

Zentrale Konzepte Evolutionärer Algorithmen

Jens Kosiol

(Foliensatz basiert weitgehend auf dem entsprechenden Foliensatz von
Eiben und Smith)

Evolution und Evolutionäre Algorithmen

Natürliche Evolution

- Eine Population von Individuen existiert in einer Umwelt mit begrenzten Ressourcen.
- Im Wettkampf um diese Ressourcen haben Individuen, die besser an die Umwelt angepasst sind, einen Vorteil.
- Die besser angepassten Individuen geben ihr genetisches Material (ggf. modifiziert durch Crossover und Mutation) überproportional oft an die nachkommende Generation weiter.
- Auch die Nachkommen nehmen am Wettkampf um die natürlichen Ressourcen (ggf. gegen die Eltern) teil.
- Durch die natürliche Selektion steigt die Fitness der Population mit der Zeit.

Evolutionäre Algorithmen

- Evolutionäre Algorithmen sind sogenannte „generate and test“ Algorithmen (heuristische Suche).
- Sie sind randomisiert und evolvieren eine Population (nicht eine einzelne Lösung).
- **Variationsoperatoren** (Crossover und Mutation) erzeugen neue Lösungen und sorgen für **Diversität**.
- **Selektion** schränkt Diversität ein und setzt **Qualität** durch.

Gemeinsames Modell evolutionärer Prozesse

- Es gibt eine **Population** von Individuen.
- Individuen haben einen **Geno-** und einen **Phänotyp**.
- Individuen haben eine **Fitness**, die durch den Phänotyp bestimmt ist.
- Auf dem Genotyp der Individuen wirken **Variationsoperatoren: Crossover und Mutation**.
- Es wirkt ein **Selektionsdruck** in Richtung höherer Fitness und zwar durch:
 - Natürliche Selektion („survival of the fittest“)
 - Sexuelle Selektion („mating of the fittest“)

Insgesamt führt dies zu einem **Optimierungsprozess** in Bezug auf auf das Fitnesskriterium.

Zwei Säulen der Evolution

In evolutionären Prozessen wirken zwei entgegengesetzte Kräfte.

Die **Diversität** der Population wird **vergrößert** durch die Variationsoperatoren

- Mutation
- Rekombination/Crossover

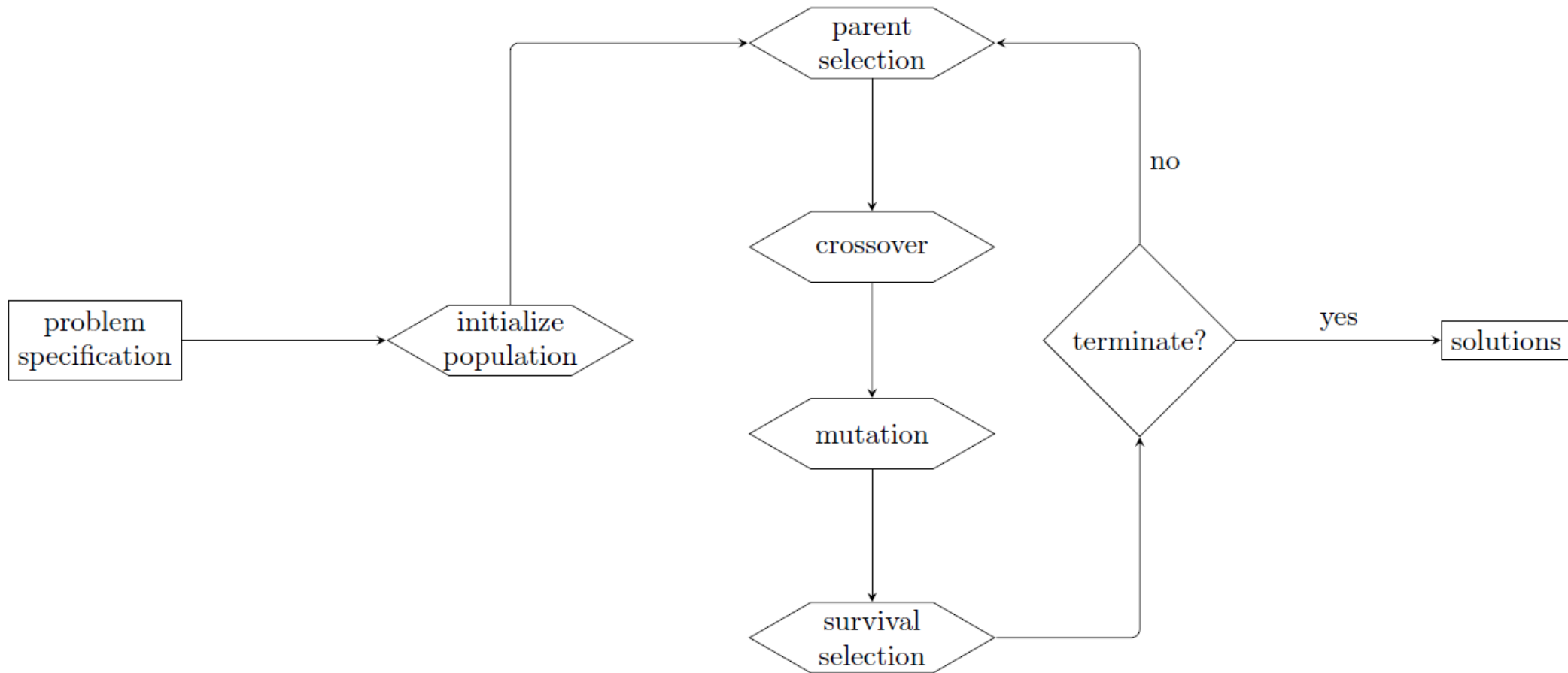
Dies erzeugt Druck in Richtung **Neuheit**.

Die **Diversität** der Population wird **verkleinert** durch die Selektionsoperatoren

- parent selection
- survivor selection

Dies erzeugt Druck in Richtung **Qualität**.

Schema eines evolutionären Algorithmus



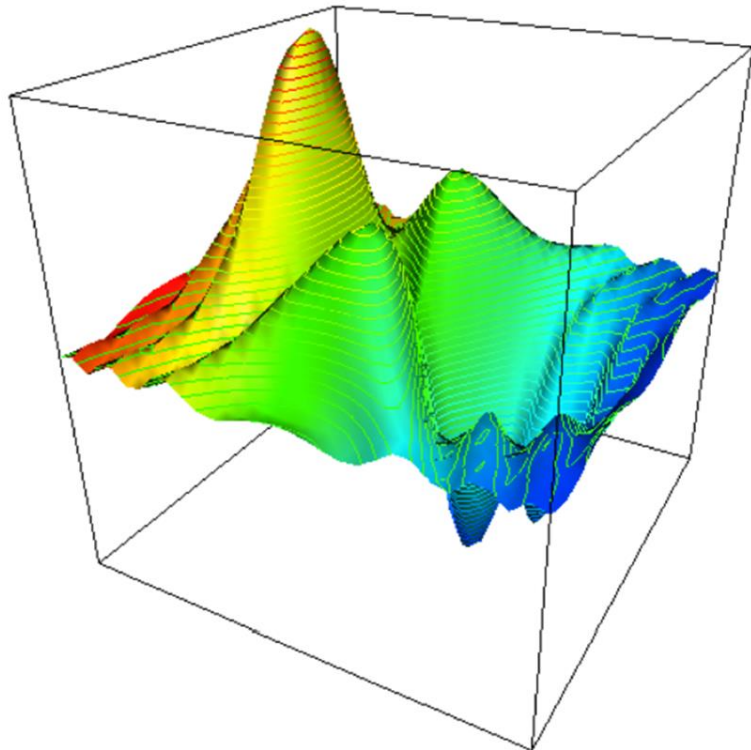
Schema in Pseudocode

```
BEGIN
  INITIALISE population with random candidate solutions;
  EVALUATE each candidate;
  REPEAT UNTIL ( TERMINATION CONDITION is satisfied ) DO
    1 SELECT parents;
    2 RECOMBINE pairs of parents;
    3 MUTATE the resulting offspring;
    4 EVALUATE new candidates;
    5 SELECT individuals for the next generation;
  OD
END
```

Redeweisen

- Die Multimenge P , auf der ein evolutionärer Algorithmus arbeitet, heißt **Population**.
- Ein Element von P heißt **Individuum, Struktur, Genotyp, String, Chromosom** oder **(potentielle) Lösung**. Die Menge aller möglichen Chromosomen ist der **Suchraum**.
- Einheiten, aus denen ein Chromosom aufgebaut ist, heißen **Gene, Eigenschaften (features), characters, decoders**.
- Die Werte, die ein Gen annehmen kann, heißen **Allele**.
- Insbesondere bei Kodierung auf Basis von Vektoren: Positionen im Vektor heißen **Loci**.

Metapher der Fitnesslandschaft



- Von Sewall Wright 1932 als Konzept zum Studium evolutionärer Phänomene vorgeschlagen
- Die z-Achse repräsentiert die Fitness, während die x - y -Ebene die genetischen Eigenschaften der Individuen erfasst.
- Jeder Punkt der Landschaft repräsentiert ein Individuum.
- Eine Population entspricht einer „Punktwolke“; durch Evolution bewegt sich die Population durch die Landschaft.

Typischer Ablauf evolutionärer Algorithmus

Phasen während der Optimierung einer eindimensionalen Fitnesslandschaft



Frühe Phase:
(quasi-)zufällige Verteilung der Population

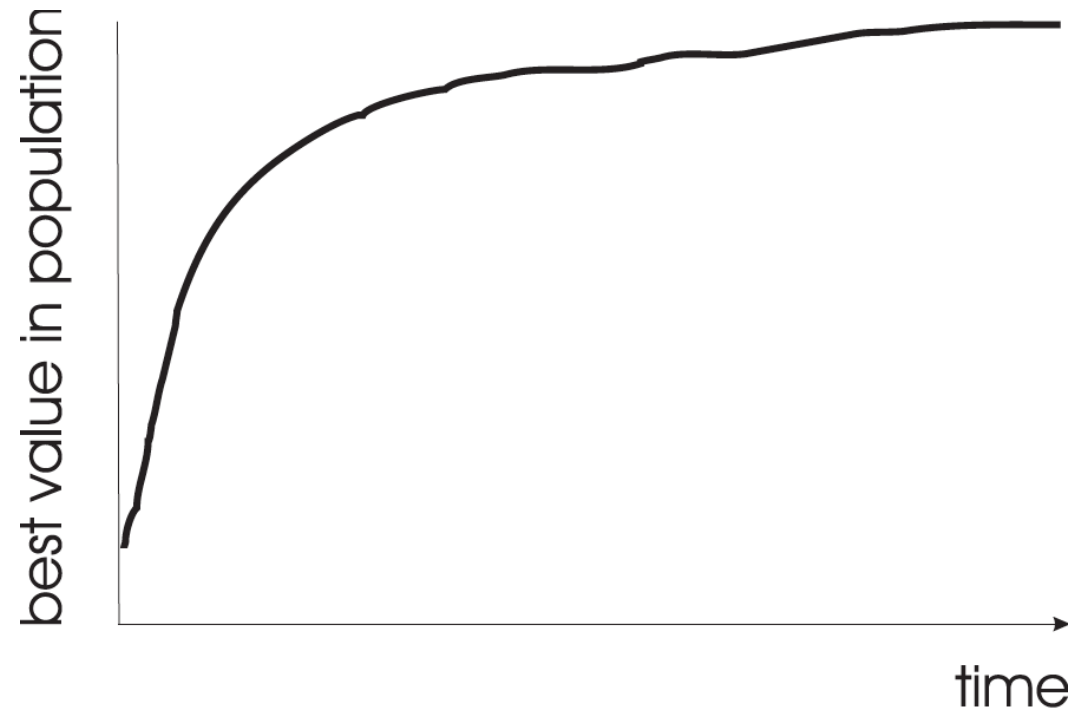


Mittlere Phase:
Population um Optima herum arrangiert



Späte Phase:
Population bei ausgeprägten Optima konzentriert

Typischer Ablauf EA: Entwicklung der Fitness



Evolutionäre Algorithmen sind “anytime algorithms”

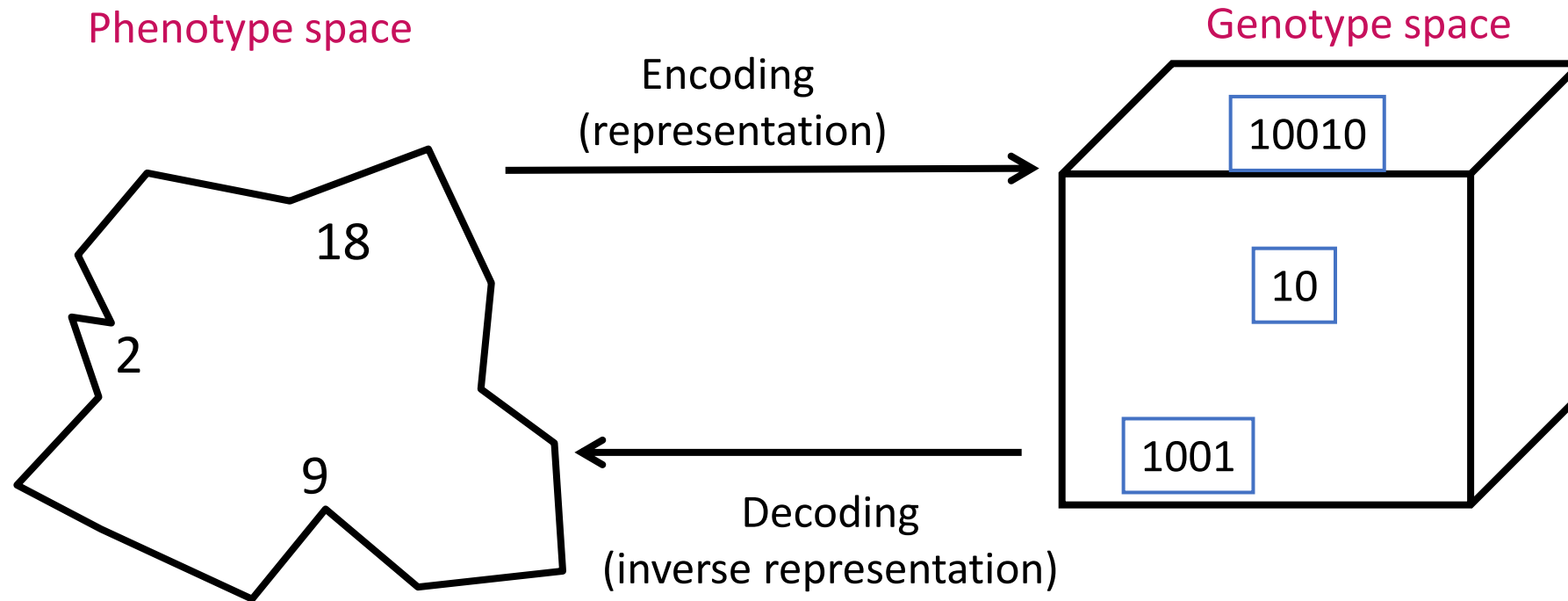
Hauptkomponenten evolutionärer Algorithmen

- Darstellung und Qualität von Lösungen
 - Kodierung
 - Evaluation der Fitness
 - Population
- Auswahl von Lösungen
 - parent selection
 - survivor selection
- Modifikation von Lösungen durch Variationsoperatoren
 - Rekombination/Crossover
 - Mutation

EA Hauptkomponenten: Repräsentation

- Zweck: Darstellung möglicher Lösungen, auf der Variationsoperatoren wirken können
- Führt zu zwei Arten der Existenz möglicher Lösungen:
 - **Phänotyp**: Objekt im eigentlichen Problemkontext
 - **Genotyp**: Kodierung dieses Objekts (**Chromosom**)
- Führt zu zwei Abbildungen:
 - Kodierung: Phänotyp \Rightarrow Genotyp
 - Dekodierung: Genotyp \Rightarrow Phänotyp
- Chromosomen enthalten **Gene**, die sich (in den meisten Repräsentationen) an festen Orten befinden, die **Loci** (Sg. **Locus**) genannt werden und mit Werten (**Allelen**) belegt werden können.

Beispiel Repräsentation



EA Hauptkomponenten: Fitnessfunktion und Evaluation

- Zweck:
 - Repräsentiert die Aufgabe, die gelöst werden soll, die Anforderungen, an die Lösungen sich anpassen sollen (übernimmt die Rolle der „Umwelt“)
 - Ermöglicht Selektion (Grundlage für den Vergleich der Fitness von Lösungen)
- Für unterschiedliche Arten von Problemen und unterschiedliche Domänen haben sich unterschiedliche Namen eingebürgert: fitness function, objective function, loss function, ...
- Ordnet jedem Phänotyp eine eindeutige, reellwertige Fitness zu; Varianten:
 - Dekodierung von Genotypen und Auswertung der Fitness in eine Funktion zusammenfassen
 - Fitness ist kein absoluter Wert und es liegt keine Funktion sondern eine **Relation (Ordnung)** vor: Gegeben zwei Lösungskandidaten, können wir auswerten, welcher besser ist.
- Typischerweise sprechen wir von Maximierungsproblemen.
 - Konversion von Minimierungsproblemen in Maximierungsprobleme ist trivial; wir schränken die Anwendbarkeit also nicht ein.

EA Hauptkomponenten: Population

- Rolle: Enthält die gegenwärtig betrachteten Lösungskandidaten als Individuen (Genotypen)
- Formal ist eine Population eine **Multimenge**: Ein Lösungskandidat kann mehrfach darin enthalten sein!
- Evolution findet auf Ebene der Population statt, nicht auf Ebene der Individuen!
 - Die Selektionsoperatoren wirken auf Populationen.
 - Die Variationsoperatoren wirken auf Individuen.
- Eine Population kann eine geometrische Struktur haben, die evolutionäre Algorithmen nutzen können.
- Wichtiges Konzept: **Diversität** einer Population; dies kann sich darauf beziehen, wie verschieden die enthaltenen Elemente in Bezug auf Fitness, Phänotyp oder Genotyp sind.

EA Hauptkomponenten: Selektion

- Rolle: Identifiziert Individuen, die
 - sich rekombinieren dürfen (Eltern werden)
 - Überleben
- Lenkt die Population in Richtung höherer Fitness, aber führt nicht selbst zu Variation
- Arbeitet normalerweise probabilistisch: Individuen mit hoher Fitness werden mit höherer Wahrscheinlichkeit ausgewählt (aber normalerweise keine Garantien)
- Der probabilistische Ansatz ermöglicht es, lokalen Optima entkommen zu können

EA Hauptkomponenten: Selektion

Parent und survivor selection

In einem evolutionären Algorithmus gibt es zwei Arten von Selektion.

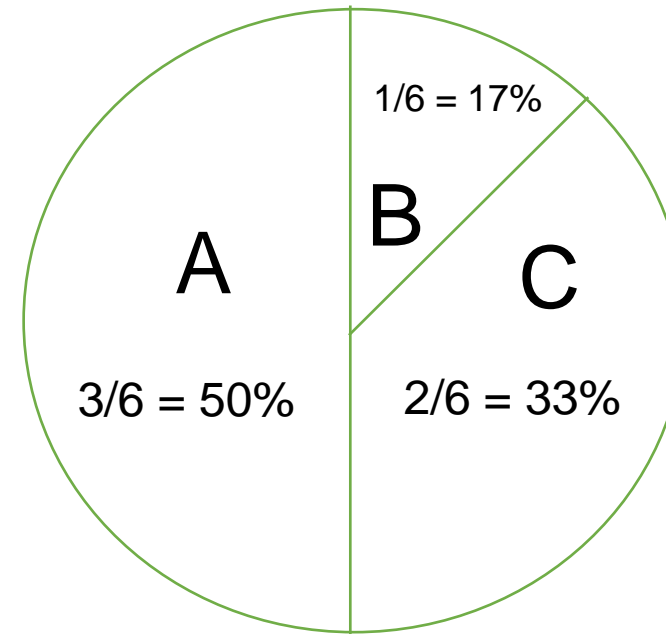
- Parent selection
 - Auswahl, welche Individuen sich fortpflanzen (ihr Genmaterial weitergeben) dürfen
 - Normalerweise probabilistisch
- Survivor selection
 - Die meisten Evolutionären Algorithmen haben eine fixe Populationsgröße → die Anzahl der Lösungen muss nach der Rekombination reduziert werden
 - Wird häufig deterministisch umgesetzt
 - Beispiel in unserem SGA: Kindgeneration ersetzt die Elterngeneration vollständig
 - Manchmal werden auch Kombinationen aus deterministischen und probabilistischen Ideen benutzt

Beispiel Selektion: Roulette Wheel Selection

fitness(A) = 3

fitness(B) = 1

fitness(C) = 2



Im Prinzip kann jeder Selektionsmechanismus sowohl für die parent als auch für die survivor selection eingesetzt werden.

EA Hauptkomponenten: Variationsoperatoren

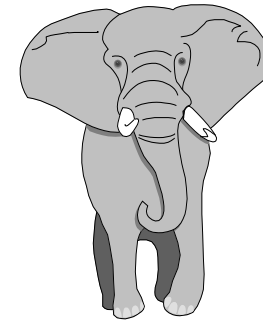
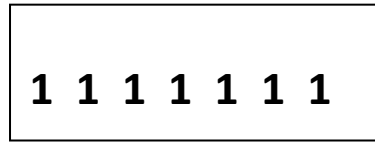
- Rolle: Neue Lösungskandidaten generieren
- Werden anhand ihrer **Stelligkeit** (Anzahl der Eingabeparameter) eingeteilt:
 - Stelligkeit = 1: **Mutationsoperatoren**
 - Stelligkeit > 1: **Rekombinationsoperatoren**
 - Stelligkeit = 2: Typischerweise **Crossover** genannt
 - Stelligkeit > 2: Im Prinzip möglich, aber kaum genutzt
- Heutzutage nutzen die meisten Evolutionären Algorithmen sowohl Mutation als auch Crossover

EA Hauptkomponenten: Mutation

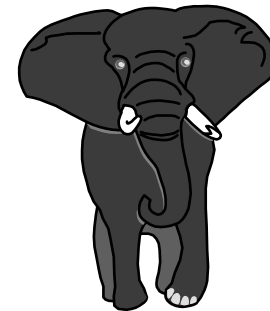
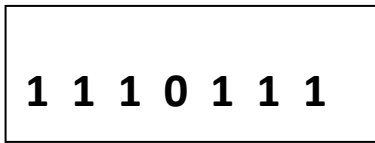
- Rolle: Mutationen führen kleine, zufällige Änderungen durch.
- Wirkt auf einem Genotypen und produziert einen anderen
- Definierende Eigenschaft nach [PBB+15]: Jeder Genotyp ist von jedem Genotyp aus durch (Hintereinanderausführung von) Mutationen erreichbar.

Intuition Mutation

before



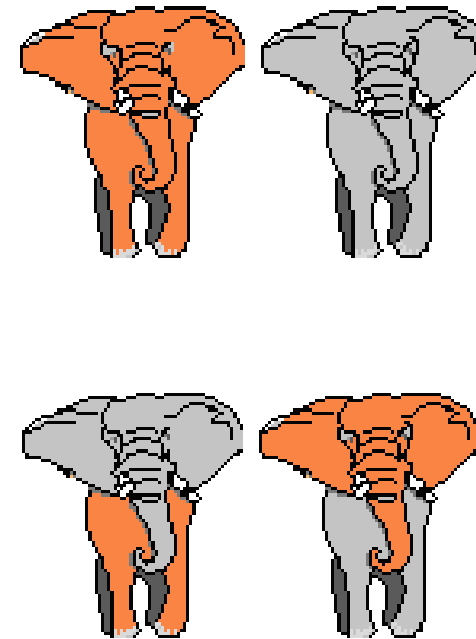
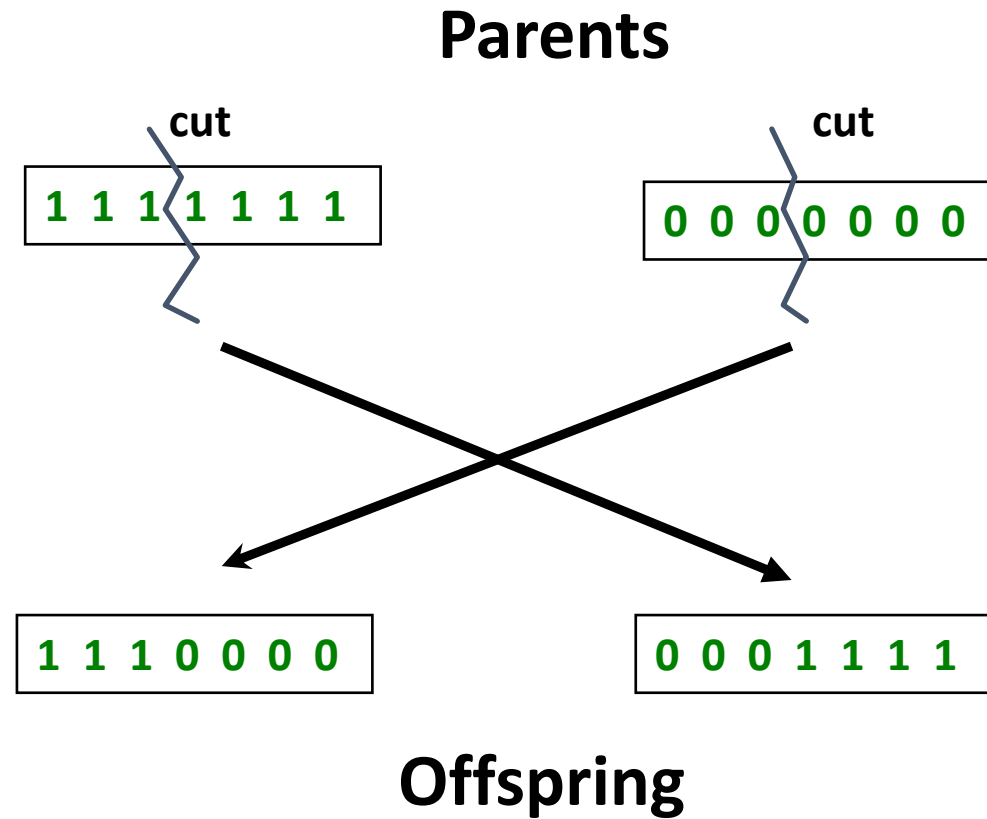
after



EA Hauptkomponenten: Rekombination

- Rolle: Verschmilzt Informationen aus Eltern zu Nachkommen
- Der Prozess, welche Information wie gemischt wird, ist wieder probabilistisch.
- Viel berechneter Nachwuchs kann weniger fit als die Eltern sein oder mit diesen übereinstimmen.
- Die Hoffnung ist, dass gute Eigenschaften von zwei Elternteilen kombiniert werden.
- Definierende Eigenschaft nach [PBB+15]: Die Frequenz, in der genetische Information (Allele) in einer Population auftaucht, wird durch Anwendung von Rekombinationsoperatoren nicht verändert. (Die Information wird nur in neuen Individuen neu gemischt.)

Intuition Rekombination



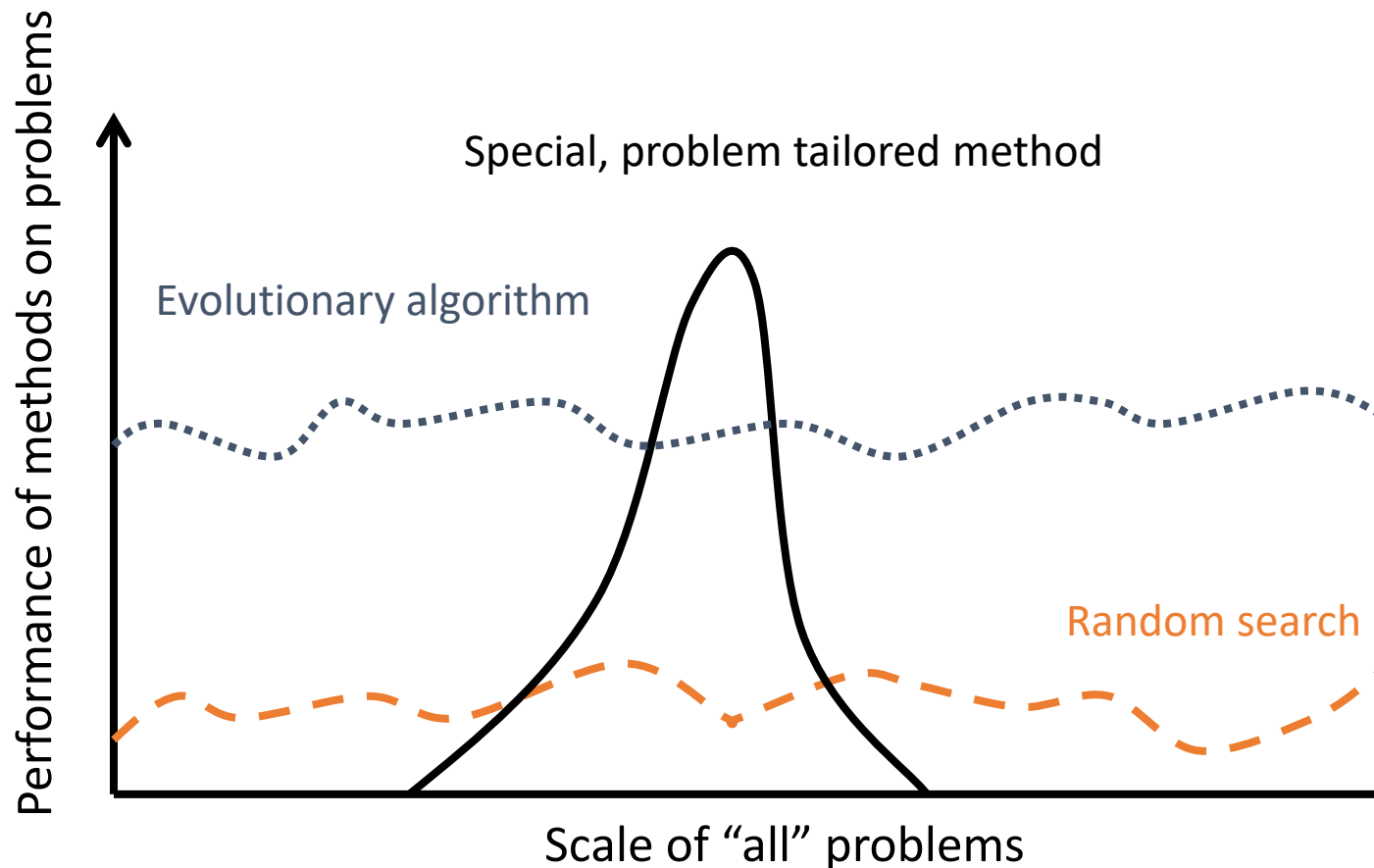
EA Hauptkomponenten: Initialisierung und Termination

- Initialisierung wird normalerweise randomisiert durchgeführt.
 - Es ist (meist) wichtig, dass die initiale Population gut im Suchraum verteilt ist.
 - Bereits existierende Lösungen oder problemspezifische Heuristiken können zur Initialisierung benutzt werden.
- Nach jeder Iteration wird auf die Terminationsbedingung geprüft.
 - Ein als ausreichend bekannter Fitnesswert wurde erreicht.
 - Eine maximal erlaubte Zahl von Iterationen/maximal erlaubte Laufzeit wurde erreicht.
 - Über eine vorher festgelegte Anzahl von Iterationen hat sich die Fitness nicht (wesentlich) verbessert.
 - Die Diversität hat ein Minimum erreicht.

Verschiedene Typen evolutionärer Algorithmen

- Historisch sind verschiedene Arten evolutionärer Algorithmen auf Grundlage verschiedener Datenstrukturen entstanden
 - Binärstrings: Genetische Algorithmen (Genetic Algorithms)
 - Vektoren reeller Zahlen: Evolutionsstrategien (Evolution Strategies)
 - Endliche Automaten: Evolutionäre Programmierung (Evolutionary Programming)
 - Bäume: Genetische Programmierung (Genetic Programming; GP)
- Heute eingesetzte Strategie:
 - Wähle Repräsentation, die zu Problem passt
 - Wähle Variationsoperatoren, die zur Repräsentation passen
- Da die Selektionsoperatoren nur auf der Fitness von Lösungen basieren, sind sie unabhängig einsetzbar von der Wahl der Repräsentation.

Veraltete Vorstellung der Funktion evolutionärer Algorithmen



No free lunch Theorem (1997): Keine metaheuristische Optimierungsmethode ist im Schnitt über alle Optimierungsprobleme besser als randomisierte Suche (in einem gewissen technischen Sinn).

Schlussfolgerung:

- Genaue Bedeutung umstritten
- Vorsichtige Schlussfolgerung klar: Auch evolutionäre Algorithmen profitieren davon, wenn man sie auf ein Problem anpasst.

Ausblick

- Auf welchen Datenstrukturen wollen wir evolutionäre Algorithmen durchführen?
 - Strings über endlichem Alphabet (Verallgemeinerung von Bitstrings)
 - Permutationen
 - Bäume und Graphen
 - Vektoren natürlicher oder reeller Zahlen
- Welche Variationsoperatoren können wir auf den jeweiligen Datenstrukturen benutzen?

Literatur

- [PBB+15] Paixão, T., Badkobeh, G., Barton, N., Çörü, s, D., Dang, D.C., Friedrich, T., Lehre, P.K., Sudholt, D., Sutton, A.M., Trubenová, B.: Toward a unifying framework for evolutionary processes. J. Theoret. Biol. **383**, 28–43 (2015). <https://doi.org/10.1016/j.jtbi.2015.07.011>