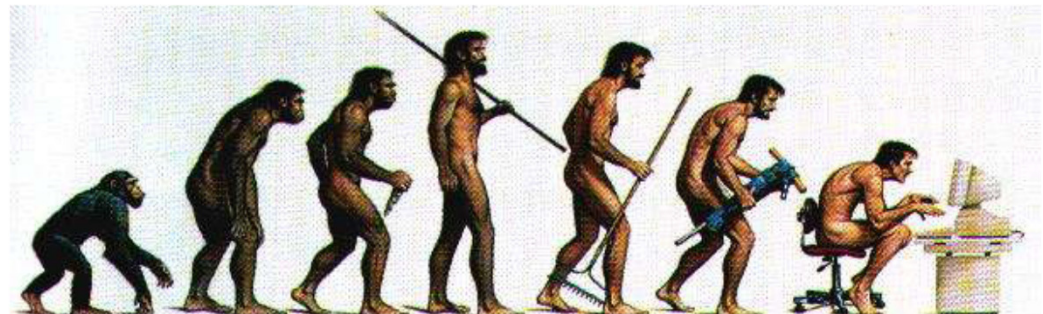


# Künstliche Intelligenz

## 5. Genetische Algorithmen

Prof. Dr. Klaus Dorer



# Übersicht

Einführung
Agentensysteme
Schwarmintelligenz
Robotik
<b>Genetische Algorithmen</b>
Entscheidungsbäume
Neuronale Netzwerke
Reinforcement Learning
Autonomes Fahren

## ■ Genetische Algorithmen

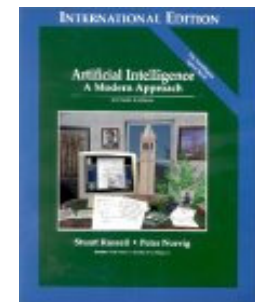
- Begriffe
- Kodierung
- Fitnessfunktion
- Selektion
- Rekombination
- Mutation
- Beispiele
- Ausflug: Statistische Signifikanz

# Ziele

- Vorteile und Nachteile Genetischer Algorithmen kennen
- Aufbau und Funktionsweise verstehen
- Probleme mit vorhandener Software lösen können

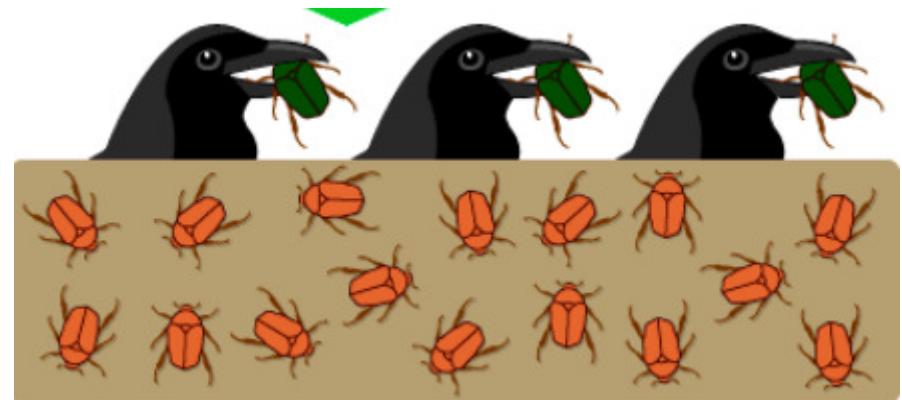
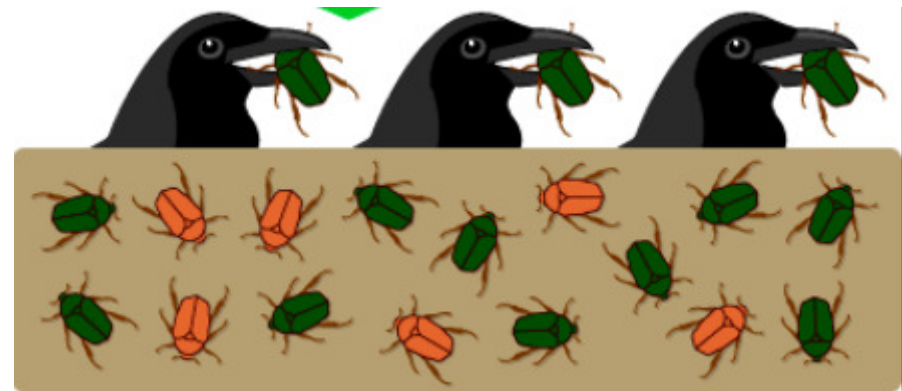
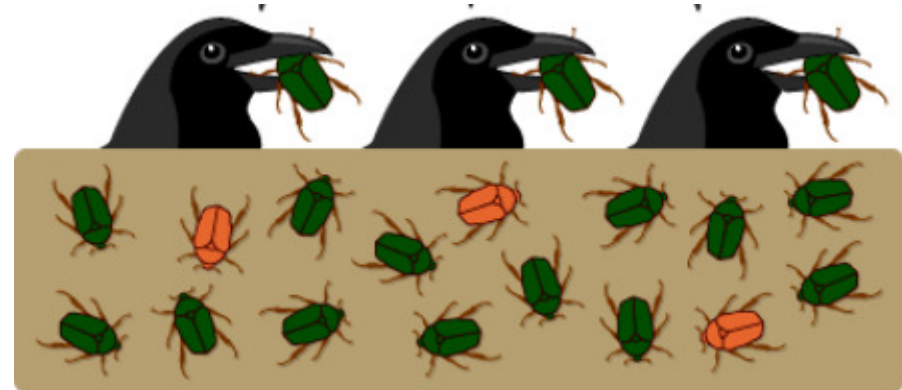
# Quellen

- Russel, Norvig, Artificial Intelligence: A Modern Approach, Prentice Hall, ISBN 0137903952, 2002.
- [www.wikipedia.de](http://www.wikipedia.de)
- [de.wikiversity.org](http://de.wikiversity.org)



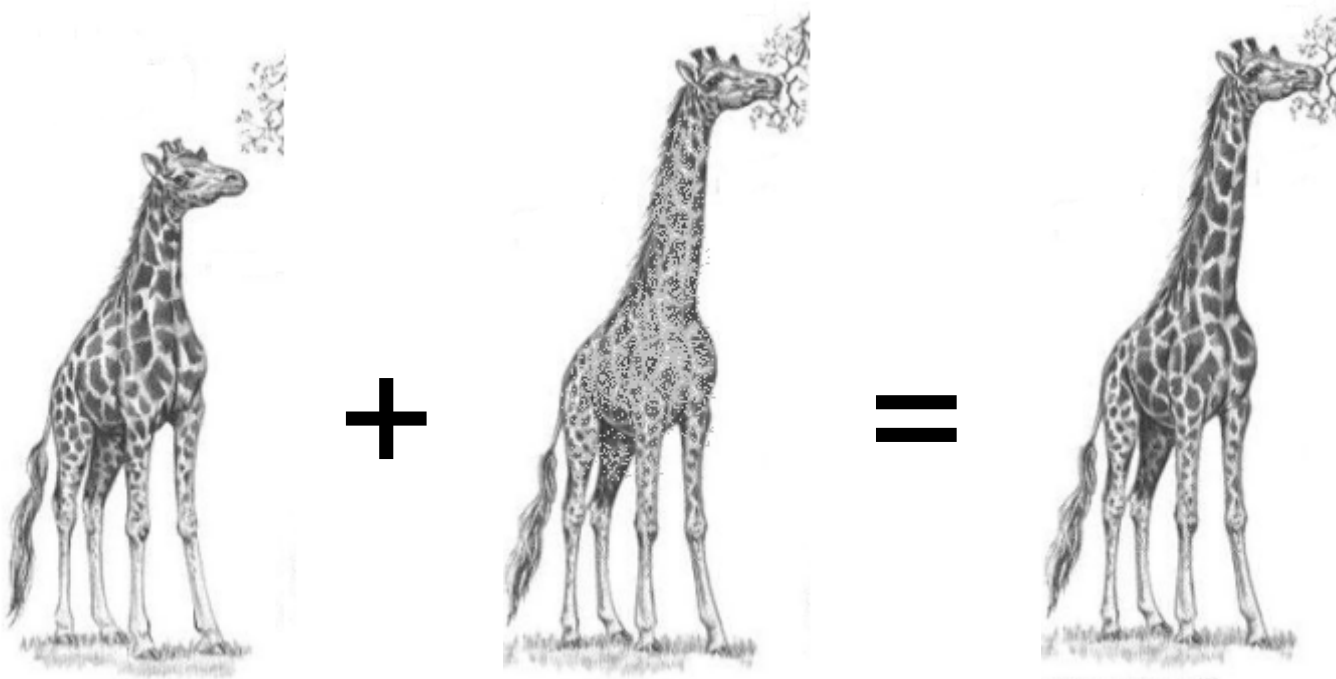
# Evolution

- Zufällige Veränderungen führen zu abweichenden Spezies
- Natürliche Selektion bevorzugt gewisse Spezies, die sich dann häufiger vermehren können
- Nach vielen Generationen hat sich eine neue Eigenschaft durchgesetzt



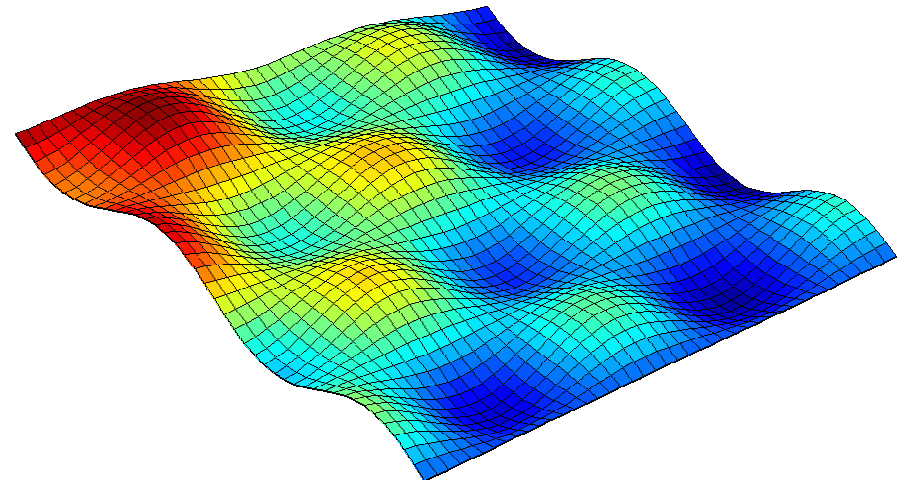
# Evolution

- Verbesserung: zweigeschlechtliche Vermehrung
  - Eigenschaften zweier Individuen werden gemischt
- Dann wieder Selektion
  - Individuen mit dem besten von beiden werden sich häufiger vermehren als Individuen mit dem schlechtesten von beiden



# Optimierung

- Gesucht wird die optimale Lösung eines Problems
- Suche ist wie ein Wandern auf einer Oberfläche
  - Beginne mit irgendeiner Lösung
  - Suche nach besseren Lösungen in der Nachbarschaft durch kleine Veränderungen
  - Position ist die aktuelle Lösung deren Höhe die Güte der Lösung
- Genetische Algorithmen
  - Starte an vielen Stellen gleichzeitig
  - Rekombination erzeugt neue Lösungen die nicht in der Nachbarschaft liegen

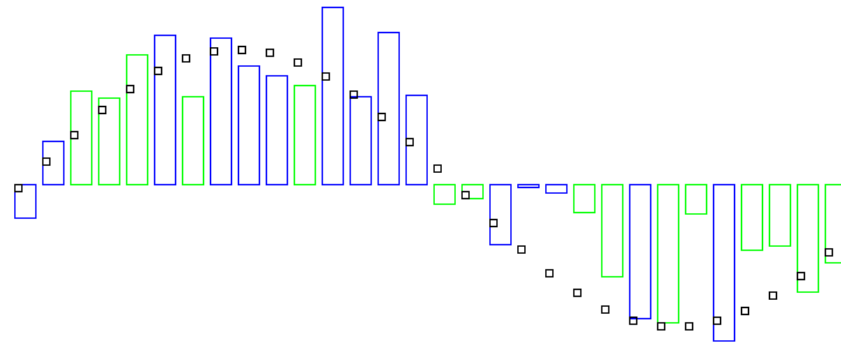


# Begriffe für Umgebungen

- Suchraum S
  - Menge aller Zustände  $s$  des Problems
- Operator (Spielzug)
  - Führt einen Zustand  $s$  in einen neuen Zustand  $s'$  über
- Nachbarzustände  $N$  von  $s$ 
  - Menge aller Zustände, die von Zustand  $s$  aus mit einem Operator erreichbar sind
- Evaluations-Funktion
  - Bildet einen Zustand auf eine Bewertung ab:  $e(s) \rightarrow \mathbb{R}$
- Globales Optimum
  - Zustand  $s$  mit bestem (höchsten/niedrigsten) Wert  $e(s)$
- Lokales Optimum
  - Güte eines Zustands  $s$  ist besser als oder gleich gut wie alle Nachbarzustände:  $e(s) \geq \max_N(e(s_n))$

# Beispiel: Funktionsapproximation

- Eine Materialform soll eine bestimmte Funktion approximieren, z.B. Sinus
  - Suchraum: an jeder Stelle kann das Material  $\pm 200$  mm dick sein
  - Operator: z.B. Wert einer Schicht um ein Delta ändern
  - Nachbarzustände: 60 bei obiger Operatordefinition
  - Evaluations-Funktion: bewertet Abstand einer Schicht zur Funktion
  - Globales Optimum: alle Werte entsprechen der Funktion
  - Lokales Optimum: vorhanden?

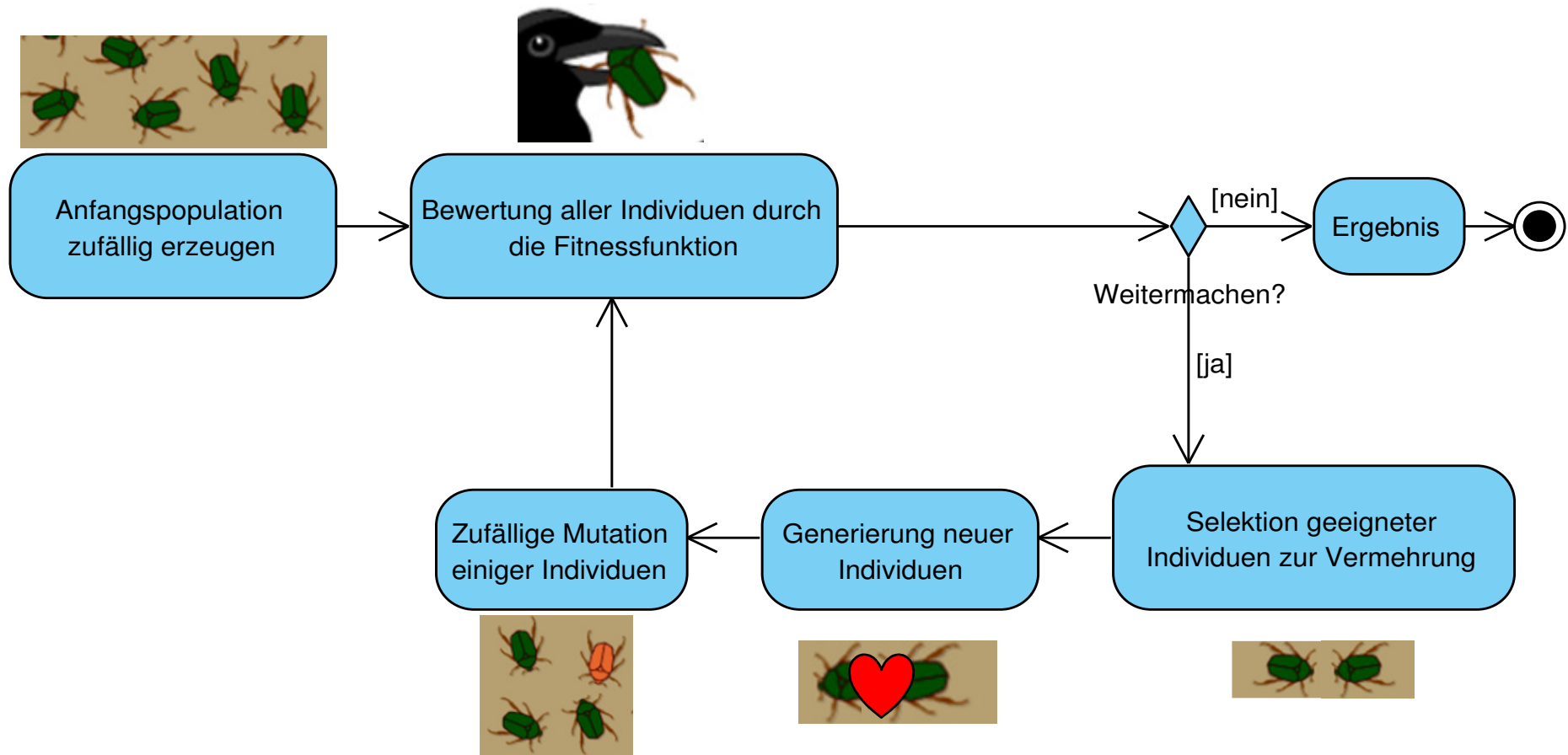




# Begriffe für genetische Algorithmen

- Statt an einer Stelle zu suchen, wird bei GA an vielen Stellen gleichzeitig gesucht
  - Jedes Individuum repräsentiert eine Lösung (Zustand)
  - Ein Individuum wird durch ein Chromosom repräsentiert, das aus einzelnen Genen besteht, deren einzelne mögliche Werte man Allele nennt
  - Eine ganze Population (Menge von Individuen) ist gleichzeitig aktiv
- Operatoren sind
  - Crossover: vermische die Chromosomen (Zustandsinformation des Individuums) zweier Individuen zu einem neuen Chromosom
  - Mutation: variiere Teile des Chromosoms eines Individuums zufällig
- Die Evaluationsfunktion
  - Wird Fitness-Funktion genannt
  - Entscheidet über die Wahrscheinlichkeit, mit der ein Individuum als Eltern Individuum ausgesucht wird (Selektion) und Kinder erzeugt

# Genetische Algorithmen



# Hauptalgorithmus

```
population = generateInitialPopulation(size);
while wantToContinue
{
    // calculate fitness of each individuum
    population.calculateFitness();
    if (population.getBest() > bestSolution)
        bestSolution = population.getBest();
    // select parents for recombination
    parents = population.selectParents(n);
    // create new individuums
    children = parents.recombineUsingCrossover();
    // take over best parents to next generation
    children.add(parents.getBestParents(m));
    // mutate new individuums with a small probability
    children.mutate(probability);
}
```

# Überblick

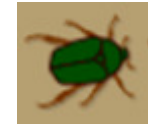
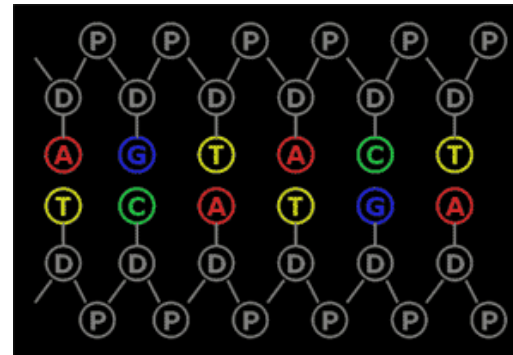
## ■ Zu definieren

- Kodierungsvorschrift (Repräsentation)
  - Problemspezifische Kodierung einer Lösung als Chromosom
- Fitnessfunktion
  - Problemspezifische Bewertung eines Individuums
- Selektionsmethode
  - Welche Individuen erzeugen Kinder
- Rekombinationsstrategie
  - Wie werden aus vorhandenen neue Chromosomen erzeugt
- Mutationsstrategie
  - Zufällige Veränderung von Genen
- Anfangspopulation
  - Meist zufällige Chromosomenkette, je nach Domäne aber komplizierteres Verfahren nötig
  - Zu viele Individuen -> mehr Rechenzeit
  - Zu wenig Individuen -> Gefahr der ‚Inzucht‘ (zu kleiner Genpool)

# Repräsentation

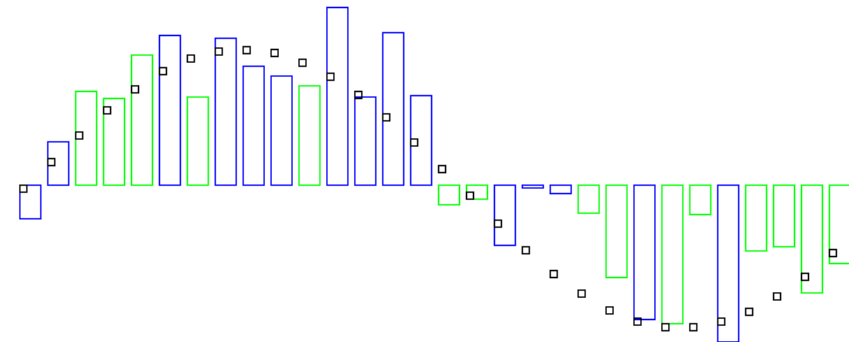
## ■ Mensch, Tiere

- DNA Doppelhelix mit 4 Symbolen G, A, C, T
- 23 Chromosomenpaare beim Menschen



## ■ Computer

- z.B. Zeichenketten oder Zahlen
- [-24.61792, 31.425507, 67.223206, 62.641907, 93.90582, 107.080444, 63.36856, 91.97037, 85.16467, 78.70178, 71.07422, 127.14902, 63.852966, 109.58435, 64.0025, -21.357605, -10.049866, -43.09497, -2.1212463, -6.8831177, -20.01773, -66.251495, -96.93176, -99.260284, -21.76715, -112.17766, -47.78308, -44.790375, -33.241516, -56.502228]



# Fitnessfunktion

- Berechnet für jedes Individuum einen Wert

- Reproduktionstauglichkeit
- Lösungsqualität

- Hängt von der Domäne ab

- Tiere

- Überlebensfähigkeit in der realen Welt



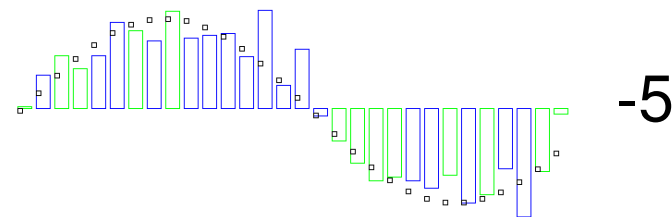
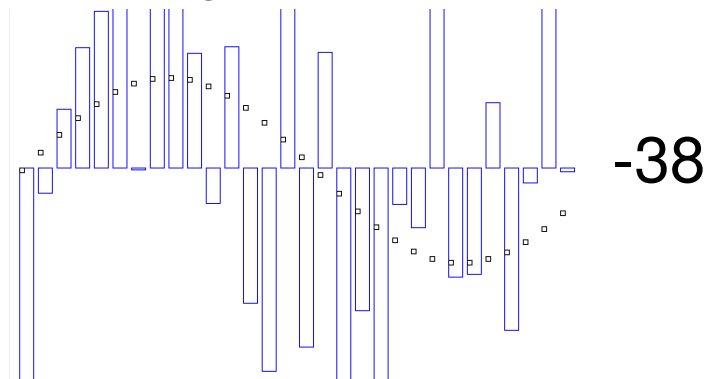
3 von 100



11 von 100 schaffen es bis zur Fortpflanzung

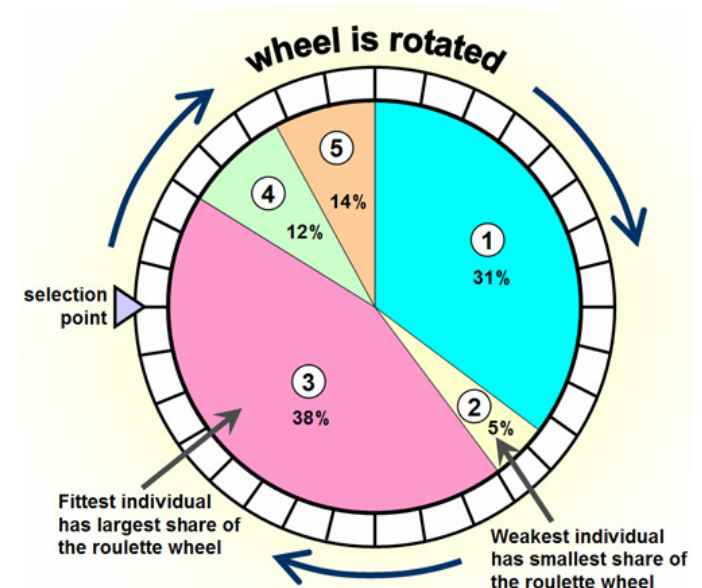
- Approximation

- Negativer Absolutfehler



# Selektion

- Auswahl der besten Individuen zur Reproduktion
- Elitismus
  - Auswahl n beste Individuen zu Eltern
  - n hängt davon ab, wie viele Kinder jedes Elternpaar erzeugt und wie viele Eltern in die nächste Generation übernommen werden
  - Kombination zu Paaren auf vielfältige Art möglich
- Monte Carlo Methode
  - Für jedes Kind werden die beiden Eltern zufällig aus der ganzen Population ausgewählt, die Wahrscheinlichkeit der Auswahl hängt ab vom Wert der Fitness-Funktion
- Rank Selection
  - Wie Monte Carlo, Wahrscheinlichkeit hängt aber vom Rang ab



# Rekombination

## ■ Crossover

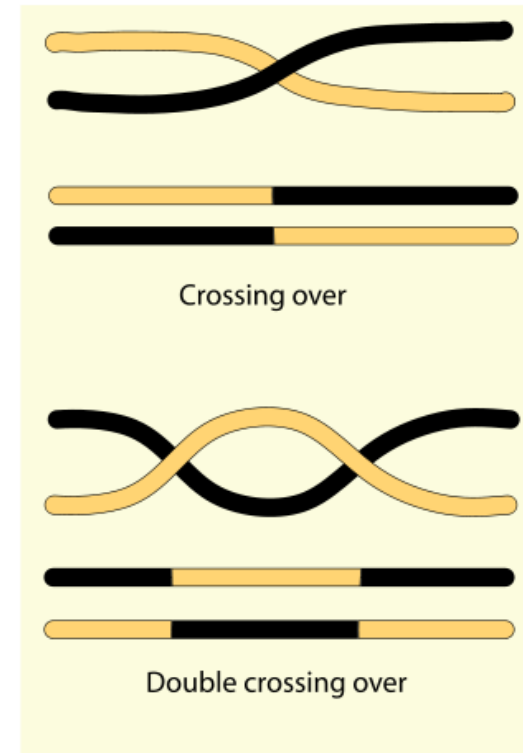
- Wähle (zufällige) Stelle im Gencode, an dem Crossover stattfinden soll
- Bilde Kinder durch wechselseitige Kombination der beiden Teile
- Beispiel

- Parent 1    1 0 1 | 0 0 1 0 1
- Parent 2    0 0 0 | 1 1 1 0 1

- Child 1      1 0 1 | 1 1 1 0 1
- Child 2      0 0 0 | 0 0 1 0 1

- Viele alternative Variationen möglich

- 
- 
- 





# Mutation

- Mit einer (geringen) Wahrscheinlichkeit wird der Gencode an (einer) zufälligen Stelle(n) verändert

- Beispiel

- Individuum 1 vorher

G A T A C C A

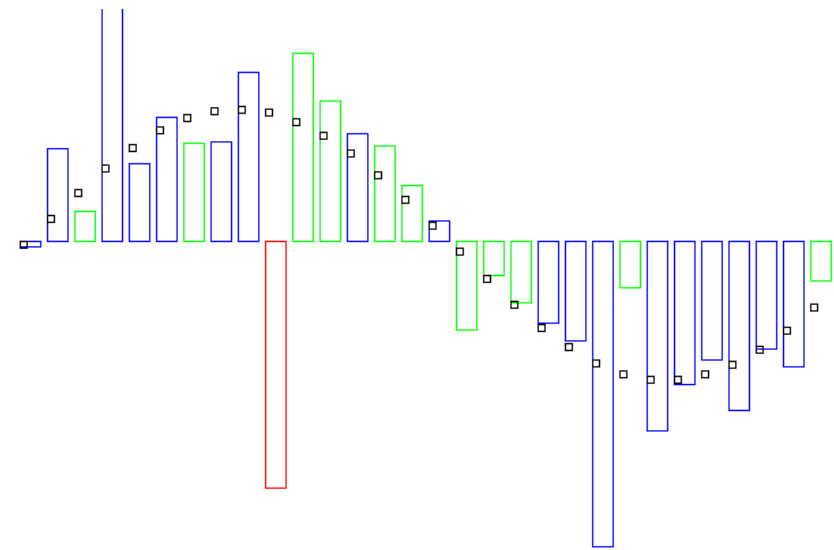


- Individuum 1 nachher

G A T A T C A

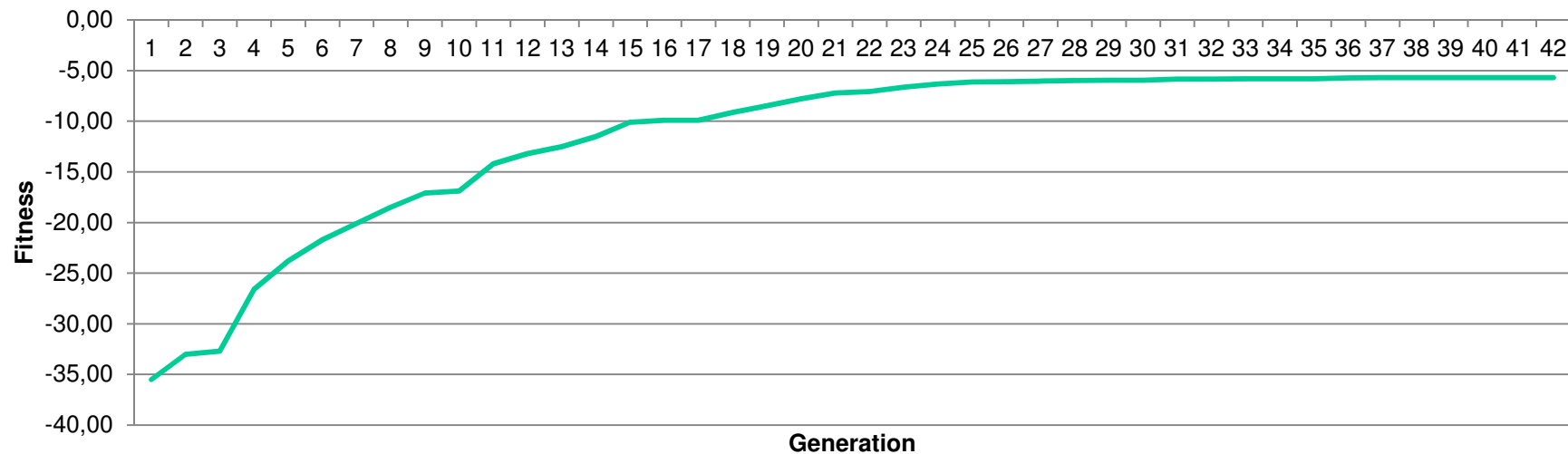


- Führt oft zu Verschlechterung  
manchmal zu Verbesserung
- Selektion entscheidet
- Meist stellt man sicher, dass  
ein gültiger neuer Gencode  
im Sinn des Problems entsteht



# Beispiel Funktionsapproximation

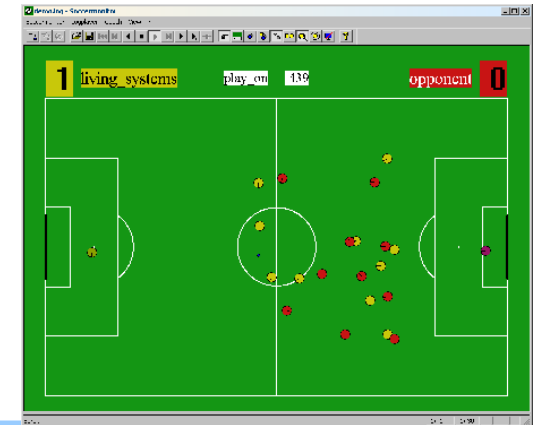
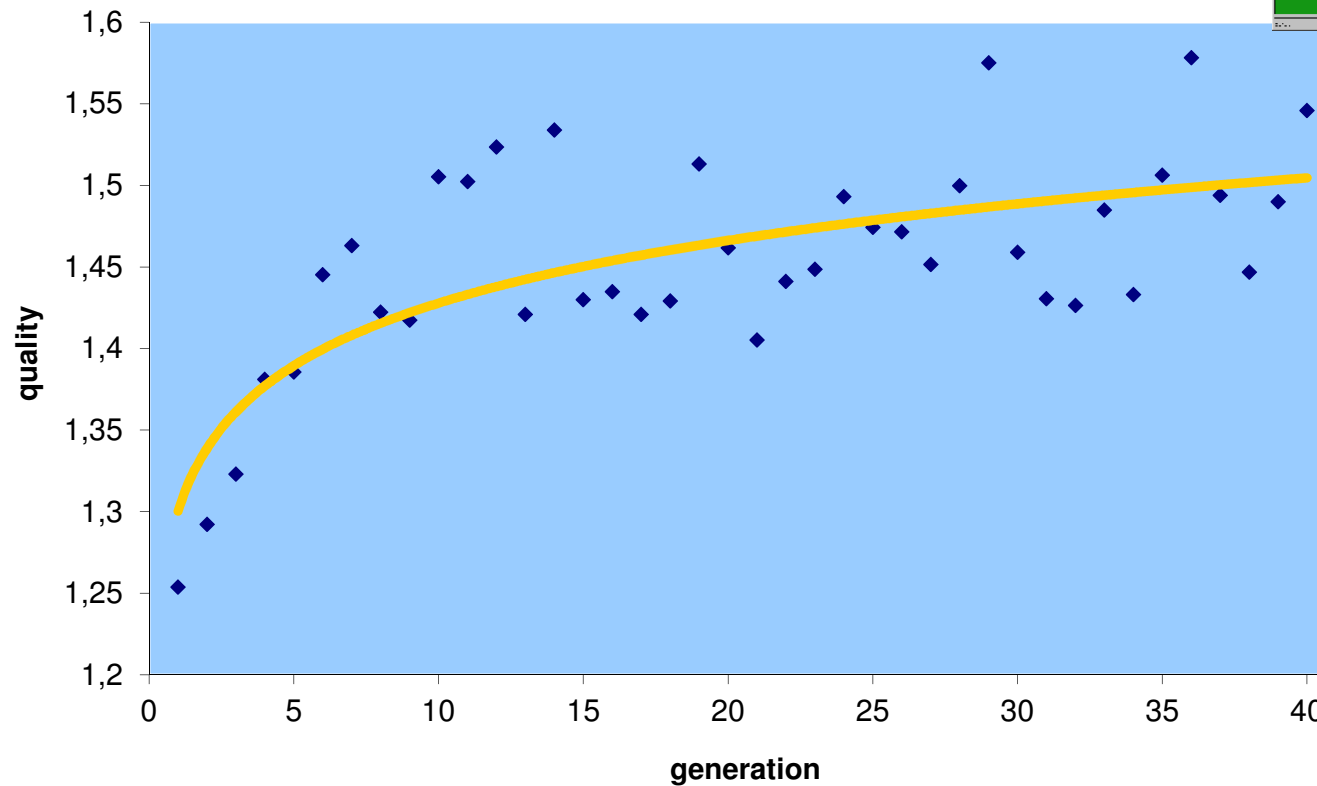
- Wie oben gezeigt
- Ziel: möglichst genaue Annäherung an Sinus Funktion



# Beispiel RoboCup

- Population: 50 Fußball Agenten (2D Simulation)
- Lernziel: Hart und präzise schießen

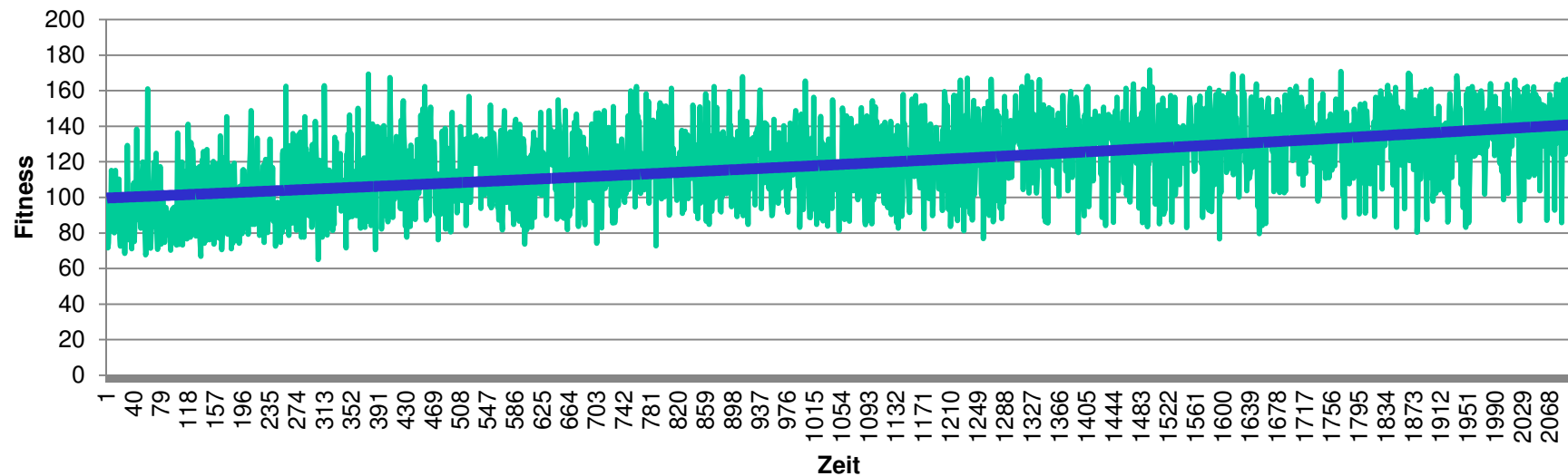
Quality of kick over time



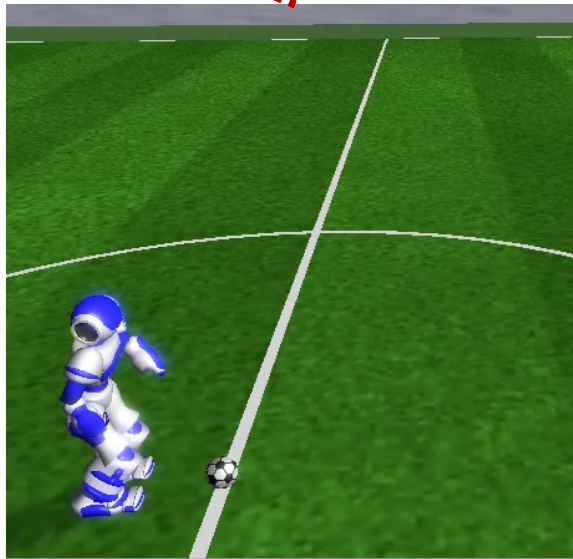
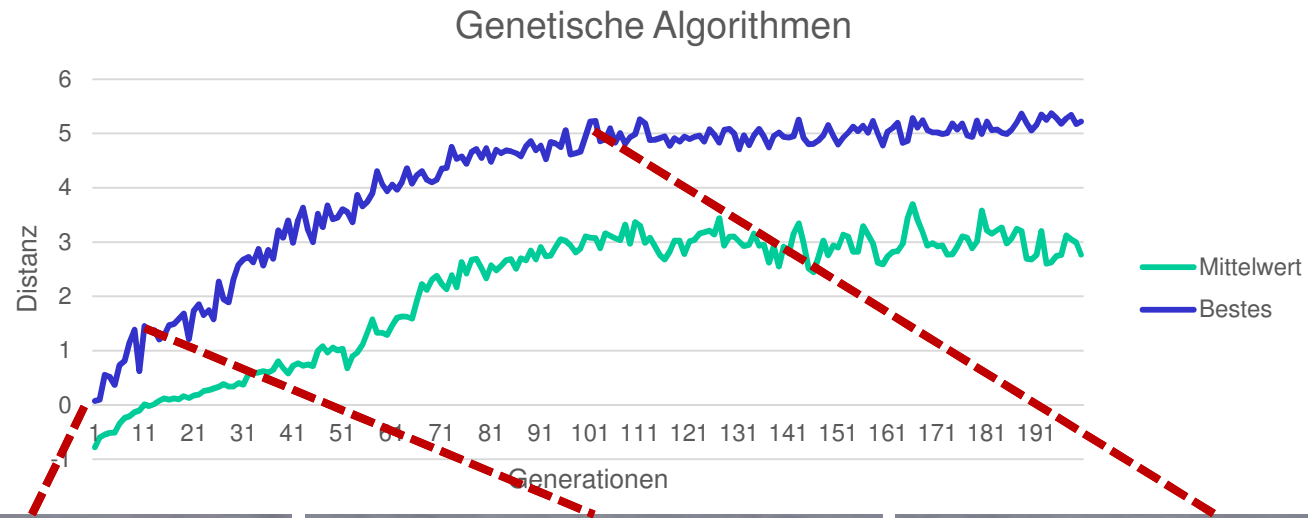
# Beispiel RoboCup

## ■ Laufen lernen (3D Simulation)

- 10 Parameter, die das Laufen beeinflussen wurden optimiert
- Ziel: möglichst schnell geradeaus laufen



# Beispiel RoboCup



# Beispiel: Transport-Netzwerk Optimierung

## ■ Aufgabe

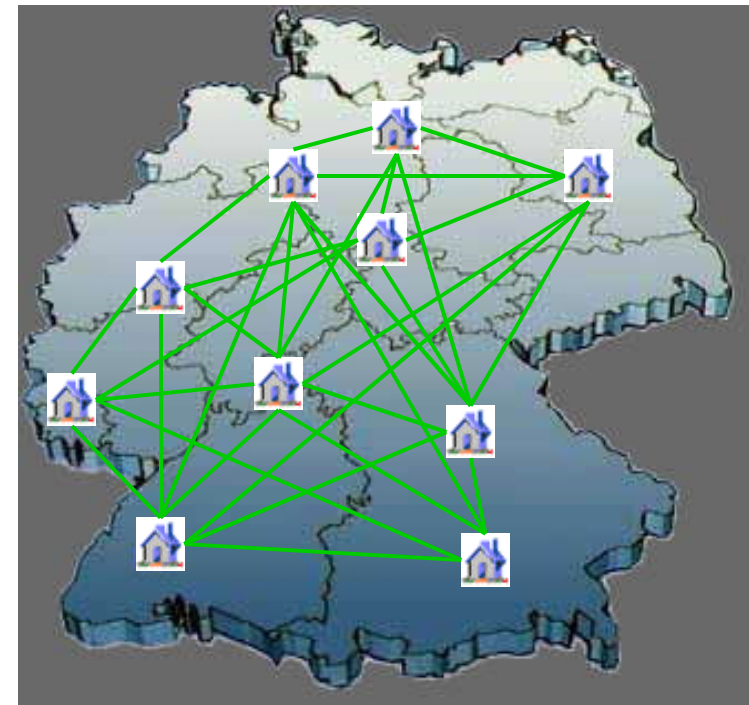
- Generiere Transportplan für alle Relationen zwischen Hubs

## ■ Ziele

- Kostenreduktion
- Kilometerreduktion
- Minimierung der Anzahl LKW

## ■ Randbedingungen

- Eingeschränkte LKW Kapazität
- Fahrzeitrestriktionen
- Früheste Abfahrtszeiten
- Späteste Ankunftszeiten
- Lade-, Ablade- und Umladezeiten
- Öffnungszeiten der Hubs



# Beispielvorgaben

## ■ Repräsentation

- Allele: Nummern der Hubs
- Position: Auftragsnummer

17, 22, 5, 0, 13, 0

## ■ Fitnessfunktion

- Anzahl der gefahrenen Kilometer

## ■ Selektion

- Beliebig

## ■ Rekombination

- Beliebige Varianten von crossover

17, 22, 5, 0, 13, 0

15, 22, 0, 7, 12, 0

---

17, 22, 0, 7, 12, 0

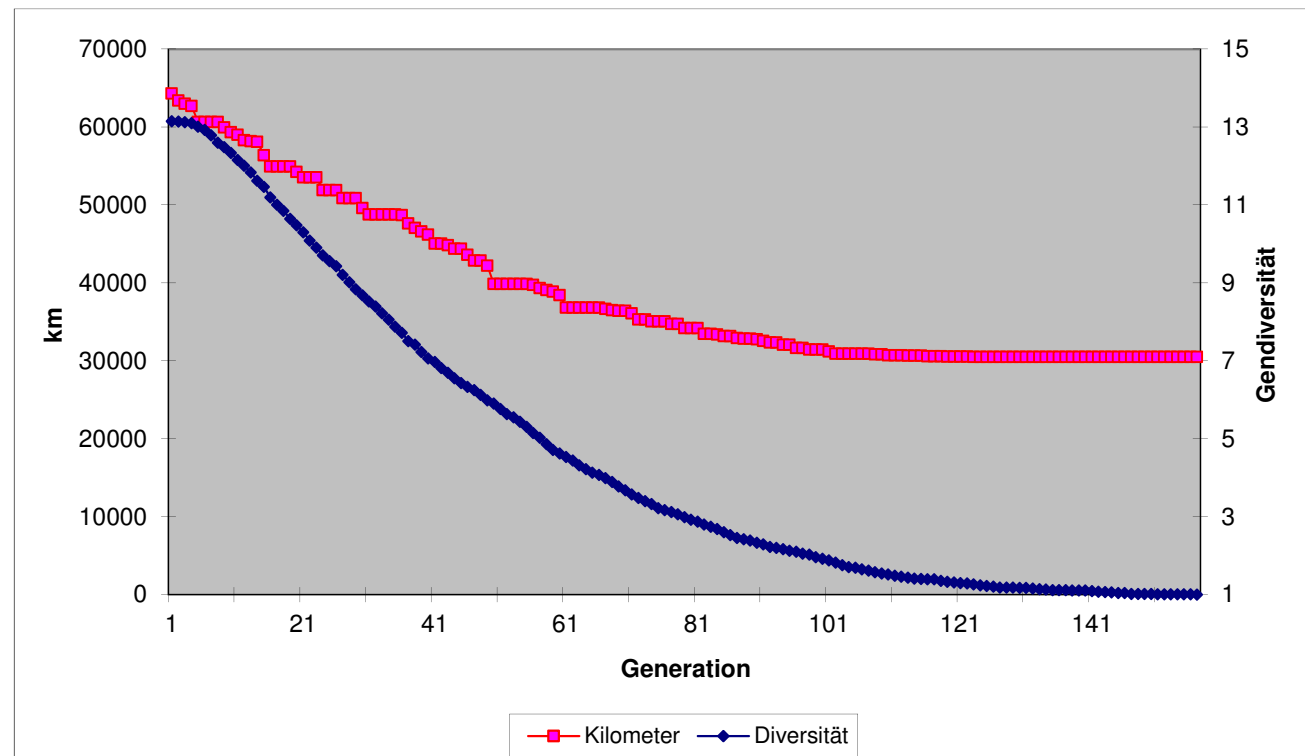
## ■ Mutation

- Beliebig, solange gültiger Hub für Auftrag verwendet wird

17, 10, 5, 0, 13, 0

# Genetische Algorithmen

- Beispiel Transportnetzwerk
  - Lernziel: Kilometerreduktion
  - Population: 720 Übernahme beste: 10
  - Mutation: keine

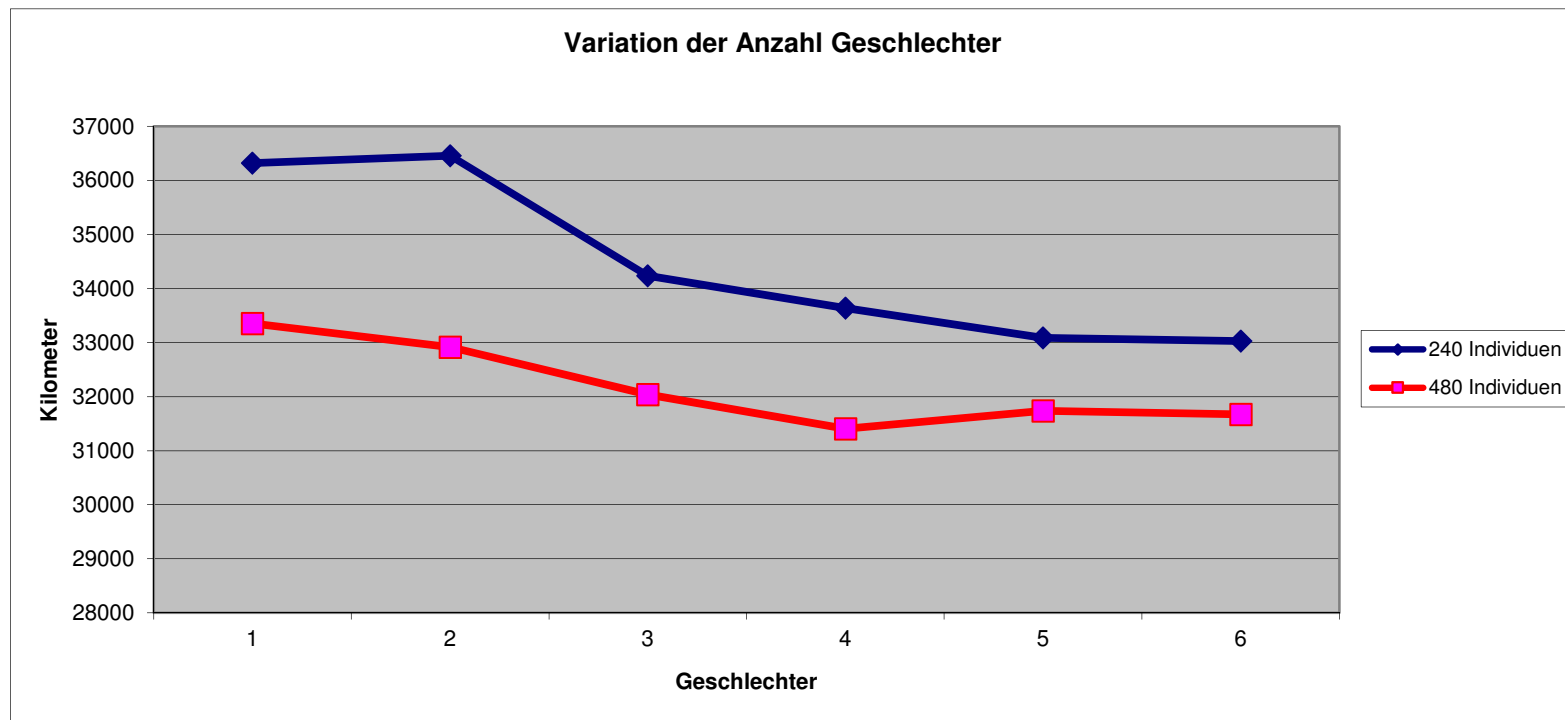




# Genetische Algorithmen

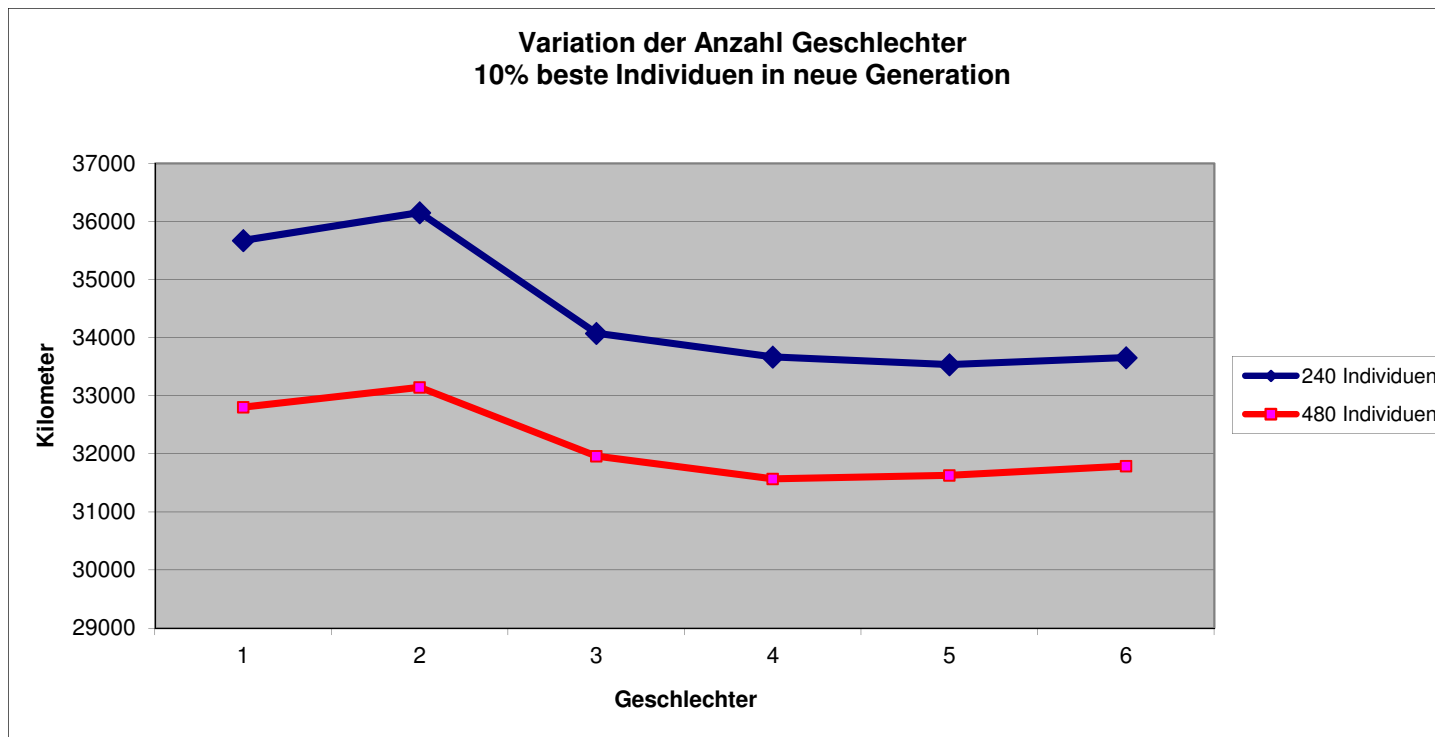
## ■ Einführung von Geschlechtern

- Als separate Gruppen von Individuen
- Anzahl der Eltern = Anzahl Geschlechter (aber mindestens 2)
- Jedes Elternteil aus anderem Geschlecht (außer bei einem Geschlecht)



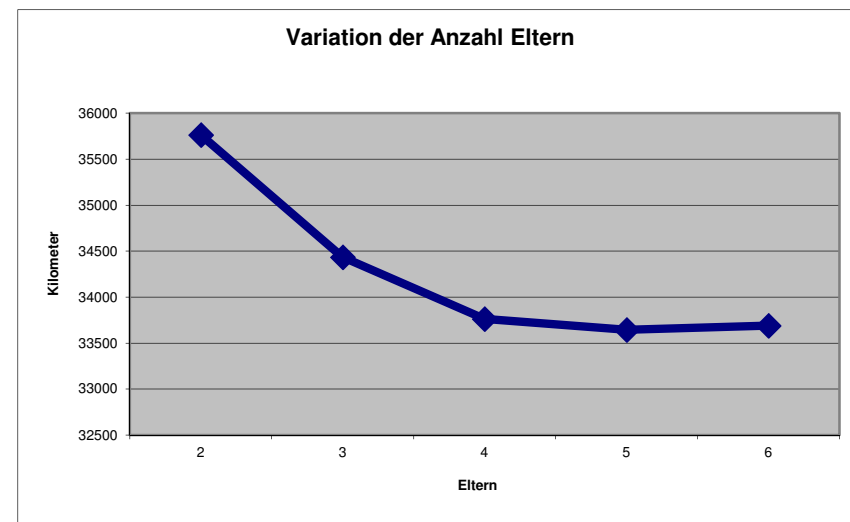
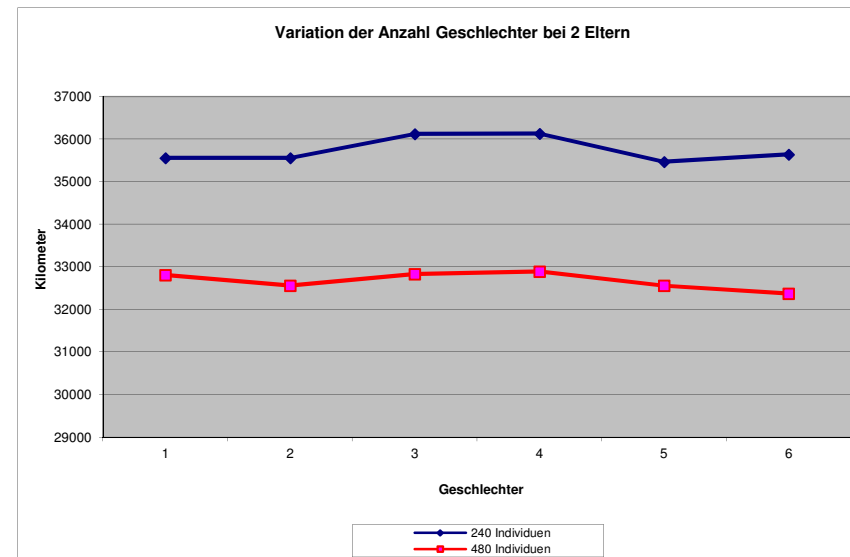
# Genetische Algorithmen

- Relative Anzahl Individuen übernehmen
  - Statt fix 10 werden 10% der besten Individuen übernommen
  - Trotzdem bleibt ein signifikanter Effekt



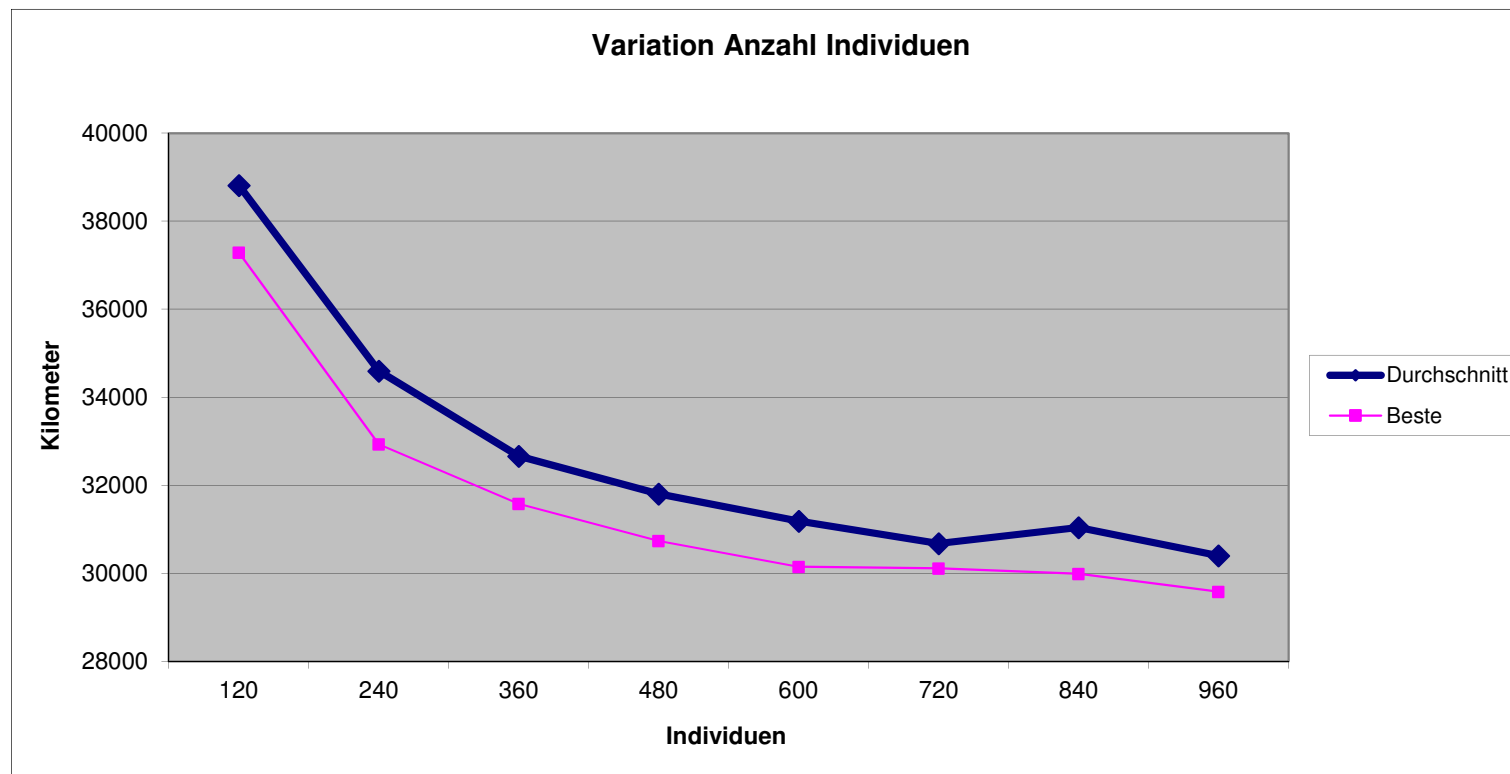
# Genetische Algorithmen

- Geschlechter und Anzahl Eltern getrennt variieren
  - Eltern aus verschiedenen Geschlechtern (falls Anzahl Eltern  $\leq$  Anzahl Geschlechter)
- Variation der Anzahl Geschlechter bei gleich bleibender Anzahl Eltern (2): kein Effekt
- Variation der Anzahl Eltern je Individuum gemittelt über 1 – 6 Geschlechter: Effekt
- Folgerung: Nicht die Anzahl Geschlechter spielen eine Rolle, sondern die Anzahl Eltern je Individuum



# Anzahl Individuen

- Wenig überraschend ist, dass die Anzahl der Individuen einen signifikanten Einfluss hat, da diese die Größe des explorierten Suchraums maßgeblich beeinflussen: Jedes Individuum entspricht einer möglichen Lösung



# Ergebnisse

- 175 Aufträge
- 33 Hubs
- Im Mittel 16 Transportpfade je Relation
- 61233 Transportkilometer

Verfahren	Ergebnis (Mittel) (km)	Ergebnis (Bestes) (km)	Laufzeit (Mittel) (sec)	Ergebnis (Bestbekannt)
Hill Climbing	33.288	33.288	9	30.657
Tabu Suche	32.647	32.647	60	32.051
Simulated Annealing	28.502	28.325	65	27.890
Genetische Algorith.	30.303	29.481	61	28.110

# Problem: Diversität

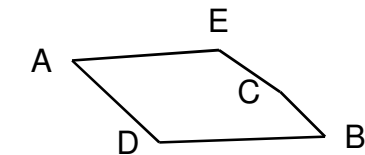
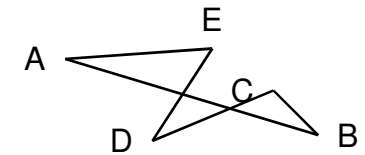
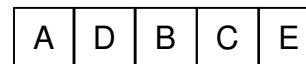
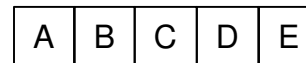
- Einerseits wollen wir besonders fitte Individuen bevorzugen
- Andererseits tauchen dadurch deren Gene immer häufiger auf, die Diversität nimmt ab
- Diversität = 1 bedeutet keine neuen Individuen mehr
- Lösungen
  - Mutation
    - zufälliges Einführen neuer Diversität
  - Repräsentation
    - Mehrere Geschlechter erhöhen die Diversität
    - Mehrere Populationen mit geringer Durchmischung
  - Selektion
    - Sicherstellung einer minimalen Diversität der Eltern

# Problem: Gen-Abhängigkeit

- Bei manchen Problemrepräsentationen sind einzelne Gene nicht unabhängig voneinander
- Beispiel: TSP (Problem des Handlungsreisenden)

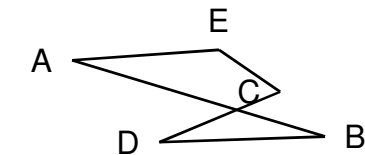
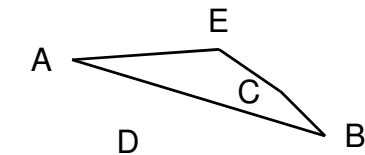
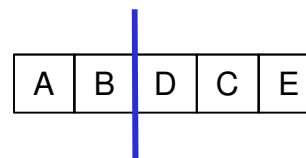
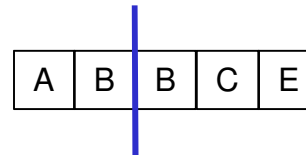
- Repräsentation

- Reihenfolge der Gene entspricht Besuchsreihenfolge



- Rekombination

- Normales Crossover erzeugt in der Regel ungültige Lösungen
- Lösung schlechte Fitness?
- Lösung domänenabhängige Strategie



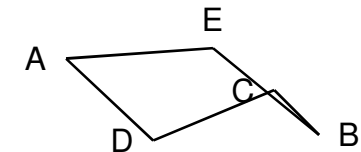
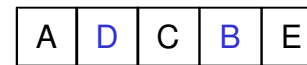
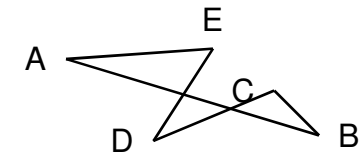
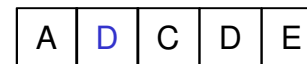
- Wähle Gene Elternteil 1 für linken Teil, wähle Gene für den rechten Teil in der Reihenfolge des 2. Elternteils

# Problem: Gen-Abhängigkeit

## ■ Beispiel: TSP (Forts.)

### ■ Mutation

- Häufig ungültige Lösungen
- Immer ungültig?
- Lösung domänenabhängige Strategie



- Statt für ein ausgewähltes Gen ein zufälliges Alele zu wählen, wählt man ein zufälliges zweites Gen, mit dem getauscht wird

## ■ Gen-Abhängigkeit kann die Konvergenz erheblich verlangsamen



# Problem: Aufwändige Fitnessfunktion

- Sehr lange Laufzeiten bei aufwändiger Fitnessfunktion
  - Funktionsapproximation
    - Einzelbewertung: im Millisekunden Bereich
    - Komplette Optimierung: einige Sekunden
  - RoboCup
    - Einzelbewertung: im Sekunden Bereich
    - Komplette Optimierung: ein Wochenende
  - Natur
    - Einzelbewertung: im Jahresbereich
    - Komplette Optimierung: Millionen Jahre

# Variante CMA-ES

- Covariance Matrix Adaptation (Hansen, 1996)
  - Einfluss von Genen wird statistisch berechnet
  - Reduziert die Zahl der nötigen Individuen und Generationen
  - Benötigt eine einigermaßen funktionierende Ausgangslösung
  - Beispiel RoboCup
    - Genetisches Lernen eines Kicks: 3 Tage
    - Verbesserung eines Kicks mit CMA-ES: 6 Stunden

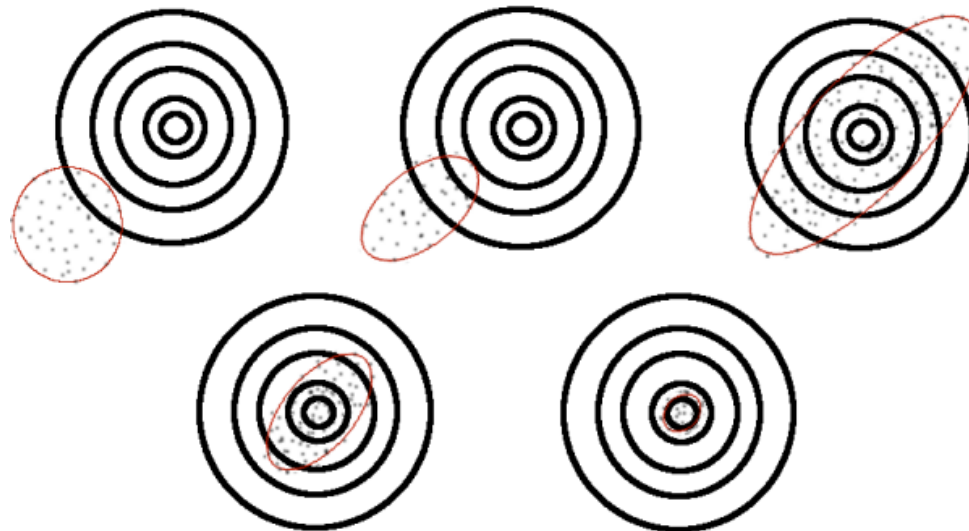


Image Source: Thesis Christoph Füller

# Genetische Algorithmen

## ■ Vorteile

- Geeignet für große Suchräume
- Sehr anschauliches Verfahren, leicht zu implementieren
- Kein Vorwissen über die Domäne nötig
- Wir erhalten eine Lösungsmenge

## ■ Nachteile

- Gleichzeitiges Suchen an mehreren Stellen ( $n$  Individuen) kostet Laufzeit und Speicher
- Bevorzugung bestimmter Gene führt leicht zu mangelnder Diversität des Genpools und führt im Extremfall dazu, dass alle Individuen einer Population identische Gene haben
- Optimalität kann nicht garantiert werden
- Nicht-deterministisch

## ■ Fürs Leben

- Evolution funktioniert nicht ohne das Sterben von Individuen

# Ausflug: Statistische Signifikanz

- Oft müssen wir zwei nicht-deterministische Verfahren vergleichen
- Nach Messung erhalten wir zwei Mittelwerte
- Frage
  - Ist Verfahren 1 besser als 2, oder ist der Unterschied zufällig?
  - Ist also  $\mu_1$  tatsächlich kleiner als  $\mu_2$  weil  $\bar{x}_1$  kleiner als  $\bar{x}_2$  ist?
- Antwort: Statistische Signifikanz Tests

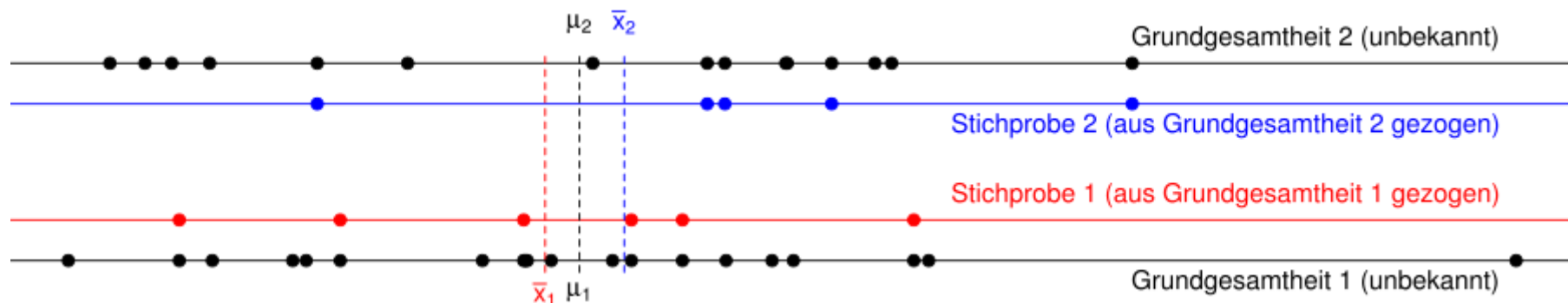


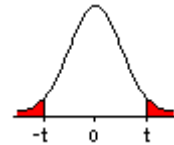
Image Source: wikipedia

# Ausflug: Statistische Signifikanz

## ■ Grundhypothese

$$H_0 : \mu_X - \mu_Y = \omega_0$$

$$H_1 : \mu_X - \mu_Y \neq \omega_0$$



Zweiseitiger Test



Einseitiger Test

$$H_0 : \mu_X - \mu_Y \leq \omega_0$$

$$H_1 : \mu_X - \mu_Y > \omega_0$$

## ■ Berechnung

$$t = \sqrt{\frac{nm}{n+m}} \frac{\bar{x} - \bar{y} - \omega_0}{s}.$$

- mit m,n Stichprobengröße Stichprobe 1 bzw. 2,
- $\bar{x}$  Mittelwert Stichprobe 1
- $\bar{y}$  Mittelwert Stichprobe 2
- $s^2 = \frac{(n-1)s_x^2 + (m-1)s_y^2}{n+m-2}$

## ■ Test


- $H_0$  wird zum Signifikanzniveau  $\alpha$  abgelehnt, wenn

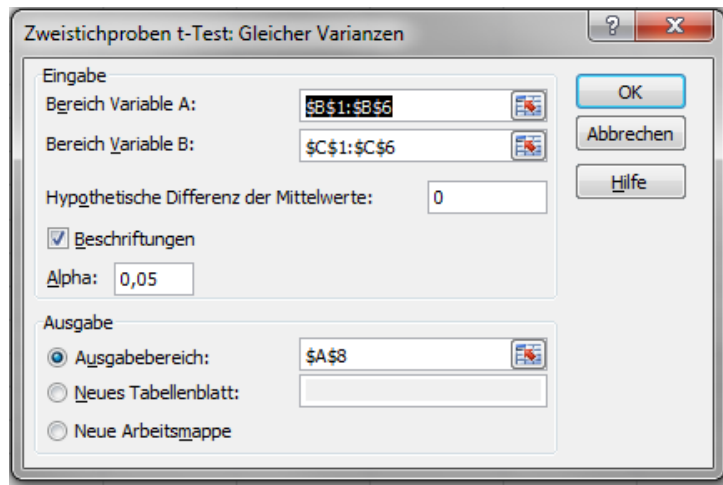
$$|t| > t(1 - \frac{1}{2}\alpha, n + m - 2).$$

$$t > t(1 - \alpha, m + n - 2)$$

# Ausflug: Statistische Signifikanz

## ■ Beispiel in Excel

- Reiter Daten->  Datenanalyse (freischalten Datei->Optionen->AddIns)
- Funktion: Zweistichproben t-Test: Gleicher Varianzen



	A	B
1		
2	17	10
3	12	2
4	21	13
5	19	15
6	9	5
7		
8	Zweistichproben t-Test unter der Annahme gleicher Varianzen	
9		
10	A	B
11	Mittelwert	15,6
12	Varianz	24,8
13	Beobachtungen	5
14	Gepoolte Varianz	27,15
15	Hypothetische Differenz der Mittelwerte	0
16	Freiheitsgrade (df)	8
17	t-Statistik	2,00276053
18	P(T<=t) einseitig	0,04008631
19	Kritischer t-Wert bei einseitigem t-Test	1,85954804
20	P(T<=t) zweiseitig	0,08017261
21	Kritischer t-Wert bei zweiseitigem t-Test	2,30600414