

Complex Systems Engineering

Evolutionäre Algorithmen

Prof. Dr. Christin Seifert

17. Oktober 2017

University of Passau, WS 2017/2018

1. Einführung
2. Biologische Grundlagen
3. Grundprinzipien
4. Beispielanwendungen
5. Zusammenfassung

Einführung

Evolution

The process by which different kinds of living organism are believed to have developed from earlier forms during the history of the earth. – Oxford Dictionary

(Biologie) stammesgeschichtliche Entwicklung von niederen zu höheren Formen des Lebendigen – Duden

Charakteristika

- Entwicklung über einen sehr langen Zeitraum hinweg
- Anpassung an Lebensbedingungen über Generationengrenzen
- Entwicklung hin zum „Höheren“ (d.h. besser angepasst, effizienter)
- Mechanismen: Mutationen und Rekombinationen im Erbgut, Selektion der Fittesten einer Generation



Abbildung 1: Affe mit Schädel von Bildhauer Hugo Reinhold (CC-SA 3.0, Darwin Monkey via Wikimedia Commons)

Beispiel Entwicklung des Linsenauges.

- Augen mit Linsen, die das Licht bündeln können, besitzen eine höhere Sehfähigkeit als Augen ohne Linsen
- Es wird angenommen, dass sich Linsen über eine Vielzahl von Generationen mit leichten Veränderungen (1% pro Generation) entwickelt haben (Mutation) und die Lebewesen, die mit besserem Sehsinn ausgestattet waren, sich tendenziell eher vermehrt haben (Selektion) [2]

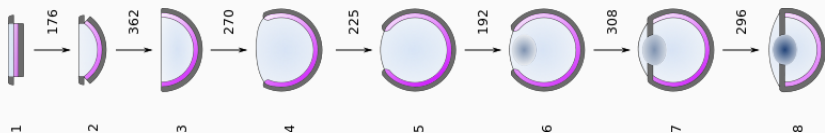


Abbildung 2: Modell der Evolution des Linsenauges. Dreistellige Zahlen geben die erwartete Anzahl der Evolutionsschritte an (CC-SA 3.0, Gagea, via Wikimedia Commons)

Beispiel Entwicklung des menschlichen Gehirns.

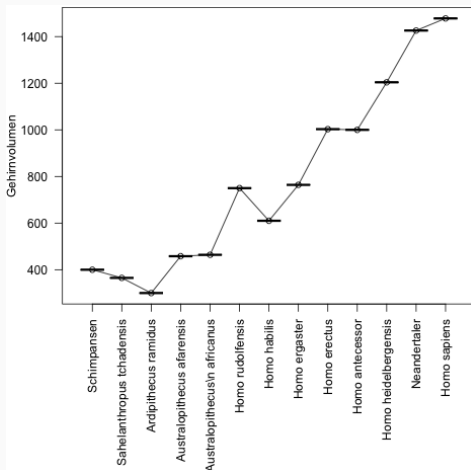


Abbildung 3: Entwicklung der Größe des Gehirns (in cm^3). Ordnung der Spezies entspricht ungefähr ihrer zeitgeschichtlichen Abfolge.

Beispiel Sichelzellenanämie.

- Mutation im Genom führt bei einigen Menschen zu Missbildung roter Blutkörperchen (Sichelzellen)
- Sichelzellen führen zu verstopfen Blutgefäßen
- Lebensbedrohlich, wenn die Gene von Vater und Mutter kommen (homozygot)
- Reine Träger des Sichelzellengens (heterozygot) sind in Gegenden mit Malaria überdurchschnittlich häufig, heterozygote Träger sind immun gegen Malaria
- Dort, wo die Mutation ein Überlebensvorteil ist, wurde sie nicht evolutionär ausselektiert

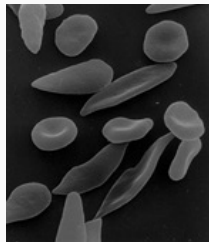


Abbildung 4:
(Public Domain, NIDDK
via Wikimedia
Commons)

- Evolutionäre Entwicklung von Lebewesen führt zu optimaler Anpassung an den Lebensraum
- Evolutionäre Algorithmen verwenden die Mechanismen der biologischen Evolution:
 - Mutation,
 - Rekombination und
 - Selektion,um technische Optimierungsprobleme zu lösen.

- Ursprung genetischer Algorithmen in den 1950/60er Jahren
- 1965 erster Algorithmus „Evolutionstrategie“ von Ingo Rechenberg (TU Berlin)
 - Erste Optimierungsansätze für technische Bauteile, z.B. Tragflächen¹
- 1966 Idee der Evolutionären Programmierung von Fogel, Owens, and Walsh
 - Mögliche Lösungen eines Problems werden als endliche Automaten repräsentiert
 - Automaten entwickeln sich, indem ihre Zustandsdiagramme mutiert (geändert werden)
 - Der am besten entwickelte (bezüglich einer Fitnessfunktion) Automat wird zur Lösung herangezogen

¹Youtube Video Sternstunden der Theorie der Evolutionstrategie – Prof. Dr. Ingo Rechenberg https://www.youtube.com/watch?v=RhJ7s_LRLAM (56:51 min)

- 1960 entwickelte John Holland den ersten genetischen Algorithmus
 - Beschreibung der Gruppe von Algorithmen und theoretisches Framework
 - Intention war die Studie der Adaptivität von biologischen Systemen (nicht wie in der Evolutionsstrategie und bei Evolutionärer Programmierung die Lösung von Problemen)
- Heute lassen sich die ursprünglichen Forschungsrichtungen nicht mehr scharf trennen
 - Zusammenfassung unter dem Überbegriff „Evolutionäre Algorithmen“ („Genetische Algorithmen“ wird synonym verwendet)

Biologische Grundlagen

EVOLUTIONSTHEORIE NACH DARWIN

- Britischer Naturforscher, Zoologe, Geologe, ...
- 1831 Weltreise mit der HMS Beagle, dabei wissenschaftliche Untersuchung von Plankton, Fossilien, Tieren, Pflanzen,..
- 1838 Theorie über die Anpassung der Arten durch Variation und natürliche Selektion
- 1859 Hauptwerk „On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life“
- Evolutionstheorie nach Darwin: „Darwinismus“

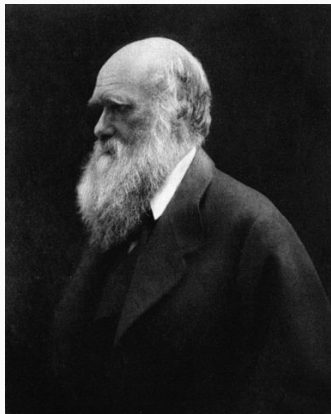


Abbildung 5: Charles Darwin
(Public Domain, Julia Margaret Cameron via Wikimedia Commons)

EVOLUTIONSTHEORIE NACH DARWIN

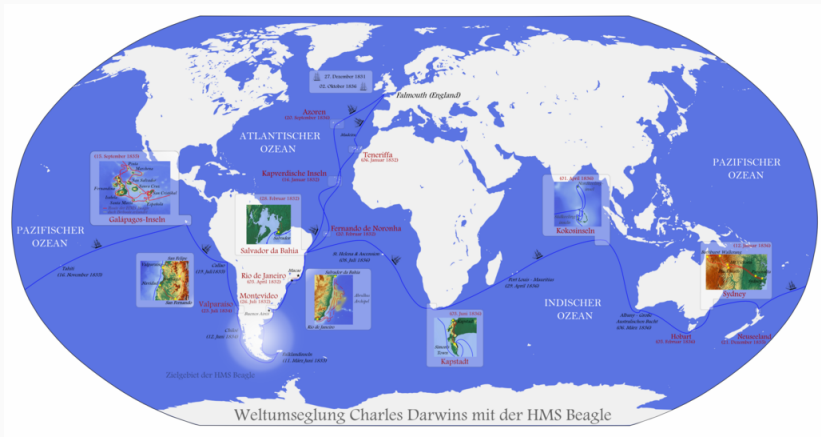


Abbildung 6: Stationen von Darwins Weltumsegelung (Public Domain, JDevil_m25 via Wikimedia Commons)

GRUNDLEGENDE BEGRIFFE

- Die Erbinformation von Lebewesen ist auf der **DNA** gespeichert
- Der gesamte Molekülstrang der DNA heißt **Chromosom**
- Die Information in der DNA ist in der Sequenz von **Nukleinsäuren** gespeichert
- Eine Sequenz von Nukleinsäuren, die zur Ausprägung eines bestimmten Merkmals führt, heißt **Gen**
- Gene auf verschiedenen Chromosomen, die zur Ausbildung desselben Merkmals führen, heißen **Allele** (z.B. Allel zur Ausprägung der Augenfarbe)
- Die Gesamtheit der DNA eines Lebewesens (alle Chromosomen) wird **Genotyp** genannt
- Das Erscheinungsbild eines Lebewesen wird **Phänotyp** genannt
- Unterschiedliche Genotypen können zum gleichen Phänotypen führen, aber nicht umgekehrt

GRUNDLEGENDE BEGRIFFE

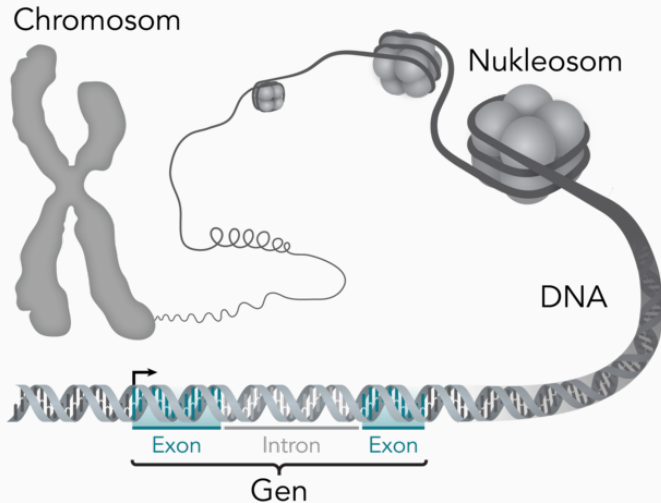


Abbildung 7: Chromosom, DNA, Gen (CC-BY 4.0, Thomas Splettstoesser via Wikimedia Commons)

Evolutionstheorie nach Darwin

Durch Variation im Erbgut und Selektion der fitesten Nachkommen entwickeln sich Arten, die besser an ihren Lebensraum angepasst sind („survival of the fittest“). Annahmen:

- Vererbung: Erbinformation der Eltern wird auf ihre Nachkommen übertragen
 - Variation: Lebewesen haben Nachkommen, die sich leicht von der Elterngeneration unterscheiden
 - Selektion: Besser an den Lebensraum angepasste Individuum überleben mit höherer Wahrscheinlichkeit und haben mehr Nachkommen
-
- Variationen im Genotyp entstehen durch **Rekombination**, und **Mutation** und nur die besten Varianten (bezüglich einer Fitnessfunktion) werden weiter vererbt (**Selektion**)

REKOMBINATION/CROSS-OVER

- Neukombination der Elternchromosomen
- Mensch hat 23 Chromosomenpaare (d.h. 46 Chromosomen), bei 2 Eltern ergeben sich $2^{23} \cdot 2^{23} = 2^{46} \approx 70$ Billionen Möglichkeiten
- Dieser Wert entsteht, da bei jedem Elternteil zuerst der Chromosomensatz getrennt wird (2^{23} Möglichkeiten) und dann die Chromosomen beider Eltern kombiniert werden (2^{23} Möglichkeiten)

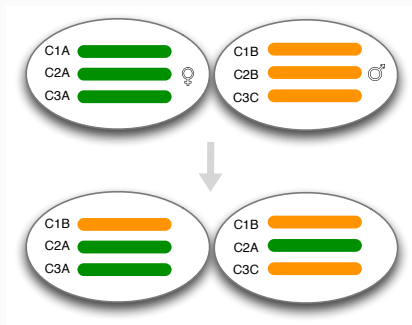


Abbildung 8: Interchromosomale Rekombination

REKOMBINATION/CROSS-OVER

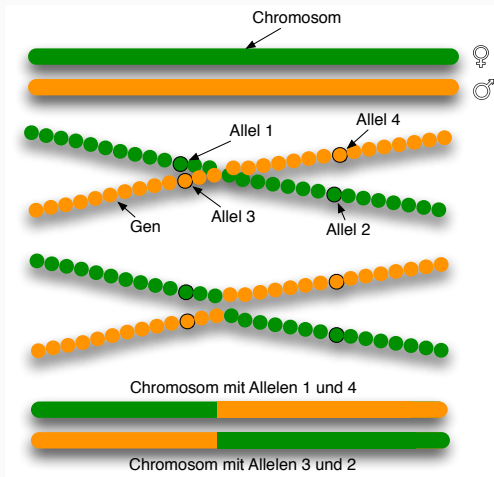
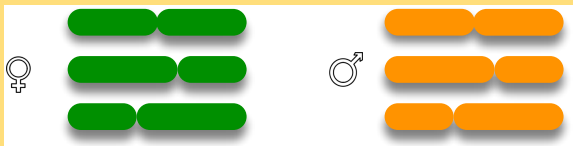


Abbildung 9: Intrachromosomale Rekombination

- Neukombination der Gene von Elternchromosomen
- Abschnitte mit mehreren Genen werden gegeneinander ausgetauscht
- Kann nicht nur einfach wie in der Abbildung, sondern mehrfach bei zwei Chromosomen auftreten

AUFGABE

Gegeben seien folgende Chromosomensätze:



1. Geben Sie einen möglichen Chromosomensatz eines Nachkommens an, der durch inter- und intrachromosomale Rekombination entsteht.
2. Wieviel Möglichkeiten gibt es für interchromosomale Rekombination der Chromosomensätze?

Lösung für 2) $2^3 - 2 = 6$

MUTATION

- Löschen, Einfügen, Verdoppeln, Invertieren oder Verschieben von DNA-Sequenzen
- Betrifft DNA-Sequenzen, keine vollständigen Gene, wie bei der Rekombination
- Mutation kann, muss aber nicht zu funktionalen Änderungen führen (stille Mutationen)

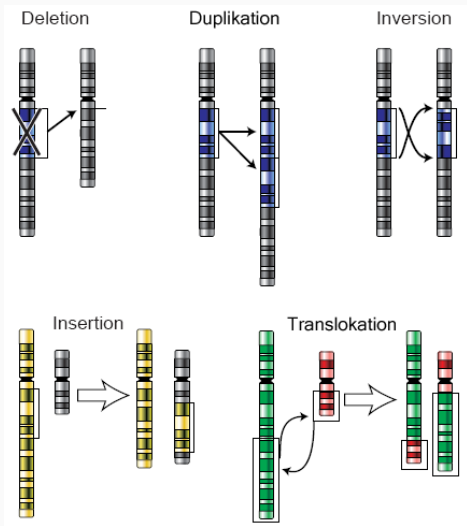


Abbildung 10: Mutationstypen (Public Domain, Yassine Mrabett via Wikimedia Commons)

Gegeben Sei folgendes Chromosom:

A T C A T G A

Geben Sie jeweils ein Beispiel für

1. Deletion
2. Duplikation
3. Inversion
4. Insertion und
5. Translokation

an. Gesucht ist das resultierende Chromosom nach der entsprechenden Mutation.

- Unter der Fitness eines Individuums (und damit dessen Genotyps) versteht man das Ausmaß seiner Angepasstheit an die Umwelt²
- Fitness ist messbar über
 - Wahrscheinlichkeit des Individuums, das fortpflanzungsfähige Alter zu erreichen, und
 - Anzahl der Nachkommen
- Fitness ist abhängig von Umweltbedingungen, ein Individuum mit hohem Fitnesswert in einem Lebensraum kann einen niedrigen Fitnesswert in einem anderen Lebensraum haben
- Individuen mit niedriger Fitness überleben weniger wahrscheinlich bis sie fortpflanzungsfähig sind und haben weniger Nachkommen
- Diese Prozess nennt man genetische Selektion

²Survival of the fittest ist somit nicht das Überleben der Stärkeren, sondern das Überleben der besser Angepassten

Grundprinzipien

- Eine Fitness-Landschaft ist eine Repräsentation von Genotypen und ihrer Fitness.
- Angenommen, jeder Genotyp ist ein String der Länge l .
- Die Distanz zwischen 2 Genotypen kann über die Hamming-Distanz berechnet werden

$$d_H(x, y) = \left| \{i = 1, \dots, l \mid x_i \neq y_i\} \right|$$

- Beispiele Hammingdistanz:

String s_1	String s_2	$d_H(s_1, s_2)$
00000	11111	5
ATCTGGC	ATCCGGC	1
Mensch	Moench	3

- Damit lässt sich der Unterschied zwischen zwei Genotypen quantifizieren.

Berechnen Sie die Hamming-Distanz folgender String-Paare:

000110 - 011110

Complex - Komplex

Komplex - Komplexe

Lösung: 2, 1, nicht definiert (Strings haben nicht die gleiche Länge)

FITNESS-LANDSCHAFT

- Eine Fitness-Landschaft ist eine Funktion im $(l+1)$ -dimensionalen Raum (l ist die Länge der Strings), wobei der Genotyp durch die ersten l Dimensionen festgelegt ist und die $(l+1)$ -ste Dimension dem Wert der Fitnessfunktion entspricht.

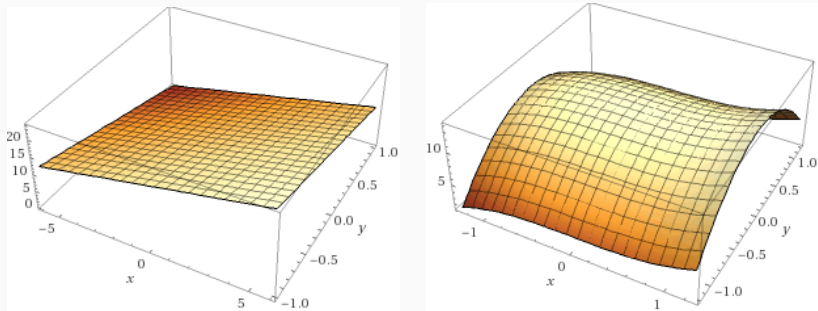


Abbildung 11: Beispiele von Fitness-Landschaften für 2-dimensionale Genotypen, hier Funktionen $z = x^3 - 6y^2 + 12$ und $z = x - 6y + 12$; x und y repräsentieren den Genotyp, z den Fitnesswert (erstellt mit wolfram alpha)

FITNESS-LANDSCHAFT

- Fitnessfunktionen können mehrere lokale Optima besitzen
- Suche nach dem optimalen Genotyp entspricht der Suche nach dem globalen Optimum der Fitnessfunktion.

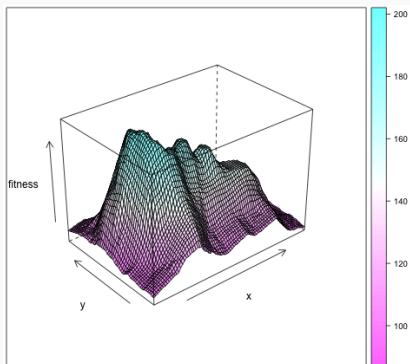


Abbildung 12: Beispiel für eine Fitnessfunktion mit mehreren lokalen Maxima

Gerüst eines Evolutionären Algorithmus

1. Beginne mit einer Anfangspopulation von n zufällig generierten Chromosomen
2. Berechne die Fitness $f(x)$ von jedem Chromosom x in der Population [**Fitness**].
3. Wiederhole, solange bis eine Population mit n Nachkommen erzeugt wurden
 - 3.1 Wähle zwei Eltern entsprechend ihrer Fitness [**Selektion**].
 - 3.2 Mit Wahrscheinlichkeit p_c rekombiniere die Eltern. Wenn Eltern nicht rekombiniert werden sind Kinder reine Kopien [**Cross-Over**].
 - 3.3 Mit Wahrscheinlichkeit p_m mutiere Kinder an jeder Stelle [**Mutation**].
 - 3.4 Füge Kinder in die Population ein.
4. Verwende neue Population für die nächste Iteration.
5. Wenn Abbruchbedingung nicht erfüllt ist, gehe zu Schritt 2.

Parameter

- Populationsgröße n
 - Bei kleinen n ist die genetische Variation in der Population gering, bei großen n ist die Laufzeit des Algorithmus lang.
 - Sehr große Populationen bringen keinen Vorteil mehr.
 - Sinnvolle Populationsgröße hängt vom speziellen Problem ab.
- Mutationswahrscheinlichkeit p_m
 - Hilft aus lokalen Minima
 - Für $p_m = 1$ werden alle Stellen in allen Chromosomen verändert. Das entspricht zufälliger Suche im Raum der Fitnessfunktion.
 - Sinnvolle Werte: 0, 01, 0, 001
- Rekombinationswahrscheinlichkeit p_c
 - Für $p_c = 1$ sind alle Kinder Kombinationen, für $p_c = 0$ werden alle Eltern übernommen.
 - Es erweist sich als sinnvoll, einige Chromosomen der Elterngeneration zu übernehmen.

KODIERUNG DER CHROMOSOMEN

- Binärcodierung: Jedes Chromosom wird als Bitstring kodiert.
 - Einfache Anwendung der Operationen Mutation und Rekombination.
 - Tendenz zu langen Chromosomen mit vielen Allelen.

0	1	0	1	0	1	1	0	0
---	---	---	---	---	---	---	---	---

- Kodierung als Vektoren reeller Zahlen: Jedes Chromosom wird als Vektor reeller Zahlen kodiert.
 - Einfache Anwendung der Operation Rekombination.

1.2	-2	0.6	2.4	0.2	1.1
-----	----	-----	-----	-----	-----

- Weitere Kodierungen möglich: Bäume, Permutationen, ...

- Selektion nach Fitnesswert:
 - Die Wahrscheinlichkeit der Selektion $p_s(x)$ eines Chromosoms x ist proportional zu dessen Fitnesswert $f(x)$.
 - Zum Beispiel $p_s(x) = \frac{f(x)}{\sum_x f(x)}$
- Selektion nach Rang des Fitnesswertes:
 - Fitnesswerte $f(x)$ werden sortiert und jedem Wert wird ein Rang $r(x)$ zugewiesen (das beste Rang 1, das zweitbeste Rang 2 usw.). Die Selektionswahrscheinlichkeit ist dann proportional zum Rang.
 - Zum Beispiel $p_s(x) = \frac{r(x)}{\sum_x r(x)}$
 - Sinnvoll, wenn die Fitness des besten Chromosoms sehr hoch ist (z.B. 80% der Gesamtsumme der Fitnesswerte). Hier haben auch Chromosomen mit weniger Fitness eine Chance, selektiert zu werden.
- Elitismus:
 - Die besten k Chromosomen einer Population werden ohne Änderung in die Nachfolgepopulation übernommen.
 - Idee ist, die besten vorhandenen Lösungen nicht zu verlieren.

REKOMBINATION

- 1-Punkt Cross-Over: Ein Index im Bitstring wird ausgewählt. Alle Bits hinter diesem Index werden getauscht, um die Kinder zu generieren.



- 2-Punkt Cross-Over: Zwei Indizes im Bitstring werden ausgewählt. Bits werden alternierend von den Eltern kopiert.



- Zufälliger Cross-Over: Mit Wahrscheinlichkeit p (z.B. $p = 0,5$) werden Bits von Elternteil 1 oder 2 übernommen.



- Für reellwertige Vektoren funktioniert Rekombination analog.

MUTATION

- Bitstring-Mutation: Jedes Bit wird mit Wahrscheinlichkeit p_m (z.B. $p = 0,01$) gekippt.



- Mutation reellwertiger Vektoren: Zu jedem Bit wird mit Wahrscheinlichkeit p_m (z.B. $p = 0,01$) ein kleiner Wert (z.B. 0,01) addiert oder subtrahiert.



Gegeben sind folgende Chromosomen:

C1: 0.4 0.5 0.6 0.7 0.8

C2: 0.0 0.1 0.2 0.3 0.9

Geben Sie jeweils ein Beispiel für

- 1-Punkt Crossover
- 2-Punkt Crossover
- Zufälligen Crossover
- Mutation

an.

Es gibt viele Variationen für die Implementierung des allgemeinen Algorithmus von Folie 25, die sich u.a. in folgendem unterscheiden:

- Wahl der Startpopulation (zufällig oder mit Vorwissen)
- Wahl der Parameter
- Wahl der Operationen (Form der Rekombination oder Kombinationen davon, Selektionsfunktion)
- Wahl der Fitnessfunktion
- Wahl des Abbruchkriteriums, z.B.
 - Bestimmte Anzahl von Iterationen
 - Die Differenz der Fitnesswerte der besten Lösung ändert sich nicht mehr.

Herausforderung ist außerdem die Wahl der Kodierung des Problems.

Beispielanwendungen

DAS PROBLEM DES HANDLUNGSREISENDEN

- Das Problem des Handlungsreisenden (auch Travelling Salesperson Problem, kurz TSP) sucht nach einer Rundreise durch Städte, sodass jede Stadt nur einmal besucht wird und die zurück gelegte Wegstrecke minimal wird.
- Das Problem ist NP-vollständig³
- Es lässt sich somit nicht in polynomieller Zeit optimal lösen und gehört zu den schwierigsten Problemen der Informatik
- Anwendungsprobleme, denen TSP zugrunde liegt, sind z.B.
 - Chip-Design (Optimierung der Verbindungen zwischen Schaltkreisen)
 - Auslieferung von Paketen durch Paketdienste
 - Besuche von Kunden eines Serviceunternehmens
 - Genomsequenzierung (DNA-Teilstränge sind „Städte“, Ähnlichkeit zwischen Teilsequenzen entspricht der Entfernung)

³Unter der unbewiesenen Annahme, dass $P \neq NP$

PROBLEM DES HANDLUNGSREISENDEN

Beispielproblem: Rundreise durch Europa

Gegeben sind die Entfernungen zwischen 21 europäischen Städten⁴

	Athens	Barcelona	Brussels	Calais	Cherbourg	Cologne	Copenhagen	...
Athens	0	3313	2963	3175	3339	2762	3276	
Barcelona	3313	0	1318	1326	1294	1498	2218	
Brussels	2963	1318	0	204	583	206	966	
Calais	3175	1326	204	0	460	409	1136	
Cherbourg	3339	1294	583	460	0	785	1545	
Cologne	2762	1498	206	409	785	0	760	
Copenhagen	3276	2218	966	1136	1545	760	0	
Geneva	2610	803	677	747	853	1662	1418	
Gibraltar	4485	1172	2256	2224	2047	2436	3196	
Hamburg	2977	2018	597	714	1115	460	460	
Hook of Holland	3030	1490	172	330	731	269	269	
Lisbon	4532	1305	2084	2052	1827	2290	2971	
Lyons	2753	645	690	739	789	714	1458	
Madrid	3949	636	1558	1550	1347	1764	2498	
Marseilles	2865	521	1011	1059	1101	1035	1778	
Milan	2282	1014	925	1077	1209	911	1537	
Munich	2179	1365	747	977	1160	583	1104	
Paris	3000	1033	285	280	340	465	1176	
Rome	817	1460	1511	1662	1794	1497	2050	
Stockholm	3927	2868	1616	1786	2196	1403	650	
Vienna	1991	1802	1175	1381	1588	937	1455	

⁴Dataset available in R: `data('eurodist', package = 'datasets')`

Kodierung

Permutation

Eine Permutation einer Menge von Objekten ist eine Anordnung aller Objekte in einer bestimmten Reihenfolge.

- Beispiele:
 - Permutationen der Menge $\{1, 2\}$ sind $(1, 2)$ und $(2, 1)$
 - Permutationen der Menge $\{A, A, N, N\}$ sind z.B. (A, N, N, A) und (N, A, A, N)
- Für eine n -elementige Menge gibt es $n \cdot (n - 1) \cdot \dots \cdot 1 = n!$ Permutationen
- Kodierung der Rundreise als Permutation
 - Jede der n Städte bekommt eine natürliche Zahl zwischen 1 und n zugeordnet (z.B. ihren Spaltenindex in der Tabelle)
 - Eine Rundreise ist dann eine Permutation der Menge der ersten n natürlichen Zahlen

PROBLEM DES HANDLUNGSREISENDEN

Operationen auf Chromosomen, die Permutationen repräsentieren

- Rekombination: Alle Werte bis zum Cross-Over-Punkt werden vom ersten Elternteil übernommen. Die restlichen Werte werden vom zweiten Elternteil kopiert, in der Reihenfolgen, wie sie dort vorkommen. Bereits vom ersten Elternteil übernommene Werte werden dabei übersprungen.



- Mutation: Zwei Indizes werden ausgewählt und die entsprechenden Werte werden vertauscht.



Fitnessfunktion

- Die Gesamtlänge der Reise soll minimiert werden
- Da die verwendete Bibliothek [3, 4]⁵ nach dem globalen Maximum sucht, wird als Fitnessfunktion $\frac{1}{l}$ verwendet, wobei l die Gesamtwegstrecke ist

Parameter

Type	=	permutation
Population size	=	10
Number of generations	=	500
Elitism	=	1 (1 Chromosom wird übernommen)
Crossover probability	=	0.8
Mutation probability	=	0.2

⁵Code adapted from <http://rpubs.com/somasdhavala/GAeg>

Ergebnis

- Länge der Rundreise: 12919 km
- Iterationen bis Konvergenz: 248
- Beste gefundene Tour als Permutation (Index der Städte):
10 20 7 11 3 4 18 5 14 12 9 ->
2 15 13 8 16 19 1 21 17 6
- Nach Umkodieren der Indizes ergibt sich als beste Tour:
Hamburg → Stockholm → Copenhagen → Hook of Holland →
Brussels → ... → Hamburg

PROBLEM DES HANDLUNGSREISENDEN

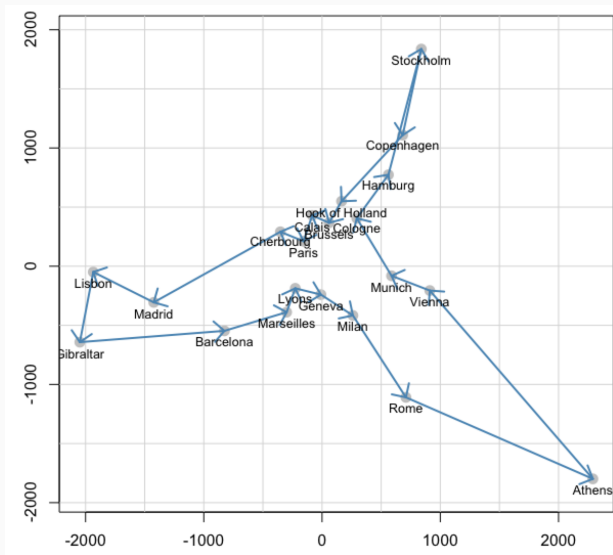


Abbildung 13: Beste Tour

PROBLEM DES HANDLUNGSREISENDEN

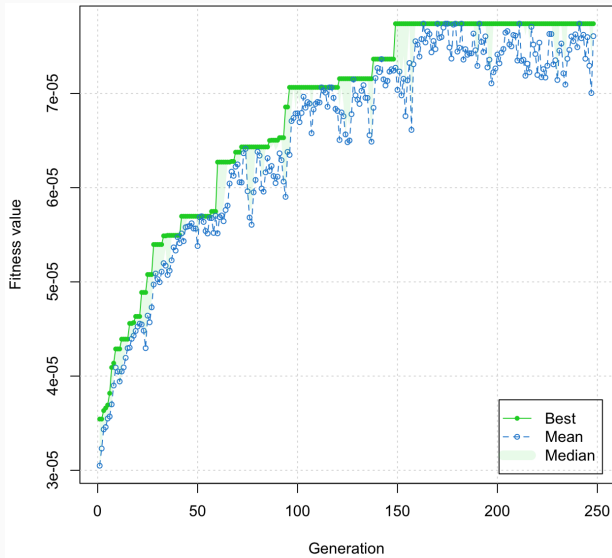


Abbildung 14: Entwicklung der Population

Gegeben seien die Dauer der Autofahrten zwischen folgenden Städten:

- Graz - Chemnitz: 7:30h
 - Graz - Passau: 3:15h
 - Chemnitz - Passau: 4:00
 - Prag - Passau: 3:00
 - Prag - Chemnitz: 2:00
 - Prag - Graz: 5:30
 - Wien - Graz: 2:00
 - Wien - Prag: 3:40
 - Wien - Chemnitz: 6:00
 - Wien - Passau: 3:00
1. Finden Sie eine Kodierung für die Rundreise: Wien - Prag - Passau - Graz - Chemnitz - Wien. Wie ist dann die Reise Graz - Wien - Prag - Chemnitz - Passau - Graz kodiert?
 2. Was ist der Fitnesswert beider Rundreisen?

RUCKSACKPROBLEM

- Kombinatorisches Optimierungsproblem
- Gegeben ein Rucksack mit einem maximalem zulässigen Gesamtgewicht θ
- Gegeben Objekte x_i , die jeweils ein Gewicht w_i und einen Wert v_i besitzen
- Gesucht ist die optimale Packung des Rucksack, sodass
 - θ nicht überschritten wird,
 - Die Summe der gepackten Werte maximal ist.
- Gehört zu den schwierigsten Problemen der Informatik (NP-vollständig)

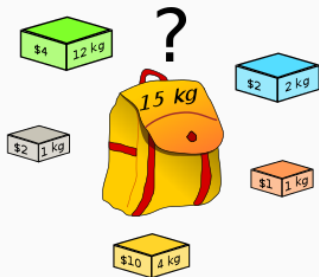


Abbildung 15: Illustration des Rucksackproblems (CC-SA 2.5, Dake, via Wikimedia Commons)

RUCKSACKPROBLEM

Beispielproblem, Wanderrucksack mit Notfallausrüstung

Objekt	Gewicht	Wert
Wasser	6	10
Äpfel	4	4
Müsliriegel	1	5
Taschenmesser	1	4
Taschenlampe	3	1
Kloppapier	2	2
Regenjacke	2	1
Reiseapotheke	5	5
Gameboy	2	4
Socken	1	2
Brot	3	5

Summe der Gewichte = 30, Summe der Werte = 43

Annahme: Maximal tragbares Gewicht = 20 ($\theta = 20$)

Kodierung

- Als Binärvektor/Bitstring
- Der Index des Bits entspricht einem bestimmten Objekt
- $x[i] = 1$ heißt, das Objekt i wird in den Rucksack übernommen

Index i	Objekt	w_i	v_i
1	Wasser	6	10
2	Äpfel	4	4
3	Müsliriegel	1	5
4	Taschenmesser	1	4
5	Taschenlampe	3	1
6	Klopapier	2	2
7	Regenjacke	2	1
8	Reiseapotheke	5	5
9	Gameboy	2	4
10	Socken	1	2
11	Brot	3	5

- Chromosom $x = 00011010100$ bedeutet also: im Rucksack befinden sich Taschenmesser, Taschenlampe, Regenjacke und Gameboy

Fitnessfunktion

- Der Wert des Rucksacks mit den ausgewählten Objekten, bzw. 0, wenn das Maximalgewicht θ überschritten wird.

$$f(x) = \begin{cases} \sum_i v_i x_i & \text{wenn } \sum_i w_i x_i \leq \theta \\ 0 & \text{sonst} \end{cases}$$

AUFGABE

Geben Sie den Wert der Fitnessfunktion für Chromosome 00011010100 und 11001001001 an.

Index i	Objekt	w_i	v_i
1	Wasser	6	10
2	Äpfel	4	4
3	Müsliriegel	1	5
4	Taschenmesser	1	4
5	Taschenlampe	3	1
6	Klopapier	2	2
7	Regenjacke	2	1
8	Reiseapotheke	5	5
9	Gameboy	2	4
10	Socken	1	2
11	Brot	3	5

- $f(00011010100) = 4 + 1 + 1 + 4 = 10$
- $f(11001001001) = 0$ (da das Gewicht der Objekte $6 + 4 + 5 + 3 + 3 = 21$ ist und den Schwellwert übersteigt)

Fitnessfunktion

- Der Wert des Rucksacks mit den ausgewählten Objekten, bzw. 0, wenn das Maximalgewicht θ überschritten wird.

$$f(x) = \begin{cases} \sum_i v_i x_i & \text{wenn } \sum_i w_i x_i \leq \theta \\ 0 & \text{sonst} \end{cases}$$

- Fitness von Chromosom $x = 000110100100$
 $f(000110100100) = 4 + 1 + 1 + 4 = 10$
- Fitness von Chromosom $x = 11001001001$
 $f(11001001001) = 0$ (da das Gewicht der Objekte $6 + 4 + 5 + 3 + 3 = 21$ ist und den Schwellwert übersteigt)

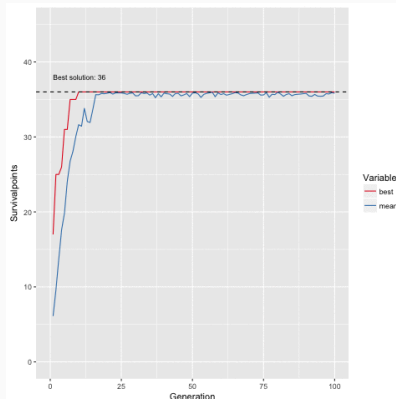
Parameter

- Populationsgröße: $n = 100$
- Maximale Anzahl Iterationen: $k = 100$
- Elitismus: $e = 0,3$ (30% der besten Lösungen werden in die nächste Generation übernommen)
- Wahrscheinlichkeit der Rekombination: $p_c = 0$ (keine Rekombination)
- Wahrscheinlichkeit von Mutation: $p_m \in \{0,01; 0,5\}$

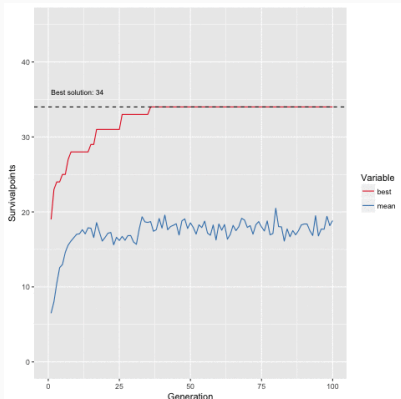
RUCKSACKPROBLEM

Ergebnisse (erstellt mit Code von

<https://www.r-bloggers.com/genetic-algorithms-a-simple-r-example/>)



$n = 100$; $k = 100$; $p_c = 0$; $p_m = 0,01$; $e = 0,3$
Beste Lösung (1 1 1 1 0 1 0 0 1 1 1)
Wasser, Äpfel, Müsliriegel, Taschenmesser,
Klopapier, Gameboy, Socken, Brot
Fitnesswert 36



$n = 100$; $k = 100$; $p_c = 0$; $p_m = 0,05$; $e = 0,3$
Beste Lösung (1 1 1 1 0 0 0 1 1 1 0)
Wasser, Äpfel, Müsliriegel, Taschenmesser,
Reiseapotheke, Gameboy, Socken
Fitnesswert 34

- Vorhersage von Proteinstrukturen auf Grundlage ihrer Aminosäuresequenz
- Entwicklung von Architekturen für Künstliche Neuronale Netze
- Design von künstlichen Strukturen, die auf Menschen natürlich wirken (Film- und Spieleentwicklung)
- Optimierung von Sensornetzwerken und Infrastrukturen
- Kombination mit Spieltheorie als evolutionären Spieltheorie

Zusammenfassung

- Evolutionäre Algorithmen (EA) sind von den Prinzipien der Evolution motiviert.
- EA werden zur Lösung von kombinatorischen Optimierungsproblemen eingesetzt (z.B. Problem des Rundreisenden, Rucksackproblem).
- EA haben eine hohe Laufzeit (Iteration durch die gesamte Population über mehrere Generationen hinweg) sind aber leicht terminierbar mit hoher (aber nicht notwendigerweise optimaler) Lösungsgüte.
- EA sind leicht parallelisierbar, der Lösungsraum wird parallel abgesucht, damit ist es weniger wahrscheinlich, in lokalen Maxima stecken zu bleiben.

Wichtige Konzepte

- Evolutionstheorie
 - Gen, Chromosom, Allel, Genotyp, Phänotyp, Fitness
 - Vererbung, Variation, natürliche Selektion
- Fitnessfunktion, Fitness-Landschaft
- Evolutionärer Algorithmus
 - Mutation
 - Selektion (nach Wert, nach Rang, Elitismus)
 - Rekombination (1-Punkt, n-Punkt, zufälliger Cross-Over)
 - Mutationswahrscheinlichkeit, Rekombinationswahrscheinlichkeit

- An Introduction to Genetic Algorithms
(PDF Version online verfügbar)
Melanie Mitchell. *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998. ISBN: 0262631857. URL: <https://svn-d1.mpi-inf.mpg.de/AG1/MultiCoreLab/papers/ebook-fuzzy-mitchell-99.pdf>
- TED Talk by Keith Downing „Evolutionary computation“<https://www.youtube.com/watch?v=D3zUmfDd79s>

Literatur

- [1] Melanie Mitchell. *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: MIT Press, 1998. ISBN: 0262631857. URL: <https://svn-dl.mpi-inf.mpg.de/AG1/MultiCoreLab/papers/ebook-fuzzy-mitchell-99.pdf>.
- [2] Dan-E. Nilsson und Susanne Pelger. "A Pessimistic Estimate of the Time Required for an Eye to Evolve". In: *Proceedings of the Royal Society of London B: Biological Sciences* 256.1345 (1994), S. 53–58. ISSN: 0962-8452. DOI: 10.1098/rspb.1994.0048. URL: <http://www.rpgroup.caltech.edu/courses/aph161/Handouts/Nilsson1994.pdf>.
- [3] Luca Scrucca. "GA: A Package for Genetic Algorithms in R". In: *Journal of Statistical Software* 53.4 (2013), S. 1–37. URL: <http://www.jstatsoft.org/v53/i04/>.
- [4] Luca Scrucca. "On some extensions to GA package: hybrid optimisation, parallelisation and islands evolution". In: *Submitted to R Journal* (2016). Pre-print available at arXiv. URL: <http://arxiv.org/abs/1605.01931>.