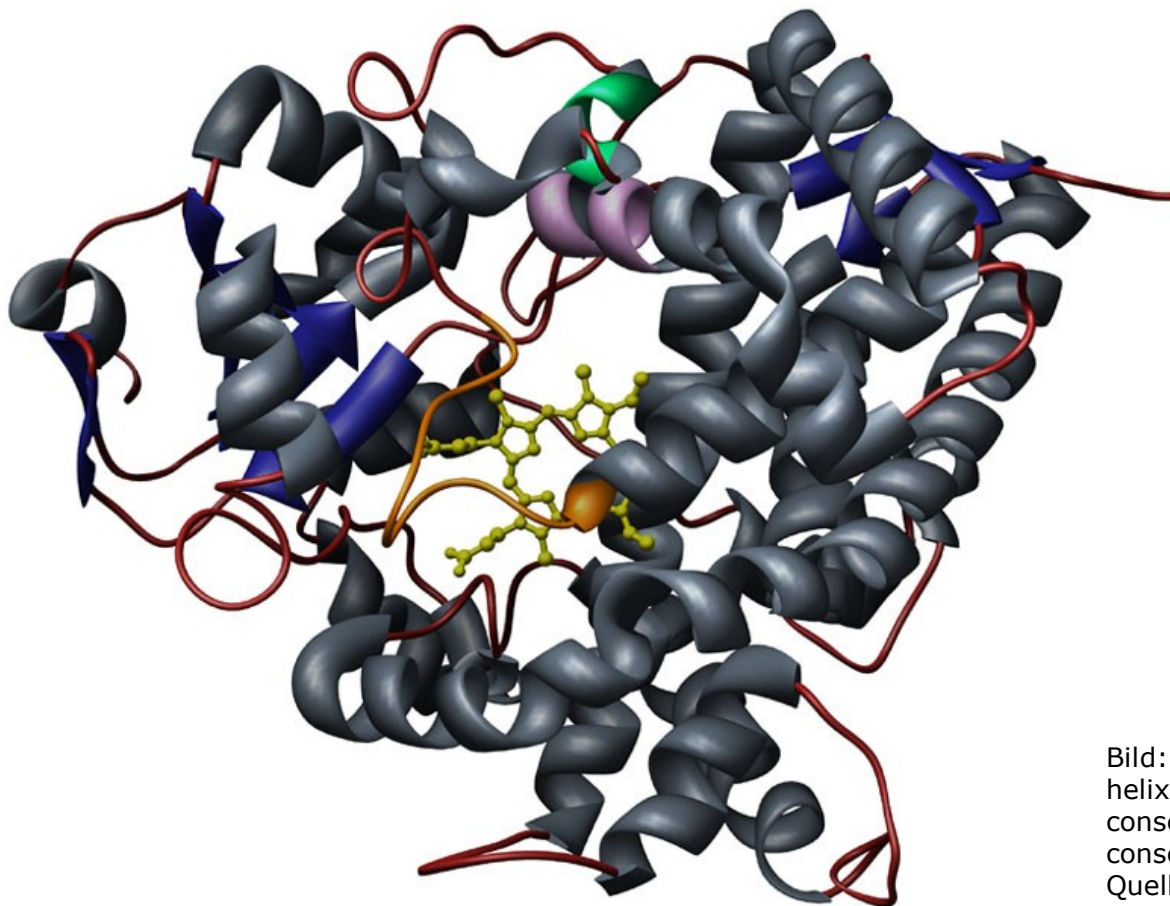


Prof. Dr. Alexander del Pino
Fachbereich Informatik
Sommersemester 2007

Genetische Algorithmen

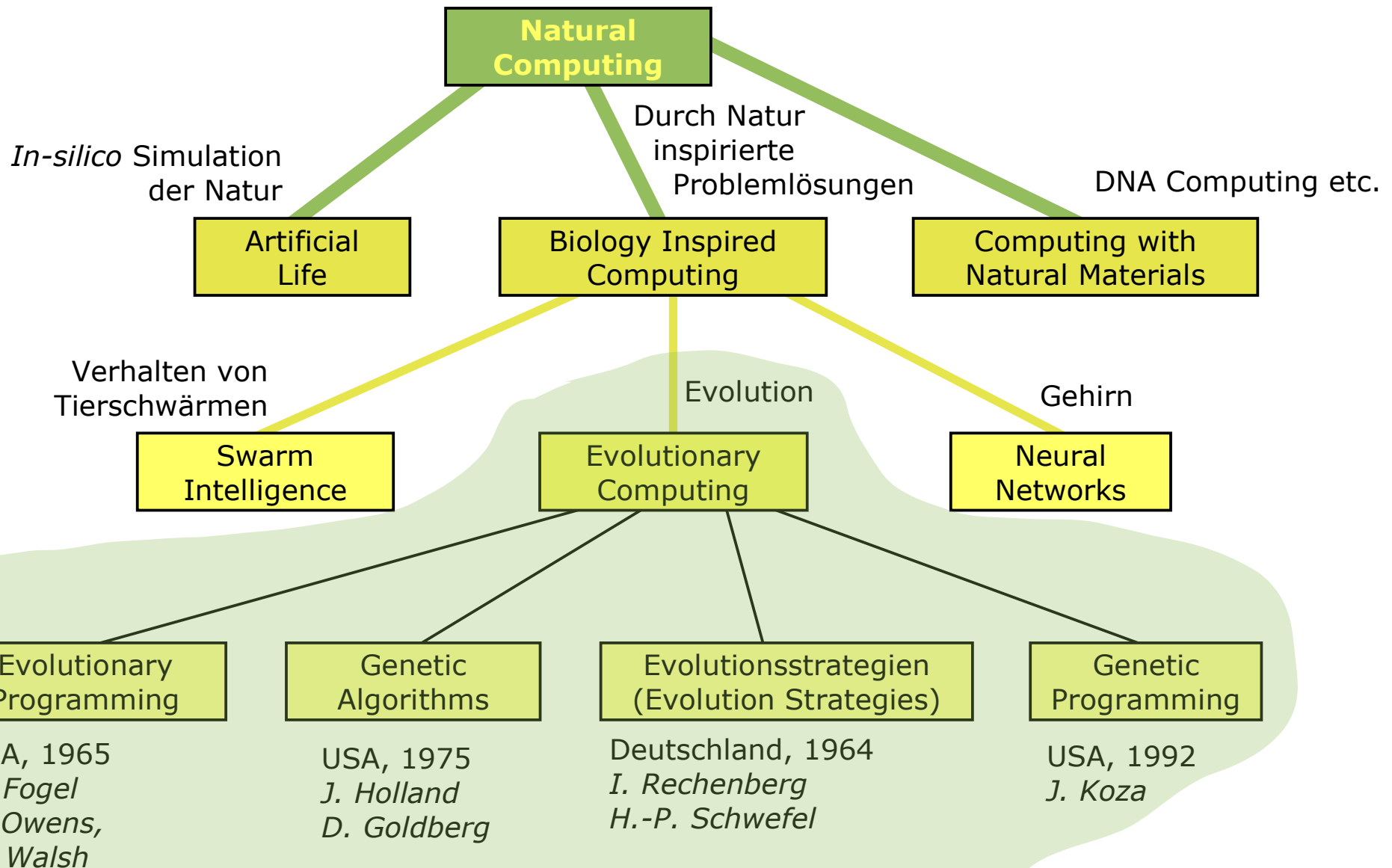


1. Teil
Einführung

Bild: CYP3A4 [1TQN, Homo sapiens] ribbon detail. Color key:
helix structures, gray; strand structures, blue; PERF
consensus, green; K-helix consensus, purple; heme-binding
consensus, orange; heme ligand, yellow
Quelle: <http://p450.kvl.dk/gallery/CYP3A4.jpg>

Einführung

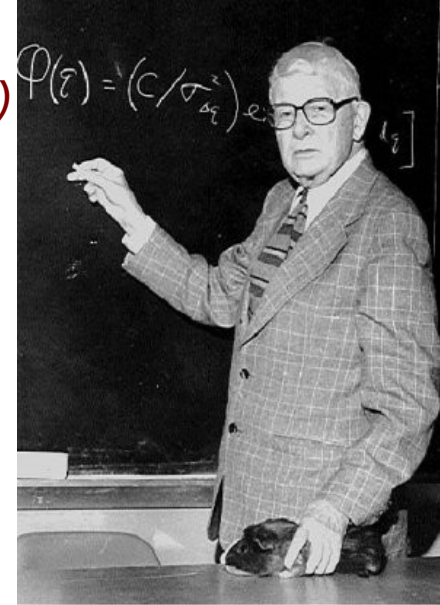
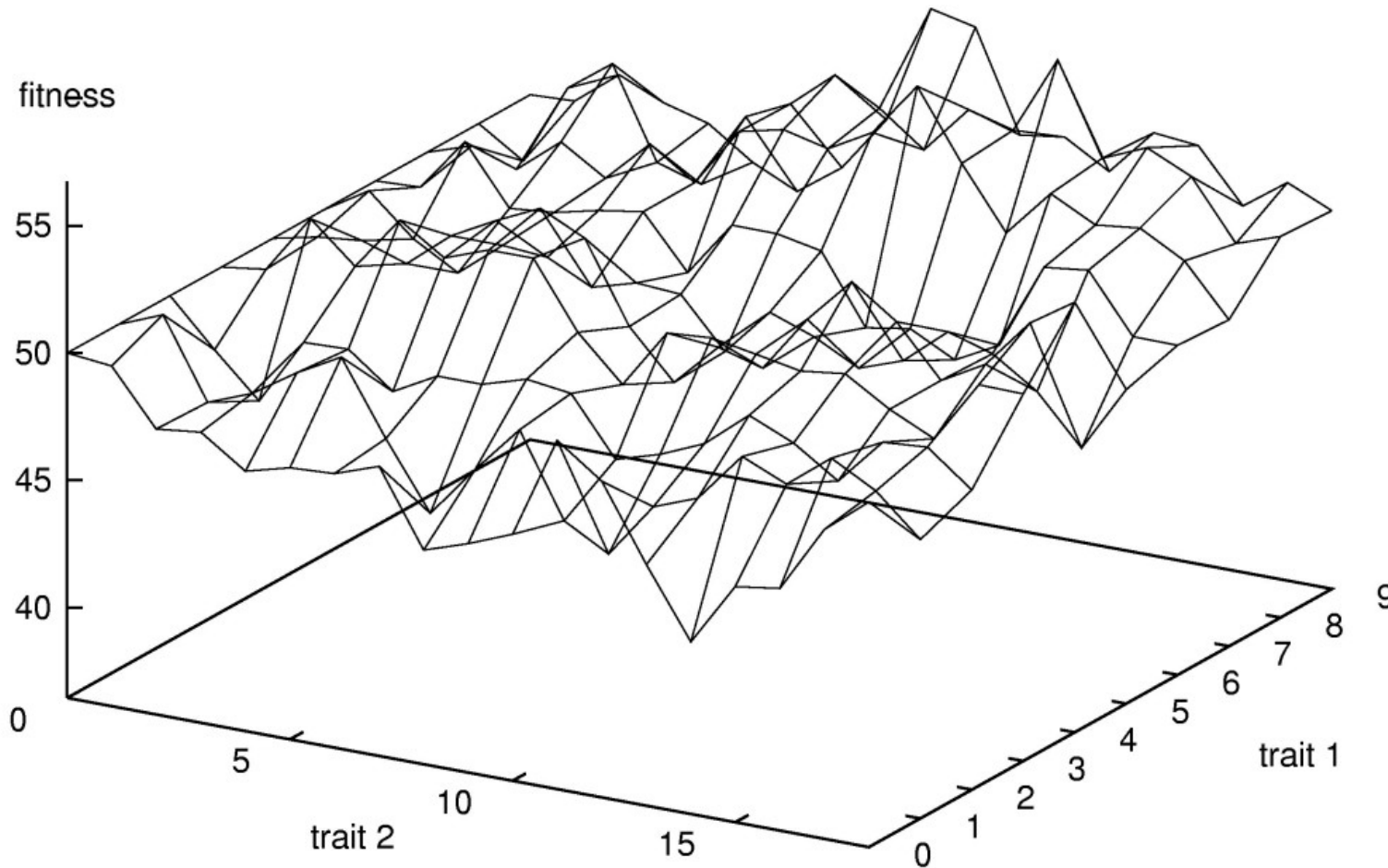
The really big picture



Einführung

Evolutionary Computation

Sewall Wright, 1930er Jahre: Population mit n Eigenschaften lebt in einer $n+1$ dimensionalen Landschaft (*adaptive Landscape*)



Einführung

Eine weitere Einordnung

Genetische Algorithmen (GA) sind ebenso wie die *neuronalen Netze* und die *Fuzzy-Systeme* ein Teilbereich des *Soft Computing*, das wiederum in die *künstliche Intelligenz* (KI) eingeordnet wird.

Professor *Lotfi A. Zadeh* von der *Berkeley University* umschreibt den Begriff *Soft Computing* wie folgt:

Soft computing differs from conventional (hard) computing in that, unlike hard computing, it is tolerant of imprecision, uncertainty, partial truth, and approximation. In effect, the role model for soft computing is the human mind.

Typische Merkmale von Soft Computing-Verfahren sind:

- Das Problem kann oftmals *nicht analytisch modelliert* werden.
- Die Lösung ist oftmals nur eine *brauchbare Näherung* an die exakte Lösung.

Einführung

Was sind genetische Algorithmen ?



Genetische Algorithmen sind *heuristische Suchverfahren*.

Sie werden zum Finden einer brauchbaren Lösung in sehr grossen Suchräumen eingesetzt.

Die Nadel ist hier.

Bildquelle: Wikipedia

Einführung

Welche anderen Suchverfahren gibt es ?

Bei der Suche nach einem Element in einem Lösungsraum gibt es folgende vier grundsätzliche Vorgehensweisen:

- Bei den *heuristischen Suchverfahren* findet man in der Regel nur eine Näherungslösung, dafür ist aber das Verhältnis zwischen der Lösungsgüte und dem dafür benötigten Rechen- bzw. Zeitaufwand oftmals sehr günstig.
- Eine *vollständige systematische Aufzählung* aller Lösungskandidaten ist praktisch nur bei sehr kleinen Suchräumen durchführbar.
- Die *Zufallssuche* funktioniert immer, ist aber oftmals sehr ineffizient.
- Bei den *analytischen Verfahren* kann eine exakte Lösung oftmals mit verhältnismäßig geringem Rechenaufwand gefunden werden – Leider sind viele interessante Probleme nicht analytisch darstellbar/lösbar.

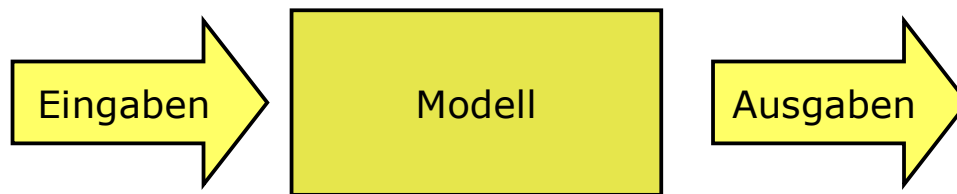
Einführung

Systeme und Probleme

Als nächstes wollen wir betrachten, welche Probleme mit Hilfe von genetischen Algorithmen gelöst werden können.

Hierzu benötigen wir zunächst das Konzept eines *Systems*.

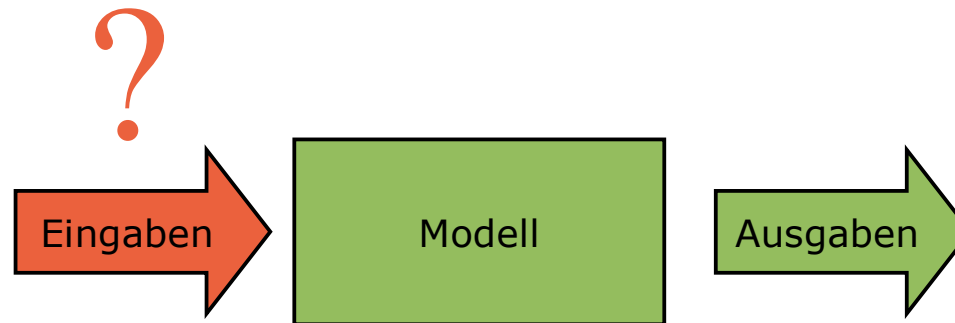
❓ Was ist eigentlich ein System ?



Mit der nötigen Abstraktion kann man jedes System auf *Eingabe*, ein *Modell* und seine *Ausgaben* reduzieren.

Wenn man das Modell kennt, versteht man wie das System arbeitet und kann die *Systemantwort*, also die Ausgaben zu vorgegebenen Eingaben berechnen.

Einführung



Optimierungsprobleme

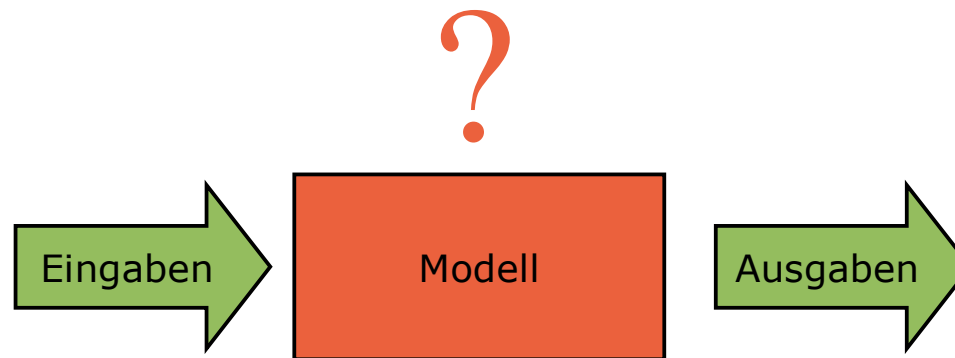
Bei Optimierungsproblemen sind das Modell und die gewünschten Ausgaben bekannt, allerdings sind die Eingaben, die zu den gewünschten Ausgaben führen unbekannt.

Das Ziel bei Optimierungsproblemen ist es, solche Eingabewerte zu finden, mit denen das System die gewünschten Ausgaben berechnet.

Beispiel:

Welches ist die kürzeste Reiseroute ?

Einführung



Modellbildungsprobleme

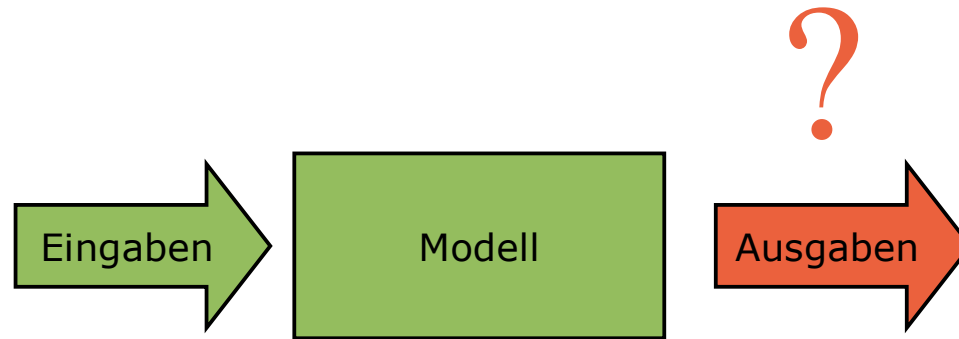
Bei Modellbildungsproblemen sind Eingaben und zugehörige Ausgaben bekannt, jedoch ist das Modell des Systems welches aus den Eingaben die zugehörigen Ausgaben berechnet unbekannt.

Das Ziel bei Modellbildungsproblemen ist es, ein Modell zu finden, welches aus den vorhandenen Eingaben die ebenfalls vorhandenen Ausgaben berechnet.

Beispiel:

Wie hängt der DAX von der Inflation, vom €/\$\$-Wechselkurs und anderen Faktoren ab ?

Einführung



Simulationsprobleme

Bei Simulationsproblemen sind die Eingaben und das Modell bekannt, jedoch sind die Ausgaben unbekannt.

Das Ziel bei Simulationsproblemen ist es, die Ausgaben zu berechnen, wenn das Modell die bekannten Eingaben im Sinne eines *was-wäre-wenn* verarbeitet.

Beispiel

Welche Einsparungen ergeben sich wenn wir von einer 3% höheren No-Show-Rate ausgehen?

Einführung

Wofür werden genetische Algorithmen eingesetzt ?

Transportprobleme

Eine *Ware* ist an verschiedenen *Quell-Standorten* in unterschiedlichen Mengen vorhanden, und soll an verschiedene *Ziel-Standorte* transportiert werden.

Jeder Ziel-Standort hat einen unterschiedlichen Bedarf an diesen Waren.

❓ Wovon hängen die Transportkosten ab?

- Dem Quell-Standort.
- Dem Ziel-Standort.
- Der gelieferten Warenmenge.

Von welchen Quell-Standorten muss welche Warenmenge an welche Ziel-Standorte geliefert werden, so dass die *Transportkosten minimal* sind ?

Hinweis: Ein Quell-Standort kann durchaus mehrere Ziel-Standorte beliefern.



Bildquelle: Wikipedia

Einführung

Wofür werden genetische Algorithmen eingesetzt ?

Rucksackprobleme

Sie planen eine Trekkingtour in den Himalaya und können verschieden *schwere Gegenstände* in einem Rucksack mitnehmen. Jeder Gegenstand hat einen spezifischen *Nutzen*.

Da der Rucksack nicht mehr als 20 Kilogramm wiegen soll, müssen Sie sich gut überlegen, welche Gegenstände Sie mitnehmen, so dass der *Nutzen maximal* und die 20 Kilogramm *nicht überschritten* werden.

Dieses Problem ist in der allgemeinen Form auch als das *0/1-Rucksack-Problem* bekannt.



Bildquelle: Wikipedia

Einführung

Wofür werden genetische Algorithmen eingesetzt ?

Problem des Handlungsreisenden

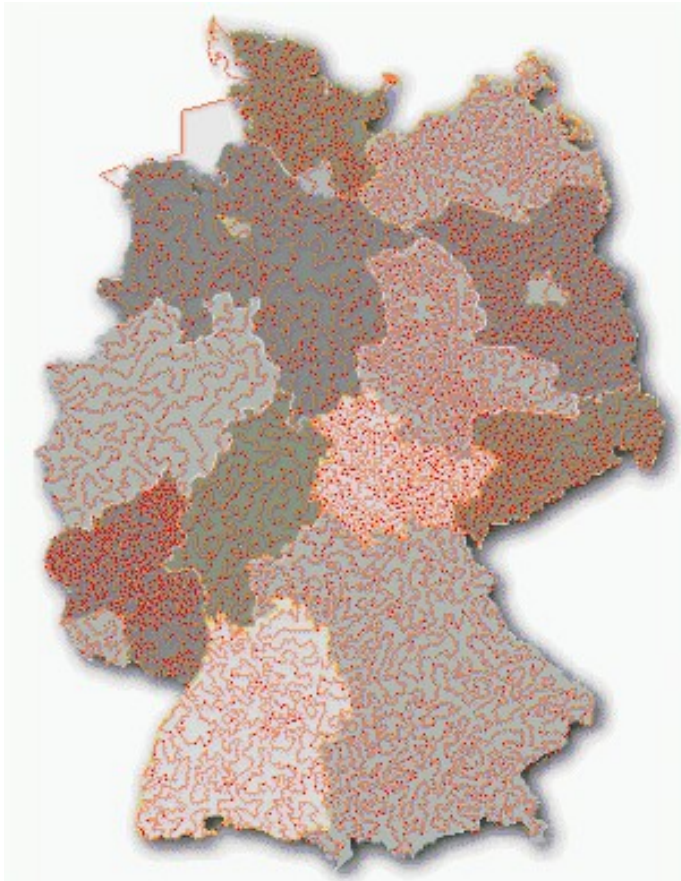


Bild: Optimale Tour durch 15112 deutsche Städte.
Berechnet im Jahr 2001 auf auf einem Rechnernetz
mit 110 Prozessoren.

Quelle: <http://www.tsp.gatech.edu/d15sol/index.html>

Ein Handlungsreisender soll eine Rundreise durch verschiedene *Städte* machen und danach wieder zu seinem Ausgangspunkt zurückkehren. Jede Stadt darf nur ein *einziges Mal* besucht werden.

Bei welcher Reihenfolge ist der *zurückgelegte Weg minimal* ?

Einführung

Wofür werden genetische Algorithmen eingesetzt ?

Sozialverhalten/Politik

- *Bonnie* und *Clyde* haben mal wieder eine Bank überfallen.
- Man kann ihnen allerdings nichts direkt nachweisen, deswegen würden beide wegen eines geringfügigen Delikts für 2 Jahre ins Gefängnis kommen.
- Die beiden werden in getrennte Zellen gebracht und können nicht miteinander kommunizieren. Dort bietet man beiden eine Kronzeugenregelung an.
- Sagt einer der beiden aus Kronzeuge gegen den anderen aus, dann kommt der Kronzeuge auf Bewährung frei, und der andere kommt 15 Jahre ins Gefängnis.



Bildquelle: Wikipedia

Einführung

Wofür werden genetische Algorithmen eingesetzt ?

Sozialverhalten/Politik

Wenn allerdings beide als Kronzeuge auftreten, ist die Kronzeugenregelung hinfällig und beide kommen für 7 Jahre ins Gefängnis.

❓ Wie sollen sich beide am besten verhalten ?

Wenn beide nicht als Kronzeuge auftreten, kommen sie mit jeweils 2 Jahren davon.

- Wie kann bei Wiederholung der Situation eine Kooperation der beiden eingeleitet werden ?

Wozu ist das Ganze gut ? Denken Sie zum Beispiel mal an die Deeskalation von Rüstungswettläufen.



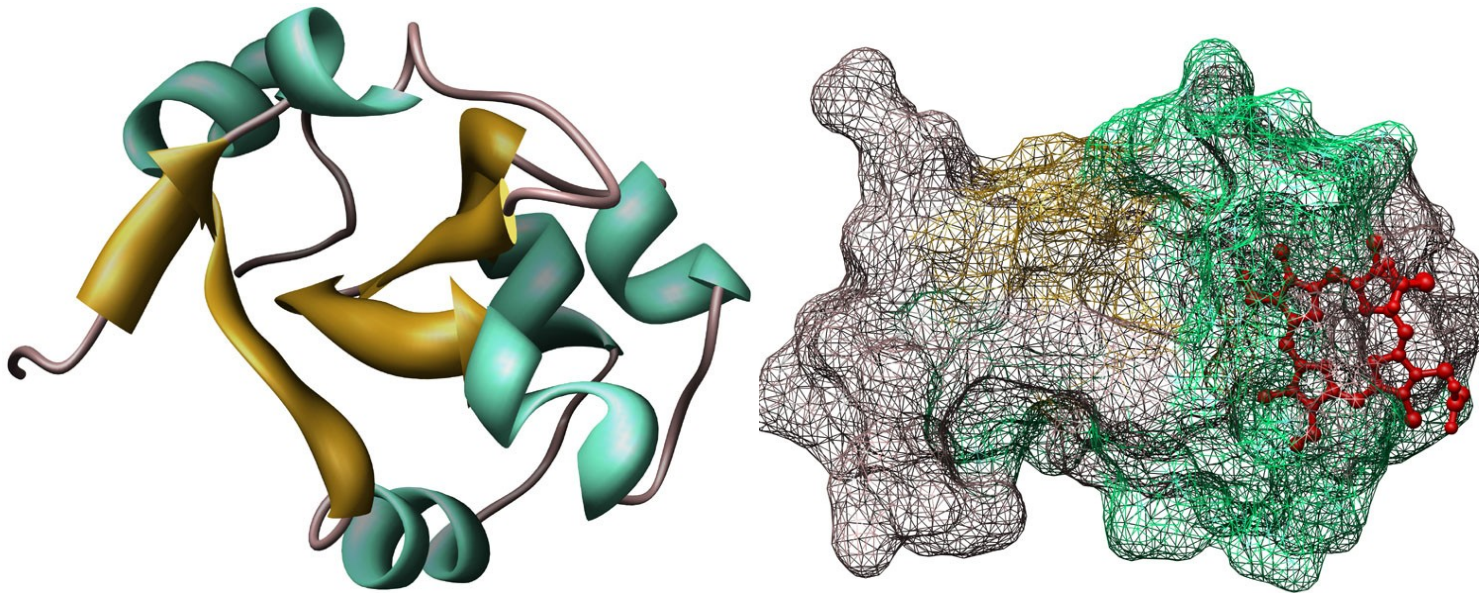
Bildquelle: Wikipedia

In der allgemeinen Form ist diese Problem als das *Gefangenendilemma* bekannt.

Einführung

Wofür werden genetische Algorithmen eingesetzt ?

Proteinfaltung



Links: Cytochrome b5 [1B5B, *Rattus norvegicus*] ribbon structure. Color key: helix, blue; strand, gold.

Rechts: Cytochrome b5 [1B5M, *Rattus norvegicus*] mesh surface structure with ball-and-stick ligand. Color key: helix, green; strand, gold; bound ligand (heme), red.

Bildquelle:
<http://p450.kvl.dk/gallery>

Proteine sind Makromoleküle und bestehen aus einer Kette von oftmals mehreren Hunderten von Aminosäuren.

Die Wirkung eines Proteins hängt maßgeblich von seiner 3-dimensionalen Raumstruktur ab. Diese ist im Normalfall durch die Sequenz der Aminosäuren festgelegt.

Einführung

Wofür werden genetische Algorithmen eingesetzt ?

Proteinfaltung

Wenn man nun die Aminosäuresequenz eines Proteins kennt, wie sieht dann die zugehörige 3-dimensionale Struktur aus ?



IBM entwickelte ab 1999 den *BlueGene* Supercomputer um Proteinfaltung besser verstehen zu können.

Die BlueGene/L Installation am *Lawrence Livermore National Laboratory* steht mit einer Linpack-Performanz von 280.6 Teraflops im Jahr 2006 auf Platz Nr. 1 der Top-500 Weltrangliste der schnellsten Computer.



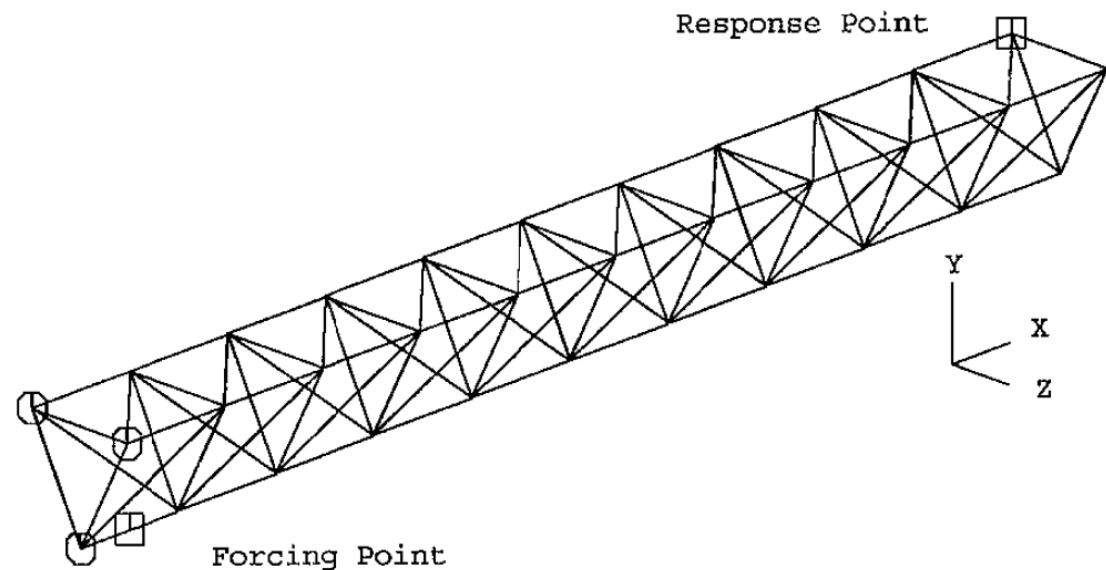
Professor *Vijay Pande* von der *Stanford University* leitet das Folding@home Projekt.

Einführung

Wofür werden genetische Algorithmen eingesetzt ?

Industrielles Design (Beispiel von Prof. A. E. Eiben)

Optimiere das Design eines Satellitenarms um Vibrationen zu minimieren.



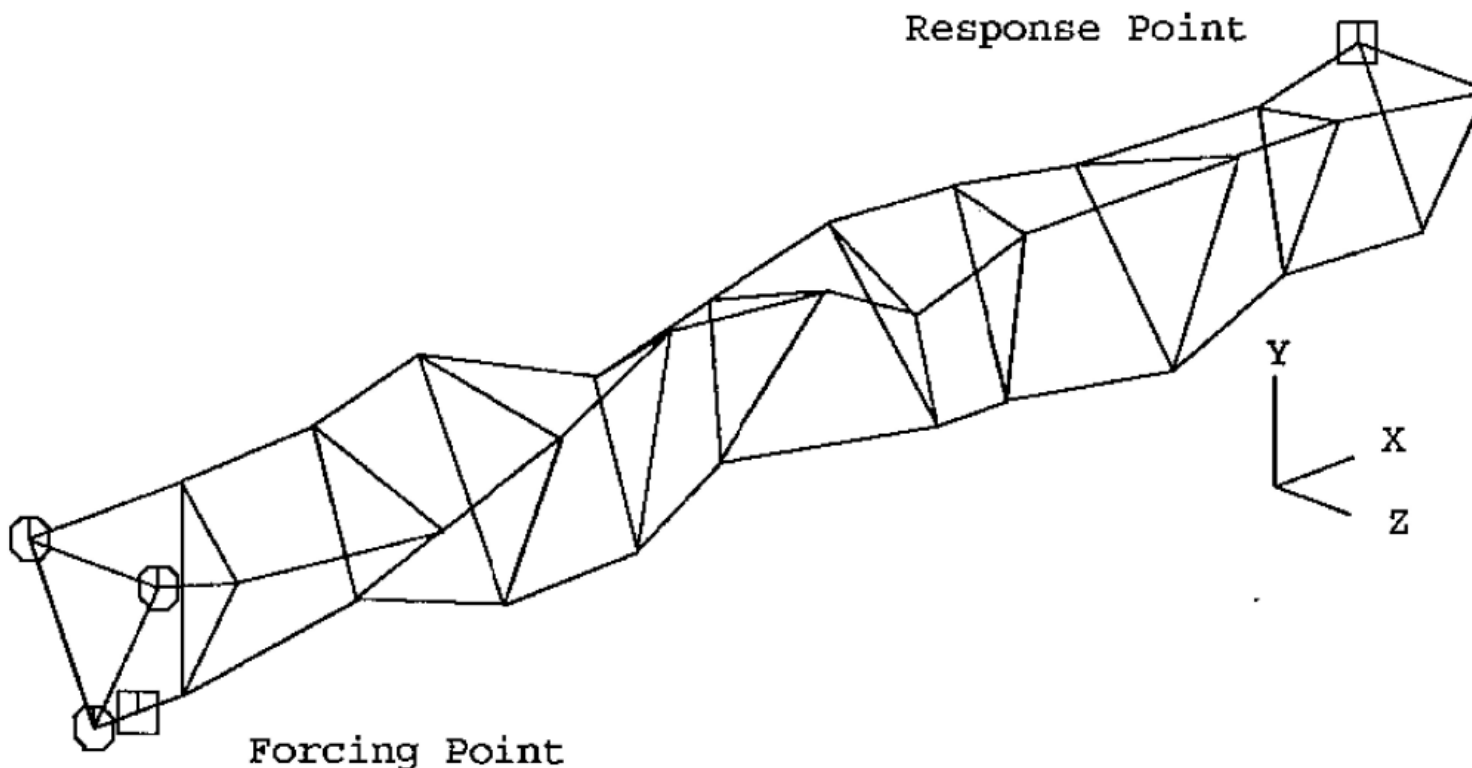
Bildquelle: A. E. Eiben, J.E. Smith: Introduction to Evolutionary Computing

Einführung

Wofür werden genetische Algorithmen eingesetzt ?

Industrielles Design

Die mit einem genetischen Algorithmus entwickelte Struktur ist über 20000% besser als traditionelle Strukturen. Sie enthält keine Symmetrien und es sind auch keine sonstigen bekannten Designprinzipien erkennbar.



Bildquelle: A. E. Eiben, J.E. Smith: Introduction to Evolutionary Computing

Einführung

Wofür werden genetische Algorithmen eingesetzt ?

Stundenplan-Erstellung (*timetabling problems*)

Hochschule Darmstadt Fachbereich Informatik		D14/211 (16 Plätze)				Stundenplan
	Montag	Dienstag	Mittwoch	Donnerstag	Freitag	
08:30 - 10:00						
10:15 - 11:45	Alexander del Pino P: Agile Software Development (english) y Master 2006 AS	Alexander del Pino P: Genetische Algorithmen y Bachelor WP Inf.				
12:00 - 13:30	Alexander del Pino P: Agile Software Development (english) y Master 2006 AS Alexander del Pino P: Quality Management (english) x Master 2006 AS	Alexander del Pino P: Genetische Algorithmen y Bachelor WP Inf.				
14:15 - 15:45				Alexander del Pino P: Softwaretechnik I Bachelor 2C		
16:00 - 17:30				Alexander del Pino P: Softwaretechnik I Bachelor 2C		
17:45 - 19:15						

Bildquelle: fbi.h-da Hinweis: Dieser Stundenplan wurde **nicht** mit einem genetischen Algorithmus erstellt.

Einführung

Wofür werden genetische Algorithmen eingesetzt ?

Stundenplan-Erstellung (*timetabling problems*)

Gegeben sei jeweils eine endliche Menge von Dozenten, von Räumen, von Veranstaltungen und von Zeitblöcken.

Gesucht ist eine Zuordnung von Dozenten, Räumen, Veranstaltungen und Zeitblöcken, so dass eine ganze Reihe von Randbedingungen erfüllt ist.

Beispiele für harte Randbedingungen (*hard constraints*):

- Ein Dozent kann zu einem Zeitblock nur einem Raum und einer Veranstaltung zugeordnet werden.
- Pflichtveranstaltungen im gleichen Semester dürfen nicht dem gleichen Zeitblock zugeordnet werden.
- Bestimmte Veranstaltungen können nur einem bestimmten Raum, z.B. CASE-Labor zugeordnet werden.

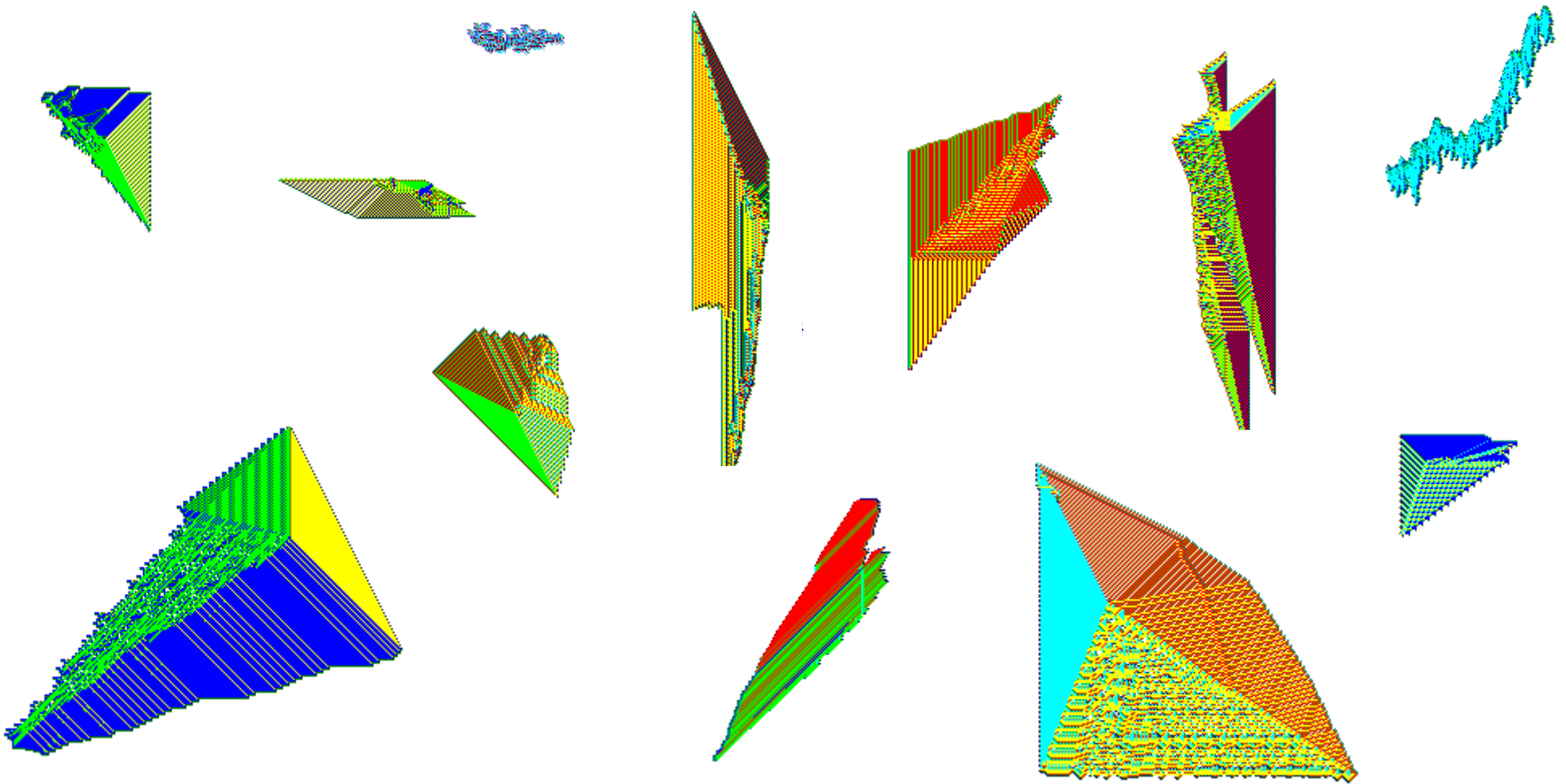
Beispiele für weiche Randbedingungen (*soft constraints*):

- Keine schweren Vorlesungen Freitags nachmittags.
- Ein Studientag für die Dozenten freihalten.

Einführung

Wofür werden genetische Algorithmen eingesetzt ?

Für das genetische Algorithmen Praktikum.



Einführung

Praktikumsaufgabe 1 - Gridmaschinen

Eine Gridmaschine ist eine Turingmaschine, die auf einem 2-dimensionalen Gitter anstatt auf einem 1-dimensionalen Band arbeitet.

Übergangsfunktion:

$$f : (Z_{alt}, S_{alt}) \rightarrow (Z_{neu}, S_{neu}, B)$$

wobei

$$B \in \{ N, NO, O, SO, S, SW, W, NW \}$$

und

$$S \in \{ 0, 1 \} \quad Z \in \{ 0, \dots, n-1, STOP \}$$

Wieviele Einsersymbole kann eine Gridmaschine mit $n=5$ Zuständen maximal auf ein leeres Gitter schreiben und dann anhalten? Finden Sie mit Hilfe eines genetischen Algorithmus eine Näherungslösung für diese Frage.

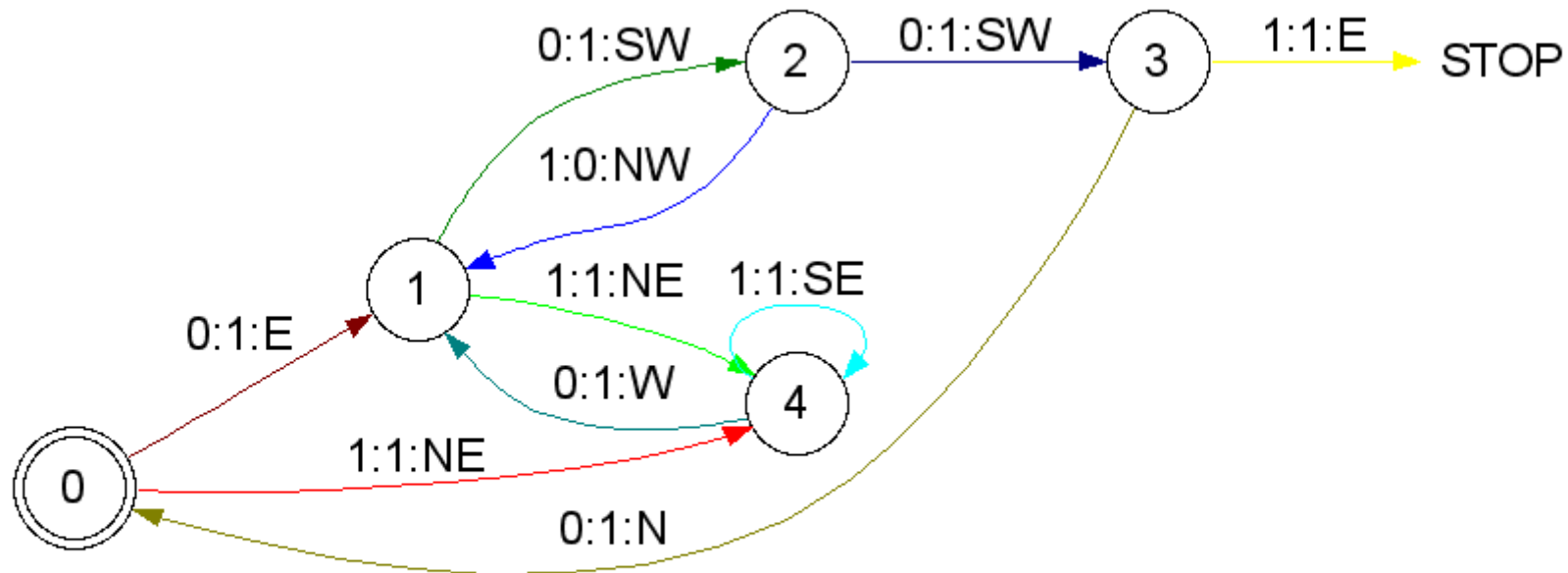
Hinweis: Dieses Problem ist nicht berechenbar. Vergleichen Sie dies auch mit dem Problem der „fleißigen Biber“.

Einführung

Praktikumsaufgabe 1 - Gridmaschinen

Beispiel

Diese Maschine produziert nach 37997 Schritten 16879 Einsersymbole wenn Sie im Zustand 0 auf einem leeren Gitter gestartet wird. Sie kann wie folgt dargestellt werden:



e3029ac2ec6d2498b838e365c19a7d2d: Stop after 37997 steps, 16879 nonzero symbols

Einführung

Praktikumsaufgabe 1 - Gridmaschinen

Beispiel

Die Übergangstabelle sieht wie folgt aus:

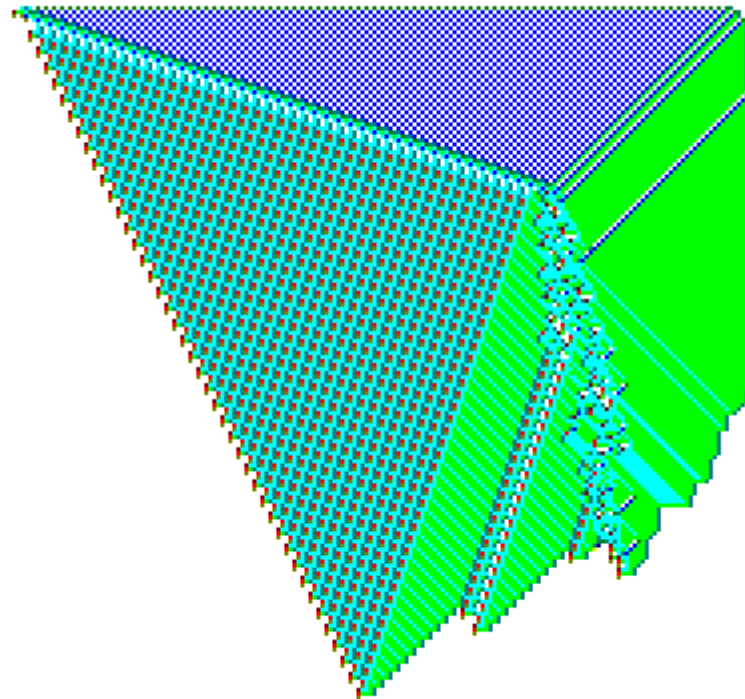
Z_{alt}	S_{alt}	Z_{neu}	S_{neu}	B
0	0	1	1	East
0	1	4	1	NorthEast
1	0	2	1	SouthWest
1	1	4	1	NorthEast
2	0	3	1	SouthWest
2	1	1	0	NorthWest
3	0	0	1	North
3	1	STOP	1	East
4	0	1	1	West
4	1	4	1	SouthEast

Einführung

Praktikumsaufgabe 1 - Gridmaschinen

Beispiel

Wenn man die Zellen mit der Farbe einfärbt, die zu der Regel gehört die dieses Einsersymbol erzeugt hat, dann sieht das Gitter wie folgt aus wenn die diese Maschine anhält:



Einführung

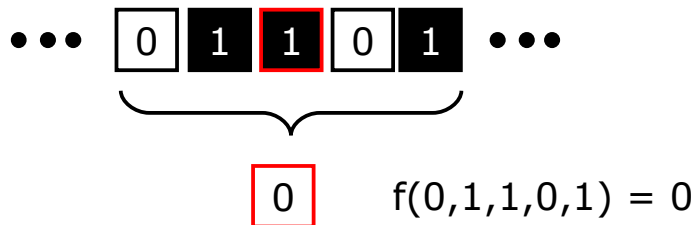
Praktikumsaufgabe 2 – Zelluläre Automaten

Ein zellulärer Automat ist ein Modell für ein System, in dem der Zustand einer Zelle nur von seinem aktuellen Zustand und dem seiner unmittelbaren Nachbarzellen abhängt.

Es gibt also nur eine lokale Beschreibung des Verhaltens, keine globale.

Beispiel:

1-dimensionaler zellulärer Automat mit je $n=2$ Nachbarzellen links und rechts, jede Zelle kann den Zustand 0 oder 1 annehmen:



Ohne Berücksichtigung von Symmetrien gibt es 2^{32} solcher Automaten.

Jeder Automat wird durch eine Tabelle mit 32 Einträgen beschrieben.

Einführung

Praktikumsaufgabe 2 – Zelluläre Automaten

Gibt es trotz der lokalen Verhaltensdefinition Automaten, die ein globales Verhalten zeigen ?

- Angenommen, ein 1-dimensionaler Automat wird gestartet nachdem jede Zelle zufällig entweder mit einer Null oder einem Einsersymbol initialisiert wurde.
- Gibt es Automaten, die im Laufe der Zeit alle Zellen mit Einsersymbolen füllen, wenn in der Anfangskonfiguration mehr Einsersymbole als Nullen waren? Andernfalls soll der Automat im Laufe der Zeit alle Zellen mit Null füllen.
- Ein solcher Automat wäre dann also in der Lage, herauszufinden, welches Symbol in einer zufälligen Anfangskonfiguration in der Mehrheit vorhanden war (*majority problem*), dies wäre ein Beispiel für eine globale Frage die mit Hilfe eines Systems beantwortet wird, welches nur durch lokale Regeln definiert ist.
- Aufgabe: Finden Sie mit Hilfe eines genetischen Algorithmus einen zellulären Automaten mit $n=2$, der das Majority-Problem am besten löst.

Einführung

Praktikumsaufgabe 2 – Zelluläre Automaten

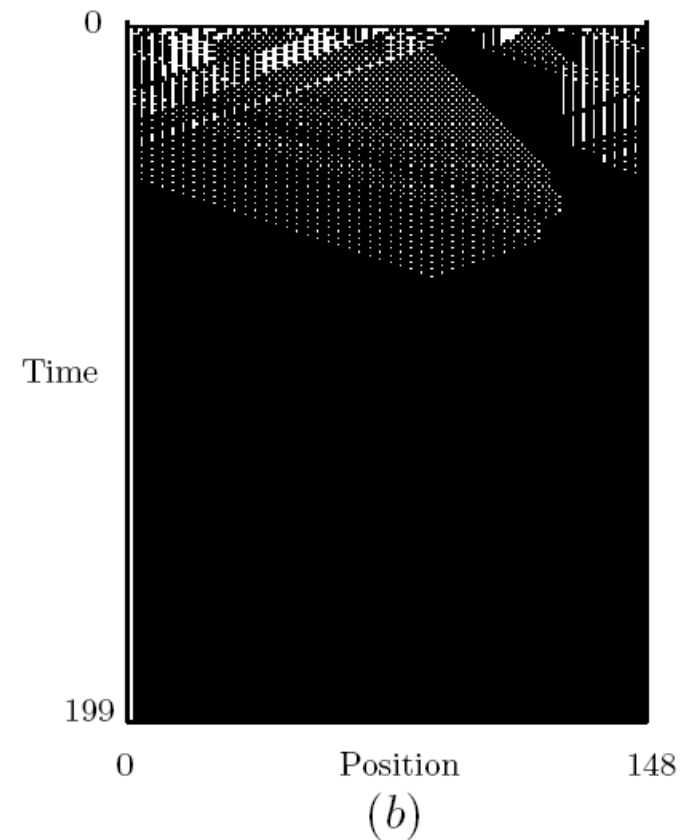
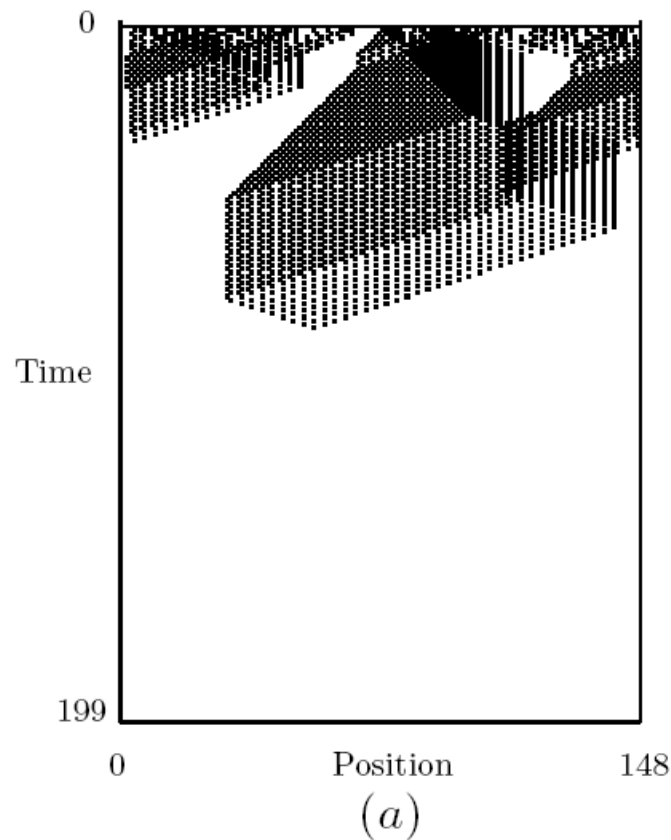
Beispiel: Ein Gitter ist 149 Zellen breit und mit kann entweder die Werte 0 („Weiß“) oder 1 („Schwarz“) annehmen. Kommen in der Anfangskonfiguration mehr weiße oder mehr schwarze Zellen vor ?

Linkes Bild:

74 schwarze Zellen

Rechtes Bild:

75 schwarze Zellen



Bildquelle: R. Breukelaar, Th. Bäck: „Using a Genetic Algorithm to Evolve Behavior in Multi-Dimensional Cellular Automata“, GECCO'05 Conference Proceedings, June 25–29, 2005

Einführung

Praktikumsaufgabe 3 - Proteinfaltung

Proteine bestehen vereinfacht gesagt aus einer Kette von Aminosäuren die entweder wasserliebend (*hydrophil*) oder wassermeidend (*hydrophob*) sind.

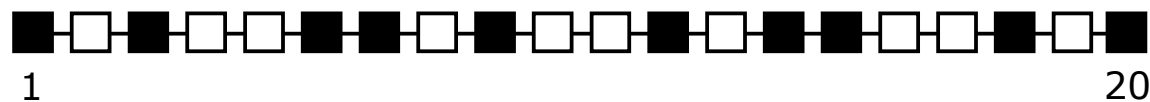
Viele Proteine werden so in eine 3-dimensionale Raumstruktur gefaltet, dass die hydrophoben Aminosäuren eher in der Mitte des Proteins sind, während die hydrophilen Aminosäuren eher außen sind.

Das **2d HP-Modell** ist eine starke Vereinfachung dieses äußerst komplizierten Vorgangs.

Betrachtet wird die Faltung (*Konformation*) einer Kette, deren Einzelelemente entweder hydrophob (schwarz) oder hydrophil (weiß) sind. Diese Faltung findet auf einem 2-dimensionalen Gitter statt:

Beispiel:

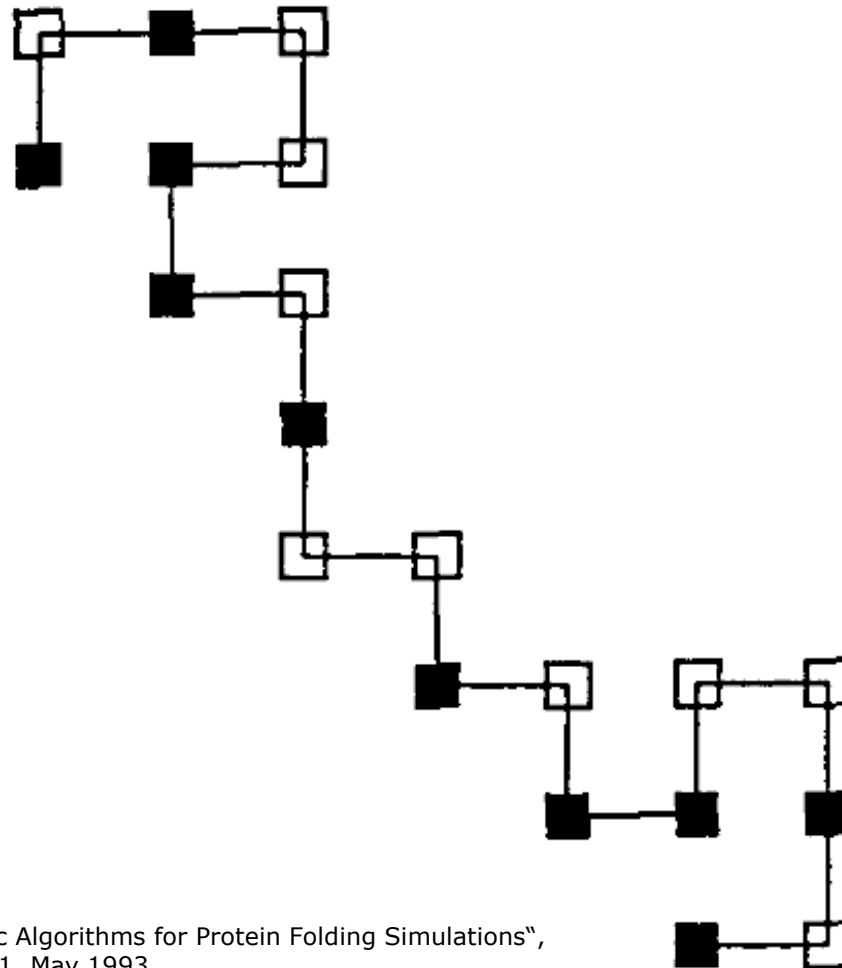
20-gliedrige Kette:



Einführung

Praktikumsaufgabe 3 - Proteinfaltung

Eine solche Kette kann an jeder Stelle im rechten Winkel gefaltet werden, so dass jedes Kettenelement auf einer Zelle in dem 2-dimensionalen Gitter platziert wird.

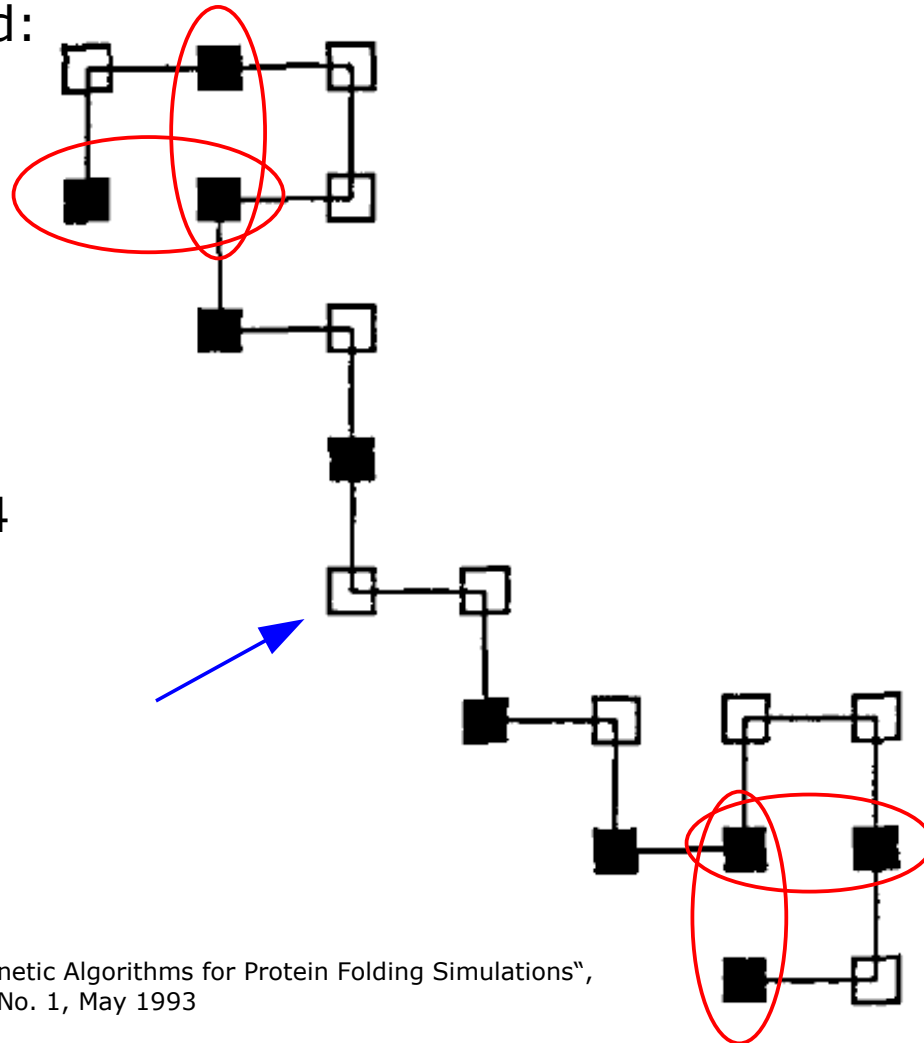


Bildquelle: Ron Unger, John Moulton: „Genetic Algorithms for Protein Folding Simulations“,
Journal of Molecular Biology, Vol. 231, No. 1, May 1993

Einführung

Praktikumsaufgabe 3 - Proteinfaltung

Die minimale Energie einer Faltung ergibt sich durch Paare von hydrophoben Elementen die an benachbarten Zellen im Gitter platziert werden ohne dass sie Sequenznachbarn sind:



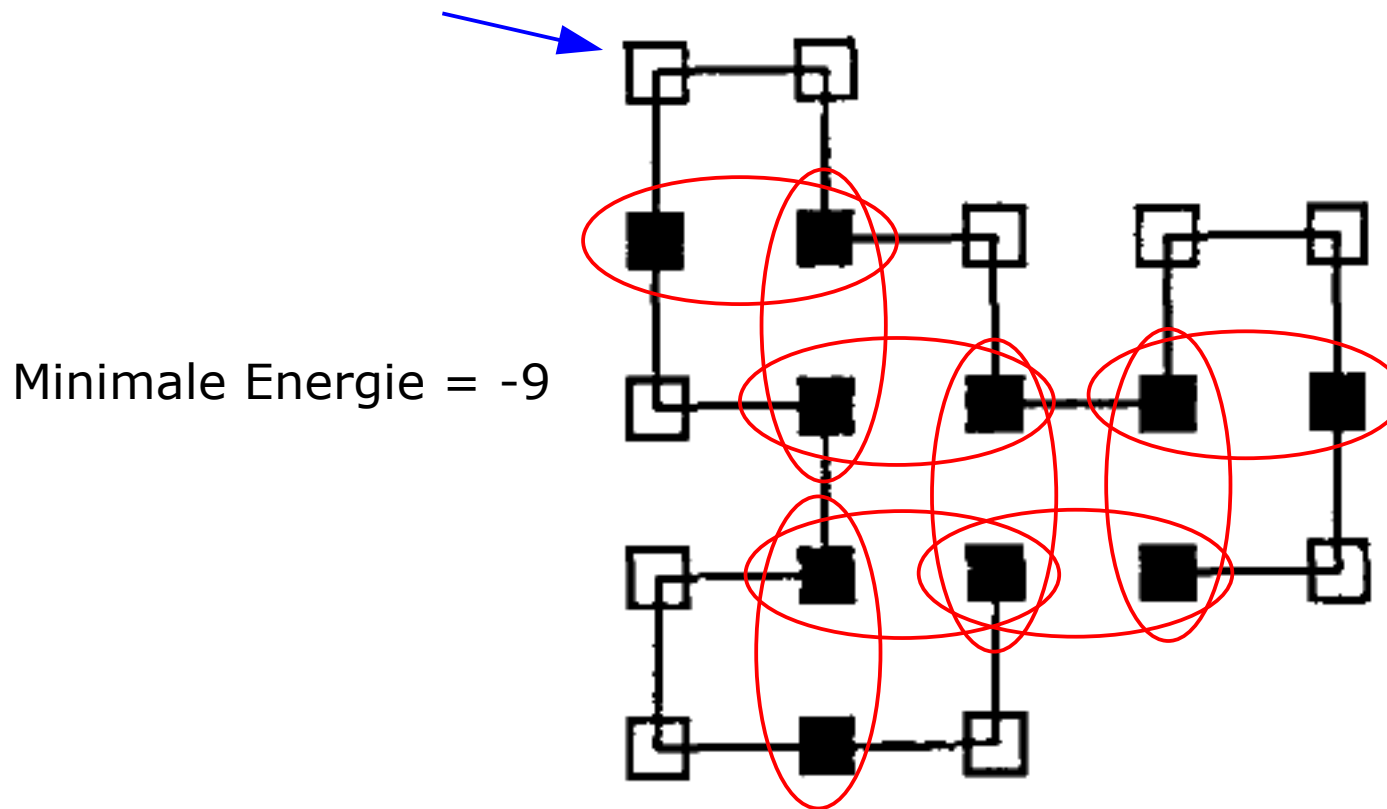
Minimale Energie = -4

Bildquelle: Ron Unger, John Moulton: „Genetic Algorithms for Protein Folding Simulations“,
Journal of Molecular Biology, Vol. 231, No. 1, May 1993

Einführung

Praktikumsaufgabe 3 - Proteinfaltung

Bei einer anderen Konformation ergibt sich eine wesentlich niedrigere Energie, diese Konformation wäre also besser:



Bildquelle: Ron Unger, John Moulton: „Genetic Algorithms for Protein Folding Simulations“,
Journal of Molecular Biology, Vol. 231, No. 1, May 1993

Einführung

Zusammenfassung Praktikumsaufgaben

Aufgabe 1 - Gridmaschinen

Finden Sie mit Hilfe eines genetischen Algorithmus eine Näherung für eine nicht berechenbare Funktion.

Aufgabe 2 – Zelluläre Automaten

Finden Sie mit Hilfe eines genetischen Algorithmus eine Näherung für einen zellulären Automaten, der ein gewünschtes globales Verhalten zeigt.

Aufgabe 3 – Proteinfaltung

Finden Sie mit Hilfe eines genetischen Algorithmus eine gute Konformation eines Proteins nach dem 2d HP-Modell.

Aufgabe 4 – Nach Absprache

Finden Sie mit Hilfe eines genetischen Algorithmus die Lösung für ein Problem Ihrer Wahl.

Einführung

Hausaufgaben

- Finden Sie heraus wann die ersten Hominiden erschienen, und schätzen Sie ab, wieviele Generationen notwendig waren bis Sie auf die Welt kamen.
- Finden Sie Beispiele wo der Begriff Evolution nicht in seiner biologischen Bedeutung verwendet wird.
- Finden Sie Beispiele für Probleme, wo eine Näherungslösung ausreichend ist.