Evolutionäre Algorithmen

Vorlesung 9

Lokale Suche
Bewertung der Standardalgorithmen

Lokale Suche

- ▷ evolutionärer Algorithmus mit Populationsgröße 1
- ▷ ein Nachkommen wird durch Mutation erzeugt
- ▷ Rekombination ist nicht möglich
- ▷ Elter oder Nachkomme wird als nächster Elter übernommen
  $\Rightarrow$ Verfahren unterscheiden sich bzgl. dieser Akzeptanzbedingung

Hillclimbing (HC)

- ▷ Nachkommen werden nur mit einer besseren Güte übernommen
- ▷ Nachteil: sehr anfällig für lokale Optima

Simulated Annealing (SA)

- ▷ Imitation des physikalischen Abkühlungsprozesses
- ▷ Wahrscheinlichkeit für Zustand x eines idealen Systems:
  $e^{-\frac{E(x)}{T}}$ mit Temperatur T und Energie $E(x)$
- ▷ schnelles Abkühlen: unregelmäßige Strukturen auf hohem Energieniveau
- ▷ langsames Abkühlen: regelmäßige Strukturen
- ▷ Wahrscheinlichkeit wird auf Akzeptanzwahrscheinlichkeit für Verschlechterungen übertragen

SA: Akzeptanzbedingung

- ▷ vordefinierter monoton fallender Abkühlungsplan: $(T_j)_{j\in\mathbb{N}}$ mit $T_j \in \mathbb{R}$ und $\lim_{j\to\infty} T_j = 0$
- ▷ Verbesserung: immer akzeptieren
- ▷ Verschlechterung: akzeptieren mit Wahrscheinlichkeit

$$e^{-\frac{|F(A),F(B)|}{T_{t-1}}}$$

- ▷ lokale Optima können immer mit gewisser Wahrscheinlichkeit verlassen werden

SA: Abkühlungsplan

- ▷ zu rasche Abkühlung: ähnliches Verhalten wie bei Hillclimbing
- ▷ zu langsame Abkühlung: geringe Konvergenzgeschwindigkeit
- ▷ gängige Vorgehensweise: hohe Starttemperatur und Abkühlen durch

$$T_{t+1} = T_t\alpha \quad \text{mit} \quad \alpha \in (0.8, 0.99)$$

- ▷ eher launisches Verfahren da Abkühlung sehr problöemabhängig ist

Threshold Accepting (TA)

- ▷ vordefinierte monoton fallende Folge: $(T_j)_{j\in\mathbb{N}}$ mit $T_j \in \mathbb{R}$ und $\lim_{j\to\infty} T_j = 0$
- ▷ in Generation t: ein neues Kindindividuum wird nur dann akzeptiert wenn es die Güte um höchstens T_t verschlechtern
- ▷ Startwert: $T_0 = \frac{1}{60}(F(C_1) + F(C_2) + F(C_3))$ mit Zufallsstichproben C_i

$$T_{t+1} = T_t\alpha \quad \text{mit} \quad \alpha \in (0.8, 0.995)$$

Sintflutalgorithmus (GD)

- ▷ absolute Akzeptanzschwelle bestimmt, welche Gütewerte nicht mehr übernommen werden
- ▷ Akzeptanzschwelle steigt linear pro Generation an

Record-to-Record-Travel (RR)

- ▷ beste bisher gefundene Güte wird gespeichert
- ▷ eine monoton fallende Folge gibt relativ zur besten Güte an, welche Werte nicht mehr akzeptiert werden

Tabu-Suche

- ▷ Geschichte der bisherigen Optimierung wird mitbetrachtet
  - – das beste Individuum wird gemerkt
  - – bisherige Lösungskandidaten beeinflussen die Nachbarschaft
- ▷ ähnlich zur $(1, \lambda)$-Strategie: Erzeugung von mehreren möglichen Kindern
- ▷ bestes Kind wird aktuelles Individuum und ersetzt ggf. das bisher beste Individuum

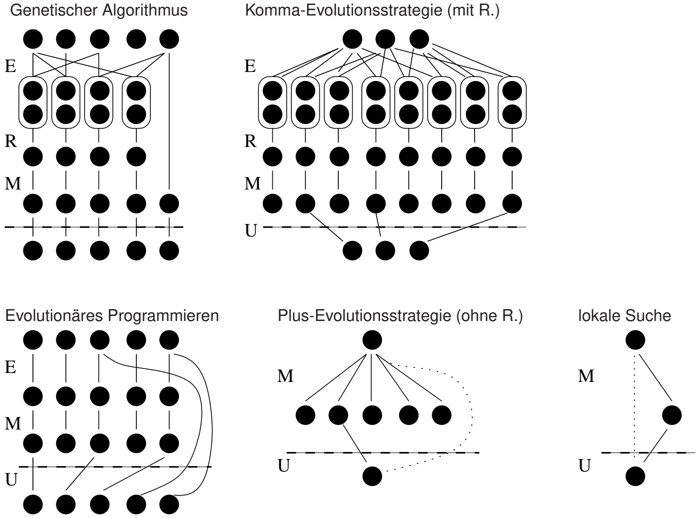
Tabu-Suche: Nachbarschaft

- ▷ Tabu-Liste
  - – FIFO-Warteschlange
  - – macht z.B. die Rückkehr zum Vorgängerindividuum unmöglich
- ▷ Bestrebungs-Liste: besonders vorteilhafte Veränderungen werden zukünftig bevorzugt
- ▷ ein neuer Lösungskandidate muss
  - – alle Tabu-Bedingungen respektieren oder
  - – wenigstens eine Bestrebungs-Bedingung erfüllen

Standardverfahren: Spiel

- ▷ Fragen zu
  - – Rolle der Operatoren
  - – Populationsgröße
  - – Selektion
  - – wenn nichts anderes hilft: Repräsentation
- ▷ Antworten: ja/nein

Standardverfahren: Übersicht



Standardverfahren: Übersicht

▷ Unterschiede:
  – Repräsentation und Kodierung
  – Rolle der Operatoren
  – Selbstanpassung

▷ dennoch: nur Varianten voneinander
⇒ unterstreicht für Anwendungen: nicht Standardverfahren sollten angepasst werden, sondern Konstruktion des „besten“ Verfahrens

EA vs. mathematische Optimierung

▷ zunächst mathematische Verfahren prüfen:
Gradientenabstieg, lineares Programmieren, Dijkstra, etc.

▷ diese sind oft nicht anwendbar, da erforderliche mathematische Formulierung nicht möglich ist

▷ ebenso problematisch bei
  – Randbedingungen
  – Mehrzieloptimierung
  – zeitabhängige oder verrauschte Zielfunktionen

EA vs. lokale Suche

▷ lokale Suche:
  – schnellere, direktere Nutzung von Neuerungen
  – keine Populationsverwaltung
  – oft besser geeignet als EA (etwa bei großem Einzugsgebiet des globalen Optimums)

▷ evolutionäre Algorithmen:
  – besser geeignet bei zerklüfteten Gütelandschaften

▷ zunächst immer erst lokale Suche ausprobieren!

Evolutionäre Algorithmen

- ▷ Vorteil: können leicht mit Heuristiken oder lokaler Suche kombiniert werden
- ▷ Nachteil: lange Berechnungszeiten (mögliches Gegenmittel: Einsatz von Parallelrechnern)
- ▷ Nachteil: garantieren nicht die Lösung des Problems
- ▷ Nachteil: letzte x% sind extrem schwierig zu erreichen
- ▷ Nachteil: Kunden möchten die gefundene Lösung und den Optimierungsprozess genre nachvollziehen