UNIKASSEL
VERSITÄT

SOFTWARE ENGINEERING
UNIVERSITÄT KASSEL

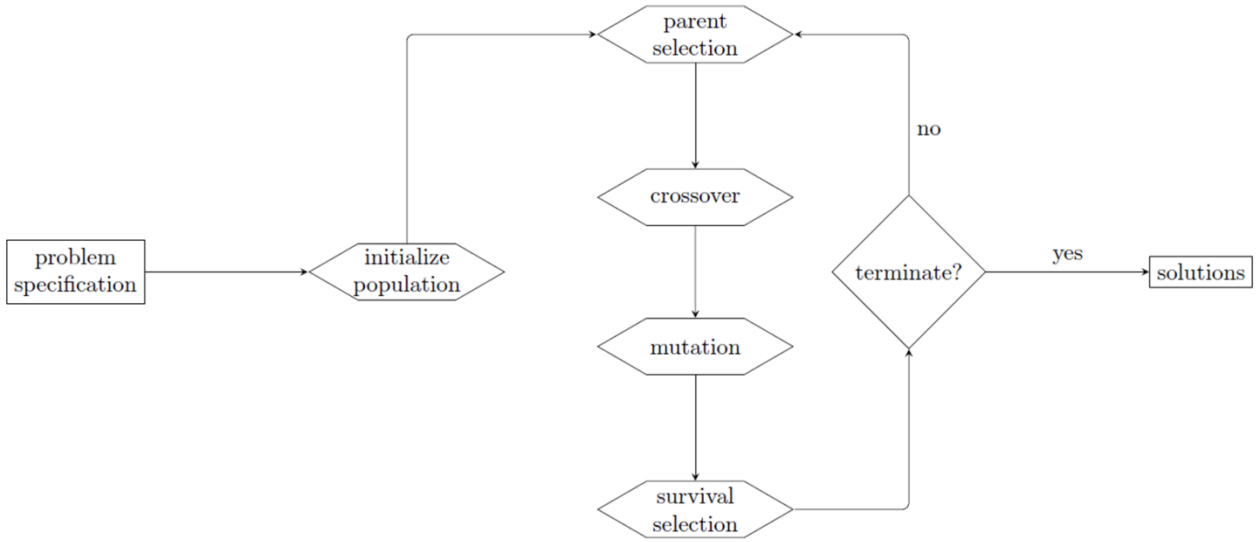
Populationsmanagement

Initialisierung, Termination, Selektion

Jens Kosiol

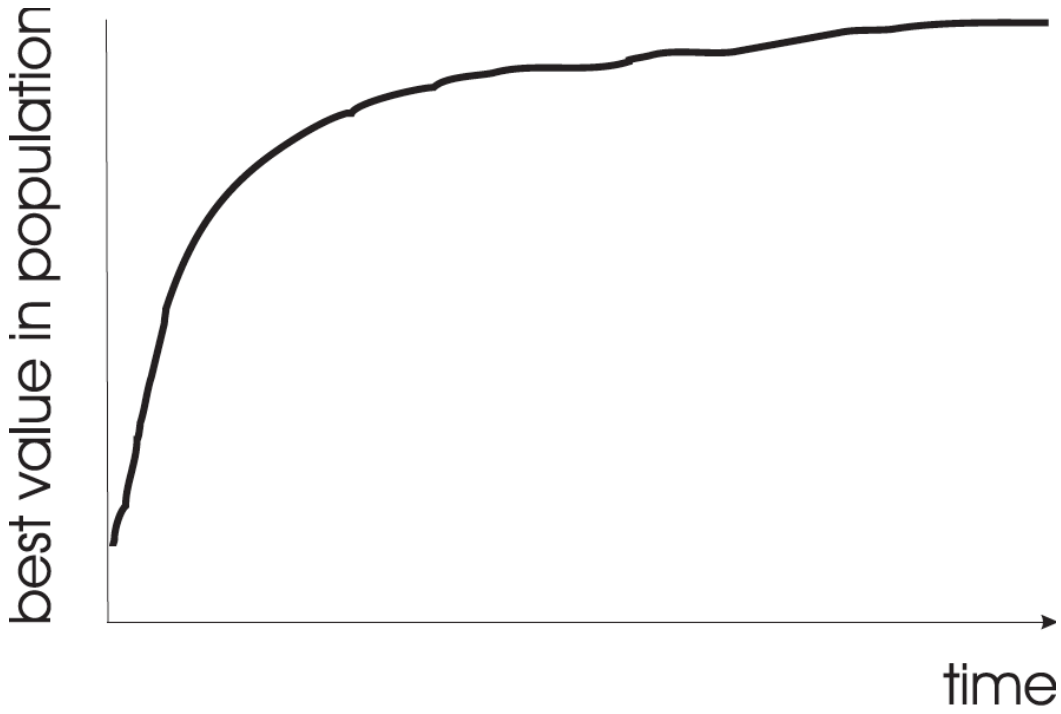
(teilweise basierend auf einem Foliensatz von Eiben/Smith)

Überblick wichtige Konzepte



- Kodierung von Lösungen und Qualitätsmessung
- Initialisierung einer Population
- Variationsoperatoren
- Selektionsoperatoren
- Terminationskriterien

Lohnt sich aufwändige Initialisierung?



Typische Fitnessentwicklung während des Laufs eines evolutionären Algorithmus

- Typischerweise lohnt sich eine aufwändige Initialisierung *nicht*, da ein evolutionärer Algorithmus (typischerweise) sehr schnell relativ große Verbesserungen der Fitness einer zufällig generierten Population erzielt und dann viel Laufzeit mit minimalen Verbesserungen verbringt.
- Bewährte Initialisierung: Nach (moderaten) Änderungen von Anforderungen (der Fitnessfunktion) mit guten Lösungen für die alten Anforderungen starten.
- Standard: Die initiale Population wird zufällig generiert.

Terminationsbedingungen

Für evolutionäre Algorithmen werden verschiedene Terminationsbedingungen benutzt:

- Feste Grenze für Laufzeit/Iterationen/Fitnessauswertungen erreicht
- Konvergenzbedingungen:
  - Keine entscheidende Verbesserung der Qualität der Population mehr über eine bestimmte Anzahl an Iterationen hinweg
  - Keine entscheidenden Änderungen innerhalb der Population mehr
- Optimale Lösung bzw. Lösung ausreichender Qualität gefunden

Termination basierend auf Konvergenz der Allele

- Idee: Die Suche wird gestoppt, wenn die Lösungen sich (auf Genotypebene) nicht weiter ändern.
- Einsetzbar für Bitstrings oder Stringkodierungen über kleinem Alphabet.
- Umsetzung: Zwei Parameter k und $0 < \theta \leq 1$, wobei k kleiner gleich der Länge der Strings der Population ist, werden benötigt. Die Suche wird terminiert, sobald auf mindestens k Positionen der Anteil derjenigen Lösungen, die an der Position das gleiche Allel haben, θ überschreitet.
- Anpassung für Vektoren reeller Zahlen: Zusätzlicher Parameter ϵ; zwei Allele gelten als gleich, falls ihre Distanz ϵ unterschreitet.

Termination basierend auf Konvergenz der Fitness

- Idee: Die Suche wird gestoppt, wenn die Qualität der Lösungen sich (auf Phänotypebene) nicht weiter verbessert.
- Solange Fitness absolut messbar ist, ist diese Methode unabhängig von der gewählten Kodierung einsetzbar.
- Umsetzung: Zwei Parameter T und $0 < \theta$ werden benötigt. Die Suche wird gestoppt, sobald sich die durchschnittliche Fitness der Population über T Iterationen hinweg um weniger als θ % verbessert hat.

Selektion im SGA

Im SGA werden in jeder Iteration aus einer Population P der Größe μ

1. μ Individuen zur Fortpflanzung ausgewählt (Parent selection per fitness proportionate selection);
2. μ Nachkommen berechnet (per Crossover und Mutation);
3. Die Ausgangspopulation P vollständig durch die neu berechneten Nachkommen ersetzt.

Wollen wir immer genau so viele Nachkommen berechnen, wie die Population groß ist? Wollen wir die Ausgangspopulation immer vollständig ersetzen? Welche Alternativen zur roulette wheel selection gibt es?

Nachteile Fitness Proportionate Selection

Fitness proportionate selection hat die folgenden Nachteile:

- (Nur anwendbar, wenn Fitnessfunktion gegeben, die maximiert werden soll und die nur positive Werte annimmt.)
- Beeinflussbar durch Translation der Fitnessfunktion.
- Wenn wenige Elemente im Vergleich eine sehr hohe Fitness haben, übernehmen sie schnell die Population (“premature convergence”).
- Wenn alle Elemente eine sehr ähnliche Fitness haben, gleicht fitness proportionate selection einer gleichverteilt zufälligen Auswahl.

Modifikationen von Fitness Proportionate Selection

Reskalierung der Fitnessfunktion f, z.B. durch Verschiebung um Minimum oder Durchschnitt:

- Ist P_t die Population der t-ten Iteration, so bestimme $f_t = \min\{f(x) \mid x \in P_t\}$ und berechne die modifizierte Fitness einer Lösung x als $f_t'(x) = f(x) - f_t$.
- Ist P_t die Population der t-ten Iteration, so bestimme den Durchschnitt $\overline{f_t}$ und die Standardabweichung σ_t der Fitness der Elemente aus P_t und berechne die modifizierte Fitness einer Lösung x als $f_t'(x) = \max(f(x) - (\overline{f_t} - c \cdot \sigma_t), 0)$; hierbei ist c eine Konstante (häufig mit $c = 2$ verwendet).

Rangbasierte Selektion

Grundidee: Selektionswahrscheinlichkeit einer Lösung ist nicht von ihrer absoluten Fitness abhängig, sondern von ihrer Position in der aktuellen Population.

Vorgehen: Gegeben eine Population P der Größe μ

1. sortiere die Elemente von P gemäß ihrer Fitness und weise ihnen Ränge von 0 (schlechteste Lösung) bis $\mu - 1$ (beste Lösung) zu;
2. berechne die Selektionswahrscheinlichkeit p_x einer Lösung $x \in P$ basierend auf ihrer Position $0 \leq i \leq \mu - 1$ (linear oder exponentiell skaliert);
3. sample die benötigte Anzahl an Lösungen mit einem Auswahlverfahren (z.B. wie bei der roulette wheel selection) auf Grundlage der bestimmten Selektionswahrscheinlichkeiten.

Lineare Skalierung für rangbasierte Selektion

Gegeben sei eine Population P der Größe μ; jeder Lösung $x \in P$ ist ihr Rang $0 \leq i \leq \mu - 1$ zugewiesen. Dann berechne die Selektionswahrscheinlichkeit P_{LRank} einer Lösung mit Rang i als

$$P_{\text{LRank}}(i) = \frac{(2-s)}{\mu} + \frac{2(s-1)}{\mu(\mu-1)}\, i,$$

wobei $1 < s \leq 2$.

Eigenschaften:

- Die Selektionswahrscheinlichkeit nimmt mit dem Rang linear zu.
- Die momentan schlechteste Lösung hat eine Selektionswahrscheinlichkeit von maximal knapp kleiner als $^1/_\mu$ (für s nahe 1); die beste Lösung eine von $^s/_\mu$. Wird μ-mal gesampelt, so erwartet man s-mal die beste Lösung zu sampeln.
- Median der Lösungen (Rang $^{\mu-1}/_2$) hat Selektionswahrscheinlichkeit $^1/_\mu$.
- Der Term $^{2-s}/_\mu$ kann als “Korrekturterm” verstanden werden, damit die Selektionswahrscheinlichkeiten zu 1 summieren.
- Der Parameter s erlaubt es, den Selektionsdruck zu beeinflussen, aber nur sehr eingeschränkt.

Vergleich fitness proportionate und rangbasierte Selektion

Individual	Fitness	Rank	P_{selFP}	P_{selLR} $(s = 2)$	P_{selLR} $(s = 1.5)$
A	1	0	0.1	0	0.167
B	4	1	0.4	0.33	0.33
C	5	2	0.5	0.67	0.5
Sum	10		1.0	1.0	1.0

Exponentielle Skalierung für rangbasierte Selektion

Eine exponentielle Skalierung sorgt für deutlich höheren Selektionsdruck:

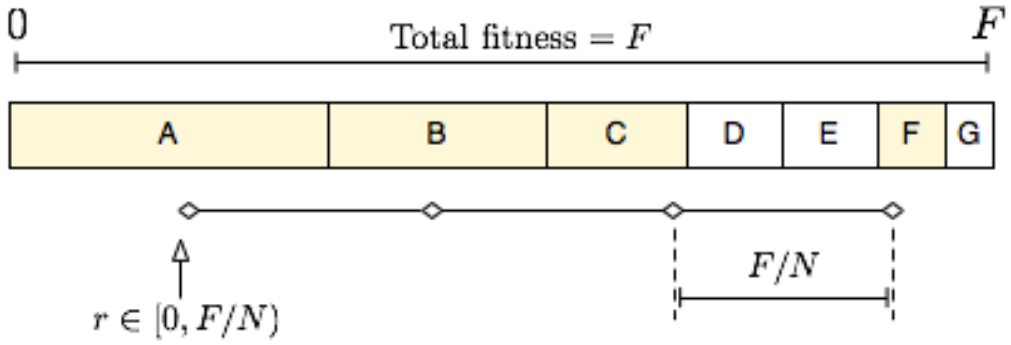
Gegeben sei eine Population P der Größe μ; jeder Lösung $x \in P$ ist ihr Rang $0 \leq i \leq \mu - 1$ zugewiesen. Dann berechne die Selektionswahrscheinlichkeit P_{ERank} einer Lösung mit Rang i als

$$P_{\mathrm{ERank}}(i) = \frac{1 - e^{-i}}{c},$$

wobei c so gewählt wird, dass die Selektionswahrscheinlichkeiten zu 1 summieren.

Stochastic universal sampling statt roulette wheel

- Sampeln per roulette wheel kann Populationen mit einer Verteilung von Elementen erzeugen, die stark von der gewählten Selektionswahrscheinlichkeit abweicht. Insbesondere können Individuen mit hoher Fitness (zu) schnell die Population übernehmen.
- Alternative: Stochastic universal sampling wählt Lösungen gemäß einer Zufallszahl im Abstand eines festes Intervalls.



Quelle: Simon.Hatthon auf Wikimedia, Lizenz: CC 2.5

Vorgehen Stochastic Universal Sampling

Gegeben seien eine Population P der Größe μ, die Selektionswahrscheinlichkeiten p_i der Elemente aus P (z.B. festgelegt proportional zur Fitness oder basierend auf dem Rang), und die kumulativen Wahrscheinlichkeiten $q_i = \sum_{j=1}^{i} p_j$. Zu sampeln seien λ Lösungen.

Vorgehen:

1. Wähle zufällig gleichverteilt einen Wert r aus dem Intervall $(0, {}^{1}/_{\lambda}]$.
2. Bis λ Lösungen gewählt sind, wiederhole das Folgende:
   1. Nehme das Element x_i in die zu berechnende Menge auf, für das gilt $q_{i-1} < r \leq q_i$ (wobei $q_0 = 0$).
   2. Setze $r = r + {}^{1}/_{\lambda}$.

Eigenschaften:

Ein Element x_i mit Selektionswahrscheinlichkeit p_i wird entweder $\lfloor \lambda p_i \rfloor$- oder $\lceil \lambda p_i \rceil$-mal gesampelt.

Tournament Selection

Gegeben sei eine Population P mit μ Elementen und λ Elemente seien zu wählen (mating pool). Tournament Selection ist abhängig von einer Turniergröße $2 \leq k \leq \mu$.

Vorgehen: Bis λ Individuen ausgewählt sind wiederhole:

1. Wähle zufällig gleichverteilt k Elemente aus P aus (mit oder ohne Zurücklegen).
2. Füge das beste dieser k Elemente der Auswahl hinzu.

(Als Variante kann zusätzlich eine Wahrscheinlichkeit p eingeführt werden, mit der das beste der k Elemente gewählt wird.)

Eigenschaften Tournament Selection

- Tournament Selection kann auch verwendet werden, wenn ein absoluter Fitnesswert unbekannt ist oder nicht existiert; es muss nur möglich sein, Lösungen paarweise zu vergleichen.
- Tournament Selection ist unabhängig von Translationen der Fitnessfunktion, ob die Fitnessfunktion zu maximieren oder minimieren ist und ob die Fitnessfunktion negative Werte annimmt.
- Durch die Turniergröße k kann der Selektionsdruck gesteuert werden: Je größer k, desto höher die Wahrscheinlichkeit für fitte Lösungen sich durchzusetzen.
- Werden die k Elemente eines Turniers ohne Zurücklegen ausgewählt, können die schlechtesten $k - 1$ Elemente nie in den mating pool gelangen. Bei Wahl mit Zurücklegen hat jedes Element eine positive Auswahlwahrscheinlichkeit.

Uniform Selection

- Manchmal wird die Parent Selection auch einfach zufällig gleichverteilt durchgeführt: λ-mal wird zufällig gleichverteilt ein Element aus der Population P gezogen.
- Selektionsdruck (Bevorzugung fitter Elemente) wird dann lediglich bei der Survivor Selection angewendet.

Survivor Selection

Situation: Auf einer Population P von μ Elementen wurde parent selection durchgeführt und anschließend λ Nachkommen berechnet (per Mutation und Crossover). Aus diesen $\mu + \lambda$ Elementen sollen nun die μ (meist) Elemente der nächsten Generation bestimmt werden.

- Name: Survivor Selection oder Replacement
- Grundsätzlich eignen sich auch die bereits besprochenen Auswahlstrategien; häufig werden jedoch spezielle Strategien verwendet. Ansätze:
  - Altersbasiert
  - Fitnessbasiert
- Unterscheidung nach „Fortpflanzungsmodellen“:
  - Generational model: In einer Iteration werden $\lambda \geq \mu$ Nachkommen berechnet und die vollständige Population P ersetzt.
  - Steady-state model: In einer Iteration werden $\lambda < \mu$ Nachkommen berechnet und gegen Teile von P ausgetauscht.

Altersbasierte Replacement-Strategien

- Wie in SGA: Jedes Element lebt eine Generation, also bei einer Population von μ Elementen werden auch $\lambda = \mu$ Nachkommen berechnet und ersetzen die komplette Population (generational model).
- Ordne die Population als FIFO-Queue und ersetze immer die ersten $\lambda < \mu$ Elemente (steady-state model).
- Kombination von Fitness und Alter: Ordne jedem Element bei Erzeugung eine Anzahl von Generationen zu, die es existieren soll (abhängig von seiner Fitness im Verhältnis zur Fitness der anderen zu der Zeit existierenden Elemente). Diese Strategie führt dazu, dass die Populationsgröße sich mit den Iterationen ändert (steady-state model).

Fitnessbasierte Replacement-Strategien

Es existiert eine ganze Reihe fitnessbasierter Strategien:

- Ersetzen der schlechtesten Lösungen
- Turnier
- Plus-Selection
- Comma-Selection
- (Elitismus)

Ersetzen der schlechtesten Lösungen (Genitor)

- Strategie: Es werden $\lambda < \mu$ Nachkommen berechnet und die λ schlechtesten Lösungen der Population ersetzt.
- Vorteile: Einfach umzusetzen und häufig schnelle Verbesserung der durchschnittlichen Fitness der Population.
- Nachteile: Fitteste Lösungen übernehmen oft sehr schnell die gesamte Population (premature convergence).
- Anpassungen:
  - Nur bei sehr großen Populationen einsetzen
  - In Verbindung mit dem Verbot von Duplikaten einsetzen

Jeder-gegen-jeden Turniere

Turnierablauf:

- Die Population P wird mit den λ berechneten Nachkommen vereinigt.
- Für jedes Element x aus dieser Menge wird ein Turnier veranstaltet:
  - q Gegner werden zufällig gleichverteilt aus der Menge gezogen.
  - Es wird gezählt, wie viele dieser q Elemente x "schlägt" (fitter ist als sie).
- Die μ Elemente, die die meisten Gegner geschlagen haben, bilden die nächste Population.

Eigenschaften:

- Der Selektionsdruck lässt durch das Ändern des Parameters q steuern: Bei hohem q sinkt die Wahrscheinlichkeit, dass ein schwaches Element viele schwache Gegner hat.
- Stammt aus der Evolutionären Programmierung

Plus-Selektion

- Notation: $(\mu + \lambda)$ selection
- Vorgehen: Aus dem Pool der $\mu + \lambda$ Elemente der Population und Nachkommen werden die μ besten ausgewählt.
- Selektionsdruck wird durch das Verhältnis von μ und λ bestimmt.
- Stammt historisch aus den Evolutionsstrategien (Optimierung auf Vektoren reeller Zahlen) und wird dort häufig mit λ/μ zwischen 5 und 7 eingesetzt.
- Selektion per Turnier nähert sich mit der Turniergröße dem Effekt von Plus-Selektion an.

Komma-Selektion

- Notation: (μ, λ) selection
- Vorgehen: Es werden $\lambda > \mu$ Nachkommen berechnet und aus den λ Nachkommen werden die μ besten als nächste Population gewählt.
- Stammt auch aus den Evolutionsstrategien und wird der Plus-Selektion im Allgemeinen vorgezogen:
  - Durch Verwerfen der Elterngeneration können lokale Optima leichter verlassen werden.
  - Kann besser mit sich verändernden Fitnessfunktionen umgehen.
  - Ist leichter mit der Selbstadaption von Parametern des evolutionären Algorithmus kombinierbar.

Elitismus

Elitismus (elitism) kann verwandte, aber leicht unterschiedliche Konzepte bezeichnen:

- Die beste Lösung wird bewahrt; falls das gewählte Selektionsverfahren das nicht zulässt, wird es so adaptiert, dass das der Fall ist.
- Selektionsverfahren, in denen Eltern mit den Nachkommen darum konkurrieren, Mitglieder der nächsten Population zu bilden (was oft, aber nicht zwangsläufig, dazu führt, dass das beste Element erhalten bleibt).

Multimodale Probleme und Diversität von Lösungen

- Unter einem multimodalen Problem versteht man ein Optimierungsproblem mit mehreren lokalen Optima.
- Herausforderung: Evolutionäre Algorithmen tendieren dazu, um eines der Optima herum zu konvergieren (genetic drift).
  - Optimum, zu dem konvergiert wird, muss nicht das globale Optimum sein.
  - In vielen praktischen Anwendungen kann es von Vorteil sein, verschiedene lokal optimale Lösungen als Ergebnisse zu erhalten.
- Maßnahmen: Diversität der Population sicherstellen
  - Implizit oder explizit
  - Basierend auf Genotyp, Phänotyp oder durch algorithmische Eigenschaften

Ansätze zur Bewahrung von Diversität

Explizite Ansätze

- Fitness sharing
- Crowding

Implizite Ansätze

- Speciation
- Inselmodelle
- Zelluläre Evolutionäre Algorithmen

Explizite Ansätze zur Bewahrung von Diversität

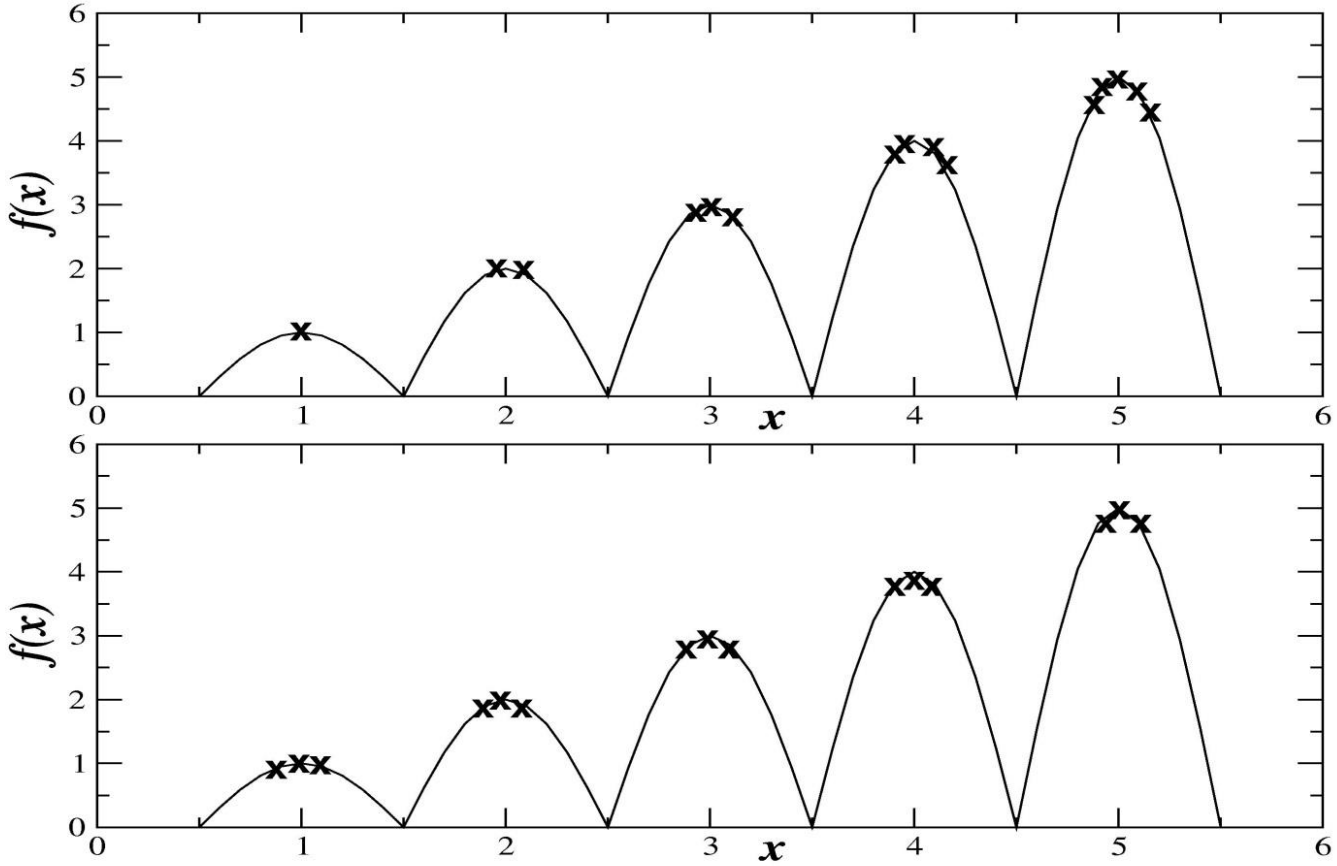
Fitness sharing

- Die Fitness jeder Lösung wird durch die Anzahl der Lösungen in ihrer Nähe gewichtet.
- Nähe kann auf Genotyp- oder Phänotypebene gemessen werden.
- Die Größe der „Nischen“, die die Fitness der Lösungen beeinflussen, wird zum Parameter des Algorithmus.
- In der Tendenz verteilen sich die Lösungen proportional zur Fitness der Nische auf verschiedene Nischen.

Crowding

- Nachkommen konkurrieren mit ihren direkten Eltern ums Überleben: Nach Rekombination und Mutation von zwei Lösungen, konkurrieren die beiden Nachkommen jeweils mit dem Elternteil, dem sie ähnlicher sind.
- Die Metrik, auf der die Distanzmessung basiert, kann auf dem Genotyp- oder dem Phänotypraum definiert sein.
- In der Tendenz verteilen sich die Lösungen gleichmäßig auf verschiedene Nischen.

Effekte fitness sharing und crowding



Erwartete Verteilung von Individuen beim Einsatz von fitness sharing

Erwartete Verteilung von Individuen beim Einsatz von crowding

x-Achse: Fitness; y-Achse: Nummer der Nische

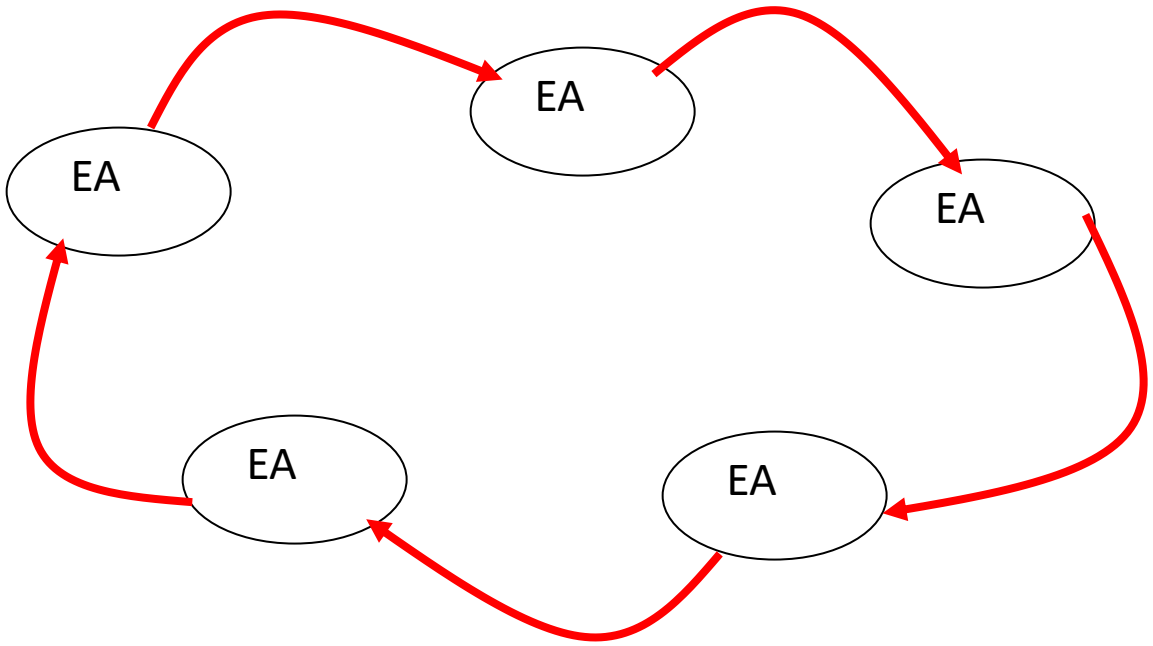
Speciation

Biologische Inspiration hinter speciation: In der Natur können sich nur Individuen, die ähnlich genug sind, miteinander fortpflanzen. Aus Arten können sich neue Arten abspalten, sodass gemeinsame Fortpflanzung nicht mehr möglich ist.

Zwei verbreitete Ansätze:

1. Elemente dürfen nur gemeinsam als Eltern in der Rekombination dienen, wenn sie (auf Genotyp- oder Phänotypebene) ähnlich genug sind.
2. Elemente erhalten Tags und nur Elemente mit dem gleichen/ähnlichen Tag/s dürfen gemeinsam als Eltern in der Rekombination dienen.
   - Tags werden zu Beginn zufällig initialisiert.
   - Tags können auch mutiert werden.
   - Gewöhnlich sind sich nach einer Weile Lösungen mit dem gleichen Tag ähnlich.

Inselmodelle



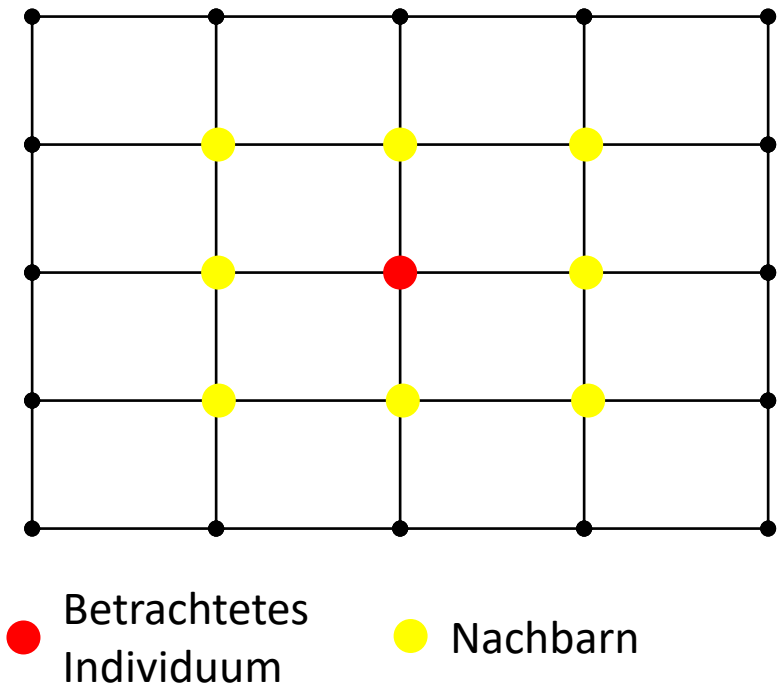
Periodischer Austausch einzelner Lösungen

- Intuition: mehrere Instanzen evolutionärer Algorithmen laufen parallel und periodisch werden einzelne Lösungen ausgetauscht.
- Austausch von Lösungen findet gewöhnlich nach einer festen Anzahl von Iterationen (einer sog. Epoche) statt.

Parameter für das Inselmodell

- Wie viele Instanzen von evolutionären Algorithmen soll man laufen lassen und in welcher Topologie für den Austausch von Individuen soll man sie anordnen (meist Kreis)?
- Wie oft sollen Individuen ausgetauscht werden?
  - Zu häufiger Austausch sorgt für Konvergenz aller Algorithmen zu den gleichen Lösungen, zu seltener verschwendet Ressourcen.
  - Meist wird eine Epoche irgendwo in der Spanne von $25 - 150$ Iterationen festgelegt.
  - Aus Effizienzgründen kann die Berechnung einzelner Algorithmen vorzeitig gestoppt werden, wenn sie konvergieren.
- Welche und wie viele Individuen sollen ausgetauscht werden?
  - Häufig werden $2 - 5$ Individuen ausgetauscht, aber die Wahl hängt auch von der Populationsgröße ab.
  - Sollen die Individuen kopiert oder verschoben werden?
  - Sollen die besten Individuen ausgetauscht werden? Es gibt Forschung, die darauf hindeutet, dass der Austausch zufälliger Individuen besser funktioniert.
- Wie sollen die Parameter der evolutionären Algorithmen eingestellt werden?
  - Für die verschiedenen Inseln können unterschiedliche Algorithmen mit unterschiedlichen Einstellungen gewählt werden.

Zelluläre Evolutionäre Algorithmen



- Die Individuen der Population werden in einer räumlichen Struktur angeordnet (häufig ein torusförmiges Gitter); Ziel: unterschiedliche Teile des Gitters durchsuchen unterschiedliche Teile des Lösungsraums.
- Nachbarschaft steuert Selektion, z.B. Ablauf einer Iteration:
  - Für jedes Element der Population:
    - Wähle per roulette wheel selection einen der Nachbarn aus.
    - Berechne einen Nachkommen per Crossover.
    - Mutiere den Nachkommen.
    - Ersetze das Ausgangselement durch den Nachkommen, falls dieser fitter ist.