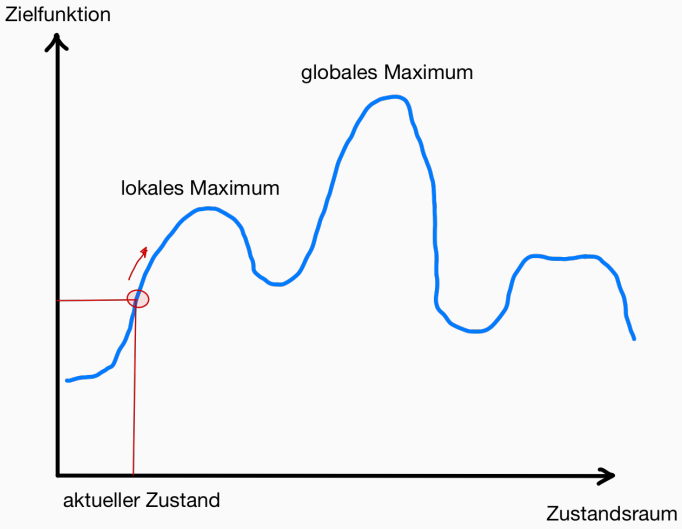
Lokale Suche: Simulated Annealing

Carsten Gips (FH Bielefeld)

Motivation



Problem: lokale Maxima und Plateaus

Gedankenexperiment: Problem "drehen" und Ball in Zustandsraum-Landschaft werfen

Pseudocode Simulated Annealing (Minimierungsproblem)

```
def simulated_annealing(problem):
    current = problem.startNode
    t = 0;  temp = schedule(t)

    while temp>0:
        temp = schedule(++t)
        neighbors = current.expandSuccessors()
        if not neighbors: return current
        working = random.choice(neighbors)
        dE = problem.value(current) - problem.value(working)
        if dE > 0 or probability(math.exp(dE/temp)):
            current = working

    return current
```

Anmerkung: Akzeptieren von Verschlechterungen

Abkühlungsplan problemabhängig wählen

- Initiale Temperatur: So hoch, daß praktisch jede Änderung akzeptiert wird
- Abkühlen: $T_k = \alpha T_{k-1}$ mit $0.8 \leq \alpha \leq 0.99$
  - Ausreichend langsam abkühlen!
  - Typisch: jede Stufe so lange halten, daß etwa 10 Änderungen akzeptiert wurden
- Stop: Keine Verbesserungen in 3 aufeinander folgenden Temperaturen

Eigenschaften: Vollständig und Optimal mit Wahrscheinlichkeit

Wrap-Up

Lokale Suchverfahren: Nur das Ergebnis zählt!

- Gradientenverfahren
  - Analogie Bergsteigen: Gehe in Richtung des stärksten Anstiegs der Kostenfunktion => **Hill-Climbing**
  - Achtung: Probleme mit lokalen Minima => **Simulated Annealing**

LICENSE

Exceptions

- Figure “Exp e.svg” by Marcel Marnitz, reworked by Georg-Johann on Wikimedia Commons (Public Domain)