



# Multi-Model Estimation of Distribution Algorithm

---

von Stefan Niemczyk

# Inhalt

1. Optimierungsverfahren
  1. Genetische Algorithmen
  2. Estimation of Distribution Algorithmen
2. Multi-Model EDA
  1. Konzept und Idee
  2. Aufbau und Ablauf
3. Evaluierung
  1. Funktionen
  2. Ergebnisse und Auswertung
4. Zusammenfassung und Ausblick

# Inhalt

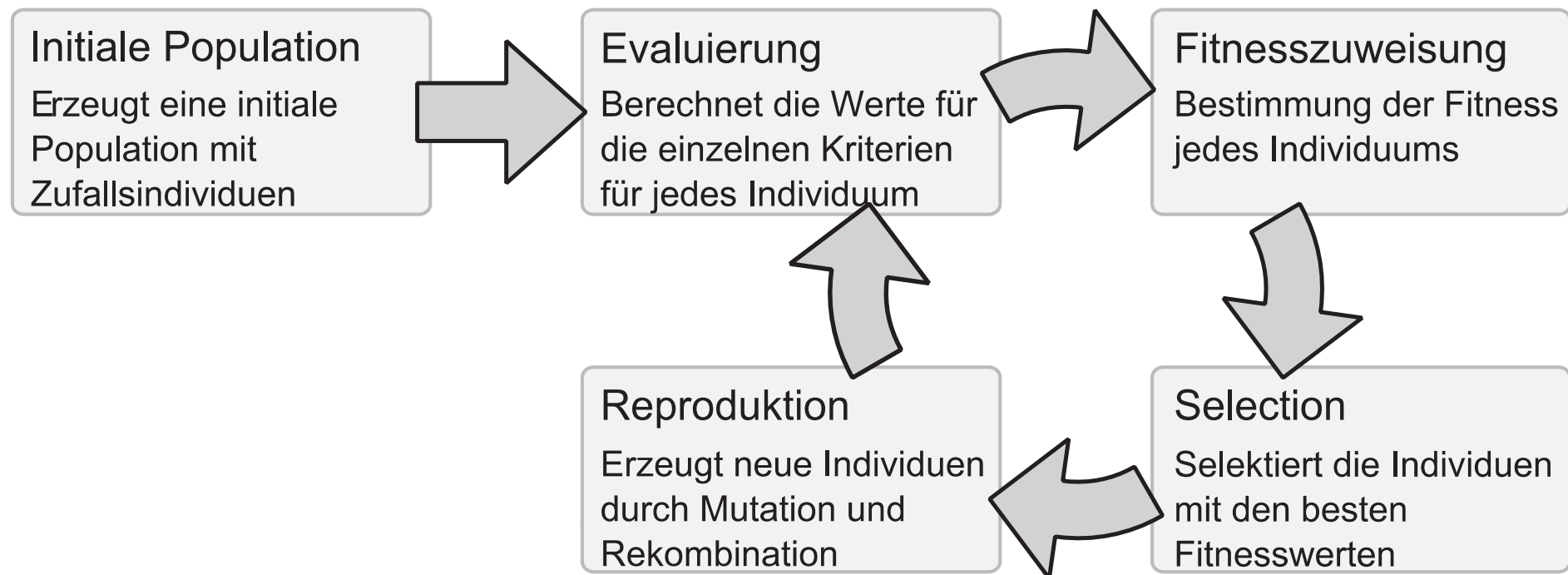
1. Optimierungsverfahren
  1. Genetische Algorithmen
  2. Estimation of Distribution Algorithmen
2. Multi-Model EDA
  1. Konzept und Idee
  2. Aufbau und Ablauf
3. Evaluierung
  1. Funktionen
  2. Ergebnisse und Auswertung
4. Zusammenfassung und Ausblick

# Optimierungsverfahren

- ◆ Evtl. weglassen
- ◆ Ansonsten was sind Optimierungsverfahren
- ◆ Wo werden sie eingesetzt

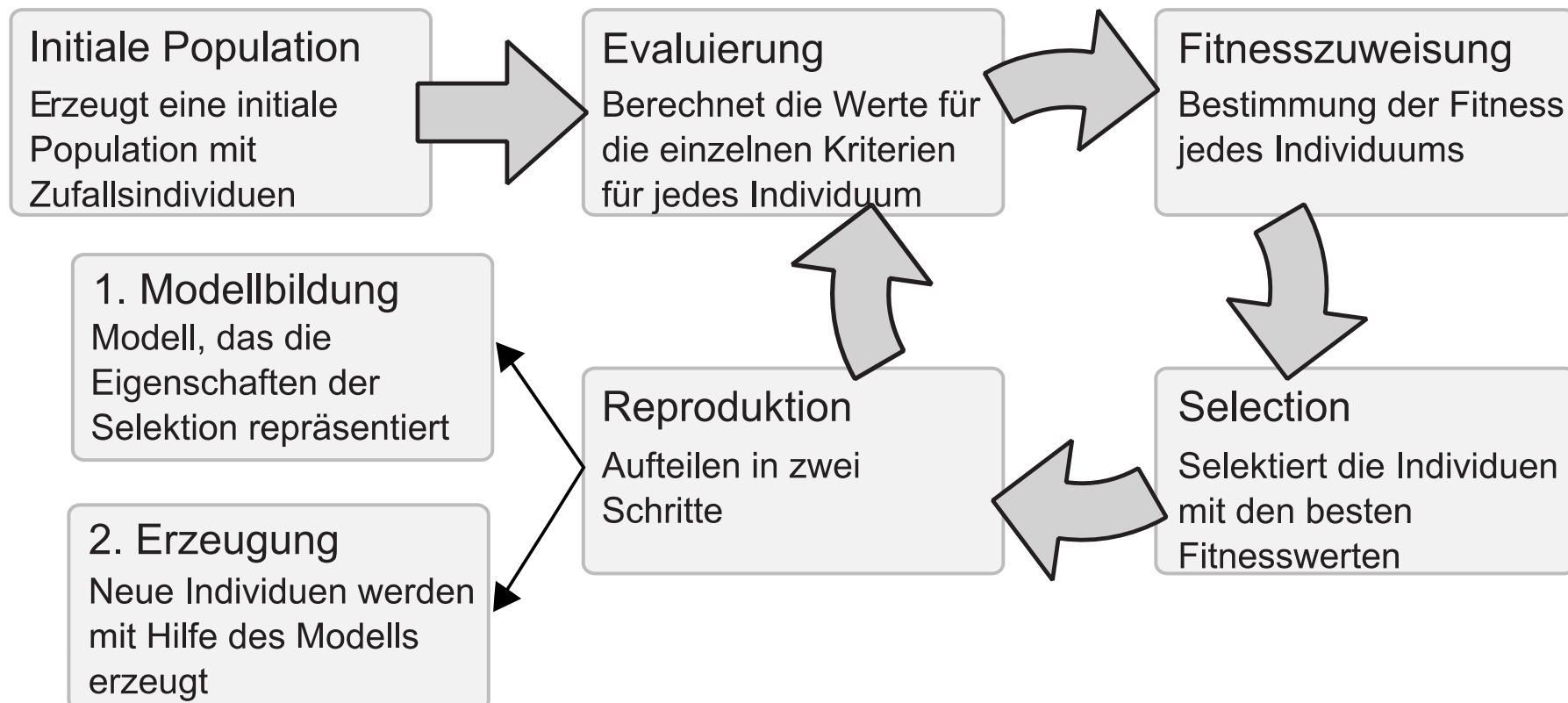
# Genetische Algorithmen

- ◆ Orientieren sich an den Theorien von Charles Darwin
- ◆ „*Survival of the fittest*“
- ◆ Lösungskandidaten entsprechen lebendigen Organismen
- ◆ Paarungsrituale (Selektion) vor der Reproduktion



## Estimation of Distribution Algorithmus

- ◆ Unterschied zu Genetischem Algorithmus:  
Es wird versucht zu lernen, wie eine ideale Lösung aussehen sollte
- ◆ Modell der selektierten Individuen wird erstellt



# Inhalt

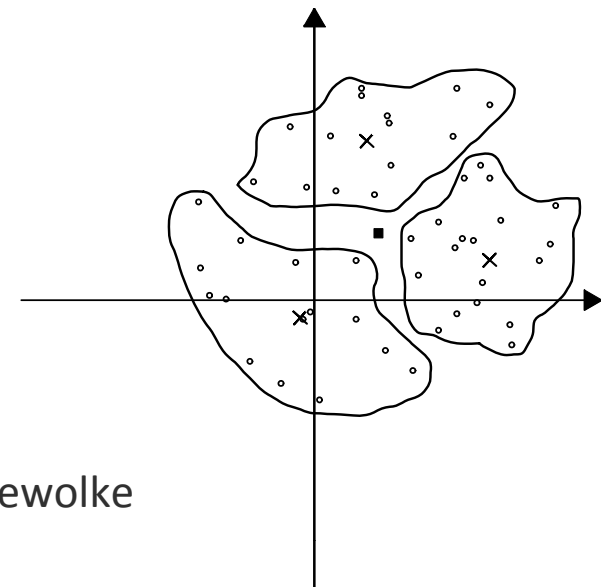
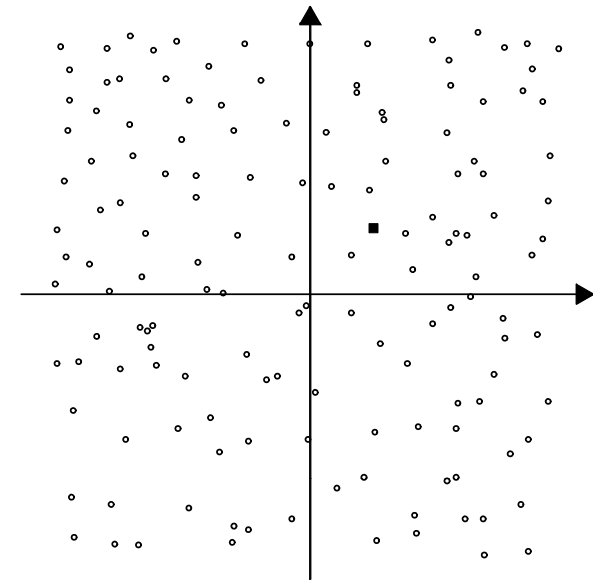
1. Optimierungsverfahren
  1. Genetische Algorithmen
  2. Estimation of Distribution Algorithmen
2. Multi-Model EDA
  1. Konzept und Idee
  2. Aufbau und Ablauf
3. Evaluierung
  1. Funktionen
  2. Ergebnisse und Auswertung
4. Zusammenfassung und Ausblick

## Konzept und Ideen

- ◆ Erweiterung eines normalen EDAs
  - ◆ Clustern ähnlicher Individuen
  - ◆ Ein Modell pro Cluster
  - ◆ Kreuzen der Modelle möglich
  
- ◆ Ziel:
  - ◆ Besseres umgehen mit hoher Neutralität
  - ◆ Verhindern vorzeitiger Konvergenz
  - ◆ Bessere Anpassungsfähigkeit durch Kreuzung von Modellen
  
- ◆ Implementierung für numerische Optimierungsprobleme

# Aufbau und Ablauf 1

1. Initiale Population wird erstellt
2. Bewerten
3. Selektieren
4. Clustern
  - ◆ Ähnliche Individuen werden zusammengefasst
  - ◆  $c$  Cluster werden gebildet
  - ◆ kMeans als Verfahren
5. Modellbildung
  - ◆ Für jedes Cluster eine mathematische Repräsentation
    - ◆ Zentroid der Wolke
    - ◆ Kovarianzmatrix für genauere Repräsentation der Punktwolke



## Aufbau und Ablauf 2

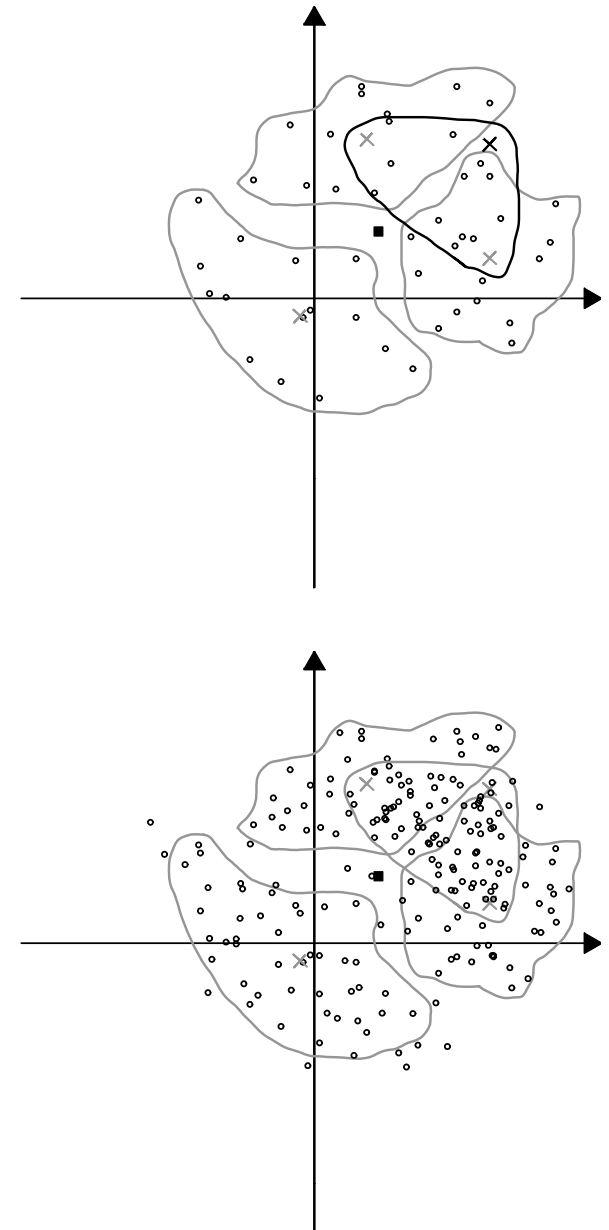
### 6. Kreuzen

- ◆ Es werden weitere Modelle durch Kreuzen gebildet
- ◆ Zwei geclusterte Modelle werden rekombiniert
  - ◆ Zentroid: zufällige Kombination der Elternzentroide
  - ◆ Konvarianzmatrix: Gebildet aus den drei Zentroiden

### 7. Erzeugen der nächsten Generation

- ◆ Jedes der n Modell erzeugt m neue Punkte
  1. Normalverteiler Zufallswert für jede Dimension d
  2. Skalierung
  3. Rotation mit Hilfe der Konvarianzmatrix
  4. Verschieben durch Addition des Zentroids
- ◆ Insgesamt  $n * m$  neue Punkte

### ◆ Livevorführung???



# Inhalt

1. Optimierungsverfahren
  1. Genetische Algorithmen
  2. Estimation of Distribution Algorithmen
2. Multi-Model EDA
  1. Konzept und Idee
  2. Aufbau und Ablauf
3. Evaluierung
  1. Funktionen
  2. Ergebnisse und Auswertung
4. Zusammenfassung und Ausblick

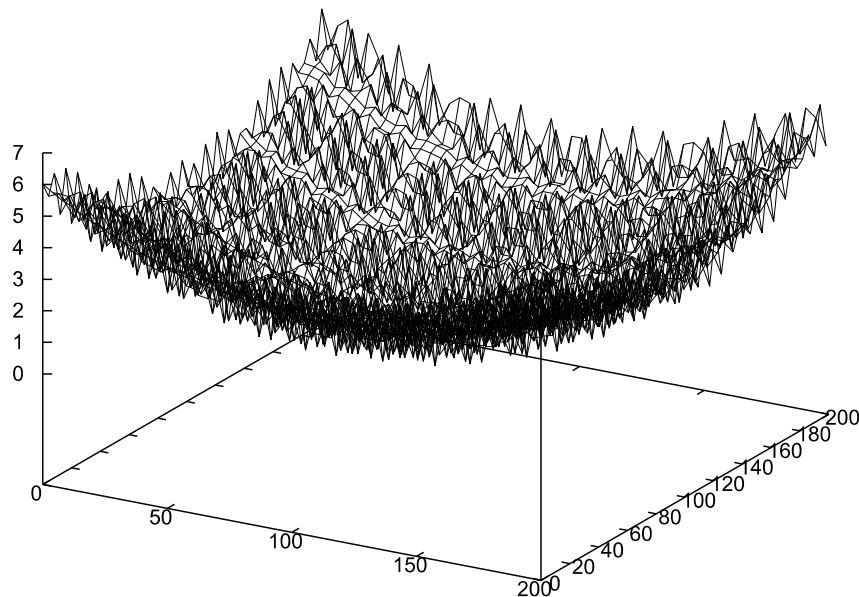
# Evaluierung

- ◆ Fünf mathematische Funktionen als Evaluierungsprobleme
- ◆ Vergleich:
  - ◆ mit normalen EDA
  - ◆ Ergebnissen von Bosman und Thierens
- ◆ Aufbau der Tests:
  - ◆ Für  $d = 5$  jeweils 30 Durchläufe,  $d = 25$  jeweils 10 Durchläufe
  - ◆ Generationsgröße: 1000 Individuen
  - ◆ Abbruch nach 10.000 Generationen oder 10 Minuten
  - ◆ Anzahl Modelle von 1 bis 10
  - ◆ Anzahl Cluster von 1 bis 10
  - ◆ Anzahl Selektion: 200, 300, 400, 500, 600

# Evaluierungsfunktionen 1

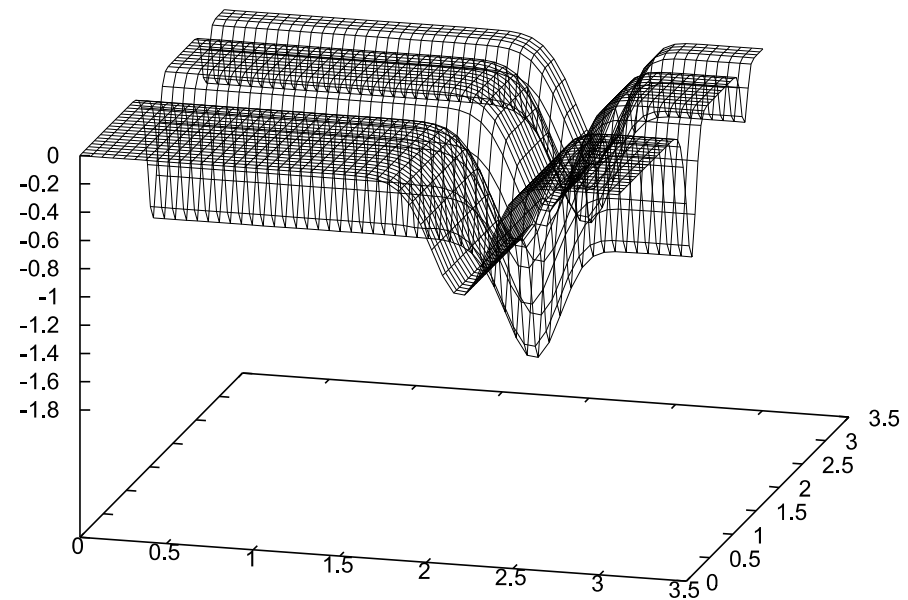
## Griewank Funktion

$$\frac{1}{4000} \sum_{i=0}^{n-1} (x_i - 100)^2 - \prod_{i=0}^{n-1} \cos\left(\frac{x_i - 100}{\sqrt{i+1}}\right) + 1$$



## Michalewicz Funktion

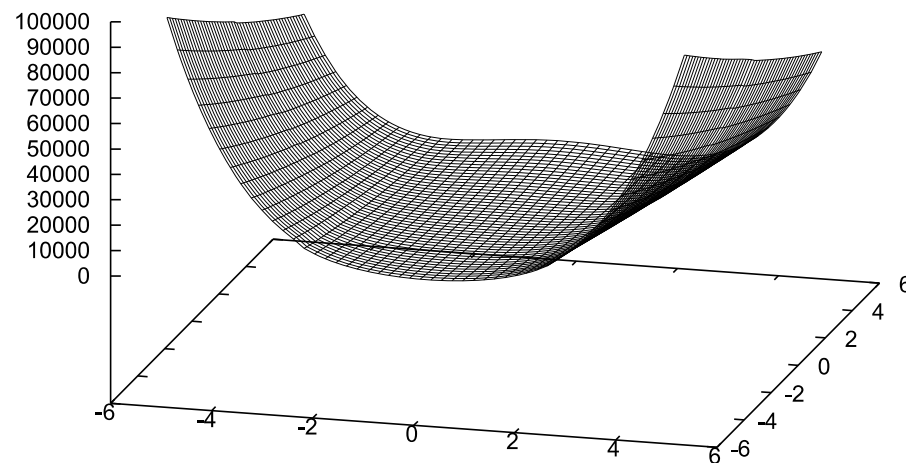
$$-\sum_{i=0}^{n-1} \sin(x_i) \sin^{20}\left(\frac{(i+1)x_i^2}{\pi}\right)$$



# Evaluierungsfunktionen 2

## Rosenbrock Funktion

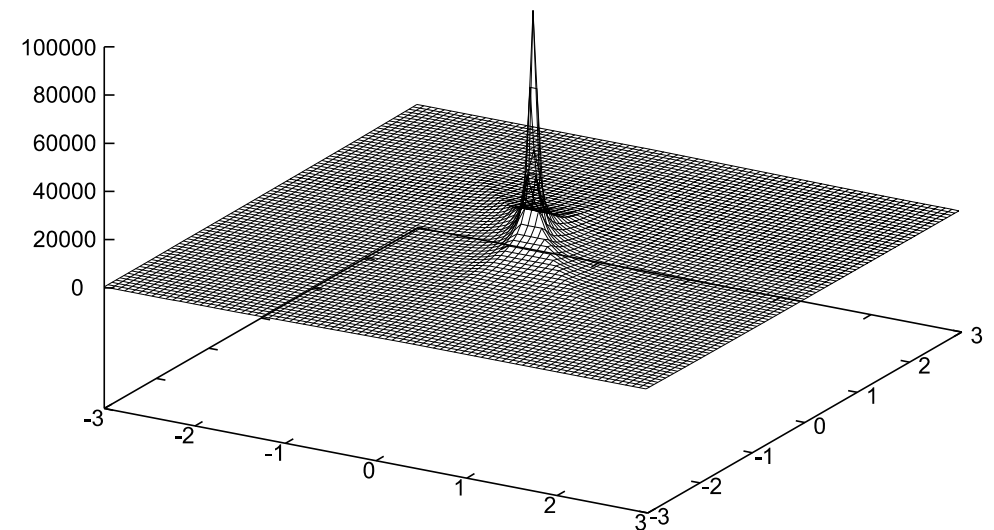
$$\sum_{i=0}^{n-2} 100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2$$



## Summation Cancellation Funktion

$$100 / \left( 10^{-5} + \sum_{i=0}^{k-1} |y_i| \right)$$

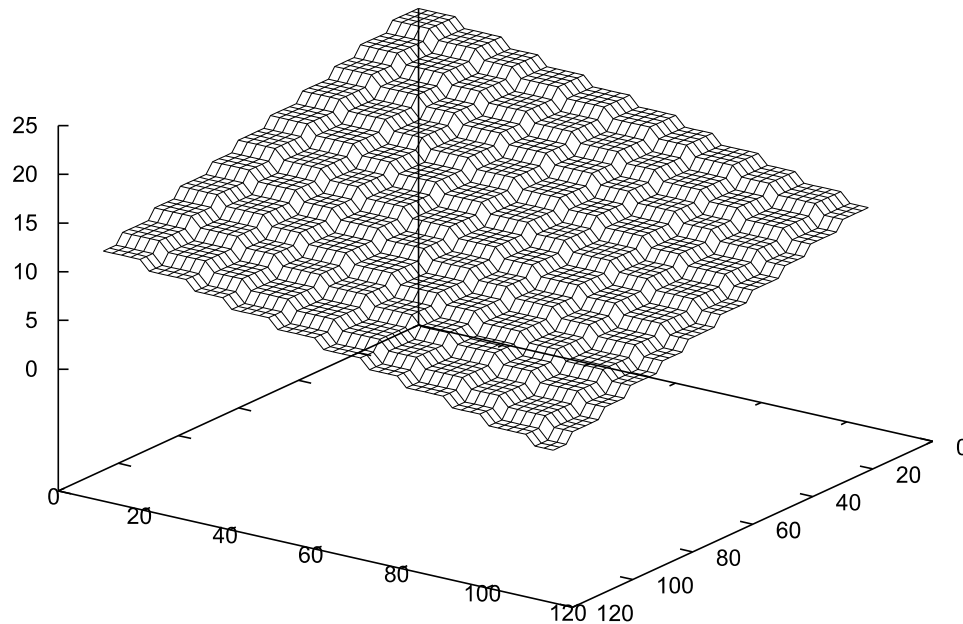
$$\text{mit } y_0 = x_0, y_i = x_i + y_{i-1}$$



# Evaluierungsfunktionen 3

## Treppen-Funktion

$$\sum_{i=0}^{n-1} \left( 10 - \frac{\text{round}(x_i)}{10} \right)$$



## Ergebnisse

- ◆ Ergebnisse leider nicht so gut, wie erwartet
- ◆ Regulärer EDA fast immer dem Multi-model EDA überlegen
- ◆ Warum?

	Multi-model EDA		Regulärer EDA	
	d = 5	d = 25	d = 5	d = 25
Griewank	Schlecht	Schlecht	Gut	Gut
Michalewicz	Schlecht	Schlecht	Schlecht	Schlecht
Rosenbrock	Schlecht	Schlecht	Schlecht	Schlecht
Summation Cancellation	Schlecht	Schlecht	Gut	Gut
Treppen	Gut	Schlecht	Schlecht	Schlecht

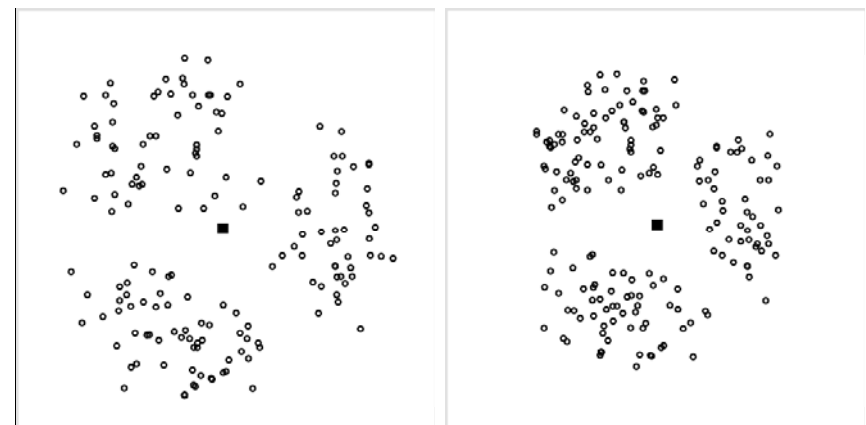
# Auswertung 1 – Griewank Funktion

- ◆ Regulärer EDA:
  - ◆ Findet Optimum recht schnell
  - ◆ Wolke bildet Kreis um das Optimum
  - ◆ Wolke wird stetig kleiner
  
- ◆ Multi-model EDA
  - ◆ Wolke schrumpft nicht weiter
  - ◆ Cluster stoßen sich gegenseitig ab
  - ◆ Vorzeitige Konvergenz wird verhindert, allerdings in einem zu großen Ausmaß

Regulärer EDA



Multi-model EDA



## Auswertung 2 – Michalewicz & Summation Cancellation Funktion

- ◆ Michalewicz Funktion
  - ◆ Beide EDAs haben keine Lösung gefunden
  - ◆ Kommen nur bis ca.  $10^{-4}$  an das Optimum heran
  - ◆ Daher vermutlich nur ein lokales Optimum gefunden
  
- ◆ Summation Cancellation Funktion
  - ◆ Verhalten des multi-model EDAs sehr ähnlich zur Griewank Funktion
  - ◆ Cluster stoßen sich gegenseitig ab
  - ◆ Durch die geringe Präzision wird der schmale Pike nicht gefunden
  - ◆ Punktwolke verformt sich, kann sogar Ring um das Optimum bilden

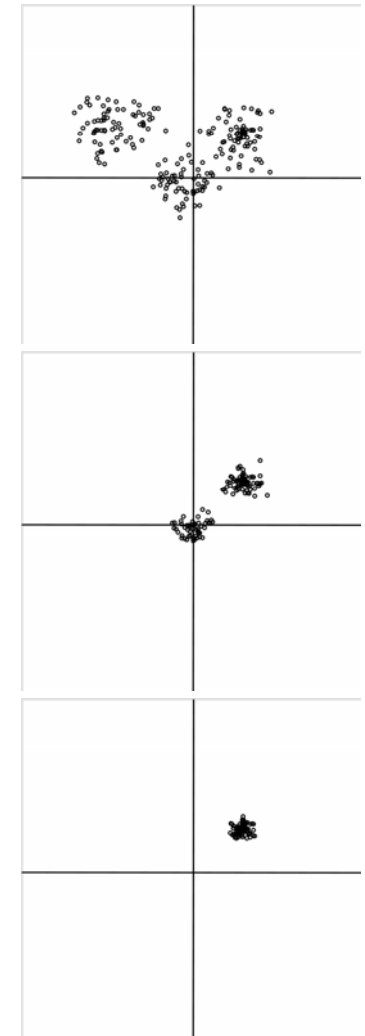
## Auswertung 3 – Rosenbrock Funktion

- ◆ Normaler EDA
  - ◆ Findet schnell den schmalen Kanal
  - ◆ Anschließend kann dieser nicht genau erkundet werden
  - ◆ Es wird nur ein lokales Optimum gefunden
  
- ◆ Multi-model EDA
  - ◆ Erkundet gleichzeitig verschiedene Stellen
  - ◆ Dadurch wird der Bereich um das globale Optimum gefunden
  - ◆ Cluster stoßen sich wieder ab
  - ◆ Durch geringere Präzision wird das Optimum selber nicht gefunden

Normaler EDA



Multi-model EDA



## Auswertung 4 – Treppen-Funktion

- ◆ Multi-model EDA deutlich besser als normaler EDA
- ◆ Treppen-Funktion extrem schweres Problem für normalen EDA
  - ◆ Auf einer Stufe sehr hohe Neutralität
  - ◆ Suchlauf degeneriert zu einem *Random Walk*
  - ◆ Neue Individuen werden nicht außerhalb der selektierten Wolke erzeugt
  
- ◆ Rekombination beim Multi-model EDA
  - ◆ Geschickte Wahl des Operators
  - ◆ Sprung in neuen Bereich des Suchraums möglich
  - ◆ Dadurch Überwindung der Stellen mit hoher Neutralität
  - ◆ Funktioniert nicht bei höheren Dimensionen

# Inhalt

1. Optimierungsverfahren
  1. Genetische Algorithmen
  2. Estimation of Distribution Algorithmen
2. Multi-Model EDA
  1. Konzept und Idee
  2. Aufbau und Ablauf
3. Evaluierung
  1. Funktionen
  2. Ergebnisse und Auswertung
4. Zusammenfassung und Ausblick

# Zusammenfassung

- ◆ Multi-model EDA
  - ◆ Verknüpft regulären EDA mit GA
  - ◆ Idee: Verhindern vorzeitiger Konvergenz
  - ◆ Rekombination als weiteren Freiheitsgrad
- ◆ Ergebnisse von den Werten her eher schlecht
  - ◆ Kaum Lösungen gefunden
  - ◆ Im Vergleich schlechter als normaler EDA
- ◆ Nach genauerer Analyse
  - ◆ Vorzeitige Konvergenz wird verhindert
  - ◆ Rekombination hilft den Algorithmus besser an das Problem anzupassen
  - ◆ Aber geringere Genauigkeit durch Clustern und Rekombination

## Ausblick

- ◆ Dynamische Adaption von Rekombination und Clustern
  - ◆ Idee: gleichzeitiges Verhindern vorzeitiger Konvergenz und hohe Präzision
  - ◆ Wenn Cluster unnötig → Anzahl verringern
  - ◆ Großer Bereich der erkundet wird → Anzahl Cluster erhöhen
- ◆ Integration eines Bayessche Netzes
  - ◆ Austausch des mathematischen Modells
  - ◆ Verbesserung der Ergebnisse