Einkriterielle Evolutionäre Algorithmen
28. 2. 2012

Carsten Franke

Vorlesungsplanung

- 21.02.2012: Einkriterielle Evolutionäre Optimierung I (CF)
- 28.02.2012: Einkriterielle Evolutionäre Optimierung II (CF)
- 06.03.2012: Test (1+2), Mehrkriterielle Evolutionäre Optimierung I (CF)
- 13.03.2012: Statistische Lerntheorie I (JP)
- 20.03.2012: Statistische Lerntheorie II (JP)
- 27.03.2012: Test (4+5), Neuronale Netze (JP)
- 10.04.2012: Support Vector Maschinen I (JP)
- 02.05.2012: Mehrkriterielle Evolutionäre Optimierung II (CF)
- 08.05.2012: Genetische Fuzzy Systeme (CF)
- 15.05.2012: Test (3+8+9), Meta-Heuristiken (ACO, PSO) (CF)
- 22.05.2012: Simulated Annealing und andere Suchmethoden (CF)
- 29.05.2012: Support Vector Maschinen II (JP)
- 05.06.2012: Test (6+7+12), Clustering (JP)
- 12.06.2012: Lernen und Spieltheorie (JP)
- 26.06.2012: 1. Termin mündliche Prüfungen
- 03.07.2012: 2. Termin mündliche Prüfungen

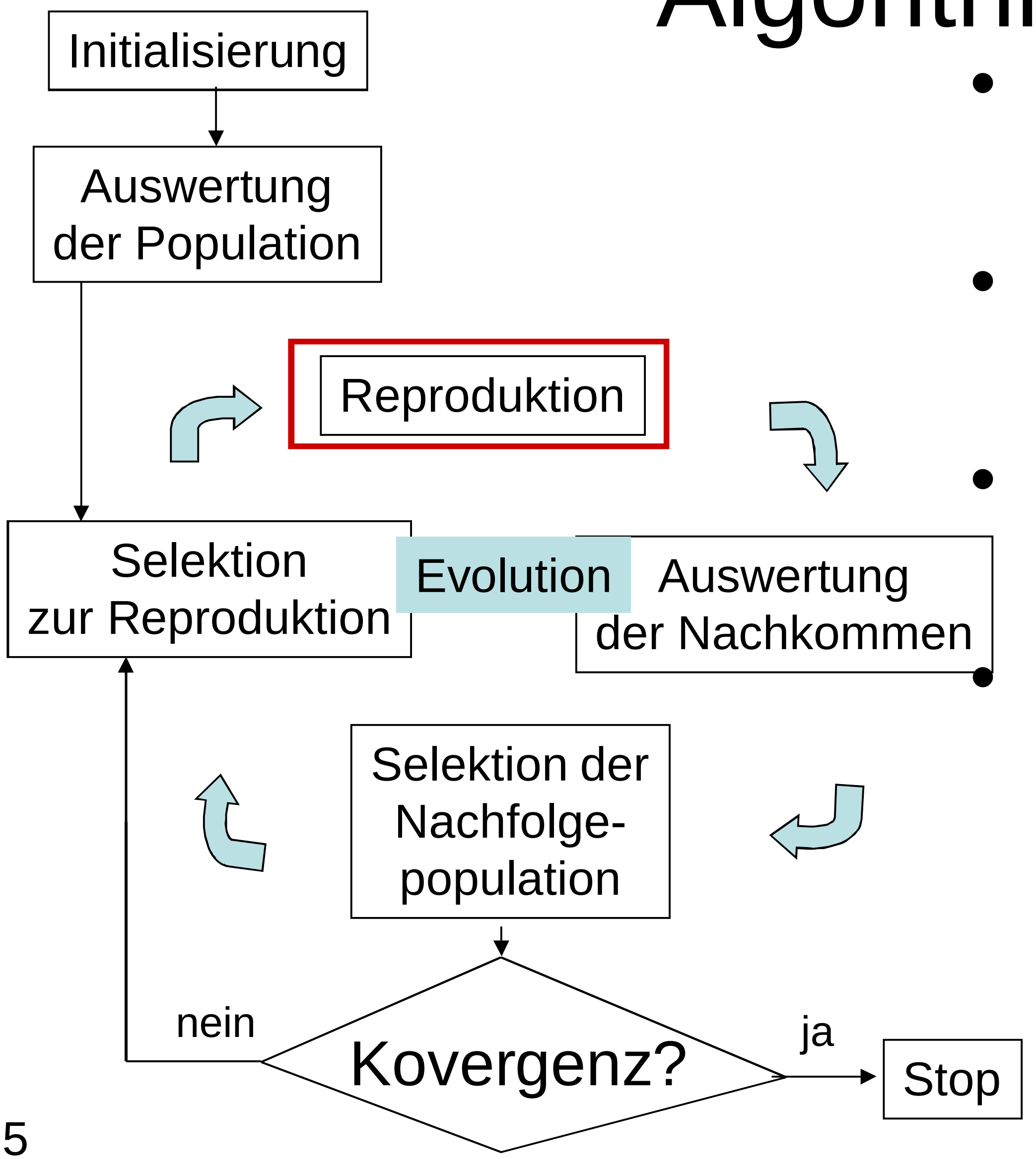
Test

- Schriftlicher Test von ca. 30 min Länge
- Erlaubte Hilfsmittel: Vorlesungsfolien + eigene Mitschriften, Taschenrechner
- **Nicht** erlaubte Hilfsmittel: Laptop, Natel

Lernziele

- Erlernen der restlichen Operatoren für Evolutionäre Algorithmen
- Ausführung eines ersten Genetischen Algorithmus
- Erlernen der Funktionsweise von Evolutionsstrategien
- Erlernen der Selbstadaptionsmechanismen für Evolutionsstrategien

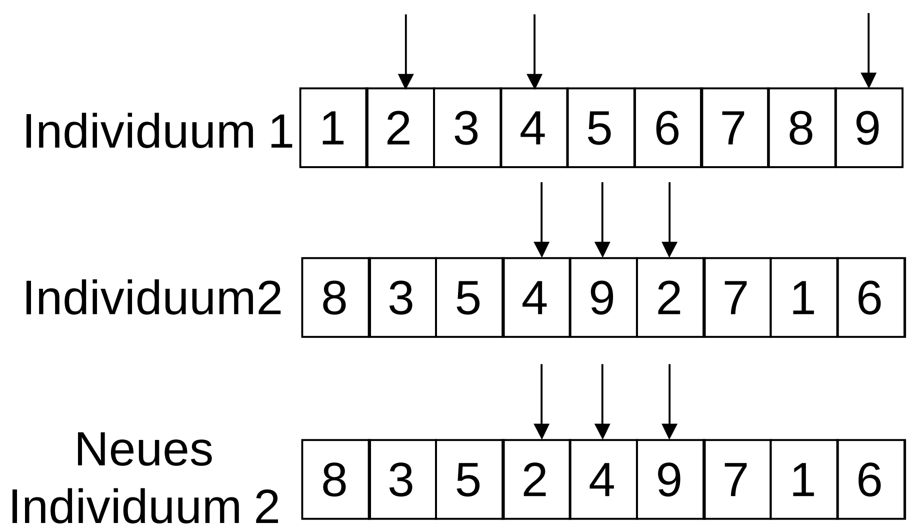
Allgemeiner evolutionärer Algorithmus



- Initialisierung:
  - Kodierung
- Evaluation:
  - Fitnesswert
- Reproduktion:
  - Rekombination, Mutation
- Konvergenz (z.B.):
  - Fixe Generationsanzahl, keine weitere Verbesserung

Rekombination von Sequenzen

1. Zufällige Auswahl von mehreren Sequenzelementen
2. Bestimmung der gleichen Elemente in der 2. Sequenz
3. Umordnung der Elemente in der 2. Sequenz gemäß der Reihenfolge in der 1. Sequenz



Übung Implementierung (war Hausaufgabe)

- Implementieren Sie:
  - 1 Funktion, die an 2 gegebenen binären Strings eine „Single-Point“ Rekombination vornimmt, wobei der Single-Point zufällig gewählt ist

Reproduktion

- Im Allgemeinen werden dabei 2 wesentliche Operatoren genutzt
  - Rekombination
  - Mutation
- Diese sind für die 3 allgemeinen Kodierungen unterschiedlich definiert

Mutation - binär

Wähle mit Wahrscheinlichkeit p_m ein Bit aus und invertiere es.

	0 1 0 0 0 0 1 1 0 0		0 1 0 0 1 0 1 1 0 0
	1 0 1 1 1 1 1 0 1 1		0 0 1 1 1 1 0 0 1 1
$\longrightarrow$	0 1 0 0 0 0 1 0 1 1	$\longrightarrow$	0 1 0 0 0 1 1 0 1 1
	0 1 0 0 0 0 1 0 1 0		0 1 0 1 0 0 1 1 1 0

Übung

- Nehmen Sie an, dass folgender binärer String gegeben ist: 01011010110
- Ein Zufallsgenerator liefert folgende Wert: 0,1; 0,4; 0,9; 0,04; 0,56; 0,33; 0,23; 0,87; 0,56; 0,83; 0,99
- Geben Sie den resultierenden String an für $p_m = 0{,}1$!

Mutation - reelwertig (W'keit:p_m)

Neuer Parametervektor: $\vec{o}'_k = \vec{o}_k + \vec{z}$

isotropic: $\vec{z} = \sigma \cdot (N_1(0,1), \ldots, N_u(0,1))$

non-isotropic: $\vec{z} = (\sigma_1 \cdot N_1(0,1), \ldots, \sigma_u \cdot N_u(0,1))$

$N(0,1)$ Bezeichnet unabhängige Samples der Normalverteilung.

Damit ergibt sich für den neuen Parametervektor folgende Dichtefunktion.

$$p(\vec{o}_i') = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{1}{2}\frac{(\vec{o}_i' - \vec{o}_i)^2}{\sigma^2}}$$

Mutation - reelwertig (W'keit:p_m)

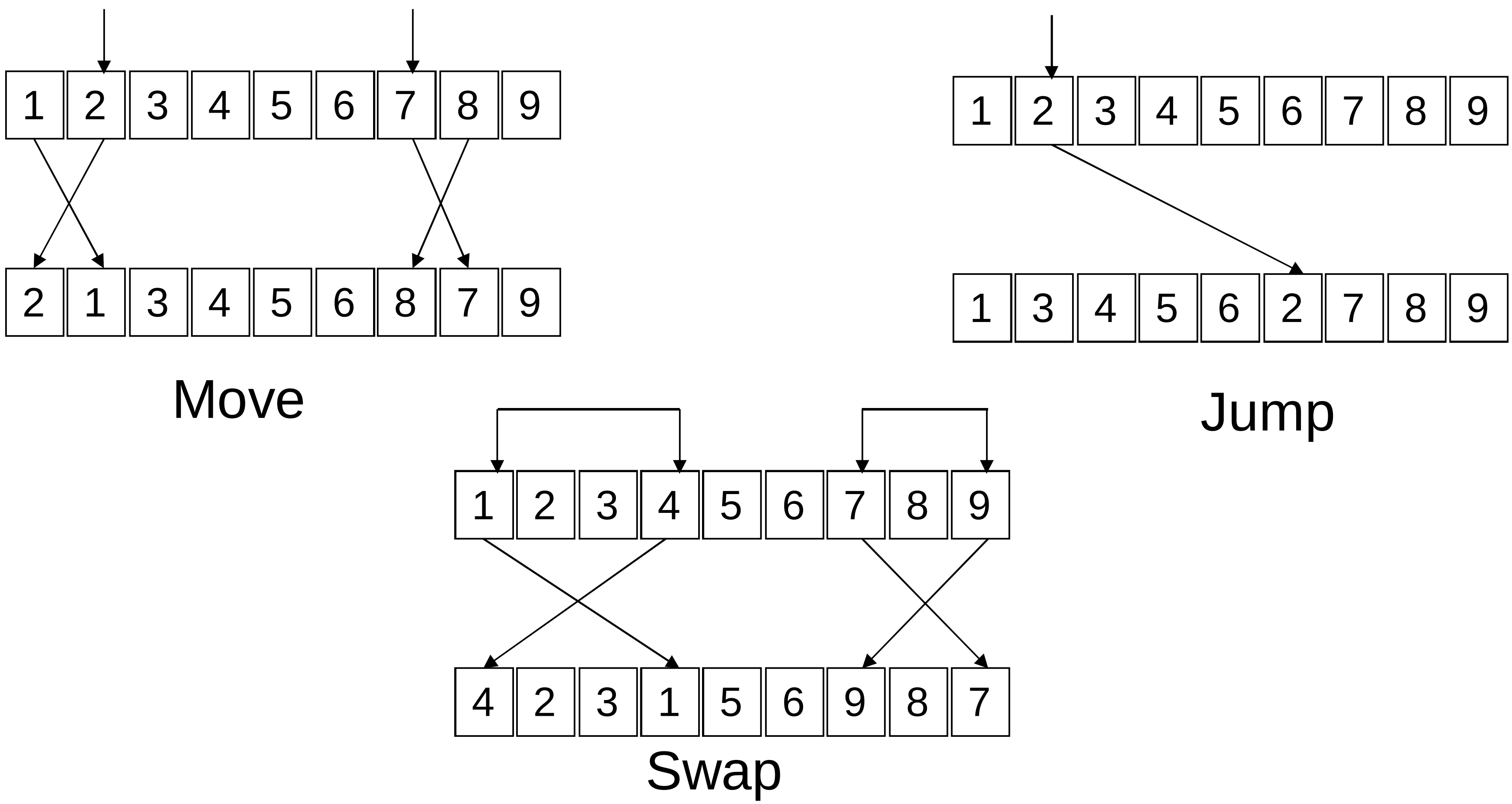
Polynomiale Mutation:

Neuer Parametervektor: $\vec{o}_i' = \vec{o}_i + (o_i^{(O)} - o_i^{(U)}) \cdot \delta_i$

Wähle: $r_i \in [0,1] \,\text{und}\, \eta_m \geq 0$

$$\delta_i = \begin{cases} (2r_i)^{1/(\eta_m+1)} - 1, & \text{wenn } r_i < 0.5 \\ 1 - (2(1-r_i))^{1/(\eta_m+1)}, & \text{sonst} \end{cases}$$

Mutationen bei Sequenzen



Mutationsanzahl und Distanz

- Move, Jump, Swap teilen die folgenden Mutationsanzahl (MZ) und Distanz (D). Beide sind geometrisch verteilt.

$$MZ = \left\lfloor \frac{ld(1-\zeta_1)}{ld(1-p_{MZ})} \right\rfloor, \text{mit } p_{MZ} = 1 - \frac{DMZ}{1+\sqrt{1+DMZ^2}}, \zeta_1 \in [0,1[$$

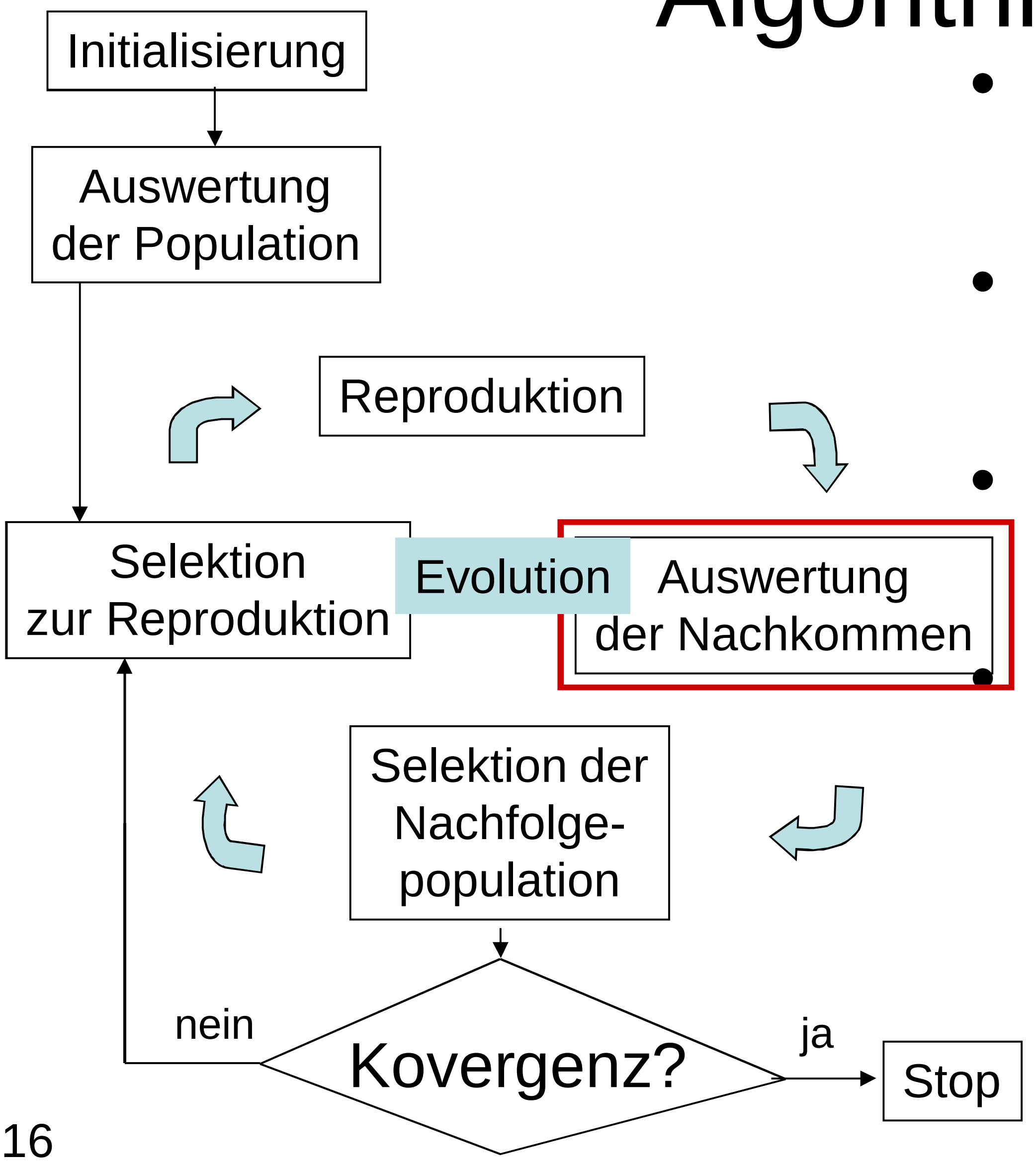
$$D = \left\lfloor \frac{ld(1-\zeta_2)}{ld(1-p_D)} \right\rfloor, \text{mit } p_D = 1 - \frac{DD}{1+\sqrt{1+DD^2}}, \zeta_2 \in [0,1[$$

DMZ: Durchschnittliche Mutationsanzahl
DD: Durchschnittliche Distanz
Richtung der Mutation wird zufällig gewählt.

Übung Implementierung

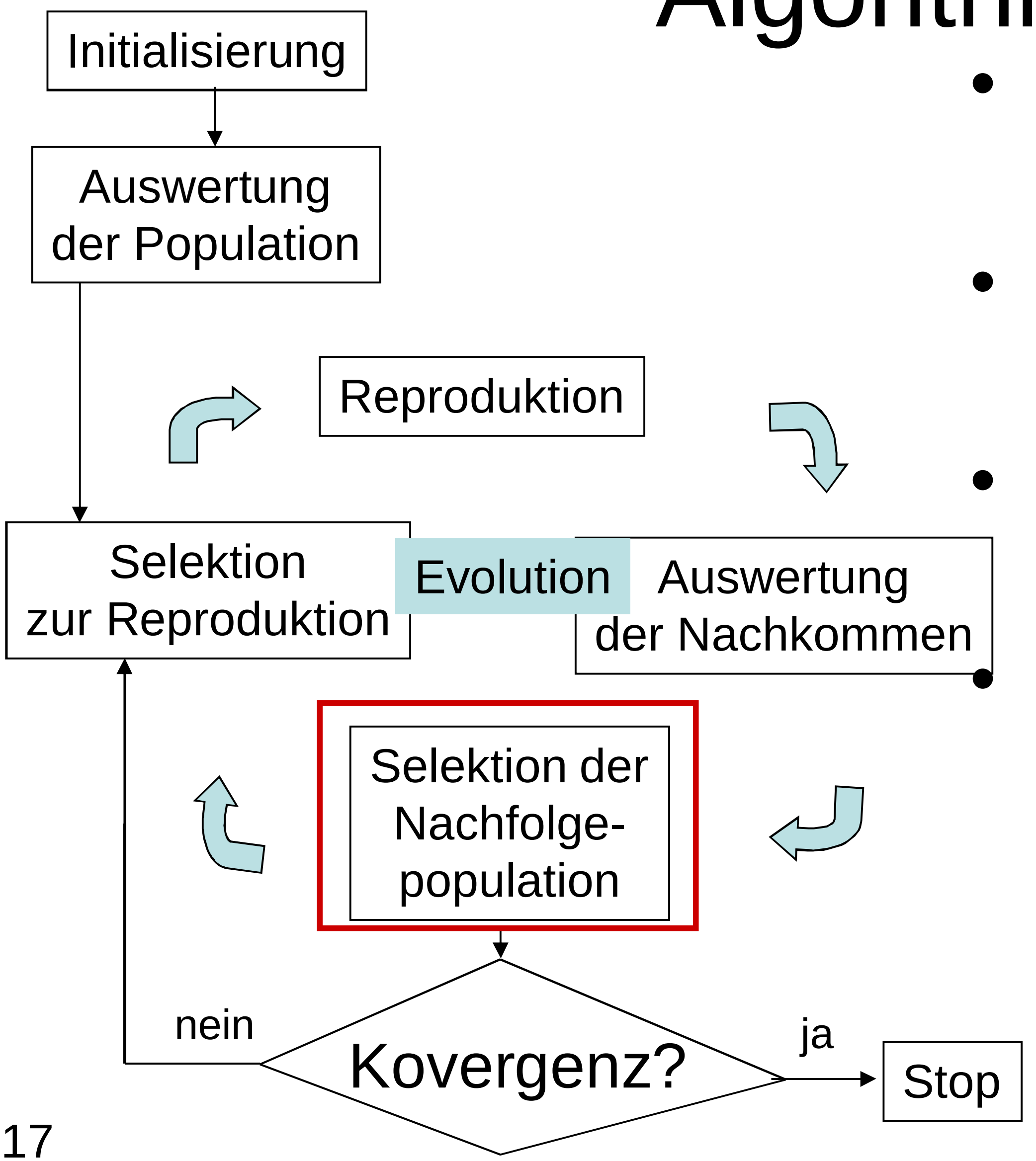
- Implementieren Sie:
  - 1 Funktion, die an 1 gegebenen binären String eine Mutation mit Wahrscheinlichkeit p_m vornimmt.
  - Wenden Sie diese Funktion auf alle lokalen Individuen an.

Allgemeiner evolutionärer Algorithmus



- Initialisierung:
  - Kodierung
- Evaluation:
  - Fitnesswert
- Reproduktion:
  - Rekombination, Mutation
- Konvergenz (z.B.):
  - Fixe Generationsanzahl, keine weitere Verbesserung

Allgemeiner evolutionärer Algorithmus



- Initialisierung:
  - Kodierung
- Evaluation:
  - Fitnesswert
- Reproduktion:
  - Rekombination, Mutation
- Konvergenz (z.B.):
  - Fixe Generationsanzahl, keine weitere Verbesserung

Selektion der Nachfolgegeneration

- Genetisch Algorithmen
  - Es werden nur die Nachkommen für die nächste Generation genutzt. Die Nachkommenanzahl wurde bei der Rekombination beachtet. $(\mu = \lambda)$
- Evolutionsstrategien
  - Verfahrensweise gemäß $(\mu, \kappa, \lambda, \rho)$

Übung

- Beschreiben Sie die Bedeutung der Parameter $(\mu, \kappa, \lambda, \rho)$!

Selektion bei der Evolutionsstrategie

μ – Populationsgrösse der Eltern
λ – Populationsgrösse der Nachkommen $(\lambda \geq \mu)$
κ – maximale Anzahl von Lebenszyklen pro Individuum
ρ – Anzahl Eltern bei der Rekombination

Algorithmus:

if $(\kappa = 1)$ then

$P_{(t+1),\mu} \leftarrow selection(P'_{t,\lambda}, \mu)$ (nicht - elitär)

else if $(\kappa = \infty)$ then

$P_{(t+1),\mu} \leftarrow selection(P'_{t,\lambda} \cup P_{t,\mu}, \mu)$ (elitär)

else

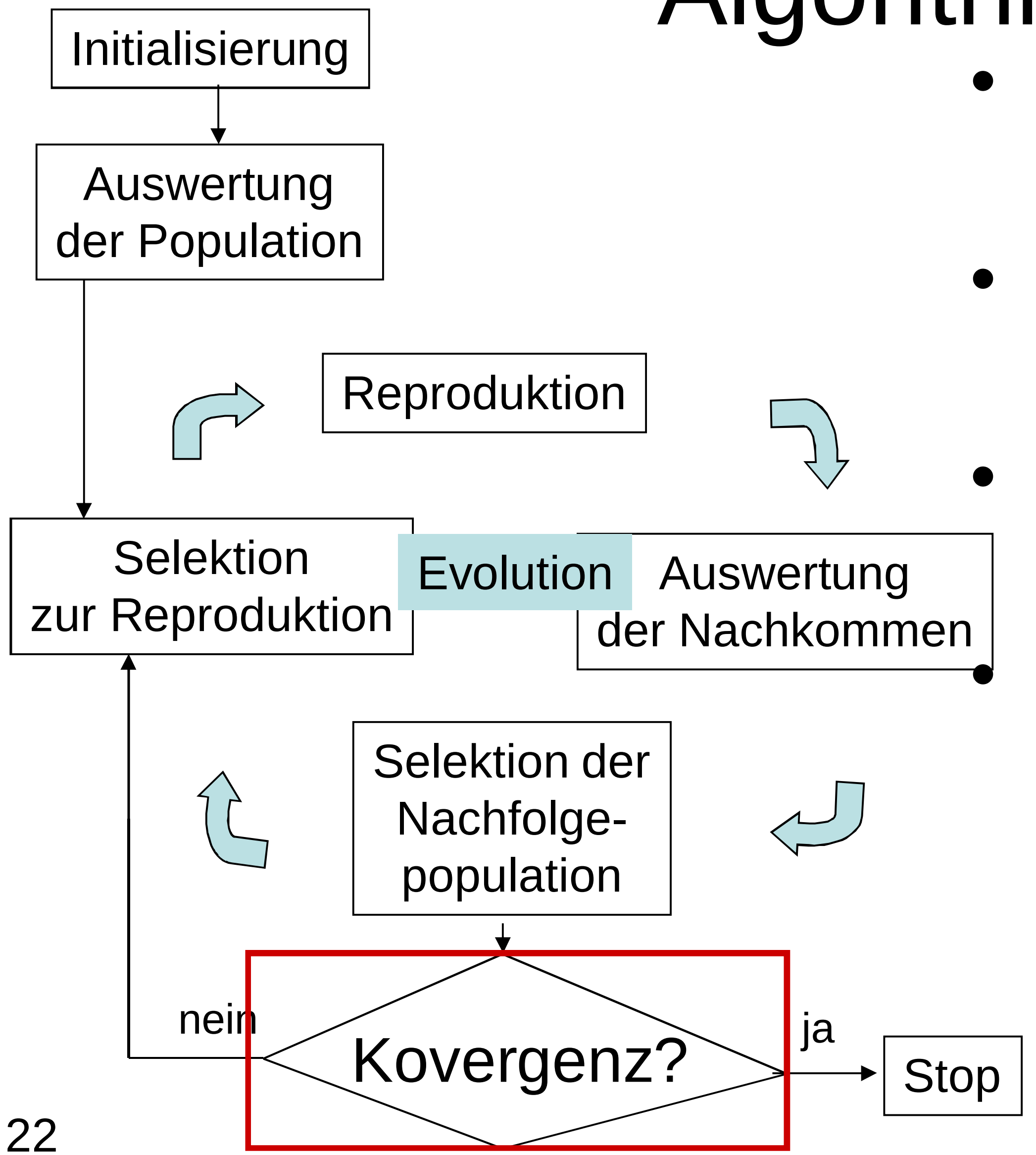
$P_{(t+1),\mu} \leftarrow selection(P'_{t,\lambda} \cup P_{t,\mu}, \mu, \kappa)$ (nicht - elitär)

selection() ist eine zu wählende Selektion, z.B. Turnier-Selektion

Übung

- Welche andere Verfahren könnten für selection() genutzt werden?

Allgemeiner evolutionärer Algorithmus



- Initialisierung:
  - Kodierung
- Evaluation:
  - Fitnesswert
- Reproduktion:
  - Rekombination, Mutation
- Konvergenz (z.B.):
  - Fixe Generationsanzahl, keine weitere Verbesserung

Konvergenz / Stopp-Kriterien

- Eine grosse Vielzahl an Möglichkeiten, gebräuchlich sind z.B.:
  - Fixe Generationenanzahl
  - Unterschreitung eines Deltas bzgl. der Verbesserung der Fitnessfunktion innerhalb einer gewissen Anzahl von Populationen

Übung – Implementiere GA bzgl. des bekannten Zylinderproblems

- Initialisiere 30 Individuen zufällig (binär)
- Werte alle 30 Individuen aus //f(x)
- For i=1 to 100 //Generationen
  - führe eine Rang-basierte Selektion aus (Individuen müssen NB erfüllen)
  - Mutation mit p_m=10% für selektierte Menge
  - keine Rekombination
  - Werte die Individuen aus
  - Speichere bestes Individuum mit f(x) in einer separaten Variable (für jede Generation)
- Zeige den Werteverlauf des jeweils besten Individuums über die 100 Generationen. Was ist die beste gefundene Lösung in (d,h) und welchen Funktionswert hat diese?

Übung

- Wiederholen Sie den vorherigen Algorithmus für p_m={1%; 0,5%; 30%}
- Lassen Sich Aussagen bzgl. p_m treffen? Wie gross sollte dieser Wert sein?

Übung – GA nun mit Rekombination

- Initialisiere 30 Individuen zufällig (binär)
- Werte alle 30 Individuen aus //f(x)
- For i=1 to 100 //Generationen
  - führe eine Rang-basierte Selektion aus (Individuen müssen NB erfüllen)
  - Rekombiniere zufällig 10 Paare (Single-Point) und ersetze die Eltern durch die Nachkommen
  - Mutation mit p_m=1% für resultierende Menge
  - Werte die Individuen aus
  - Speichere bestes Individuum mit f(x) pro Generation
- Zeige den Werteverlauf des besten Individuums über die 100 Generationen. Was ist die beste gefundene Lösung in (d,h) und welchen Funktionswert hat diese?

Rahmenparameter

- Objektparameter (Problemkodierung): $\vec{o}_k$
- Strategieparameter (Mutation/ Rekombination/ Selektion): $\vec{s}_k$
- Fitness: $F(\vec{o}_k)$
- Populationsgröße: $\mu = |P_{t,\mu}|$
- Anzahl Kinder: λ
- Anzahl Eltern bei Rekombination: ρ
- Maximale Anzahl von Generationen pro Individuum: κ

	Genetische Algorithmen	Evolutionsstrategie
Individuum	$a_k = (\vec{o}_k, F(\vec{o}_k))$	$a_k = (\vec{o}_k, \vec{s}_k, F(\vec{o}_k))$
Eltern / Nachkommen	$\mu = \lambda$	$\mu / \lambda = 1/7$
Generationen per Individuum	$\kappa = 1$	$\kappa \in [1, \infty[$
Strategieparameter	fix	selbst-anpassend

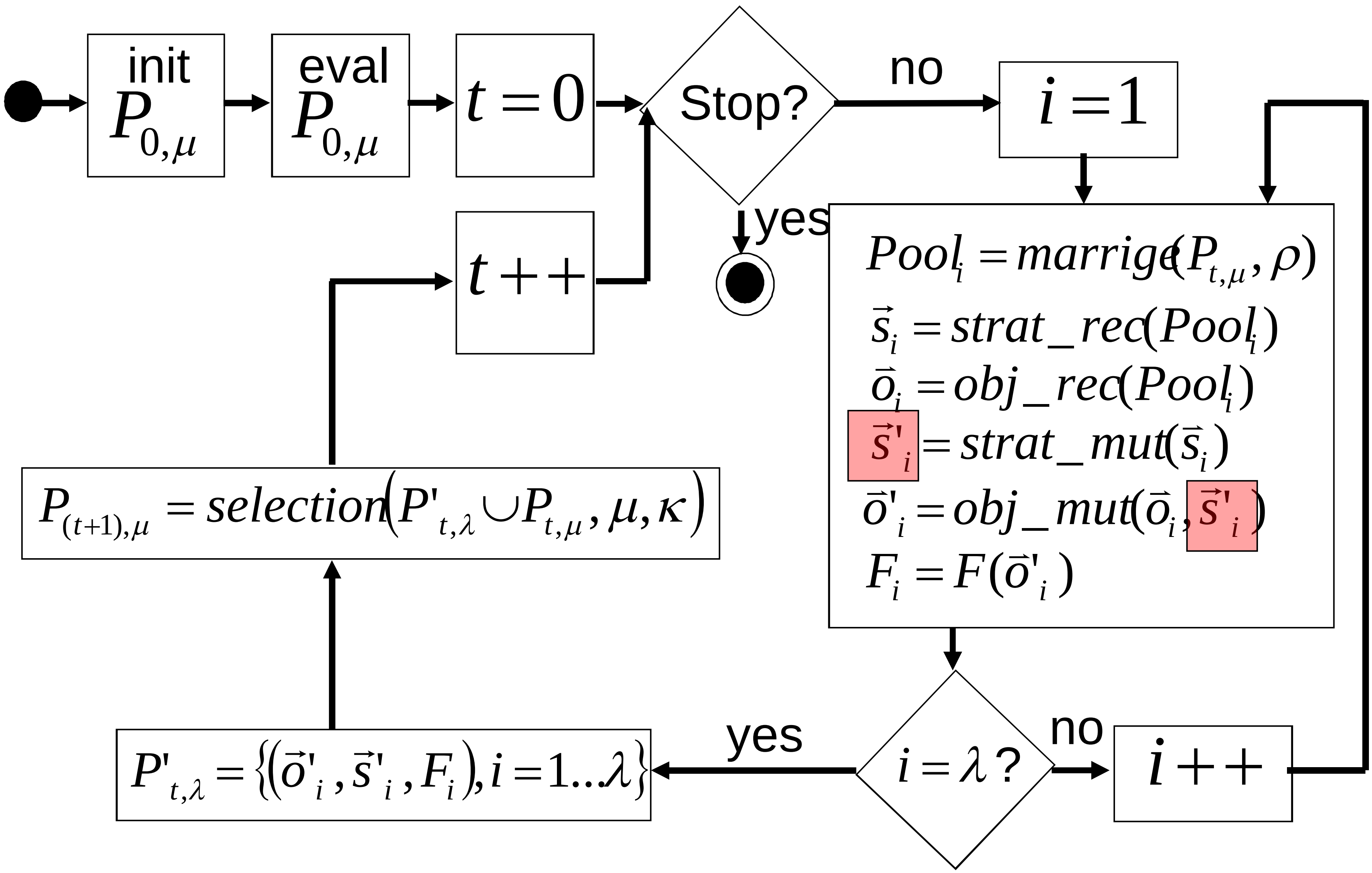
Übung

- Beschreiben Sie <u>eine</u> mögliche "Selektion der Nachfolgepopulation" für einen Genetischen Algorithmus!
- Beschreiben Sie <u>eine</u> mögliche "Selektion der Nachfolgepopulation" für eine Evolutionsstrategie!

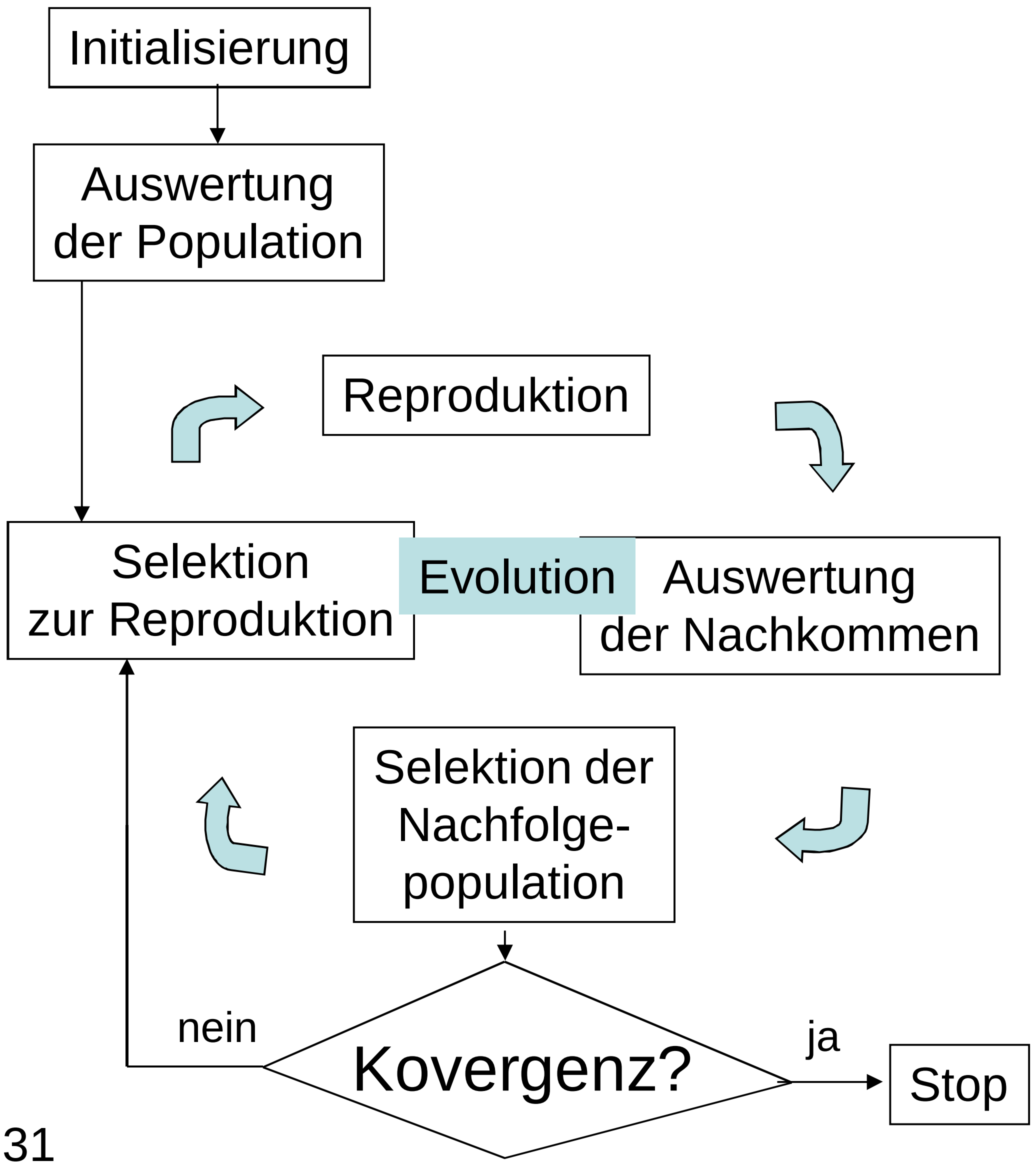
Strategieparameter - Evolutionsstrategien

- Exogene Strategieparameter
  - konstant während der gesamten Optimierung
  - z.B. $(\mu, \kappa, \lambda, \rho)$
- Endogene Strategieparameter
  - passen sich während der Optimierung an
  - Beispiele
    - Mutations- / und Rekombinationswahrscheinlichkeiten
    - Schrittweitenparameter bei einzelnen Operatoren

Ablauf einer $(\mu, \kappa, \lambda, \rho)$ Evolutionsstrategy



Übung



Wo finden Sie entsprechende Elemente in der Beschreibung der Evolutionsstrategie?

Selbstanpassung - Evolution Strategies

- Der Erfolg einer ES hängt stark von der Selbstanpassung der Strategieparameter ab, die auf veränderte Suchräume reagiert.

z.B. Mutation $\vec{o}'_k = \vec{o}_k + \vec{z}$

$$\vec{z} = (\boxed{\sigma_1} \cdot N_1(0,1), \ldots, \boxed{\sigma_u} \cdot N_u(0,1))$$

(u – Anzahl der Objektparameter)

- 1/5 Regel (original nur für $(\mu = 1, \lambda = 1, \kappa = \infty)$ Strategien)

$$P_S = \frac{\#\text{erfolgreiche Mutationen}}{\#\text{aller Mutationen}} \qquad \sigma' = \begin{cases} \sigma / a, \text{ wenn } P_S > 1/5 \\ \sigma \cdot a, \text{ wenn } P_S < 1/5 \\ \sigma, \text{ wenn } P_S = 1/5 \end{cases}$$

mit $0.85 \leq a < 1$

Übung

- Nehmen Sie die Nutzung der 1/5 Regel an. Welche Aussage(n) ist/sind richtig?
  - Wenn 1/5 der letzten Mutationen erfolgreich war, dann wir danach eher lokal gesucht.
  - Wenn 1/5 der letzten Mutationen erfolgreich war, dann wird danach eher global gesucht.
  - Bei vielen erfolglosen Mutationen wird das Suchgebiet vergrössert um bessere Lösungen zu finden.

Selbstanpassung - Evolution Strategies

- Isotropic Mutation $\vec{z} = \boxed{\sigma} \cdot (N_1(0,1), \ldots, N_u(0,1))$

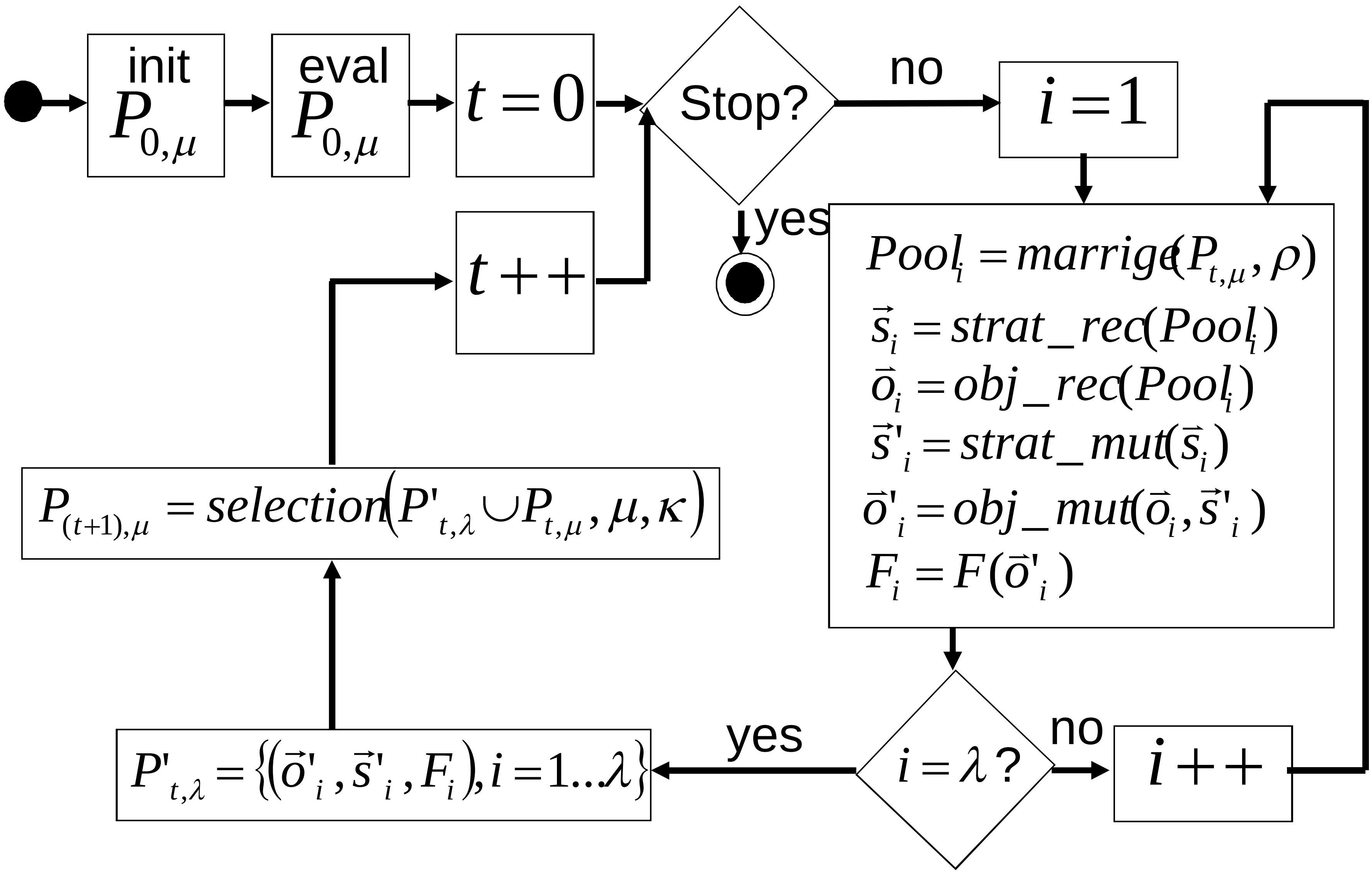
$\vec{\sigma}' = \sigma \cdot e^{\tau \cdot N(0,1)}$ Lernraten (exogene Parameter) $\tau = \frac{1}{\sqrt{u}}$ $\tau = \frac{1}{\sqrt{2 \cdot u}}$ hoch multi-modal

- Non-Isotropic Mutation $\vec{z} = (\boxed{\sigma_1} \cdot N_1(0,1), \ldots, \boxed{\sigma_u} \cdot N_u(0,1))$

$$\vec{\sigma}' = e^{\tau_0 N(0,1)} (\sigma_1 \cdot e^{\tau_1 N_1(0,1)}, \ldots, \sigma_u \cdot e^{\tau_1 N_u(0,1)})$$

Lernraten (exogene Parameter) $\tau_0 = \frac{1}{\sqrt{2 \cdot u}}$ $\tau_1 = \frac{1}{\sqrt{2 \cdot \sqrt{u}}}$

Übung: Wo erfolgt die Selbstanpassung?



Übung

- Implementieren Sie das bekannte Zylinderproblem mittels einer 100- Generationen Evolutionsstrategie

$$\left(\mu, \kappa, \lambda, \rho\right) = (7{,}15{,}49{,}3)$$

- Objektparameter
  - Durschnittliche Rekombination
  - Isotropic Mutation (Start 1%) mit $\tau = \frac{1}{\sqrt{u}}$
- Strategieparameter
  - Diskrete Rekombination
  - Non-Isotropic Mutation (Start 1%) mit $\tau_0 = \frac{1}{\sqrt{2 \cdot u}}$ $\tau_1 = \frac{1}{\sqrt{2 \cdot \sqrt{u}}}$
- Zeigen Sie den Verlauf des besten Individuums pro Generation.

Wann sollte was wie genutzt werden?

- Genetische Algorithmen: für diskrete Suchräume
- Evolutionsstrategien: für kontinuierliche Suchräume
  - Falls ES in diskreten Suchräumen genutzt wird:

$$(\mu, \lambda, \kappa = 1, \rho)$$

- NICHT als reines Black-Box Optimierungstool
  - Problemkodierung ist sehr wichtig
  - Mutation/Rekombination/Selektion Methoden
  - Methodenparameter, z.B. Mutationsstärke, Rekombinationswahrscheinlichkeiten

Aber im Allgemeinen sehr mächtig und flexibel!

Hausaufgabe

- Beenden Sie alle Beispiele, die Sie während der Vorlesung nicht fertig gestellt haben!
- Überprüfung zu Beginn der nächsten Vorlesung.