

Vorlesung

Simulated Annealing

Prof. Karsten Held

IFP – AG Computational Materials Science

Problem

Finde *Minimum* einer Funktion $E(\mathbf{x})$

für komplexes Problem d.h. $\mathbf{x} = \{x_1 \dots x_d\}$ mit $d \gg 1$

Lösung

Simulated annealing (Kirkpatrick et al.'83)

es gibt viele andere ...

Beispiel: Travelling-Salesman-Problem I

Problem

Handelsreisender muss *d* Städte besuchen.
Wegstrecke soll minimiert werden.



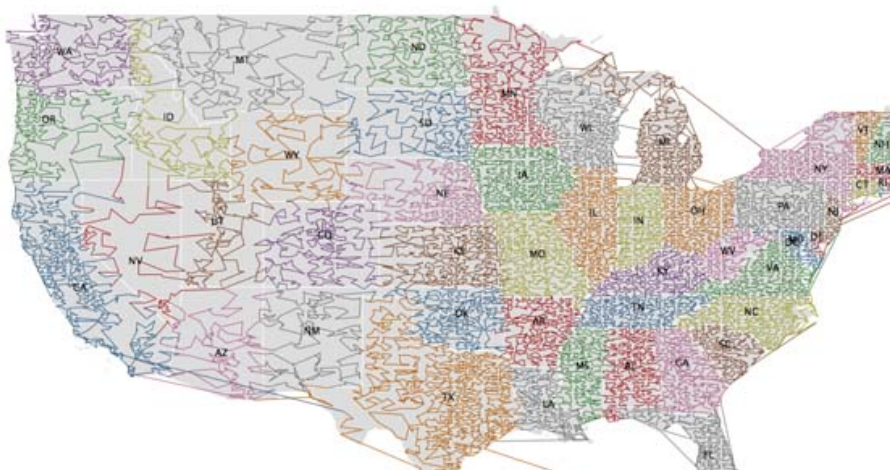
Brute force
(alles Ausprobieren):
Aufwand $t \sim d!$

Beispiel: Travelling-Salesman-Problem II

US-Präsidentschaftskandidat soll alle ZIP-Codes besuchen

$d = 3700$

Aufwand $t \sim 3700!$



Beispiel: Travelling-Salesman-Problem III

Travelling-Salesman-Problem ist *das* NP-Problem*
(NP: nicht polynomial)

Brute force: Aufwand $t \sim d!$

Graphentheorie/dynamical programming: $t \sim 2^d \text{polynom}(d)$

Computerwissenschaft

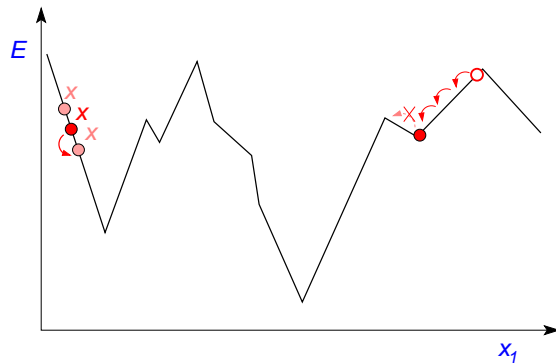
unterscheidet P (polynomial) und NP-Probleme

NP-Probleme werden auch mit schnelleren Rechnern (und Moore's law) nicht lösbar sein!

* *nicht bewiesen, Annahme da nach vielen Jahren kein Algorithmus gefunden.*

Hill climbing (Bergsteigen)

lokale Optimierung; hier eigentlich “valley descent” (Talabstieg)



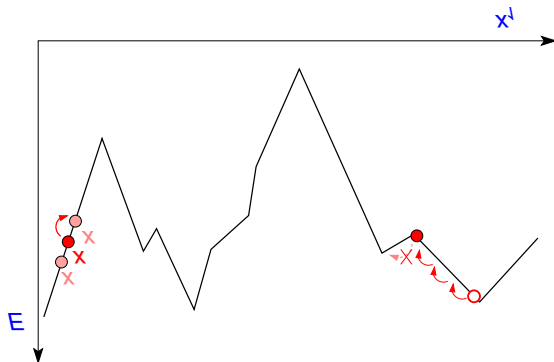
Nachteil: Wir können in lokalem Minimum stecken bleiben!

verwandt: **Steepest descent (Gradientenverfahren)**

$$\mathbf{x}' = \mathbf{x} - \alpha \frac{\partial E(\mathbf{x})}{\partial \mathbf{x}} \text{ (statt fixe Schrittweite)}$$

Hill climbing (Bergsteigen)

lokale Optimierung; hier eigentlich “valley descent” (Talabstieg)



Nachteil: Wir können in lokalem Minimum stecken bleiben!

verwandt: **Steepest descent (Gradientenverfahren)**

$$\mathbf{x}' = \mathbf{x} - \alpha \frac{\partial E(\mathbf{x})}{\partial \mathbf{x}} \text{ (statt fixe Schrittweite)}$$

Simulated Annealing

Idee

Lokale Optimierung

***aber** erlaube* auch lokal schlechtere Lösungen,
um **andere** lokale/globale Minima zu erreichen*

* mit gewisser Wahrscheinlichkeit

Algorithmus Simulated Annealing

- 1 Wähle Anfangs-Konfiguration x
- 2 Tue $N_{\text{warm-up}}$ mal: **sweep**
- 3 Tue N_{mess} mal
 - **sweep**
 - $|m|_{\text{sum}+} = |m|_{\{\sigma_I\}}$
- 4 Berechne Mittelwerte und Fehler ...

Sweep

Tue d (Dimensionen des Vektors x) mal

- Wähle x' "lokaler" Nachbar von x
- Berechne $\Delta E = E(x') - E(x)$, $r = e^{-\frac{\Delta E}{T}}$
- Wenn Zufallszahl $\text{rand} \in [0, 1) < \min\{1, r\}$,
 $x = x'$; sonst behalte x

Algorithmus Simulated Annealing

- 1 Wähle Anfangs-Konfiguration x und Temperatur $T = T_i$
- 2 Tue $N_{\text{warm-up}}$ mal: **sweep**
- 3 Solange $T > T_f$: $T = T/\Lambda$, $\Lambda > 1$, gehe zu (2)
- 4 x ist approx. Minimum

Sweep

Tue d (Dimensionen des Vektors x) mal

- Wähle x' "lokaler" Nachbar von x
- Berechne $\Delta E = E(x') - E(x)$, $r = e^{-\frac{\Delta E}{T}}$
- Wenn Zufallszahl $\text{rand} \in [0, 1) < \min\{1, r\}$,
 $x = x'$; sonst behalte x

Algorithmus Simulated Annealing

- 1 Wähle Spin-Konfiguration $x = \{\sigma_I\}$
- 2 Tue $N_{\text{warm-up}}$ mal: **sweep**
- 3 Tue N_{mess} mal
 - **sweep**
 - $|m|_{\text{sum}+} = |m|_{\{\sigma_I\}}$
- 4 Berechne Mittelwerte und Fehler ...

Sweep

Tue N (Anzahl der Gitterplätze) mal

- Wähle Gitterplatz i , x' ist gleich x außer $\sigma_i \rightarrow \sigma'_i$
- Berechne $\Delta E = E_{\{\sigma_1 \dots \sigma'_i \dots \sigma_N\}} - E_{\{\sigma_I\}}$, $r = e^{-\frac{\Delta E}{T}}$
- Wenn Zufallszahl $\text{rand} \in [0, 1) < \min\{1, r\}$,
 ändere ($\sigma_i \rightarrow \sigma'_i$); sonst behalte σ_i

Algorithmus Simulated Annealing

- 1 Wähle Spin-Konfiguration $x = \{\sigma_I\}$ und Temperatur $T = T_i$
- 2 Tue $N_{\text{warm-up}}$ mal: **sweep**
- 3 Solange $T > T_f$: $T = T/\Lambda$, $\Lambda > 1$, gehe zu (2)
- 4 $x = \{\sigma_I\}$ ist approx. Minimum

Sweep

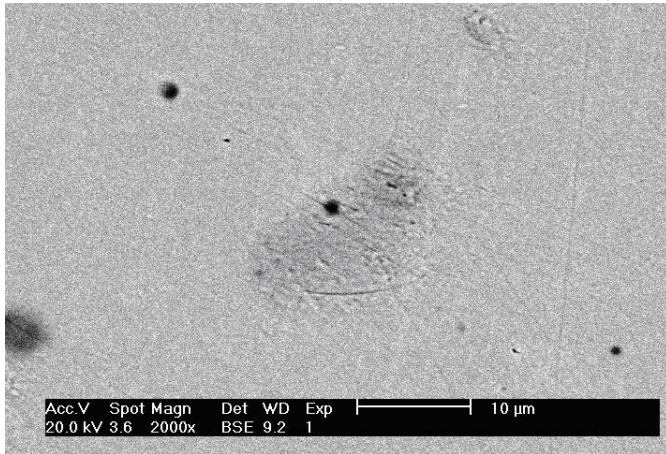
Tue N (Anzahl der Gitterplätze) mal

- Wähle Gitterplatz i , x' ist gleich x außer $\sigma_i \rightarrow \sigma'_i$
- Berechne $\Delta E = E_{\{\sigma_1 \dots \sigma'_i \dots \sigma_N\}} - E_{\{\sigma_I\}}$, $r = e^{-\frac{\Delta E}{T}}$
- Wenn Zufallszahl $\text{rand} \in [0, 1) < \min\{1, r\}$,
ändere ($\sigma_i \rightarrow \sigma'_i$); sonst behalte σ_i

Warum annealing (Glühen/Ausheilen)?

Simulated annealing simuliert Prozess aus Festkörperphysik und Werkstoffkunde:

z.B. $\text{Ce}_3\text{Pd}_{20}\text{Si}_6$ Kristalle (von A. Prokofiev IFP)



Warum annealing (Glühen/Ausheilen)?

Simulated annealing simuliert Prozess aus Festkörperphysik und Werkstoffkunde:

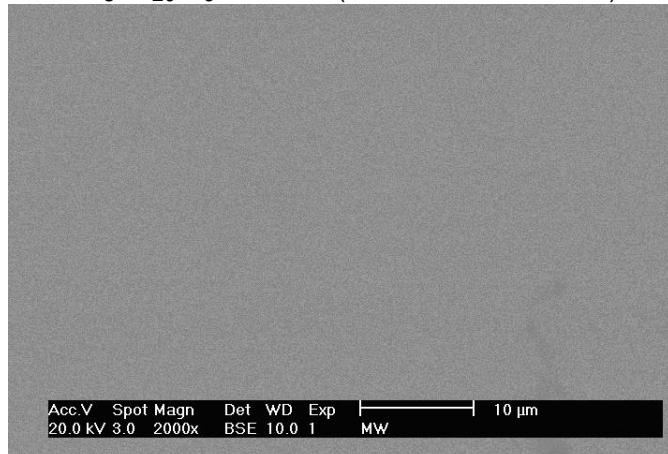
z.B. $\text{Ce}_3\text{Pd}_{20}\text{Si}_6$ Kristalle (von A. Prokofiev IFP)



Warum annealing (Glühen/Ausheilen)?

Simulated annealing simuliert Prozess aus Festkörperphysik und Werkstoffkunde:

z.B. $\text{Ce}_3\text{Pd}_{20}\text{Si}_6$ Kristalle (von A. Prokofiev IFP)



Vor- und Nachteile von Simulated Annealing

Vorteile

- + bleibt nicht in **lokalem Minimum** hängen
- + **flexibel**, einfach
anwendbar auf jedes Minimierungsproblem $E(x)$
- + anwendbar auf nicht-lineare Modelle, Rauschen, chaotisches Verhalten
- + findet **globales Minimum**, wenn ∞ -**lange** simuliert wird
- + oft $E_{SA}(x) \approx E_{min}$; dies ist oft ausreichend

Nachteile

- globales Minimum nicht in endlicher Zeit
- **Parameter T_i , T_f , Λ , Nachbarschaftsdef.** müssen an Problem(klasse) angepasst werden (metaheuristisch)

Alternativen I

Brute force

- + findet globales Minimum
- nur in exponentiell langer Zeit $t \sim d!$ (Trav.Salesm.)

Dynamical programming

- + findet globales Minimum
- + immerhin $d \sim 40 - 50$ möglich
- nur in exponentiell langer Zeit $t \sim 2^d$ (Trav.Salesm.)
- nur $d \sim 40 - 50$ möglich

Hill climbing/steepest descent

- + sehr einfach
- findet nur lokales Minimum
- f. Gradientenverfahren $E'(x)$ benötigt

Alternativen II

Genetic algorithm

parents: $\mathbf{x} = \{x_1, x_2, x_3, \dots, x_d\}$, $\mathbf{y} = \{y_1, y_2, y_3, \dots, y_d\}$,

→ children: $\mathbf{x}' = \{y_1, x_2, x_3, \dots, x_d\}$, $\mathbf{y}' = \{x_1, y_2, y_3, \dots, x_d\}$,

$\mathbf{z}' = \{x_1, x_2 \pm \Delta, y_3, \dots, x_d\}$

(survival of the) **fittest** → **neue parents**

- + sucht in verschiedensten lokalen Minima gleichzeitig
- findet **globales Minimum** nicht unbedingt

Neuronale Netzwerke

ähnlich Ising-Modell,

aber Kopplungen J_{ij} (**Neuronen**) werden **trainiert/gelernt**

(so dass für Trainings-Set die Lösungen verbessert werden)

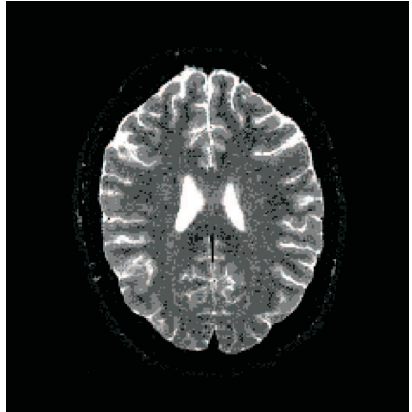
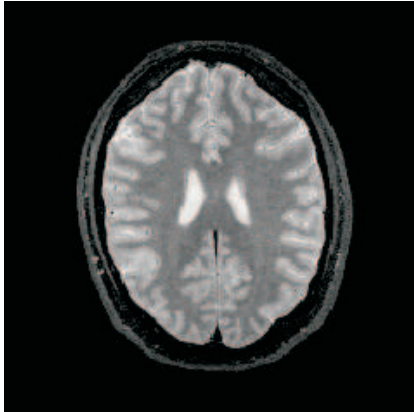
- + flexibel, **adaptiv**, approx. Lösung
- findet **globales Minimum** nicht unbedingt
- keine Parameteranpassung/Modell-Bildung aber **Training**

Übung: Segmentierung von NMR-Bildern

Aufgabe: Entscheide was ist $\sigma_i \in \underbrace{\{\text{BG, WM, GM, CSF, BS}\}}_{=\{1\dots 5\}}$

T_1 -Zeit gewichtet

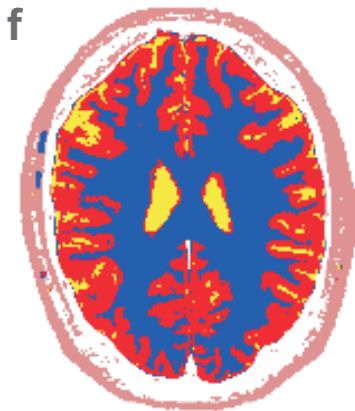
T_2 -Zeit gewichtet



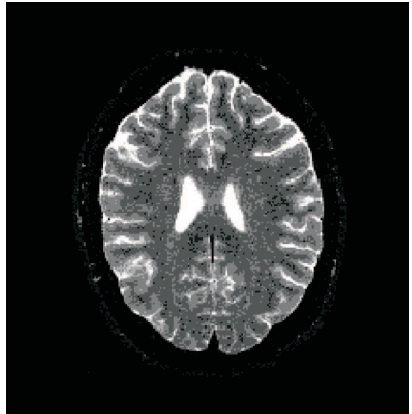
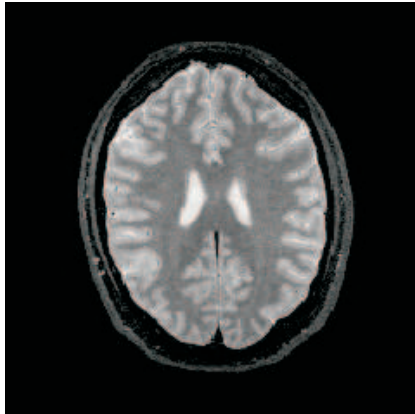
Übung: Segmentierung von NMR-Bildern

Aufgabe: Entscheide was ist $\sigma_i \in \underbrace{\{\text{BG, WM, GM, CSF, BS}\}}_{=\{1\dots 5\}}$

Segmentiertes Bild mit **Simulated annealing**



Übung: Segmentierung von NMR-Bildern



Rauschen (a.u.) (Gerätespezifisch: ± 30)

T1: WM: 823 ± 70 , GM: 1059 ± 95 , CSF: 1363 ± 177 , SB: 456 ± 120

T2: WM: 426 ± 59 , GM: 602 ± 102 , CSF: 1223 ± 307 , SB: 167 ± 69

Held et al.'96

Modell-Bildung

Gegeben: NMR Intensität $\mathbf{z} = \{z_i\}$

Gesucht: Gewebeart $\mathbf{x} = \{\sigma_i\}$, $\sigma_i \in \{BG, WM, GM, CSF, BS\}$

$$H = -J \sum_{\langle ij \rangle} \delta_{\sigma_i, \sigma_j} + \sum_i E_{z_i}(\sigma_i) \quad (1)$$

$$P(\{\sigma_i\}) \sim e^{-H(\{\sigma_i\})/(T)} \quad (2)$$

$T = 1$: samplen gem. Wahrsch.

$T = 0$: wahrscheinlichste Struktur (Ziel der Übung)

J : gleiche Nachbarn wahrscheinlicher gleiches Gewebe

$E_{z_i}(\sigma_i)$: Information aus dem gemessenen NMR-Bild
(\equiv Magnetisches Feld im Ising-Modell)

Bayes-Statistik

Frage

Was ist $E_{z_i}(\sigma_i)$?

bzw. $P(\sigma_i | z_i) = e^{-E_{z_i}(\sigma_i)/T}$

hier: $T = 1$ (reale Wahrscheinlichkeit)

Bayes-Statistik

$$P(z_i | \sigma_i) = \frac{1}{\sqrt{2\pi}\Sigma_{\sigma}} e^{-(z_i - \bar{z}_{\sigma_i})^2 / (2\Sigma_{\sigma_i}^2)} \quad (3)$$

\bar{z}_{σ_i} Mittelwert NMR-Intensität von Gewebe σ_i ; $\Sigma_{\sigma_i}^2$: Varianz

Aber z_i gegeben σ_i gesucht! bzw. $P(\sigma_i | z_i)$

$$P(\sigma | \mathbf{z}) P(\mathbf{z}) = P(\mathbf{z} | \sigma) P(\sigma) \quad (4)$$

$$P(\sigma | \mathbf{z}) \sim \underbrace{P(\sigma)}_{e^{-J \sum \langle ij \rangle \delta_{\sigma_i, \sigma_j}}} \underbrace{P(\mathbf{z} | \sigma)}_{e^{-\sum_i E_{z_i}(\sigma_i)}} \quad (5)$$

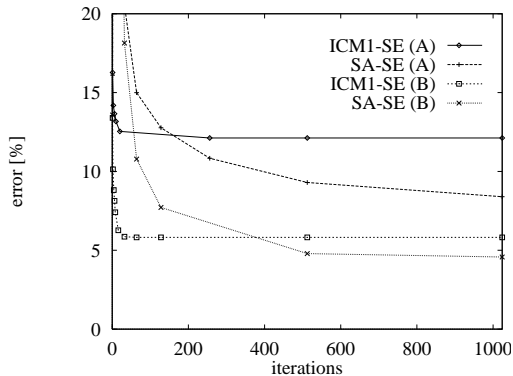
$P(\sigma)$ *a priori* Wahrsch. (kein NMR-Bild)

$P(\sigma | \mathbf{z})$ *a posteriori* Wahrsch. (mit NMR-Bild \mathbf{z})

$$H = -J \sum_{\langle ij \rangle} \delta_{\sigma_i, \sigma_j} + \sum_i (z_i - \bar{z}_{\sigma_i})^2 / (2\Sigma_{\sigma_i}^2) + \ln(\Sigma_{\sigma_i}) \quad (6)$$

Übung: Segmentierung von NMR-Bildern

Fehler: Simulated Annealing vs. Hill climber (ICM)



Literatur

Simulated Annealing

- S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, (1983).
"Optimization by Simulated Annealing". Science 220, 671 (4598);
www.sciencemag.org/content/220/4598/671.abstract
- W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling,
Numerical Recipes in Fortran77 (second edition,
Cambridge University Press, 1995). apps.nrbook.com
- F. Buseti, *Simulated annealing overview*,
[163.18.62.64/wisdom/Simulated annealing overview.pdf](http://163.18.62.64/wisdom/Simulated%20annealing%20overview.pdf)

Segmentation of MR images

K. Held *et al.*, IEEE Trans. Med. Imag. **16**, 878 (1997);
arxiv.org/abs/0903.3114