Unterscheidung:

Statische

Dynamische zeitbezogene

Interpretation eines Chromosoms:

- | | |
|-------------------------|----------------------------|
| ■ Genmodell | ■ Aktionsmodell |
| ■ Chromosom | ■ Aktionskette (AK) |
| ■ Chromosomentyp | ■ Aktionskettentyp, AK-Typ |
| ■ Gentyp | ■ Aktionstyp |
| ■ Gene | ■ Aktionen |
| ■ Entscheidungsvariable | ■ Parameter |

Aktionsmodell (1):

- Aktionsketten werden durch einen Simulator interpretiert oder „ausgeführt“.
- Jede *Aktion der AK* führt zu einer Aktion in der simulierten oder realen Welt und greift in den Prozessablauf ein.
- Während der Aktionsausführung kann es zu Ereignissen kommen, welche die Fitness beeinflussen.
- Herstellung des **Zeitbezugs** durch einen **aktionsbezogenen Zeittakt**:

Aktion $a_1 \rightarrow t_0$

Aktion $a_2 \rightarrow t_0 + \Delta t$

...

Aktion $a_n \rightarrow t_0 + (n-1) \cdot \Delta t$

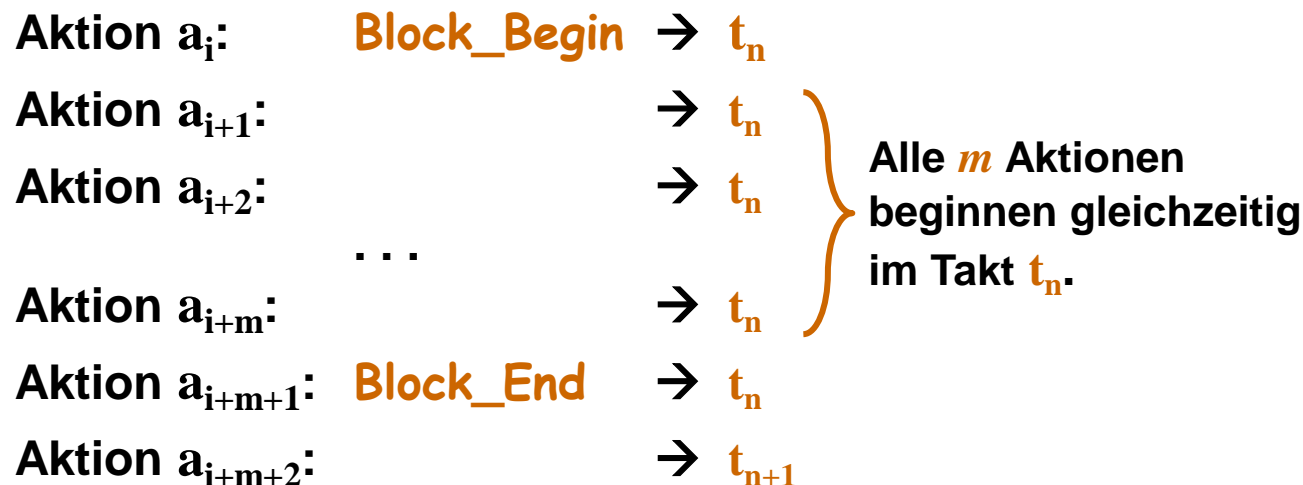
Für jede Aktion steht also eine (simulierte) Ausführungszeit Δt zur Verfügung.

Aktionsmodell (2):

Zwei Elemente des **Standardmodells zur Zeitsteuerung**:

1. **Block_Begin** und **Block_End**

Alle ***m*** Aktionen zwischen **Block_Begin** und **Block_End** starten im gleichen Zeittakt:



Aktionsmodell (3):

2. **Unchanged**

Verschiebung der Ausführung der nachfolgenden Aktion um **n** Takte (Integer-Parameter **n**).

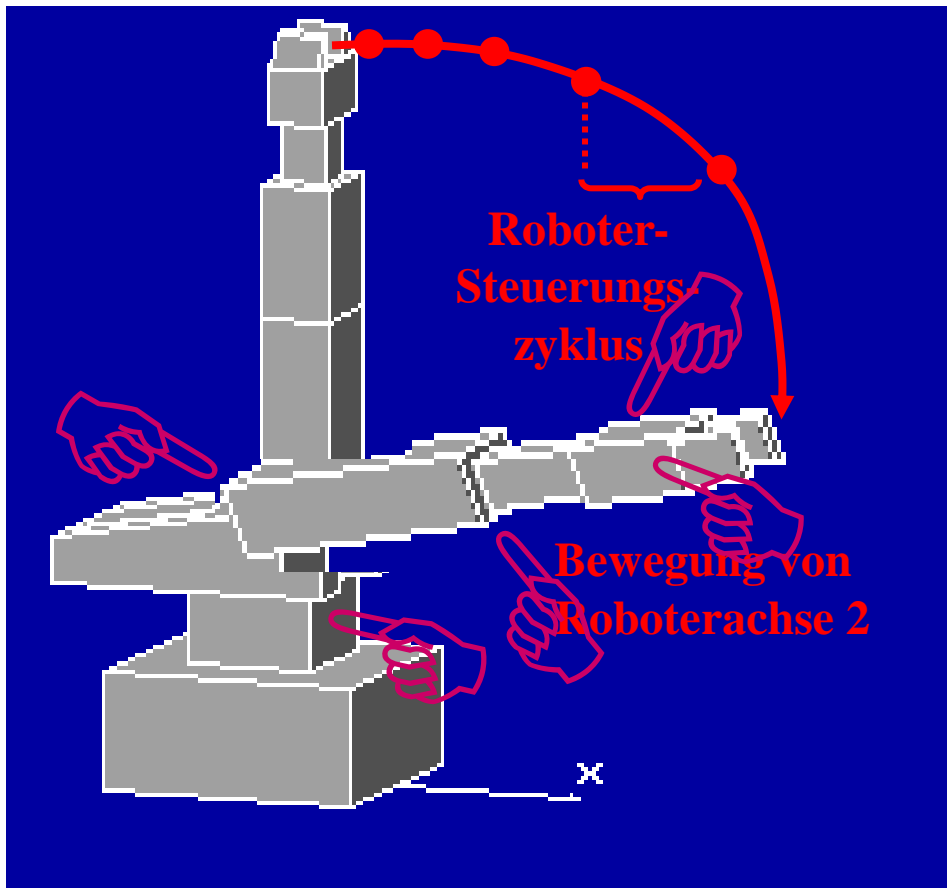
Beibehaltung der Einstellungen aller vorangegangenen Aktionen für die nächsten **n** Zyklen.

$$\begin{array}{ll} \text{Aktion } a_i: & \rightarrow t_i \\ \text{Aktion } a_{i+1}: & \text{Unchanged } n \rightarrow t_i + \Delta t \\ \text{Aktion } a_{i+2}: & \rightarrow t_i + (n+1) \cdot \Delta t \end{array}$$

Damit können beliebige zeitliche Abläufe zum Starten und Beenden von Aktionen modelliert werden.

Anwendungsbeispiel für das Aktionskonzept: [Roboterbahnplanung](#) (1)

Ansteuerung der Motoren eines Industrieroboters mit rotatorischen Achsen:



beispielhafter Achsbefehl:

MotorAn_2 mit 12 Grad/Sekunde,
mit 48 Grad/Sekunde²

Mitsubishi-Roboter RV-M1
mit 5 Achsen:

- Rumpf
- Schulter
- Ellenbogen
- Hand, knicken
- Hand, drehen

Roboterbahnplanung (2):

Achs-Befehle:

1. **MotorAn_<nr> mit Geschwindigkeit=<g_wert>, mit Rampe=<r_wert>**

Bewegung des Motors der Achse <nr>

mit einer Zielgeschwindigkeit von <g_wert> Grad/Sekunde

mit einer Rampe von <r_wert> Grad/Sekunde² zur Erreichung derselben

2. **MotorAus_<nr> mit Rampe=<r_wert>**

Anhalten des Motors der Achse <nr>

mit einer Rampe von <r_wert> Grad/Sekunde²

Damit gibt es pro Achse zwei Bewegungsaktionen.



Warum nicht zwei Motor-Aktionen mit der Achsnummer als Parameter?



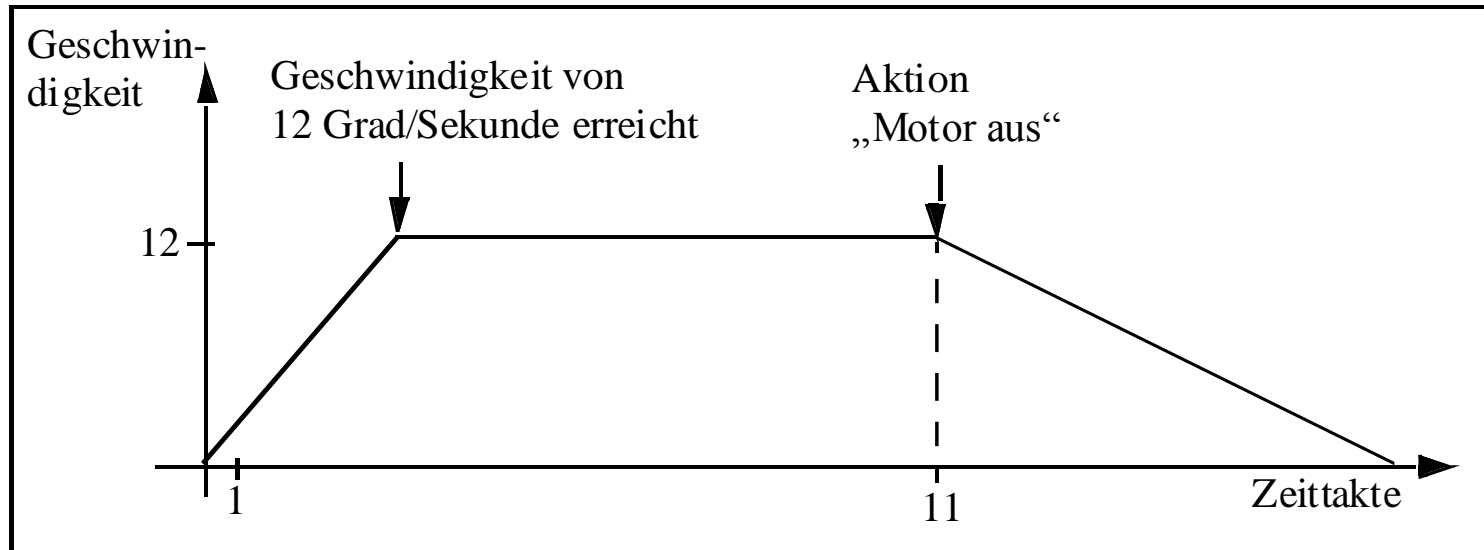
Roboterbahnplanung (3):

Beispiel: Interpretation folgender drei Aktionen:

MotorAn_2 mit Geschwindigkeit=12, mit Rampe=48 Takt 0

Unchanged 10 Takt 1

MotorAus_2 mit Rampe=17 Takt 11



GLEAM – Aufbau – Ablaufschema und Pseudocode

Pseudocode von GLEAM:

Initialisierung und Bewertung der Startpopulation

REPEAT (Generationen-Schleife)

FOR alle Individuen der Population

Wähle *ranking-basiert* einen Partner innerhalb der Nachbarschaft

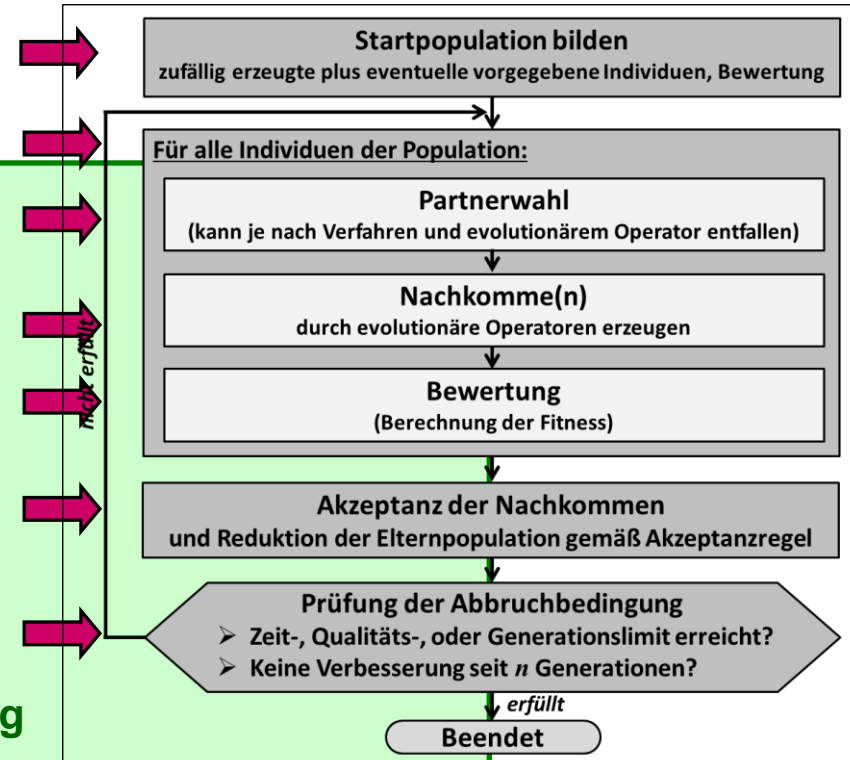
FOR alle Gruppen genetischer Operatoren

Nachkommenerzeugung und Bewertung

Akzeptanz oder Zurückweisung des besten Nachkommen gemäß Akzeptanzregel und positivenfalls Ersetzung des Elter

UNTIL Abbruchkriterium erfüllt (Zeit, Qualität, Stagnation, ...)

Liefere bestes Individuum (und weitere, falls vorgegeben) als Ergebnis



Allgemeiner EA

Übliche Werte des Ranking-Parameters sp im Bereich zwischen 1.4 und 1.6.

Übersicht über die **genetischen Operatoren**:

- Standard-**Mutationen**, Beachtung der Wertebereiche des Genmodells
- Standard-**Crossoveroperatoren**, Beachtung des Chromosomentyps
- Schnittstelle für anwendungsspezifische Operatoren

Die Operatoren werden zu Gruppen zusammengefasst, die

- zwei Nachkommen durch Crossover erzeugen oder
- zwei Nachkommen durch Crossover + anschließenden Mutationen erzeugen oder
- ein mutiertes Elterklon erzeugen.

Dabei hat jede Gruppe und jeder Operator jeweils eine eigene Ausführungswahrscheinlichkeit.

Konsequenzen:

1. Die Anzahl der Nachkommen pro Paarung variiert.
2. Klone können zufallsbedingt unverändert bleiben und werden gelöscht.

Mutation **relative Parameteränderung** (1):

Ziele:

- Einhaltung der Bereichsgrenzen
- Gleiche Wahrscheinlichkeit für Vergrößerung und Verkleinerung unabhängig vom aktuellen Wert
- Kleine Änderungen erheblich wahrscheinlicher als große (Vergleichbar mit der von der ES her bekannten **Glockenkurve** der **Normalverteilung**)
- Schnelle Berechenbarkeit

Vorgehensweise:

1. Einteilung des verfügbaren Gesamtänderungsbereichs in 10 gleichgroße Teilbereiche
2. Der erste Teil bildet das **erste Änderungsintervall**,
der erste und zweite Teil das **zweite Änderungsintervall**,
der erste, zweite und dritte Teil das **dritte Änderungsintervall**, usw.
3. Jedem Änderungsintervall wird die gleiche Wahrscheinlichkeit von 10% zugeordnet.



Mutation **relative Parameteränderung** (2):

Algorithmus:

1. Entscheide gleichverteilt, ob vergrößert ($vz = 1$) oder verkleinert ($vz = -1$) wird.
Daraus ergibt sich der Gesamtänderungsbereich.
2. Wähle das Änderungsintervall gleichverteilt aus.
3. Wähle den Änderungswert Δw aus diesem Intervall gleichverteilt aus.
4. Berechne neuen Parameterwert: $w_{neu} = w_{alt} + vz \cdot \Delta w$

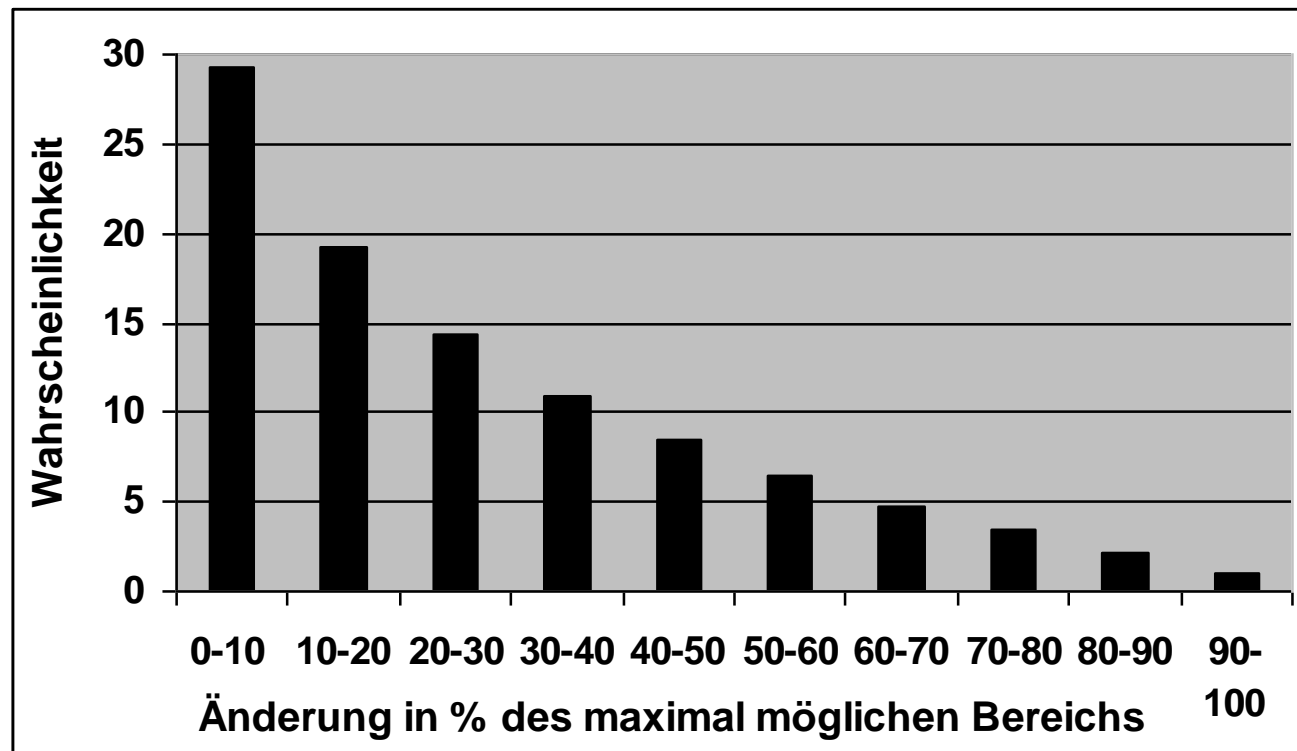
Einteilung und Algorithmus ergeben folgende summierte Wahrscheinlichkeiten für den

1. Teilbereich (0 - 10% Änderung): $10\% \cdot (1 + 1/2 + 1/3 + \dots + 1/10) = 29,3\%$
2. Teilbereich (10 - 20% Änderung): $10\% \cdot (1/2 + 1/3 + \dots + 1/10) = 19,3\%$
3. Teilbereich (20 - 30% Änderung): $10\% \cdot (1/3 + \dots + 1/10) = 14,3\%$
- ...
10. Teilbereich (90 - 100% Änderung): $10\% \cdot (1/10) = 1\%$



Mutation [relative Parameteränderung \(3\)](#):

Einteilung und Algorithmus ergeben folgende summierte Wahrscheinlichkeiten der einzelnen Teilbereiche:

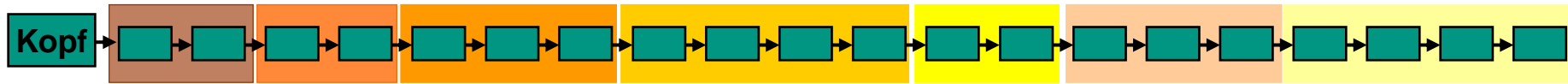


→ *Kleine Änderungen erheblich wahrscheinlicher als große*

Übersicht über die gen- oder aktionsbezogenen Mutationen:

Gen- oder aktionsbezogene Standard-Mutationen	Chromosomen- oder AK-Typ		
	Typ 1	Typ 2	Typ 3
Änderung des Parameterwerts	ja	ja	ja
Neuer Parameterwert	ja	ja	ja
Änderung aller Parameter eines Gens	ja	ja	ja
Erneuerung aller Parameter eines Gens	ja	ja	ja
Verschiebung		ja	ja
Ersetzung			ja
Einfügung			ja
Verdoppelung			ja
Löschung			ja

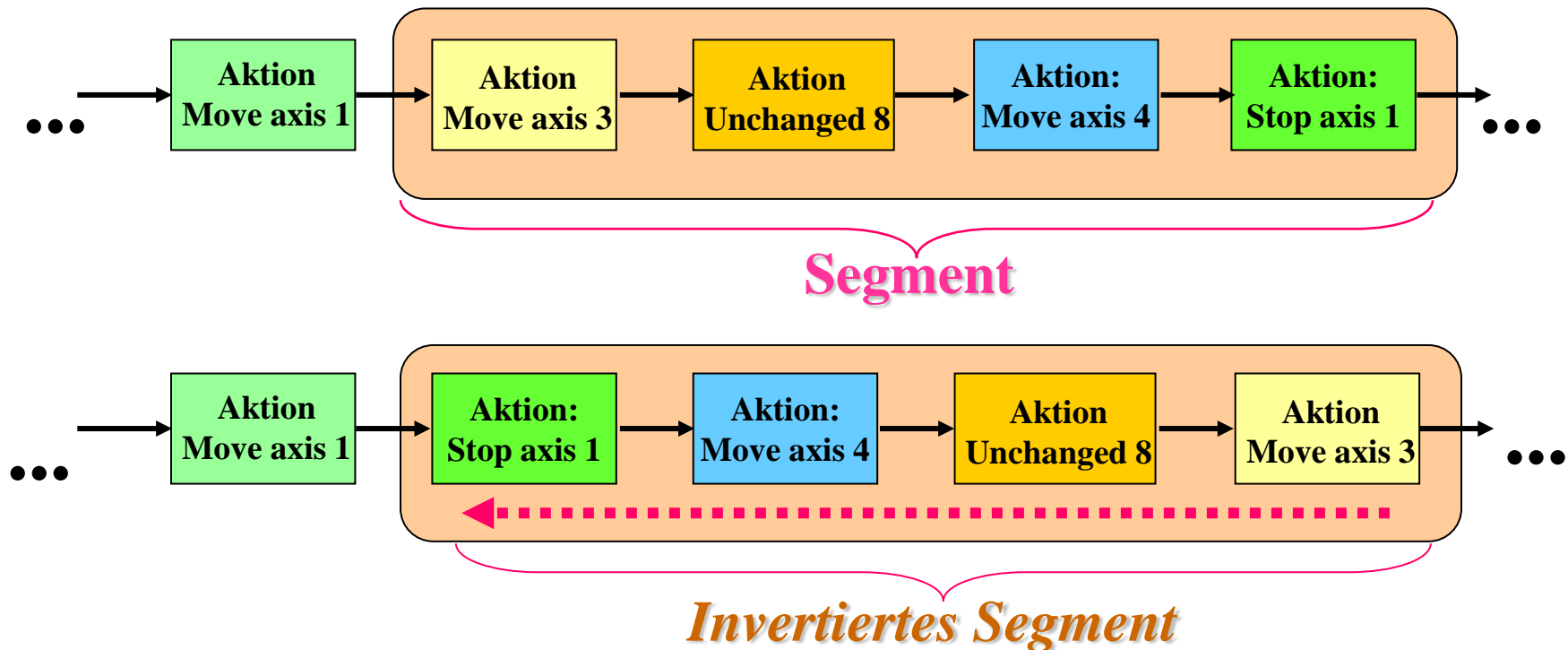
Übersicht über die segmentbezogenen Mutationen:



Segmentbezogene Standard-Mutationen	Chromosomen- oder AK-Typ		
	Typ 1	Typ 2	Typ 3
Parameteränderung der Gene eines Segments	ja	ja	ja
Erneuerung von Parametern der Gene eines Segments	ja	ja	ja
Inversion (Umkehrung der Genreihenfolge)		ja	ja
Verschmelzung benachbarter Segmente	ja	ja	ja
Verschmelzung nicht benachbarter Segmente		ja	ja
Teilung eines Segments	ja	ja	ja
Verschiebung		ja	ja
Ersetzung			ja
Einfügung			ja
Verdoppelung			ja
Löschung			ja

Segmentmutationen:

Invertierung der Aktions- oder Genreihenfolge eines Segments:



Wirkung einer *einzelnen* Mutation: Achse 1 stoppt sofort,
Achse 4 startet früher und läuft länger und
Achse 3 startet später und läuft kürzer

GLEAM – Aufbau – Genetische Operatoren

Crossoveroperatoren (1):

GLEAM enthält insgesamt vier Standard-Crossoveroperatoren, die alle auf den **Segmentgrenzen** aufbauen.

Drei allgemeine Crossoveroperatoren, welche die Chromosome an den Segmentgrenzen aufteilen:

- **1-Punkt-Crossover**
- **n-Punkt-Crossover**
- **Segmentaustausch** (Austausch genau eines Segments)

Sie sind bei allen Chromosomentypen sinnvoll.

Die drei Operatoren können **illegale Nachkommen** erzeugen:

Bei welchen Chromosomen- od. AK-Typen? Beispiele?



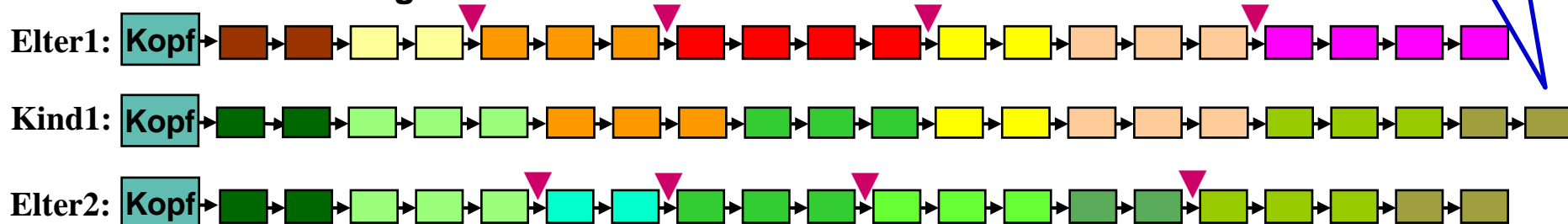
Crossoveroperatoren (2):

n-Punkt-Crossover:

Algorithmus:

1. Bildung des 1. Kindes durch abwechselndes Kopieren einer ausgewürfelten Anzahl von Segmenten der Eltern.
2. Die Segmente jeweils anderen Elter bilden das 2. Kind.
3. Je nach Chromosomentyp Verschiebung überzähliger Gene auf das jeweils andere Kind (Reparatur)

Beispiel: 1. Zwei Segmente von Elter2
2. Ein Segment von Elter1
3. Ein Segment von Elter2
4. Zwei Segmente von Elter1
5. Zwei Segmente von Elter2



Crossoveroperatoren (3):

Crossoveroperator für kombinatorische Probleme:

Segmentorientierte Variante des **Order-based Crossover** (OX):
(siehe Folie 12, Kap. 4.4)

- Die Segmentierung des 1. Elter bestimmt die Anzahl und Länge der Sequenzen.
- Diese Sequenzen werden ohne Rücksicht auf die Segmentierung des 2. Elters zur Bildung der Kinder genutzt.
- Kind 1 erbt die Segmentstruktur des ersten Elter und Kind 2 entsprechend.

Können dabei illegale Nachkommen entstehen?



Für welche Chromosomentypen sind diese Operatoren sinnvoll?



Reparaturmechanismen (1):

Unzulässige Nachkommen können im Allgemeinen entstehen durch:

- Änderungen von Entscheidungsvariablen oder Parametern:
 - Verletzung der Unter- bzw. Obergrenzen
bei GLEAM ausgeschlossen
 - Werte in unzulässigen Bereichen innerhalb der Definitionsgrenzen
meist Sache der Bewertung. So auch bei GLEAM.
- Mutationen zur Genverschiebung bei kombinatorischen Aufgaben mit einzuhaltenden Reihenfolgen: entweder Sache der Bewertung oder Reparatur (genotypische Reparatur, Genetic Repair)
- die allgemeinen Crossoveroperatoren (Gene zu viel oder zu wenig)
entweder Sache der Bewertung oder genotypische Reparatur (GLEAM)

Problem bei der Reparatur:

Sinnvolle Änderungen können nicht auf mehrere Schritte verteilt werden, wenn die Zwischenschritte zu unzulässigen Phänotypen führen.

Alternativen: Bestrafung unzulässiger Phänotypen, phänotypische Reparatur

Reparaturmechanismen (2):

Genetic Repair (1):

Fall: Mutationsbedingte **unzulässige Genreihenfolgen**

Lösungsalternativen:

1. Falsch positionierte Gene bis zu einer zulässigen Position verschieben:
(**genotypische Reparatur**)
 - in Richtung Chromosomenanfang (AK-Kopf)
 - in Richtung Chromosomenende (AK-Ende)
2. Bei der Interpretation so lange hinten an stellen, bis das Gen zulässig wird.
(**phänotypische Reparatur**)

Die **phänotypische Reparatur**

- erlaubt die Aufteilung einer zulässigen/positiven Änderung auf mehrere Zwischenschritte (genetische Operatoren) und
- muss auch bei der Ergebnisauswertung erfolgen!

Reparaturmechanismen (3):

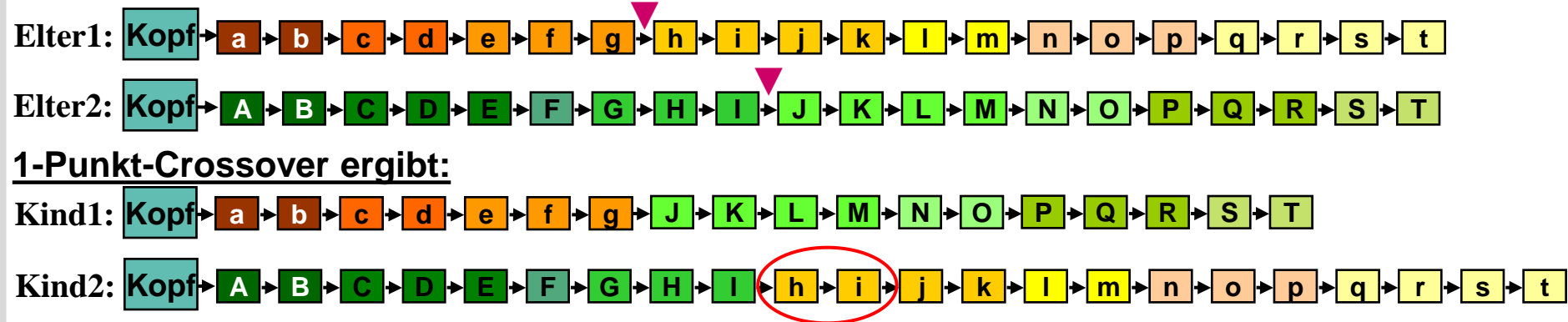
Genetic Repair (2):

Fall: **Zu viel oder zu wenig Gene** auf Grund von Crossover:
(GLEAM: nur bei Chromosomentyp 1 und 2 möglich)

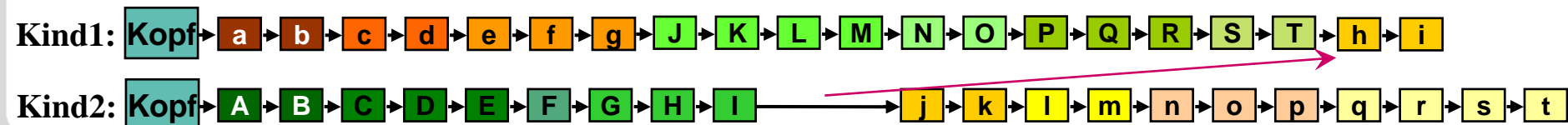
Lösung:

Verschiebung überzähliger Gene auf das jeweils andere Kind

Beispiel (basierend auf 1-Punkt-Crossover):



Genetic Repair ergibt:



Bewertung mit der gewichteten Summe und Straffunktionen (1):

1. Normierung der Bewertung der einzelnen Kriterien

- Abbildung der Werteskala der Kriterien auf eine einheitliche Fitness-Skala: $0 \dots f_{max}$
- Verwendung von 6 Standard-Normierungsfunktionen:
 - linear
 - exponentiell
 - gemischt linear und exponentiell

2. Bildung der Summe aller Kriterien → Rohfitness

3. Berechnung der Straffunktionen, soweit zutreffend → Straffaktoren $\in [0, 1]$

4. Multiplikation der Rohfitness mit allen Straffaktoren → Endfitness

Ausblick: Kaskadierte Gewichtete Summe

- Gruppierung der Kriterien nach Prioritäten
- Jedes Kriterium erhält einen Schwellwert.
- Zunächst tragen nur die Kriterien mit höchster Priorität zur gewichteten Summe bei.
- Wenn alle Kriterien einer Gruppe den jeweiligen Schwellwert überschritten haben, werden die Kriterien der Gruppe mit der nächst niedrigeren Priorität aktiviert. [Jak14]

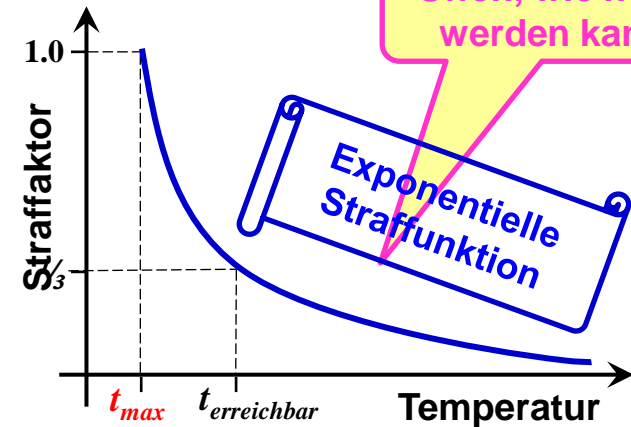
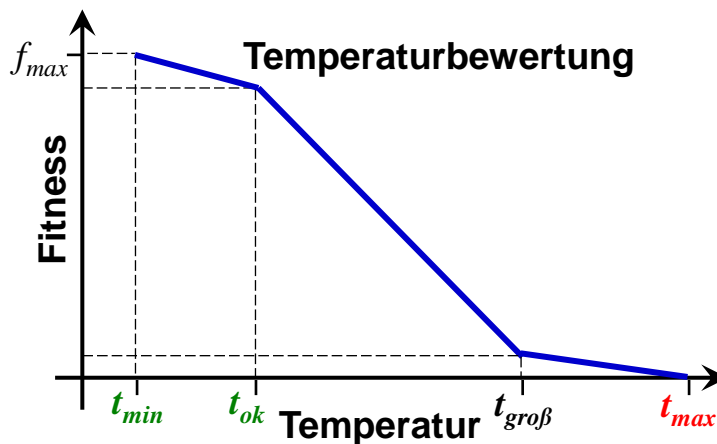
Bewertung mit der gewichteten Summe und Straffunktionen (2):

Beispiel für den Einsatz von Normierungs- und Straffunktion:

Situation: Die Temperatur soll möglichst gering sein und einen Maximalwert t_{max} nicht überschreiten. Werte bis zu t_{ok} sind unproblematisch. Es gibt weitere Kriterien.

Lösung: Normierung im Bereich bis zu t_{max} mit einer linearen Funktion. Die Fitnessunterschiede zwischen minimaler Temperatur und t_{ok} sind gering. Danach starker Abfall bis zu einem noch akzeptablen Wert $t_{groß}$.

Bei Überschreitung von t_{max} wirkt folgende Straffunktion:



Warum eine Straffunktion und nicht nur Abwertung auf 0 ab t_{max} ?



Bewertung mit der gewichteten Summe und Straffunktionen (3):

Kriterien und Hilfskriterien:

■ Bewertungskriterien:

- Ergeben sich aus der Aufgabenstellung (**primäre Ziele**).

■ Hilfskriterien:

- sollen die Erreichung primärer Kriterien unterstützen:
 - z.B. Bessere Erreichung einer Senkung der **Energiespitzen** durch
 - Bewertung des Energiespitzenwertes
 - Bewertung der Anzahl aller Spitzen, die ein Limit überschreiten
 - Bewertung des Gesamtenergieverbrauchs aller ein Limit überschreitenden Spitzen
- sollen übergeordneten Zielen dienen:
 - z.B. dient die Bewertung der *Chromosomenlänge* bei Chromosomen vom Typ 3 der sanften Begrenzung der dynamischen Chromosomenlänge.

Beseitigung oder
Verkleinerung einer
von mehreren gleich
großen Spitzen
???

Akzeptanzregeln:

Der beste Nachkomme ersetzt das Elter gemäß einer der folgenden Regeln:

1. Akzeptiere immer (always)
Akzeptiere immer den besten Nachkommen.
2. Akzeptiere immer, elitäre Strategie (always, ES)
Akzeptiere den besten Nachkommen, wenn entweder das Elter nicht das Deme-Beste ist oder der Nachkomme besser als sein Elter ist.
3. Lokal Schlechtestes (local least)
Akzeptiere den besten Nachkommen, wenn er besser als das schlechteste Deme-Mitglied ist.
4. Lokal Schlechtestes, elitäre Strategie (local least, ES)
Akzeptiere den besten Nachkommen, wenn er besser als das schlechteste Deme-Mitglied ist UND wenn entweder das Elter nicht das Deme-Beste ist oder der Nachkomme besser als sein Elter ist.
5. Elter-Verbesserung (better parent)
Akzeptiere den besten Nachkommen, wenn er besser als das Elter ist.

Abbruchkriterien:

Stagnationsbezogene Abbruchkriterien bei einem Nachbarschaftsmodell:
(neben Zeit, Fitness oder Generationen, vgl. Folie 19, Kap. 4.4)

- **GDV**

Generationen ohne **Deme-Verbesserung** (= Verbesserung des (Deme-)Besten)

- **GAk**

Generationen ohne **Akzeptanz** (= keine Nachkommenakzeptanz (im Deme))

Welches Kriterium führt schneller zum Abbruch und ist damit schärfer?



Stagnation vs. Konvergenz:

Stagnation: Ausbleiben von Verbesserungen

Konvergenz: Genotypische Ähnlichkeit aller Individuen einer Population
Kann durch den Hammingabstand der Individuen bestimmt werden.
(aufwändig)

