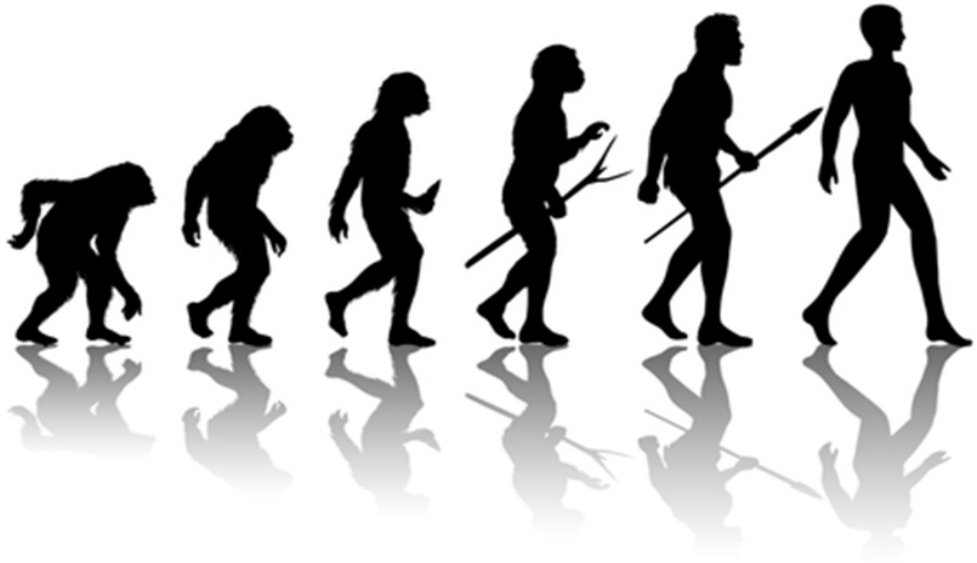
DHBW
Duale Hochschule
Baden-Württemberg
Mannheim

Optimierung industrieller Prozesse



Gedeon, Fabian, Alex, Philip, Daniel

12.04.2017

DHBW Mannheim

Agenda

Betrachtete Algorithmen

Genetischer Algorithmus

- Funktionsweise ist der Natur entlehnt.
- Initialisierung einer Population.
- Bewertung der Individuen mithilfe einer Fitnessfunktion.
- **Operationen:**
  - Selektion (der Eltern)
  - Crossover
  - Mutation
  - Ersetzung
- **Parameter:**
  - Popualtionsgröße & Generationsanzahl
  - Mutationsrate
  - Crossoverrate

Funktionsweise des Algorithmus

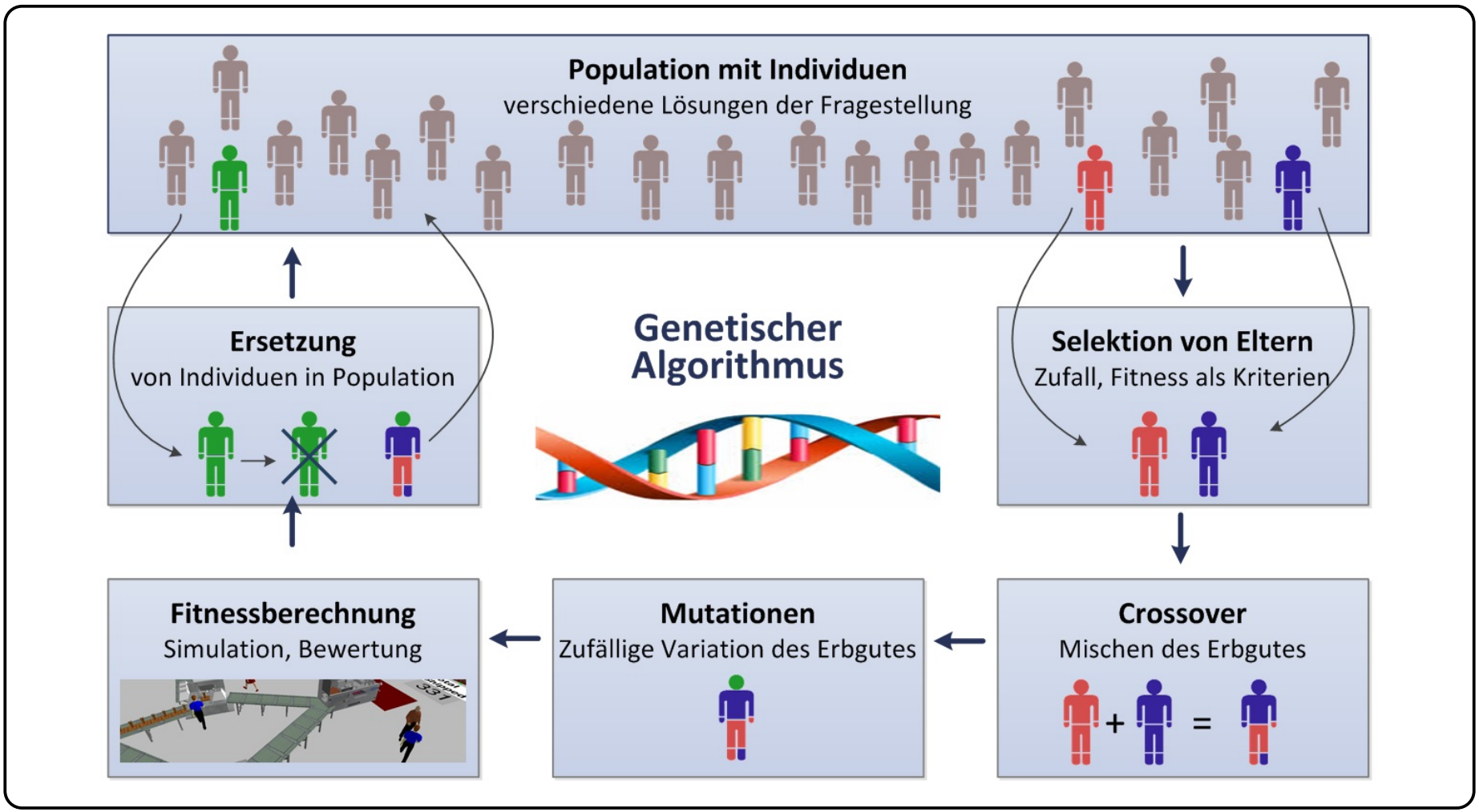


Figure 1: Funktionsweise des genetischen Algorithmus

Realisierung

- Crossover-Strategien:
  - **Gewichteter Durchschnitt:** Berechnung des gewichteten Durchschnitts (basierend auf der Fitness der Eltern) für jedes Gen des Kindes.
  - **Singlepoint Crossover:** Kind erhält zwei unterschiedliche Teile des Vektors der Eltern. Trennung an einem Punkt.
  - **Multipoint Crossover:** Kind erhält beliebig viele unterschiedliche Teile des Vektors der Eltern. Trennung an beliebig vielen Punkten.
- Mutations-Strategien:
  - **Vertauschen zweier Parameter:** Mutationswahrscheinlichkeit für gesamtes Individuum. Vertauschen einer festen Anzahl von Genen.
  - **Mutation mit fester Anzahl:** Mutationswahrscheinlichkeit für gesamtes Individuum. Zufällige Werte für eine feste Anzahl von Genen.
  - **Mutation mit variabler Anzahl:** Mutationswahrscheinlichkeit für jedes Gene einzeln. Zufällige Werte für eine feste Anzahl von Genen.
  - **Gauss Mutation:** Mutationswahrscheinlichkeit für gesamtes Individuum. Addieren eines Gauss- verteilten Zufallswertes.

Anpassungen

- Anpassung der Mutationsrate über e-Funktion (Optimierung für 5000 Iterationen):

$$e^{-x*0.0009-1}$$

- Werte mit „not feasible“ werden mit einer Strafe von 1.000.000 für den Fitnesswert versehen.
- Versuch mit fixer Anzahl an Mutationen konsistente Änderungen und Data Space Exploration zu erreichen.
- Truncation Selection mit 25% der Population

Parameter

- Populationsgröße: 2^{13}
- Generationsanzahl: 5000
- Mutationsrate (variabel): 0.08
- Mutationsrate (fix): 0.3 für 4 Individuen

Particle Swarm Algorithmus

- Vorbild dieses Algorithmus is das biologische Schwarmverhalten (z.B. Vögel oder Bienen)
- Die Population bewegt sich durch den Suchraum und orientiert sich am aktuell besten Individuum
- Bewertung der Individuen mithilfe einer Fitnessfunktion.
- **Operationen:**
  - Initialisierung des Schwarms
  - Fitnessfunktion
  - Anpassen der Geschwindigkeit (Velocity)
  - Schwarmoptimum bestimmen
- **Parameter:**
  - Populationsgröße
  - Lokale Optimierung (C1)
  - Globale Optimierung (C2)

Funktionsweise des Algorithmus

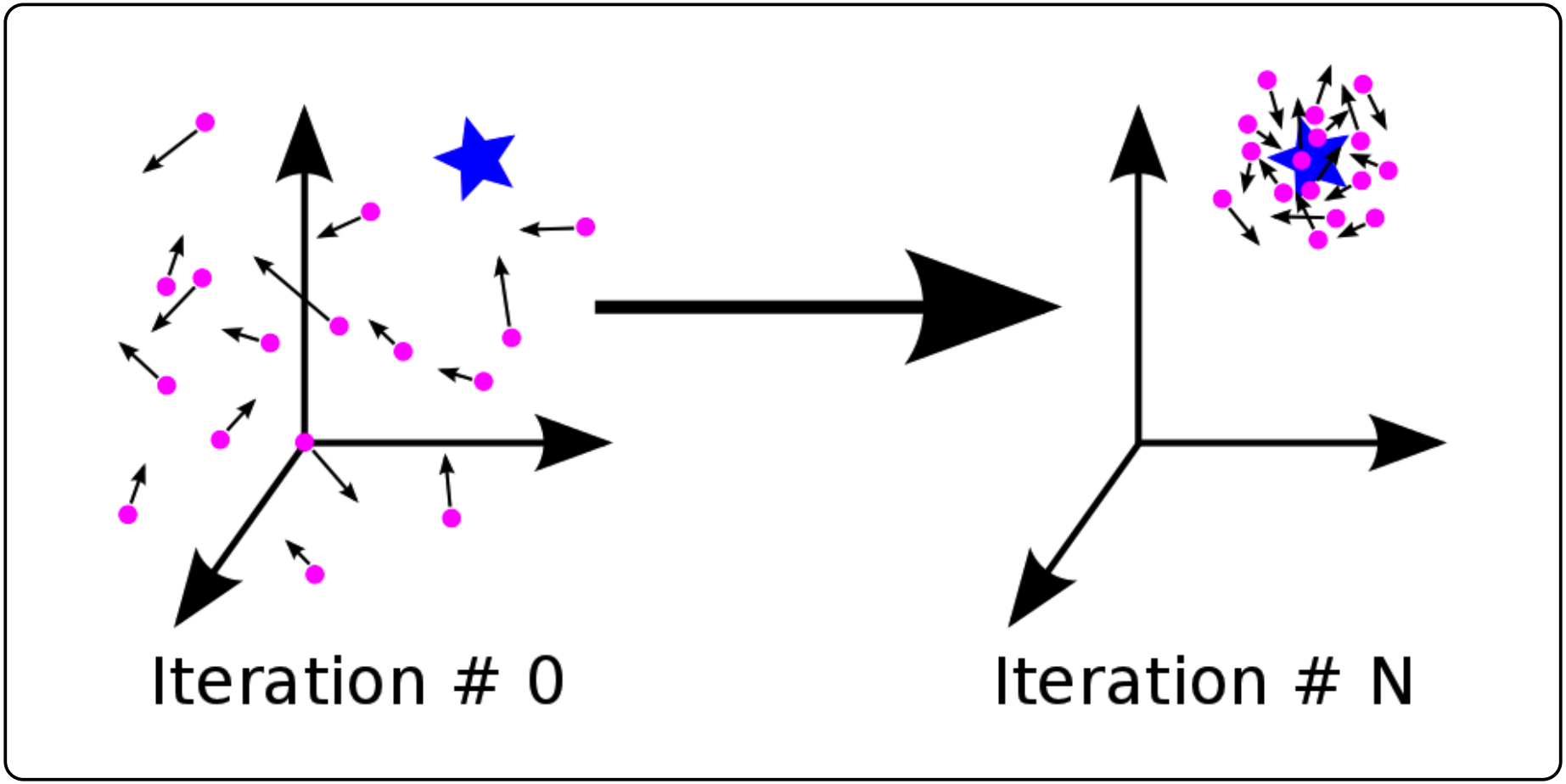


Figure 2: Funktionsweise des Particle Swarm Algorithmus

Vorgehensweise

1: $p_g := \infty$
2: **for all** $x_i \in S$ **do**
3: Initialisiere x_i auf Position p_i in A
4: Setze lokales Optimum auf p_i
5: Wenn $f(p_{b,i}(t)) \leq f(p_g(t))$ setze $p_g := p_{b,i}$
6: Setze v_i auf eine zufällige Geschwindigkeit
7: **end for**
8: **while** *Abbruchbedingung* $\neq$ *true* **do**
9: **for all** $x_i \in S$ **do**
10: Berechne neue Geschwindigkeit für x_i
11: Berechne neue Position für x_i
12: Berechne neues Optimum $p_{b,i}$ und p_g
13: **end for**
14: **end while**
15: In p_g befindet sich das Optimum

Figure 3: Vorgehensweise des Particle Swarm Algorithmus in Pseudo Code

Parameter

- Populationsgröße: 10^4
- Iterationen: 1000 (Abbruchbedingung)
- C1: 1
- C2: 2

Erkenntnisse

Genetischer Algorithmus

- Gute Näherung der Testfunktionen von Stiblinski-Tang und Rastrigin.
- Schlechte Näherung der Testfunktion von Rosenbrock.
- Gute Näherung der unbekannten Funktion
- Ergebnisse: `https://github.com/DaWe1992/OIP/blob/master/Results.md`
- Keine Eignung für alle Optimierungsprobleme

Particle Swarm Algorithmus

- Nachvollziehbares Vorgehen
- Gute Näherung der Testfunktionen von Rosenbrock
- Gute Näherung der unbekannten Funktion
- Ergebnisse: `https://github.com/DaWe1992/OIP/blob/master/Results.md`
- Im Gegensatz zum GA aktualisieren die Partikel ihre Kennzahlen selbstständig und verfügen über einen internen Speicher für das lokale Optimum
- Orientierung im Gegensatz zum GA nur an einem Indikator (Global Best) -> one way information sharing mechanism

Kommunikation

Anbindung an RabbitMQ

- RabbitMQ Client kapselt Funktionalität von Sender und Receiver.
- RabbitMQ Client bietet „Send and Wait“ $\Longrightarrow$ Blockierender Aufruf.
- Wiederverwendbarkeit für sämtlich Algorithmen.

Technische Limitation

- Maximales Senden von 20k Nachrichten pro Sekunde.