

Künstliche Intelligenz

Vorlesung 4: Evolutionäre Algorithmen



LITERATUR LISTE

-  Bäck, T. and Schwefel, H. (1993).
An overview of evolutionary algorithms for parameter optimization.
Evolutionary Computation, 1(1):1–23.
-  Banzhaf, W., Nordin, P., Keller, R. E., and Francone, F. D. (1998).
Genetic Programming — An Introduction: On the Automatic Evolution of Computer Programs and Its Applications.
Morgan Kaufmann Publisher, Inc. and dpunkt-Verlag, San Francisco, CA, USA
and Heidelberg, Germany.
-  Darwin, C. (1859).
On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life.
John Murray, London, United Kingdom.

LITERATUR LISTE

-  Dawkins, R. (1986).
The Blind Watchmaker.
Norton, New York, NY, USA.
-  Dawkins, R. (1989).
The Selfish Gene.
Oxford University Press, United Kingdom, 2nd edition.
-  Dawkins, R. (1990).
Der blinde Uhrmacher: ein neues Plädoyer für den Darwinismus.
Deutscher Taschenbuch-Verlag, Munich, Germany.
-  Dawkins, R. (1998).
Das egoistische Gen.
Rowohlt, Reinbek bei Hamburg, Germany.

LITERATUR LISTE

-  Dorigo, M. and Stützle, T. (2004).
Ant Colony Optimization.
MIT Press, Cambridge, MA, USA.
-  Gerdes, I., Klawonn, F., and Kruse, R. (2004).
Evolutionäre Algorithmen.
Vieweg, Wiesbaden, Germany.
-  Michalewicz, Z. (1996).
Genetic Algorithms + Data Structures = Evolution Programs.
Springer-Verlag, New York, NY, USA, 3rd (extended) edition.
-  Nissen, V. (1997).
Einführung in evolutionäre Algorithmen: Optimierung nach dem Vorbild der Evolution.
Vieweg, Braunschweig/Wiesbaden, Germany.



LITERATUR LISTE



Vollmer, G. (1995).

Der wissenschaftstheoretische status der evolutionstheorie. einwände und gegenargumente.

In Vollmer, G., editor, *Biophilosophie*, page 92–106. Reclam, Stuttgart, Germany.



Weicker, K. (2007).

Evolutionäre Algorithmen.

Teubner Verlag, Stuttgart, Germany, 2nd edition.



COMPUTATIONAL INTELLIGENCE

- Evolutionäre Algorithmen
- meist **modellfreie** Ansätze
- Approximation statt exakte Lösung
- schnelles Finden einer brauchbaren Lösung



OPTIMIERUNGSPROBLEME

Definition (Optimierungsproblem)

Ein *Optimierungsproblem* (Ω, f, \succ) ist gegeben durch einen Suchraum Ω , eine Bewertungsfunktion $f : \Omega \rightarrow \mathbb{R}$, die jedem Lösungskandidaten einen Güterwert zuweist, sowie einer Vergleichsrelation $\succ \in \{<, >\}$. Dann ist die *Menge der globalen Optima* $\mathcal{H} \subseteq \Omega$ definiert als

$$\mathcal{H} = \{x \in \Omega \mid \forall x' \in \Omega : f(x) \succeq f(x')\}.$$

- gegeben: ein Optimierungsproblem (Ω, f, \succ)
- gesucht: ein Element $x \in \Omega$, das die Funktion f (global) optimiert

Evolutionäre Algorithmen werden oft zum Lösen von Optimierungsproblemen genutzt.



PRINZIPIELLE LÖSUNGSANSÄTZE

analytische Lösung:

- sehr effizient, aber nur seltenen anwendbar

vollständige Durchforstung:

- sehr ineffizient, daher nur bei sehr kleinen Suchräumen anwendbar

blinde Zufallssuche:

- immer anwendbar, aber meist sehr ineffizient

gesteuerte Suche:

- Voraussetzung: Funktionswerte ähnlicher Elemente aus Suchraum Ω sind sich ähnlich



ANWENDUNGSGEBIETE

Parameteroptimierung

- z.B. Krümmung von Rohren für minimalen Widerstand
- allgemein: Finden eines Parametersatzes, sodaß gegebene reellwertige Funktion ein (möglichst globales) Optimum annimmt

Packprobleme

- z.B. Füllen eines Rucksacks mit maximalem Wert
- oder Packen möglichst weniger Kisten mit gegebenen Gütern

Wegeprobleme

- z.B. Problem des Handlungsreisenden (z.B. Bohren von Platinen)
- Reihenfolge anzufahrender Ziele, Fahrtroutenoptimierung, Verlegen von Leiterbahnen auf Platinen/integrierten Schaltkreisen



ANWENDUNGSGEBIETE

Anordnungsprobleme

- z.B. Steinerproblem (engl. facility allocation problem):
- Positionierung von Verteilerknoten z.B. in einem Telefonnetz

Planungsprobleme

- z.B. Ablaufpläne (Scheduling), Arbeitspläne, Operationenfolgen
- (auch Optimierung in Compilern – Umordnung der Befehle)

Strategieprobleme

- z.B. Gefangenendilemma und andere Modelle der Spieltheorie
- Verhaltensmodellierung von Akteuren im Wirtschaftsleben

biologische Modellbildung

- z.B. Netspinner (beschreibende Regeln zum Spinnennetz-Bau)
- EA optimiert Parameter, Vergleich mit Realität \Rightarrow gutes Modell



MOTIVATION

- EAs basieren auf biolog. Evolutionstheorie.
- grundsätzliches Prinzip:
 - Durch zufällige Variation entstehende, vorteilhafte Eigenschaften werden durch natürliche Auslese ausgewählt.
 - Individuen mit vorteilhaften Eigenschaften haben bessere Fortpflanzungs- und Vermehrungschancen - **differentielle Reproduktion**
- Evolutionstheorie erklärt Vielfalt und Komplexität der Lebewesen
- erlaubt Vereinigung aller Disziplinen der Biologie



NATÜRLICHE EVOLUTION

- Grundverständnis für die Zusammenhänge und die Komplexität der natürlichen Evolution – mit dem Ziel deren Nachahmung durch die evolutionären Algorithmen zu verstehen.
- Die Evolutionsfaktoren werden in ihrer grundsätzlichen Arbeitsweise verstanden.
- In einem ersten Abstraktionsschritt können Vorgänge der natürlichen Evolution simuliert werden.



NATÜRLICHE EVOLUTION ALS VORBILD FÜR OPTIMIERUNGSPROBLEME

- Seit den 1950er Jahren
- Vorgänge und Begriffe aus der Biologie werden entlehnt → simulierte Evolution → möglichst gute Näherungslösungen für eine gegebenes Problem zu entwickeln.
- Wichtigste Begriffe: Individuum, Population, Selektion, Mutation, Rekombination, Genotyp, Fitness.
- Evolutionäre Algorithmen orientieren sich an der Evolution lebender Organismen.



EVOLUTIONSFAKTOREN

- **Genom**: Gesamtheit aller Gene eines Organismus
- **Genotyp**: exakte genetische Ausstattung, also der individuelle Satz von Genen, die im Zellkern getragen ist
- **Phänotyp**: oder Erscheinungsbild ist in der Genetik die Menge aller Merkmale eines Organismus. Er bezieht sich nicht nur auf morphologische, sondern auch auf physiologische Eigenschaften und auf Verhaltensmerkmale.
- Genom + Phänotyp = Individuum
- Ein einzelnes Gen im Genom kann meist verschiedene Werte annehmen. Jede dieser Ausprägungen wird als ein **Allel** bezeichnet. Ein Beispiel wäre bei einem Gen für die Haarfarbe ein Allel für blonde und ein Allel für schwarze Haare.
- Die Gesamtheit aller Allele in einer Population wird auch als **Genpool** bezeichnet.



EVOLUTIONSFAKTOREN

- **Artbegriff:** Eine Art wird durch diejenigen Populationen definiert, deren Individuen zu einem gemeinsamen Genpool gehören und sich miteinander paaren können.
- Population mit Individuen → Evolution = Änderung des Lebens. Findet genau dann statt, wenn sich die Häufigkeit der Allele, die sog. **Genfrequenz** verändert.



HERLEITUNG DER EVOLUTIONSFAKTOREN

- Mutation/Vervielfältigungsfehler → neue Allele werden eingeführt
- Rekombination
- Selektion: Veränderung der Allelenhäufigkeit durch unterschiedlich viele Nachkommen der einzelnen Allele.
- Die Selektion kann durch den Selektionswert bzw. Fitnesswert gemessen werden. Die relative Fitness eines Genotyps G ist über die Anzahl der überlebenden Nachkommen in einer Population definiert als

$$\text{Fitness}(G) = \frac{\text{Anzahl der Nachkommen von } G}{\text{Anzahl der Nachkommen von } G'}$$

wobei G' der Genotyp mit den meisten Nachkommen in der Population ist.



HERLEITUNG DER EVOLUTIONSFAKTOREN

- Implizit wird bei dem Fitnesswert angenommen, dass ein Genotyp, der besser an seine Umwelt angepasst ist, mehr Nachkommen erzeugt. Damit ist der Fitnesswert ein abgeleitetes Maß für die Tauglichkeit eines Individuums.
- Genfluss: die Genhäufigkeiten in der Population werden direkt durch Zu- oder Abwanderung von Individuen einer anderen Population derselben Art verändert.
- Gendrift: wird insbesondere bei kleinen Populationen beobachtet. Aufgrund von Zufallseffekten sterben Allele einzelner Gene aus. Gendrift bewirkt somit eine deutliche Reduktion der Vielfalt in einer Population. Gerade in sehr kleinen Populationen mit weniger als 100 Individuen ist Gendrift ein wesentlicher Evolutionsfaktor, wenn z. B. ein neu entstandener Lebensraum durch sehr wenige Individuen besiedelt wird. In sehr großen Populationen ist Gendrift vernachlässigbar.



ANPASSUNG ALS RESULTAT DER EVOLUTION

- Die Anpassung einer Population an ihre Umwelt führt zu verschiedenen charakteristischen Phänomenen wie der Besetzung von ökologischen Nischen, Wechselbeziehung zwischen Arten und dem Baldwin-Effekt.
- Durch Evolutionsfaktoren ist eine Population in der Lage, sich Veränderungen in der Umwelt anzupassen und den Lebensraum zu behaupten.



BIOLOGISCHE KONZEPTE IN EVOLUTIONÄREN ALGORITHMEN

Begriff	Biologie	Informatik
Individuum	Lebewesen	Lösungskandidat
Chromosom	DNS-Histon-Protein-Strang	Zeichenkette
	legt „Bauplan“ bzw. (Teil der Eigenschaften) eines Individuums in kodierter Form fest	
	meist mehrere Chromsomen je Individuum	meist nur ein Chromosom je Individuum
Gen	Teilstück eines Chromosoms	ein Zeichen
	grundlegende Einheit der Vererbung, die eine (Teil-)Eigenschaft eines Individuums festlegt	
Allel (Allelomorph)	Ausprägung eines Gens	Wert eines Zeichens
	je Chromosom gibt es nur eine Ausprägung eines Gens	
Locus	Ort eines Gens	Position eines Zeichens
	in einem Chromosom gibt es an jedem Ort genau ein Gen	



BIOLOGISCHE KONZEPTE IN EVOLUTIONÄREN ALGORITHMEN

Begriff	Biologie	Informatik
Phänotyp	äußeres Erscheinungsbild eines Lebewesens	Umsetzung/Implementierung eines Lösungskandidaten
Genotyp	genetische Konstitution eines Lebewesens	Kodierung eines Lösungskandidaten
Population	Menge von Lebewesen	Familie/Multimenge von Chromosomen
Generation	Population zu einem Zeitpunkt	
Reproduktion	Erzeugen von Nachkommen aus einem oder mehreren (meist zwei) Lebewesen	Erzeugen von (Kind-)Chromosomen aus 1 oder mehreren (Eltern-)Chromosomen
Fitness	Tauglichkeit/Angepaßtheit eines Lebewesens	Güte/Tauglichkeit eines Lösungskandidaten
	bestimmt Überlebens- und Fortpflanzungschancen	



ELEMENTE EINES EVOLUTIONÄREN ALGORITHMUS

Kodierungsvorschrift für Lösungskandidaten

- problemspezifische Kodierung der Lösungskandidaten
- keine allgemeinen Regeln
- später: einige Aspekte, zur Beachtung bei Wahl einer Kodierung

Methode, die Anfangspopulation erzeugt

- meistens Erzeugen zufälliger Zeichenketten
- auch komplexere Verfahren je nach gewählter Kodierung

Bewertungsfunktion (Fitnessfunktion) für die Individuen

- stellt Umgebung dar und gibt Güte der Individuen an
- meist identisch mit zu optimierender Funktion
- enthält auch zusätzliche Elemente (z.B. Nebenbedingungen)



ELEMENTE EINES EVOLUTIONÄREN ALGORITHMUS

Auswahlmethode basierend auf Fitnessfunktion

- bestimmt Individuen für Erzeugung von Nachkommen
- wählt Individuen unverändert in nächste Generation

Genetischen Operatoren, die Lösungskandidaten ändern

- Mutation - zufällige Veränderung einzelner Gene
- Crossover - Rekombination von Chromosomen
- richtig: **crossing over** (Meiose-Vorgang, Zellteilungsphase)
- Chromsomen werden zerteilt und dann überkreuzt zusammengefügt

Parameterwerte (Populationsgröße, Mutationsgeschwindigkeit, etc.)

Abbruchkriterium, z.B.

- festgelegte Anzahl von Generationen berechnet
- festgelegte Anzahl von Generationen keine Verbesserung
- vorgegebene Mindestlösungsgüte erreicht



FORMALE DEFINITIONEN: DEKODIERUNGSFUNKTION

Für jedes Optimierungsproblem: unterschiedliche Darstellung der Lösungskandidaten.

- EAs trennen Suchraum Ω (sog. Phänotyp) von Darstellung des Lösungskandidaten in Individuum (sog. Genotyp \mathcal{G})
- Bewertungsfunktion f ist definiert auf Ω
- Mutation und Rekombination sind auf \mathcal{G} definiert
- Bewertung eines im Genotyp vorliegenden Individuums durch Abbildung in Ω

Definition (Dekodierungsfunktion)

Eine Dekodierungsfunktion $\text{dec}: \mathcal{G} \rightarrow \Omega$ ist eine Abbildung vom Genotyp \mathcal{G} auf den Phänotyp Ω .



FORMALE DEFINITIONEN: INDIVIDUUM

Individuum besteht i.A. aus drei Dingen:

- 1 **Genotyp** $A.G \in \mathcal{G}$ eines Individuums A
- 2 **Zusatzinformationen** oder **Strategieparameter** $A.S \in \mathcal{Z}$
 - z.B. Parametereinstellungen für Operatoren
 - Raum \mathcal{Z} aller möglichen Zusatzinformationen
 - $A.S$ sind ebenso wie $A.G$ durch Operatoren modifizierbar
- 3 **Güte** oder **Fitness** $A.F \in \mathbb{R}$.



INDIVIDUUM

Definition

Ein *Individuum* A ist ein Tupel $(A.G, A.S, A.F)$ bestehend aus dem eigentlichen Lösungskandidaten, dem Genotyp $A.G \in G$, den optionalen Zusatzinformationen $A.S \in Z$ und dem Gütwert $A.F = f(\text{dec}(A.G)) \in \mathbb{R}$.

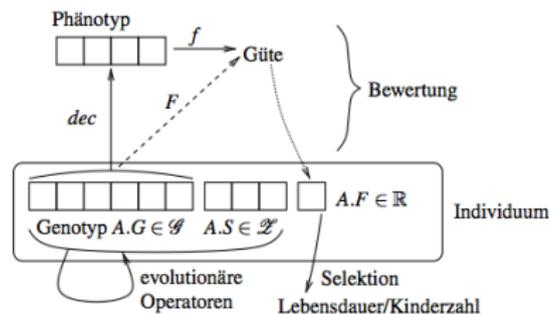


Abbildung 1: Unterschiedliche Aspekte eines Individuums: Genotyp $A.G$, Phänotyp $\text{dec}(A.G)$, Zusatzinformationen $A.S$ und Güte $A.F$.

BEMERKUNG

- Die Bewertung benutzt lediglich den Genotyp $A.G$ und speichert den dabei erhaltenen Wert im Güteattribut $A.F$, die evolutionären Operatoren können den Genotyp $A.G$ und die zusätzlichen Informationen $A.S$ nutzen und ggf. verändern und die Selektion leitet ausschließlich aus dem Gütewert $A.F$ die Überlebenswahrscheinlichkeit oder die Reproduktionsrate eines Individuums ab.
- Die genotypischen Räume sind sehr vielfältig
- Im Weiteren ist $G = M^*$ - die Menge aller Sequenzen beliebiger Länge mit Elementen aus der Menge M .
- Die Menge M wird auch als Basiswertebereich der einzelnen Komponenten bezeichnet.



FORMALE DEFINITIONEN: OPERATOREN

- Ξ (X_i): Menge aller Zustände des Zufallsgenerators
- keine Beschreibung der Veränderung des aktuellen Zustands $\xi \in \Xi$.

Definition (Operatoren)

Für ein durch \mathcal{G} kodiertes Optimierungsproblem und \mathcal{Z} die Menge der Zusatzbedingungen wird ein *Mutationsoperator* definiert durch die Abbildung

$$\text{Mut}^\xi: \mathcal{G} \times \mathcal{Z} \rightarrow \mathcal{G} \times \mathcal{Z}.$$

Analog wird ein *Rekombinationsoperator* mit $r \geq 2$ Eltern und $s \geq 1$ Kindern ($r, s \in \mathbb{N}$) definiert durch die Abbildung

$$\text{Rek}^\xi: (\mathcal{G} \times \mathcal{Z})^r \rightarrow (\mathcal{G} \times \mathcal{Z})^s.$$



FORMALE DEFINITIONEN: SELEKTIONSOPERATOR

- Eingabe: Population mit r Individuen, wobei s gewählt werden
- Selektion verändert/erfindet keine neuen Individuen
- Selektion bestimmt Indizes der Individuen nur durch Gütewerte



SELEKTIONSOPERATOR

Definition

Ein *Selektionsoperator* wird auf eine Population $P = (A^{(1)}, \dots, A^{(r)})$ angewandt:

$$\text{Sel}^\xi : (G \times Z \times \mathbb{R})^r \rightarrow (G \times Z \times \mathbb{R})^s$$

$$(A^{(i)})_{1 \leq i \leq r} \mapsto (A^{(IS^\xi(c_1, \dots, c_r)_k)})_{1 \leq k \leq s} \text{ mit } A^{(i)} = (a_i, b_i, c_i).$$

Die dabei zugrunde gelegte Indexselektion hat die Form

$$IS^\xi : \mathbb{R}^r \rightarrow \{1, \dots, r\}^s.$$



BEISPIELE: SELEKTIONSOPERATOR

- Elternpopulation bestehe aus Individuen $A^{(1)}, A^{(2)}, \dots, A^{(5)}$
- jeweiligen Güterwerte der Individuen seien
 1. $A^{(1)}.F = 2.5$
 2. $A^{(2)}.F = 1.9$
 3. $A^{(3)}.F = 3.7$
 4. $A^{(4)}.F = 4.1$
 5. $A^{(5)}.F = 2.4$
- Selektion wählt mit $IS^\xi : \mathbb{R}^5 \rightarrow \{1, \dots, 5\}^3$ Indizes 4, 3 und 1 bzw. Individuen $A^{(4)}, A^{(3)}$ und $A^{(1)}$



BEISPIEL

Hier werden als Indexselektion die drei Indizes mit den größten Gütewerten in der fünfelementigen Population gewählt. Formal wird dies beschrieben durch die Funktion

$$IS^5 : \mathbb{R}^5 \rightarrow \{1, \dots, 5\}^3$$
$$\langle c_1, c_2, c_3, c_4, c_5 \rangle \mapsto \{i, j, k\} (=: I) \text{ mit } i \neq j \text{ und } i \neq k \text{ und } j \neq k \text{ und}$$
$$\min\{c_i, c_j, c_k\} \geq \max\{c_m \mid 1 \leq m \leq 5 \text{ und } m \notin I\}.$$

Wichtig ist, dass es keine Einschränkungen hinsichtlich der Abbildung IS gibt. So kann sowohl eine deterministische Auswahl der besten Individuen als auch eine probabilistische realisiert sein, die Individuen zufällig auswählt. Auch können Individuen mehrfach gewählt werden und sogar $s > r$ ist möglich.



BEISPIEL

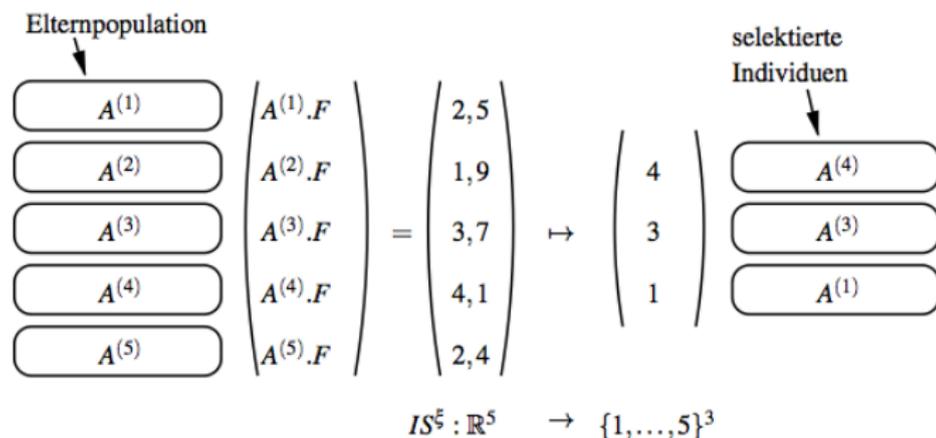


Abbildung 2: Das Beispiel demonstriert für $r = 5$ und $s = 3$, wie die Selektion auf die Indexselektion zurückgeführt wird. In diesem Beispiel würden die drei besten Individuen ausgewählt werden.

GENERISCHER EVOLUTIONÄRER ALGORITHMUS

Definition

Ein *generischer evolutionärer Algorithmus* zu einem *Optimierungsproblem* (Ω, f, \succ) ist ein 8-Tupel $(G, dec, Mut, Rek, IS_{Eltern}, IS_{Umwelt}, \mu, \lambda)$. Dabei bezeichnet μ die Anzahl der Individuen in der Elternpopulation und λ die Anzahl der erzeugten Kinder pro Generation. Ferner gilt

$$Rek: (G \times Z)^k \rightarrow (G \times Z)^{k'}$$

$$IS_{Eltern}: \mathbb{R}^\mu \rightarrow \{1, \dots, \mu\}^{\frac{k}{k'} \cdot \lambda} \text{ mit } \frac{k}{k'} \cdot \lambda \in \mathbb{N}$$

$$IS_{Umwelt}: \mathbb{R}^{\mu+\lambda} \rightarrow \{1, \dots, \mu + \lambda\}^\mu.$$



EA-SCHEMA

EA-SCHEMA(Optimierungsproblem (Ω, f, \succ))

- 1 $t \leftarrow 0$
 - 2 $P(t) \leftarrow$ erzeuge Population der Größe μ
 - 3 bewerte $P(t)$
 - 4 **while** Terminierungsbedingung nicht erfüllt
 - 5 **do** $\lceil P' \leftarrow$ selektiere Eltern für λ Nachkommen aus $P(t)$
 - 6 $P'' \leftarrow$ erzeuge Nachkommen durch Rekombination aus P'
 - 7 $P''' \leftarrow$ mutiere die Individuen in P''
 - 8 bewerte P'''
 - 9 $t \leftarrow t + 1$
 - 10 $\lfloor P(t) \leftarrow$ selektiere μ Individuen aus $P''' \circ P(t-1)$
 - 11 **return** bestes Individuum aus $P(t)$
-



GENETISCHER VS. EVOLUTIONÄRER ALGORITHMUS

BEGRIFFLICHE TRENNUNG

Genetischer Algorithmus:

- Kodierung: Zeichenkette aus Nullen und Einsen \Rightarrow Chromosomen ist Bitstring (Wort über Alphabet $\{0, 1\}$)

Evolutionärer Algorithmus:

- Kodierung: problemspezifisch (Zeichenkette aus Buchstaben, Graphen, Formeln, etc.)
- genetische Operatoren: anhand Kodierung und Problems definiert



HANDLUNGSREISENDENPROBLEM

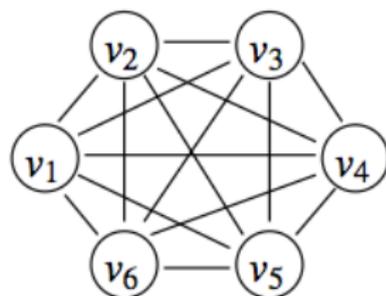
Sei $G = (V, E, \gamma)$ zur Berechnung der Kosten. Die Knotenmenge $V = \{v_1, \dots, v_n\}$ repräsentiert n verschiedene Städte, die paarweise durch Straßen in der Kantenmenge $E \subseteq V \times V$ verbunden sind. Jeder dieser Straßen ist eine Fahrtzeit $\gamma: E \rightarrow \mathbb{R}$ zugeordnet. Das **Handlungsreisendenproblem** ist dann definiert als Tupel $(S_n, f_{TSP}, <)$, wobei der Raum aller Permutationen S_n die unterschiedlichen Besuchsreihenfolgen repräsentiert. Die zu minimierende Bewertungsfunktion f_{TSP} ist definiert für $(\pi_1, \dots, \pi_n) \in S_n$ als

$$f_{TSP}((\pi_1, \dots, \pi_n)) = \gamma((v_{\pi_n}, v_{\pi_1})) + \sum_{j=2}^n \gamma((v_{\pi_{j-1}}, v_{\pi_j})).$$

Ein Handlungsreisendenproblem heißt ferner genau dann **symmetrisch**, wenn für alle $(v_i, v_j) \in E$ sowohl $(v_j, v_i) \in E$ als auch $\gamma((v_i, v_j)) = \gamma((v_j, v_i))$ erfüllt sind.



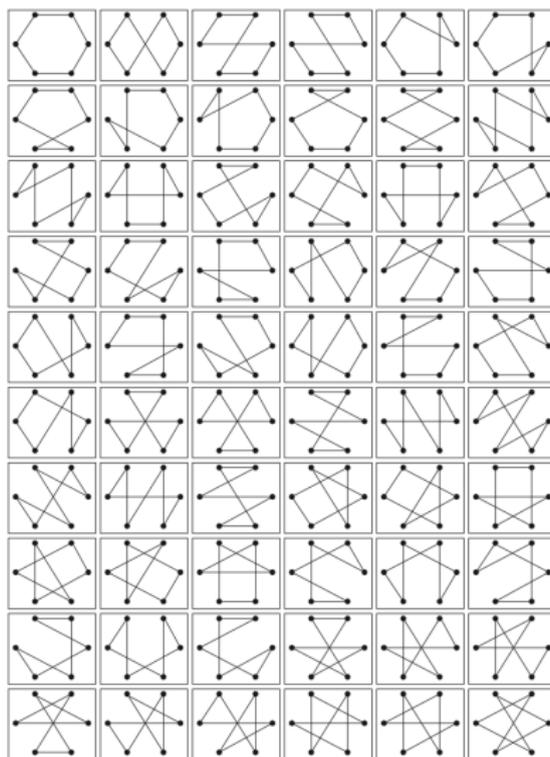
BEISPIEL



Kante e	$\gamma(e)$	Kante e	$\gamma(e)$	Kante e	$\gamma(e)$
(v_1, v_2)	5	(v_2, v_3)	10	(v_3, v_5)	17
(v_1, v_3)	8	(v_2, v_4)	4	(v_3, v_6)	8
(v_1, v_4)	11	(v_2, v_5)	9	(v_4, v_5)	6
(v_1, v_5)	3	(v_2, v_6)	12	(v_4, v_6)	5
(v_1, v_6)	7	(v_3, v_4)	6	(v_5, v_6)	11

Abbildung 3: Ungerichteter Graph eines beispielhaften symmetrischen Handlungsreisendenproblems, bei dem es zwischen allen Paaren von Städten eine Straße gibt. Die Tabelle gibt die Kosten bzw. Fahrtzeiten der einzelnen Straßen wieder.

BEISPIEL: 60 MÖGLICHE RUNDREISEN



BEISPIEL

- Jede der dargestellten Rundreisen steht dabei für zwölf verschiedene Rundreisen, die an jeder der 6 Städten mit zwei unterschiedlichen Fahrtrichtungen beginnen kann.
- Wenn man die Rundtouren weglässt, die sich nur durch die Fahrtrichtung oder die Startstadt unterscheiden, gibt es im vorliegenden Beispiel genau 60 verschiedene Lösungen.
- Bei 101 Städten sind es bereits $4,6631 \cdot 10^{157}$, allgemein $\frac{1}{2} \cdot (n - 1)!$ für n Städte.



BEMERKUNG

- Das Handlungsreisendenproblem zeichnet sich durch eine strikt vorgegebene und damit offensichtliche Struktur der Lösungskandidaten aus: eine Permutation der Indizes der Städte.
- Dies ist beispielsweise nicht der Fall, wenn eine Brückenkonstruktion gewichtsmminimal so optimiert werden soll, dass sie dennoch eine vorgegebene maximale Last tragen kann.
- Hier sind verschiedene Ansätze denkbar, wie ein Lösungskandidat die Struktur eines solchen Tragwerks beschreibt.
- Auch solche Probleme lassen sich erfassen, indem der Suchraum Ω entsprechend definiert wird.



BEWERTUNGSFUNKTION

- Die Bewertungsfunktion ist **der wichtigste Bestandteil eines Problems**, aus dem ein Optimierungsalgorithmus die Richtung der Optimierung ableitet. Daher muss in vielen Anwendungen diesem Aspekt hinreichend viel Aufmerksamkeit gewidmet werden.
- In einigen Fällen ist mit der Suche nach geeigneten Kriterien für die Erfassung der Güte eines Lösungskandidaten der größte Teil der Arbeit erledigt.



EVOLUTIONÄRER ALGORITHMUS FÜR DAS HANDLUNGSREISENDENPROBLEM

- Ziel ist es, ein Handlungsreisendenproblem mit 101 Städten schnell und mit ausreichender Qualität zu lösen
- Für eine vollständige Suche müssen wir $4,6631 \cdot 10^{157}$ Rundreisen untersuchen.
- Das Universum enthält etwa 10^{78} Atome und seit dem Urknall sind etwa 10^{19} Sekunden verstrichen.
- Wir müssen entscheiden, wie der konkrete Raum der Individuen aussehen soll, auf dem der Algorithmus arbeitet: $\Omega = \mathcal{S}_n$, d.h. die Individuen sind Permutationen.
- Operatoren zur Variation der Lösungskandidaten: jeder Operator muss aus Permutationen wieder gültige Permutationen erzeugen



TSP: 101 STÄDTE

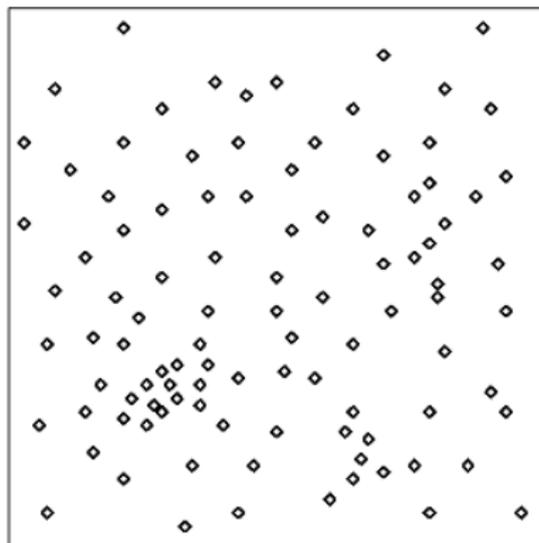


Abbildung 4: Das Bild zeigt die Positionen der Städte für eine Beispielinstanz des Handlungsreisendenproblems mit 101 Städten.

EVOLUTIONÄRER ALGORITHMUS FÜR DAS HANDLUNGSREISENDENPROBLEM

- Wir müssen einen Mutationsoperator entwerfen: z.B. die VERTAUSCHENDE MUTATION für eine geringfügige Veränderung

VERTAUSCHENDE-MUTATION(Permutation $A = (A_1, \dots, A_n)$)

```
1  $B \leftarrow A$ 
2  $u_1 \leftarrow$  wähle Zufallszahl gemäß  $U(\{1, \dots, n\})$ 
3  $u_2 \leftarrow$  wähle Zufallszahl gemäß  $U(\{1, \dots, n\})$ 
4  $B_{u_1} \leftarrow A_{u_2}$ 
5  $B_{u_2} \leftarrow A_{u_1}$ 
6 return  $B$ 
```

- Wähle zwei Zufallszahlen $u_1 = 2$ und $u_2 = 6$. Aus dem Individuum $(1, 2, 3, 4, 5, 6, 7, 8)$ entsteht das Individuum $(1, 6, 3, 4, 5, 2, 7, 8)$



EVOLUTIONÄRER ALGORITHMUS FÜR DAS HANDLUNGSREISENDENPROBLEM

- Mutation = eine kleine Veränderung charakterisiert
- Frage: ist die Mutation durch Tausch zweier Zahlen die kleinstmögliche Veränderung hinsichtlich des Handlungsreisendenproblems?
- Antwort: INVERTIERENDE MUTATION - invertiert (umkehrt) ein Teilstück der Permutation.
- Wähle zwei Zufallszahlen $u_1 = 2$ und $u_2 = 6$. Aus dem Individuum $(1, 2, 3, 4, 5, 6, 7, 8)$ entsteht das Individuum $(1, 6, 5, 4, 3, 2, 7, 8)$



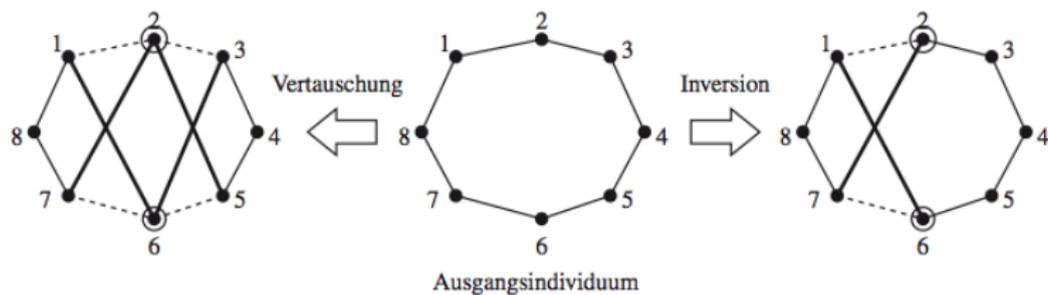
INVERTIERENDE MUTATION

INVERTIERENDE-MUTATION(Permutation $A = (A_1, \dots, A_n)$)

- 1 $B \leftarrow A$
 - 2 $u_1 \leftarrow$ wähle Zufallszahl gemäß $U(\{1, \dots, n\})$
 - 3 $u_2 \leftarrow$ wähle Zufallszahl gemäß $U(\{1, \dots, n\})$
 - 4 **if** $u_1 > u_2$
 - 5 **then** \square vertausche u_1 und u_2
 - 6 **for each** $j \in \{u_1, \dots, u_2\}$
 - 7 **do** $\square B_{u_2+u_1-j} \leftarrow A_j$
 - 8 **return** B
-



BEIDE OPERATOREN VERANSCHAULICHT



EVOLUTIONÄRER ALGORITHMUS FÜR DAS HANDLUNGSREISENDENPROBLEM

- Vertauschende Mutation: vier Kanten werden gestrichen und vier neue Kanten eingefügt. Das bedeutet, dass bei der Bewertung des neuen Individuums vier Kantengewichte abgezogen und vier Kantengewichte zur Güte des Ausgangsindividuum hinzuaddiert werden.
- Es werden lediglich zwei Kanten durch zwei neue Kanten ersetzt. Bezüglich der Bewertungsfunktion nimmt dieser Operator offensichtlich eine kleinere Veränderung an einer Rundreise vor.
- Auf der Basis dieser Überlegung werden wir in unserem evolutionären Algorithmus für das Handlungsreisendenproblem dem Operator INVERTIERENDE MUTATION den Vorzug geben.



EVOLUTIONÄRER ALGORITHMUS FÜR DAS HANDLUNGSREISENDENPROBLEM

- Wir brauchen noch einen Operator, der die Rekombination simulieren soll: Mische die Eigenheiten der Eltern und trage sie auf das Kindindividuum über
- Frage: Wie kann man möglichst große Teile der in den Elternindividuen vorliegenden Rundreisen in ein neues Individuum vererben, so dass keine gänzlich neue Rundtour entsteht?



EIN ERSTER VERSUCH: ORDNUNGSREKOMBINATION

- Wir übernehmen ein beliebig langes Präfix der einen Rundtour und fügen die restlichen Städte gemäß ihrer Reihenfolge in der anderen elterlichen Rundreise an

ORDNUNGSREKOMBINATION(Permutationen $A = (A_1, \dots, A_n)$ und $B = (B_1, \dots, B_n)$)

```
1   $j \leftarrow$  wähle zufällig gemäß  $U(\{1, \dots, n-1\})$ 
2  for each  $i \in \{1, \dots, j\}$ 
3  do  $\sqsubset C_i \leftarrow A_i$ 
4  for  $i \leftarrow 1, \dots, n$ 
5  do  $\sqsupset$  if  $B_i \notin \{C_1, \dots, C_j\}$ 
6     then  $\sqsupset$   $j \leftarrow j+1$ 
7      $\sqsubset$   $\sqsubset C_j \leftarrow B_i$ 
8  return  $C$ 
```

- Durch die Abfrage in der zweiten for-Schleife wird auch hier die ausschließliche Erzeugung von gültigen Permutationen garantiert.



BEISPIEL

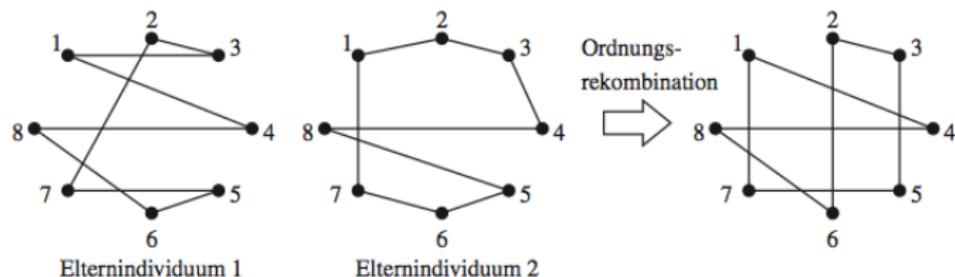


Abbildung 5: Die ORDNUNGSREKOMBINATION übernimmt vom Elternindividuum (1, 4, 8, 6, 5, 7, 2, 3) die ersten vier Städte. Die noch fehlenden Städte werden gemäß ihrer Reihenfolge im zweiten Elternindividuum (1, 2, 3, 4, 8, 5, 6, 7) aufgefüllt. So ergibt sich das Individuum (1, 4, 8, 6, 2, 3, 5, 7).

BEMERKUNG

- Wie man an diesem Beispiel sieht, kann es durchaus vorkommen, dass das Ergebnis des Operators stark von den Eltern abweicht - hier wurden zwei Kanten eingefügt, die in keinem der beiden Elternindividuen vorkamen.
- Der Name ORDNUNGREKOMBINATION rührt daher, dass die Ordnung bzw. Reihenfolge der Städte erhalten bleibt.
- Für das Handlungsreisendenproblem würde ein idealer Rekombinationsoperator ausschließlich Kanten der Eltern benutzen.
- Dieser Anforderung kommt ein zweiter Rekombinationsoperator, die KANTENREKOMBINATION, sehr nahe, welche die gemeinsamen Adjazenzliste beider Eltern betrachtet, mit dem Startknoten eines der beiden Elternindividuen beginnt und iterativ gemäß der Adjazenzinformation den nächsten Knoten mit den wenigsten weiteren Wahlmöglichkeiten aussucht.



EIN ZWEITER VERSUCH: KANTENREKOMBINATION

KANTENREKOMBINATION(Permutationen $A = (A_1, \dots, A_n)$ und $B = (B_1, \dots, B_n)$)

```
1  for each Knoten  $v \in \{1, \dots, n\}$ 
2  do  $\sqsubset Adj(v) \leftarrow \emptyset$ 
3  for each  $i \in \{1, \dots, n\}$ 
4  do  $\sqsupset Adj(A_i) \leftarrow Adj(A_i) \cup \{A_{(i \bmod n)+1}\}$ 
5      $Adj(A_{(i \bmod n)+1}) \leftarrow Adj(A_{(i \bmod n)+1}) \cup \{A_i\}$ 
6      $Adj(B_i) \leftarrow Adj(B_i) \cup \{B_{(i \bmod n)+1}\}$ 
7      $\sqsubset Adj(B_{(i \bmod n)+1}) \leftarrow Adj(B_{(i \bmod n)+1}) \cup \{B_i\}$ 
8   $C_1 \leftarrow$  wähle zufällig gemäß  $U(\{A_1, B_1\})$ 
9  for  $i \leftarrow 1, \dots, n-1$ 
10 do  $\sqsupset K \leftarrow \{m \in Adj(C_i) \mid \#(Adj(m) \setminus \{C_1, \dots, C_i\}) \text{ minimal}\}$ 
11     if  $K \neq \emptyset$ 
12     then  $\sqsubset C_{i+1} \leftarrow$  wähle gleichverteilt zufällig aus  $K$ 
13      $\sqsubset$  else  $\sqsubset C_{i+1} \leftarrow$  wähle gleichverteilt zufällig aus  $\{1, \dots, n\} \setminus \{C_1, \dots, C_i\}$ 
14 return  $C$ 
```



BEISPIEL: KANTENREKOMBINATION

Ausgangssituation:

$Adj(1) = \{2, 3, 4, 7\}$	$Adj(2) = \{1, 3, 7\}$	$Adj(3) = \{1, 2, 4\}$	wähle zufällig $C_1 = 1$ $\Rightarrow (1, \dots)$
$Adj(4) = \{1, 3, 8\}$	$Adj(5) = \{6, 7, 8\}$	$Adj(6) = \{5, 7, 8\}$	
$Adj(7) = \{1, 2, 5, 6\}$	$Adj(8) = \{4, 5, 6\}$		

1. Iteration:

	$Adj(2) = \{3, 7\} \Leftarrow$	$Adj(3) = \{2, 4\} \Leftarrow$	wähle $C_2 = 3$ $\Rightarrow (1, 3, \dots)$
$Adj(4) = \{3, 8\} \Leftarrow$	$Adj(5) = \{6, 7, 8\}$	$Adj(6) = \{5, 7, 8\}$	
$Adj(7) = \{2, 5, 6\} \Leftarrow$	$Adj(8) = \{4, 5, 6\}$		

2. Iteration:

	$Adj(2) = \{7\} \Leftarrow$	$Adj(6) = \{5, 7, 8\}$	wähle $C_3 = 2$ $\Rightarrow (1, 3, 2, \dots)$
$Adj(4) = \{8\} \Leftarrow$	$Adj(5) = \{6, 7, 8\}$		
$Adj(7) = \{2, 5, 6\}$	$Adj(8) = \{4, 5, 6\}$		

Abbildung 6: Ein Ablaufprotokoll für eine Rekombination zwischen den Elternindividuen (1, 2, 3, 4, 8, 5, 6, 7) und (1, 4, 8, 6, 5, 7, 2, 3) veranschaulicht die Arbeitsweise. Die Pfeile \Leftarrow markieren die Knoten, die an der jeweiligen Stelle auswählbar sind. Die Pfeile \Leftarrow kennzeichnen die Knoten, die zwar durch eine Kante erreichbar wären, aber vom Algorithmus zugunsten der anderen Knoten verworfen werden.



BEISPIEL: KANTENREKOMBINATION

3. Iteration:

$Adj(4) = \{8\}$ $Adj(7) = \{5, 6\} \leftarrow$	$Adj(5) = \{6, 7, 8\}$ $Adj(8) = \{4, 5, 6\}$	$Adj(6) = \{5, 7, 8\}$	es folgt $C_4 = 7$ $\Rightarrow (1, 3, 2, 7, \dots)$
--	--	------------------------	---

4. Iteration:

$Adj(4) = \{8\}$	$Adj(5) = \{6, 8\} \leftarrow$ $Adj(8) = \{4, 5, 6\}$	$Adj(6) = \{5, 8\} \leftarrow$	wähle $C_5 = 6$ $\Rightarrow (1, 3, 2, 7, 6, \dots)$
------------------	--	--------------------------------	---

5. Iteration:

$Adj(4) = \{8\}$	$Adj(5) = \{8\} \leftarrow$ $Adj(8) = \{4, 5\} \leftarrow$		es folgt $C_6 = 5$ $\Rightarrow (1, 3, 2, 7, 6, 5, \dots)$
------------------	---	--	---

6. Iteration:

$Adj(4) = \{8\}$	$Adj(8) = \{4\} \leftarrow$		es folgt $C_7 = 8$ $\Rightarrow (1, 3, 2, 7, 6, 5, 8, \dots)$
------------------	-----------------------------	--	--

7. Iteration:

$Adj(4) = \{8\} \leftarrow$			es folgt $C_8 = 4$ $\Rightarrow (1, 3, 2, 7, 6, 5, 8, 4)$
-----------------------------	--	--	--

Abbildung 7: Ein Ablaufprotokoll für eine Rekombination zwischen den Elternindividuen $(1, 2, 3, 4, 8, 5, 6, 7)$ und $(1, 4, 8, 6, 5, 7, 2, 3)$ veranschaulicht die Arbeitsweise. Die Pfeile \leftarrow markieren die Knoten, die an der jeweiligen Stelle auswählbar sind. Die Pfeile \leftarrow kennzeichnen die Knoten, die zwar durch eine Kante erreichbar wären, aber vom Algorithmus zugunsten der anderen Knoten verworfen werden.



BEISPIEL: KANTENREKOMBINATION

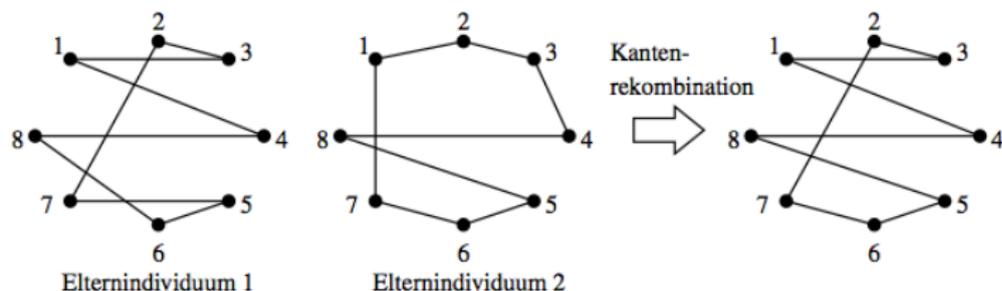


Abbildung 8: Für das ausführliche Beispiel aus Abb. 7,8 werden hier die Elternindividuen und das durch die KANTENREKOMBINATION entstandene Kindindividuum gezeigt.

SELEKTION

- Gibt der Optimierung eine Richtung
- Dies soll ohne größere weiterführende Überlegungen in einer Umweltselktion geschehen, die die besten Individuen aus den Eltern und den neu erzeugten Kindern übernimmt.
- Wir wählen eine Elternpopulation der Größe 10 und erzeugen pro Generation 40 neue Individuen.
- Damit besteht die nächste Elternpopulation nur aus den 10 besten Individuen.
- Die Auswahl der Eltern in der Elternselektion findet zufällig gleichverteilt statt.



SELEKTION

- Die Mutation nimmt eine sehr gezielte kleine Veränderung vor, die durch ihre hohe Anpassung an das Problem zusammen mit der Selektion bereits einen guten iterativen Optimierungsfortschritt verspricht.
- Die Rekombination bemüht sich zwar nach Möglichkeit einzelne Details der Elternindividuen zu benutzen, kann aber dennoch sehr starke Eingriffe in die Struktur eines Lösungskandidaten mit sich bringen.
- Da zusätzlich der Berechnungsaufwand für die Rekombination größer ist als für die Mutation, erzeugen wir nur 30% der neuen Individuen mit der Rekombination und einer anschließenden Mutation.



SELEKTION

- Die restlichen Individuen werden nur mittels einer Mutation erzeugt.
- Somit ergibt sich der Algorithmus zur Lösung des Handlungsreisendenproblems.
- Als Abbruchkriterium wurde hier eine Grenze von maximal 2 000 Generationen gesetzt.



ALGORITHMUS

EA-HANDLUNGSREISENDENPROBLEM(Zielfunktion F , Anzahl der Städte n)

```
1  $t \leftarrow 0$ 
2  $P(t) \leftarrow$  Liste mit 10 gleichverteilt zufälligen Permutationen aus  $U(\mathcal{S}_n)$ 
3 bewerte alle  $A \in P(t)$  mit Zielfunktion  $F$ 
4 while  $t \leq 2000$ 
5 do  $\lceil P' \leftarrow \langle \rangle$ 
6   for each  $i \in \{1, \dots, 40\}$ 
7   do  $\lceil A \leftarrow$  wähle gleichverteilt zufällig erstes Elter aus  $P(t)$ 
8     if  $u < 0,3$  für eine Zufallszahl  $u$  gewählt gleichverteilt gemäß  $U([0, 1])$ 
9     then  $\lceil B \leftarrow$  wähle gleichverteilt zufällig zweites Elter aus  $P(t)$ 
10       $\lceil A \leftarrow$  KANTENREKOMBINATION  $(A, B)$ 
11       $A \leftarrow$  INVERTIERENDE-MUTATION  $(A)$ 
12       $\lceil P' \leftarrow P' \circ \langle A \rangle$ 
13      bewerte alle  $A \in P'$  mit Zielfunktion  $F$ 
14       $t \leftarrow t + 1$ 
15       $\lceil P(t) \leftarrow$  10 beste Individuen aus  $P' \circ P(t-1)$ 
16 return bestes Individuum aus  $P(t)$ 
```



BEISPIEL

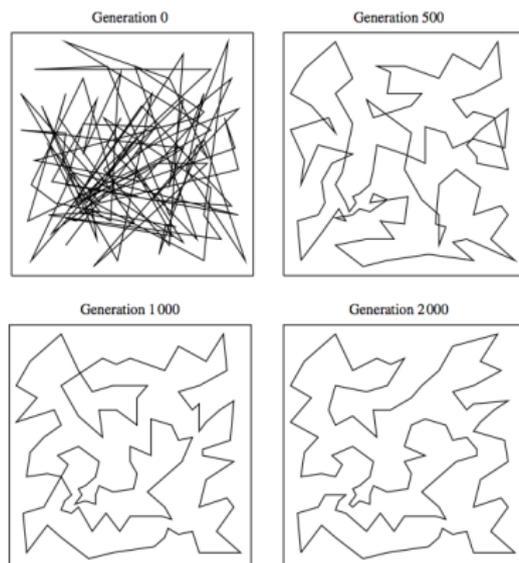


Abbildung 9: Für die Optimierung mit der Kantenrekombination und der invertierenden Mutation werden die besten gefundenen Rundreisen in den Generationen 0, 500, 1 000 und 2 000 dargestellt.

BEISPIEL

- Die besten Rundreisen der Generationen 0, 500, 1000 und 2000 demonstrieren, wie die Länge der Tour durch Entfernung von Überkreuzungen verringert wird.
- Das Endergebnis hat die Länge 670 und ist damit bereits sehr nahe an dem bekannten Bestwert 629 - die Abweichung beträgt 6,1%.
- Tatsächlich haben durch diesen Algorithmus insgesamt 80010 bewertete Individuen ausgereicht, um ein sehr gutes Ergebnis zu erlangen.
- Verglichen mit der Anzahl aller Rundreisen $4,6631 \cdot 10^{157}$ ist dies ein verschwindend geringer Teil des Suchraums, was auch den letzten Skeptiker von der Arbeitsweise der evolutionären Algorithmen überzeugen sollte.



BEISPIEL

- Der Handlungsreisendenproblem EA-Algorithmus lässt sich in diesem Formalismus als Tupel

$$(\mathcal{S}_n, id, \text{INVERTIERENDE-MUTATION: } \mathcal{S}_n \times \{\perp\} \rightarrow \mathcal{S}_n \times \{\perp\},$$

$$\text{KANTENREKOMBINATION: } (\mathcal{S}_n \times \{\perp\})^2 \rightarrow \mathcal{S}_n \times \{\perp\},$$

$$\text{gleichverteilt zufällig: } \mathbb{R}^{10} \rightarrow \{1, \dots, 10\}^{80},$$

$$\text{Wahl der Besten: } \mathbb{R}^{50} \rightarrow \{1, \dots, 50\}^{10}, 10, 40\}$$

darstellen.



STANDARDOPTIMIERUNGSVERFAHREN

- Skalares Optimierungsproblem
Maximiere/minimiere $f(x)$ unter der Nebenbedingung $x \in X$
- Vektoroptimierung (Pareto-Optimierung)
Mehrere Zielgrößen gleichzeitig optimieren
- Spezialfälle:
lineare Optimierung, ganzzahlige Optimierung, nichtlineare Optimierung
- besonders kompliziert
Globale nichtlineare Optimierung ist mathematisch nicht gelöst



BEISPIEL: NEWTONVERFAHREN

SKALARE OPTIMIERUNG

- Auffinden eines Minimum/Maximums einer eindimensionalen Funktion
- Entspricht dem Auffinden der Nullstelle der ersten Ableitung
- Grundidee:
 1. Beliebigen Punkt x_0 wählen
 2. Tangente in diesem Punkt an Graphen legen
 3. Schnittpunkt Tangente – x -Achse ist neuer Punkt
 4. Wiederhole Schritt 3 bis $|f(x)| < \epsilon$

$$x_{n+1} = x_n - \frac{f(x_n)}{f'(x_n)}$$



BEISPIEL: SIMPLEX-ALGORITHMUS

LINEARE OPTIMIERUNG UNTER NEBENBEDINGUNGEN

- Lösung des Optimierungsproblems

$$\max_{x \in \mathbb{R}^n} \{c^T x \mid Ax \leq b, x \geq 0\}$$

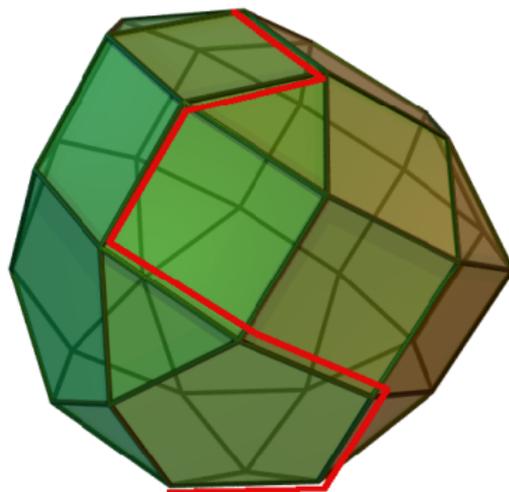
- A Matrix mit reellen Einträgen, entspricht Koeffizienten im Gleichungssystem
- b Vector mit Beschränkungen
- c Zielfunktionsvektor
- Ziel: Finden eines Punktes x , der maximalen Zielfunktionswert hat



BEISPIEL: SIMPLEX-ALGORITHMUS

LINEARE OPTIMIERUNG UNTER NEBENBEDINGUNGEN

- Da das Problem ein lineares Ungleichungssystem ist, befindet sich die optimale Lösung am Rand des konvexen Lösungsbereiches.
- 1. Phase
 - Finden einer Basislösung
- 2. Phase
 - Verbessern der Lösung entlang der Kanten



METAHEURISTIKEN

- Die Nichtlineare globale Optimierung ist in der Mathematik quasi ungelöst, ist aber in realen Anwendungen sehr oft anzuwenden.
- Erfolgreiche Anwendung von sogenannten Metaheuristiken.



WAS IST EINE METAHEURISTIK?

- Algorithmus zur näherungsweise Lösung eines kombinatorischen Optimierungsproblems definiert abstrakte Folge von Schritten, die auf beliebige Problemstellungen anwendbar sind
- Die einzelnen Schritte müssen aber problemspezifisch implementiert werden
⇒ problemspezifische Heuristik



EINSATZ VON METAHEURISTIKEN

- bei Problemen, wo kein effizienterer Lösungsalgorithmus bekannt ist
 - z.B. kombinatorische Optimierungsprobleme
 - Finden einer optimalen Lösung ist in der Regel nicht garantiert
 - gute Lösungen können beliebig schlecht verglichen mit optimaler Lösung sein
 - Erfolg und Laufzeit hängen ab von
 - Problemdefinition und
 - Implementierung der einzelnen Schritte
- ⇒ EAs sind auch Metaheuristiken



LOKALE SUCHALGORITHMEN

- geg.: Optimierungsproblem (Ω, f, \succ)
- ges.: finde Element $x \in \Omega$, das f optimiert (max. oder min.)
- o.B.d.A.: finde ein Element $x \in \Omega$, das f maximiert
(ist f zu minimieren, dann betrachten wir $f' \equiv -f$)

⇒ **lokale Suchalgorithmen** zum Finden von lokalen Optima in Ω

- **Voraussetzung:** $f(x_1)$ und $f(x_2)$ unterscheiden sich für ähnliche $x_1, x_2 \in \Omega$ nicht zu sehr (keine großen Sprünge in f)
- anwendbar für beliebige Ω zum Finden von lokalen Optima



LOKALE SUCHALGORITHMEN

- lokale Suchalgorithmen = Sonderfall der EAs
 - Population: 1 Lösungskandidaten \Rightarrow verschiedene Konsequenzen
 - Rekombinationsoperator nicht sinnvoll, da nur ein Individuum
 - Veränderungen: Mutations- bzw. Variationsoperator
 - Selektion: neu erzeugtes Individuum anstatt Elternindividuum in nächster Generation?
- \Rightarrow ein Basisalgorithmus für alle lokalen Suchalgorithmen
- Varianten durch unterschiedliche Akzeptanzkriterien
 - Individuen besitzen i.d.R. keine Zusatzinformationen
 $\mathcal{Z} = \emptyset$
 - Genotyp \mathcal{G} ist (wie gehabt) problemabhängig



Algorithm 1 Basialgorithmus der lokalen Suche

Input: Zielfunktion F

Output: Lösungskandidaten A

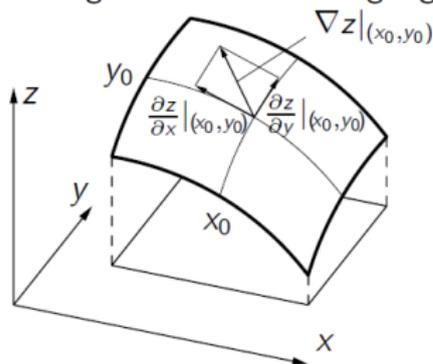
```
1:  $t \leftarrow 0$ 
2:  $A(t) \leftarrow$  erzeuge Lösungskandidat
3: bewerte  $A(t)$  durch  $F$ 
4: while Terminierungsbedingung nicht erfüllt {
5:    $B =$  variiere  $A(t)$ 
6:   bewerte  $B$  durch  $F$ 
7:    $t \leftarrow t + 1$ 
8:   if  $Akz(A(t-1).F, B.F, t)$  {                               /* Akzeptanzbedingung */
9:      $A(t) \leftarrow B$ 
10:  } else {
11:     $A(t) \leftarrow A(t-1)$ 
12:  }
13: }
14: return  $A(t)$ 
```

- Akz : untersch. implementiert bei versch. lokalen Suchalgorithmen



GRADIENTENVERFAHREN

- **Voraussetzung:** $\Omega \subseteq \mathbb{R}^n$ und $f : \Omega \rightarrow \mathbb{R}$ ist differenzierbar
 - **Gradient:** Differentialoperation, die Vektorfeld erzeugt
- ⇒ liefert Vektor in Richtung der stärksten Steigung einer Funktion



- Illustration des Gradienten von $z = f(x, y)$ am Punkt (x_0, y_0)

$$\nabla z|_{(x_0, y_0)} = \left(\frac{\partial z}{\partial x}|_{(x_0, y_0)}, \frac{\partial z}{\partial y}|_{(x_0, y_0)} \right)$$

GRADIENTENVERFAHREN

Idee: von zufälligem Startpunkt, gehe kleine Schritte in Ω jeweils in Richtung des stärksten Anstiegs der Funktion bis (lokales) Maximum erreicht

1. wähle (zufälligen) Startpunkt $\mathbf{x}^{(0)} = (x_1^{(0)}, \dots, x_n^{(0)})$
2. bestimme Gradienten am aktuellen Punkt $\mathbf{x}^{(t)}$

$$\nabla_{\mathbf{x}} f(\mathbf{x}^{(t)}) = \left(\frac{\partial}{\partial x_1} f(\mathbf{x}^{(t)}), \dots, \frac{\partial}{\partial x_n} f(\mathbf{x}^{(t)}) \right)$$

3. gehe kleines Stück in Richtung des Gradienten

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \eta \nabla_{\mathbf{x}} f(\mathbf{x}^{(t)})$$

η : Schrittweitenparameter („Lernrate“ in KNNs)

4. wiederhole Schritte 2 und 3 bis Abbruchkriterium erfüllt ist (z.B. best. Anzahl Schritte ausgeführt, aktueller Gradient sehr klein)



PROBLEME

Wahl des Schrittweitenparameters

- zu kleiner Wert \Rightarrow lange Laufzeit bis Maximum erreicht
- zu großer Wert \Rightarrow Oszillationen, Hin- und Herspringen in Ω
- Lösungen: Momentterm, adaptiver Schrittweitenparameter (siehe Vorlesung „Neuronale Netze“)

Hängenbleiben in lokalen Maxima

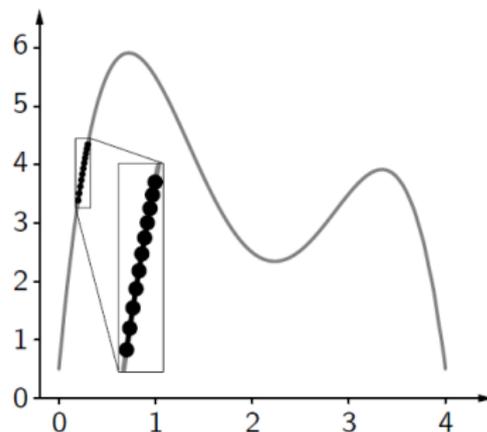
- aufgrund lokaler Steigungsinformationen, vielleicht nur lokales Maximum erreichbar
- Problem kann *nicht* prinzipiell behoben werden
- Chancenverbesserung: mehrfaches Ausführen mit verschiedenen Startwerten



BEISPIELE

$$f(x) = -\frac{5}{6}x^4 + 7x^3 - \frac{115}{6}x^2 + 18x + \frac{1}{2}$$

t	x_t	$f(x_t)$	$f'(x_t)$	Δx_t
0	0.200	3.388	11.147	0.011
1	0.211	3.510	10.811	0.011
2	0.222	3.626	10.490	0.010
3	0.232	3.734	10.182	0.010
4	0.243	3.836	9.888	0.010
5	0.253	3.932	9.606	0.010
6	0.262	4.023	9.335	0.009
7	0.271	4.109	9.075	0.009
8	0.281	4.191	8.825	0.009
9	0.289	4.267	8.585	0.009
10	0.298	4.340		

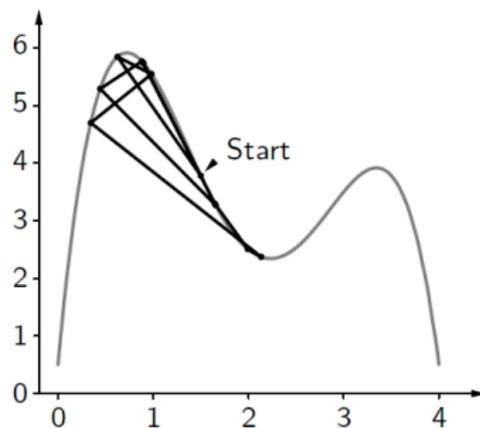


Gradientenaufstieg mit Startwert 0.2 und $\eta = 0.001$

BEISPIELE

$$f(x) = -\frac{5}{6}x^4 + 7x^3 - \frac{115}{6}x^2 + 18x + \frac{1}{2}$$

t	x_t	$f(x_t)$	$f'(x_t)$	Δx_t
0	1.500	3.781	-3.500	-0.875
1	0.625	5.845	1.431	0.358
2	0.983	5.545	-2.554	-0.639
3	0.344	4.699	7.157	1.789
4	2.134	2.373	-0.567	-0.142
5	1.992	2.511	-1.380	-0.345
6	1.647	3.297	-3.063	-0.766
7	0.881	5.766	-1.753	-0.438
8	0.443	5.289	4.851	1.213
9	1.656	3.269	-3.029	-0.757
10	0.898	5.734		

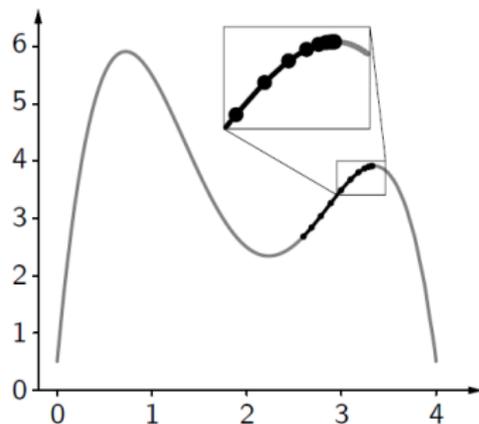


Gradientenaufstieg mit Startwert 1.5 und $\eta = 0.25$

BEISPIELE

$$f(x) = -\frac{5}{6}x^4 + 7x^3 - \frac{115}{6}x^2 + 18x + \frac{1}{2}$$

t	x_t	$f(x_t)$	$f'(x_t)$	Δx_t
0	2.600	2.684	1.707	0.085
1	2.685	2.840	1.947	0.097
2	2.783	3.039	2.116	0.106
3	2.888	3.267	2.153	0.108
4	2.996	3.492	2.009	0.100
5	3.097	3.680	1.688	0.084
6	3.181	3.805	1.263	0.063
7	3.244	3.872	0.845	0.042
8	3.286	3.901	0.515	0.026
9	3.312	3.911	0.293	0.015
10	3.327	3.915		



Gradientenaufstieg mit Startwert 2.6 und $\eta = 0.05$

ZUFALLSAUFSTIEG

Idee: falls f nicht differenzierbar, bestimme Richtung, in der f ansteigt durch Auswerten zufälliger Punkte in Umgebung des aktuellen Punktes

1. wähle (zufälligen) Startpunkt $x_0 \in \Omega$
2. wähle Punkt $x' \in \Omega$ „in der Nähe“ von x_t (z.B. durch zufällige, aber kleine Veränderung von x_t)
3. setze

$$x_{t+1} = \begin{cases} x' & \text{falls } f(x') > f(x_t), \\ x_t & \text{sonst} \end{cases}$$

4. wiederhole Schritte 2 und 3 bis Abbruchkriterium erfüllt



ZUFALLSAUFSTIEG

Pseudocode für **Akzeptanzbedingung** im **Basisalgorithmus**:

Algorithm 2 Akzeptanzbedingung für Zufallsaufstieg

Input: Elterngüte $A.F$, Kindgüte $B.F$, Generation t

Output: **true** oder **false**

1: **return** $B.F \succ A.F$

- **Problem:** Hängenbleiben in lokalen Maxima
- alle folgenden Verfahren versuchen, dieses Problem zu verringern



SIMULIERTES AUSGLÜHEN

WIEDERHOLUNG

1. wähle (zufälligen) Startpunkt $x_0 \in \Omega$
2. wähle Punkt $x' \in \Omega$ „in der Nähe“ des aktuellen Punktes x_t (z.B. durch zufällige, aber kleine Veränderung von x_t)
3. setze

$$x_{t+1} = \begin{cases} x' & \text{falls } f(x') \geq f(x_t), \\ x' \text{ mit Wahrscheinlichkeit } p = e^{-\frac{\Delta f}{kT}} & \\ x_t \text{ mit Wahrscheinlichkeit } 1 - p & \text{sonst.} \end{cases}$$

$\Delta f = f(x_t) - f(x')$ Qualitätsverringering der Lösung
 $k = \Delta f_{\max}$ (Schätzung des) Umfangs der Funktionswerte
 T Temperaturparameter (sinkt im Laufe der Zeit)

4. wiederhole Schritte 2 und 3 bis Abbruchkriterium erfüllt

für kleine T Verfahren geht nahezu in Zufallsaufstieg über



SIMULIERTES AUSGLÜHEN

EA

Algorithm 3 Akzeptanzbedingung für Simuliertes Ausglühen

Input: Elterngüte $A.F$, Kindgüte $B.F$, Generation t

Output: true oder false

```
1: if  $B.F \succ A.F$  {
2:   return true
3: } else {
4:    $u \leftarrow$  wähle zufällig aus  $U([0, 1])$  /* Zufallszahl zw. 0 und 1 */
5:   if  $u \leq \exp\left(-\frac{A.F - B.F}{kT_{t-1}}\right)$  {
6:     return true
7:   } else {
8:     return false
9:   }
10: }
```



SCHWELLWERTAKZEPTANZ

Idee: ähnlich wie beim simulierten Ausglühen auch Akzeptanz schlechterer Lösungen, allerdings mit oberer Schranke für Verschlechterung

1. wähle (zufälligen) Startpunkt $x_0 \in \Omega$
2. wähle Punkt $x' \in \Omega$ „in der Nähe“ des aktuellen Punktes x_t (z.B. durch zufällige, aber kleine Veränderung von x_t)
3. setze

$$x_{t+1} = \begin{cases} x' & \text{falls } f(x') \geq f(x_t) - \theta, \\ x_t & \text{sonst.} \end{cases}$$

θ Schwellenwert für das Akzeptieren schlechterer Lösungen
(wird im Laufe der Zeit (langsam) gesenkt)
($\theta = 0$ entspricht normalem Zufallsaufstieg)

4. wiederhole Schritte 2 und 3 bis Abbruchkriterium erfüllt



SCHWELLWERTAKZEPTANZ

Algorithm 4 Akzeptanzbedingung für Schwellwertakzeptanz

Input: Elterngüte $A.F$, Kindgüte $B.F$, Generation t

Output: `true` oder `false`

```
1: if  $B.F \succ A.F$  oder  $A.F - B.F \leq \theta$  {  
2:   return true  
3: } else {  
4:   return false  
5: }
```



SINFLUTALGORITHMUS

Idee: ähnlich wie beim simulierten Ausglühen auch Akzeptanz schlechterer Lösungen, allerdings mit unterer Schranke

1. wähle (zufälligen) Startpunkt $x_0 \in \Omega$
2. wähle Punkt $x' \in \Omega$ „in der Nähe“ des aktuellen Punktes x_t (z.B. durch zufällige, aber kleine Veränderung von x_t)

3. setze

$$x_{t+1} = \begin{cases} x' & \text{falls } f(x') \geq \theta_0 + t \cdot \eta, \\ x_t & \text{sonst.} \end{cases}$$

θ_0 untere Schranke für Lösungskandidaten bei $t = 0$
(anfänglicher „Wasserstand“)

η Schrittweitenparameter („Geschwindigkeit des Regens“)

4. wiederhole Schritte 2 und 3 bis Abbruchkriterium erfüllt



SINTFLUTALGORITHMUS

Algorithm 5 Akzeptanzbedingung für Sintflut-Algorithmus

Input: Elterngüte $A.F$, Kindgüte $B.F$, Generation t

Output: **true** oder **false**

```
1: if  $B.F > \theta_0 + \eta \cdot t$  {  
2:   return true  
3: } else {  
4:   return false  
5: }
```



REKORDORIENTIERTES WANDERN

Idee: ähnlich wie beim Sintflut-Algorithmus wird steigender Wasserpegel genutzt, jedoch an Güte des bisher besten gefundenen Individuums gekoppelt

- mögliche Verschlechterung: stets in Relation zum besten gefundenen Individuum
- nur bei Verbesserung: aktuelles Individuum von Bedeutung
- wie bei Schwellwertakzeptanz: monoton fallende Folge reeller Zahlen regelt Übernahme eines schlechten Individuums



REKORDORIENTIERTES WANDERN

Algorithm 6 Akzeptanzbedingung für Rekordorientiertes Wandern

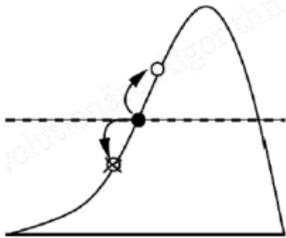
Input: Elterngüte $A.F$, Kindgüte $B.F$, t , beste gefundene Güte F_{best}

Output: **true** oder **false**, F_{best}

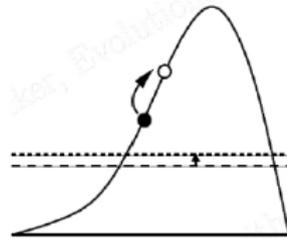
```
1: if  $B.F \succ F_{\text{best}}$  {  
2:    $F_{\text{best}} \leftarrow B.F$   
3:   return true,  $F_{\text{best}}$   
4: } else {  
5:   if  $B.F - F_{\text{best}} < T_t$  {  
6:     return true,  $F_{\text{best}}$   
7:   }  
8: }  
9: return false,  $F_{\text{best}}$ 
```



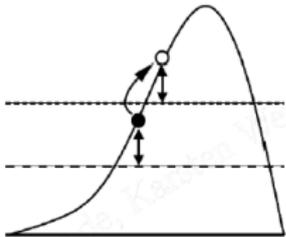
VERGLEICH LOKALE SUCHALGORITHMEN



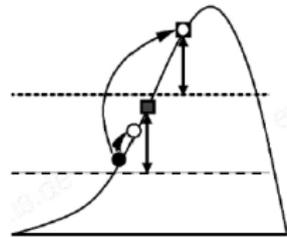
Hillclimbing



Sintflut-Algorithmus



Schwellwertakzeptanz



Rekordorientiertes Wandern