

Methoden und Anwendungen der Optimierung (MAO)

Kapitel 5: Metaheuristiken – Genetische Algorithmen

Univ.-Prof. Dr. Michael Schneider
Christian Schröder

Deutsche Post Chair – Optimization of Distribution Networks (DPO)
RWTH Aachen University

schroeder@dpo.rwth-aachen.de

WS 2017/18



Deutsche Post
Chair - Optimization of
Distribution Networks

RWTHAACHEN
UNIVERSITY

Gesamtgliederung

- 1 Einführung: Heuristiken, Komplexität
- 2 Greedy Algorithmen
- 3 Lösungsqualität und Approximation
- 4 Lokale Suche
- 5 **Metaheuristiken**
 - Einführung
 - ILS
 - VND, VNS
 - Tabu Search
 - Large Neighborhood Search
 - **Genetische Algorithmen**

5 Metaheuristiken – Genetische Algorithmen

Struktur genetischer Algorithmen

- **Initialisierung:** Erzeugung einer Population von Individuen (Startpopulation): zufällig, oder random-greedy
- **Evaluation:** Auswertung der Fitness der Individuen: bestimmt die Fortpflanzungswahrscheinlichkeit
- **Rekombination:** Crossover-Operator lässt aus zwei Chromosomen zwei neue Chromosomen entstehen
- **Reproduktion:** Klonen
- **Mutation:** Mutation verhindert eine frühzeitige Konvergenz des Algorithmus und stellt die „genetische“ Vielfalt sicher

Schlüsselfaktoren zur Konstruktion genetischer Algorithmen

- Repräsentation des Lösungsraums
 - Binär-Codierung oder anwendungsspezifisch
 - Prinzip des minimalen Alphabets
 - Repair vs. Strafkosten
- Selektion
 - proportional zur Fitness
 - nach Rang
- Populationsmodelle
 - diskrete Generationen
 - überlappende Generationen

Rekombination / Crossover

- wichtigste Methode der Evolution, Mutation nur mit geringer Wahrscheinlichkeit
- Eltern mit einer hohen Fitness produzieren ein Kind mit noch besserer Fitness
- 1-Punkt-Crossover kann nicht einfach auf Permutationsprobleme angewandt werden: ungültige Lösungen

Genetische Algorithmen für TSP: Problemrepräsentationen

- Adjazenz-Repräsentation: Stadt j wird an Stelle i gespeichert, wenn in Lösung Kante $\{i, j\}$ existiert
 - Beispiel: 3 5 7 6 4 8 2 1 \rightarrow 1-3-7-2-5-4-6-8-1
 - Problem:

Genetische Algorithmen für TSP: Problemrepräsentationen

- Adjazenz-Repräsentation: Stadt j wird an Stelle i gespeichert, wenn in Lösung Kante $\{i, j\}$ existiert
 - Beispiel: 3 5 7 6 4 8 2 1 \rightarrow 1-3-7-2-5-4-6-8-1
 - Problem: Zyklen, z. B. 3 5 7 6 2 4 1 8

Genetische Algorithmen für TSP: Problemrepräsentationen

- Adjazenz-Repräsentation: Stadt j wird an Stelle i gespeichert, wenn in Lösung Kante $\{i, j\}$ existiert
 - Beispiel: 3 5 7 6 4 8 2 1 \rightarrow 1-3-7-2-5-4-6-8-1
 - **Problem:** Zyklen, z. B. 3 5 7 6 2 4 1 8
 - empirisch schlechte Resultate

Genetische Algorithmen für TSP: Problemrepräsentationen

- Adjazenz-Repräsentation: Stadt j wird an Stelle i gespeichert, wenn in Lösung Kante $\{i, j\}$ existiert
 - Beispiel: 3 5 7 6 4 8 2 1 \rightarrow 1-3-7-2-5-4-6-8-1
 - **Problem:** Zyklen, z. B. 3 5 7 6 2 4 1 8
 - empirisch schlechte Resultate
- Ordinal-Repräsentation (erlaubt die Anwendung des 1-Punkt-Crossover-Operators)
 - umständlich
 - empirisch schlechte Resultate

Genetische Algorithmen für TSP: Problemrepräsentationen

- Adjazenz-Repräsentation: Stadt j wird an Stelle i gespeichert, wenn in Lösung Kante $\{i, j\}$ existiert
 - Beispiel: 3 5 7 6 4 8 2 1 \rightarrow 1-3-7-2-5-4-6-8-1
 - **Problem:** Zyklen, z. B. 3 5 7 6 2 4 1 8
 - empirisch schlechte Resultate
- Ordinal-Repräsentation (erlaubt die Anwendung des 1-Punkt-Crossover-Operators)
 - umständlich
 - empirisch schlechte Resultate
- Pfad-Repräsentation: 1 2 3 4 5 6 7 8
 - natürlich
 - empirisch gute Resultate, aber spezielle Crossover nötig

Genetische Algorithmen für TSP: Crossover

- Partially-Matched Crossover-Operator (PMX)
- Order Crossover-Operator (OX)
- Cyclic Crossover-Operator (CX)

Partially-Matched Crossover-Operator (PMX)

- Zufällig Wahl zweier Crossover-Punkte, Austausch der Chromosomteile, die zwischen den Crossover-Punkten liegen
- Rest der Chromosome der Eltern wird so gut wie möglich erhalten

Beispiel:

Elternteil 1:	a	b	c	d	e	f	g	h	i	j
Elternteil 2:	c	f	g	a	j	b	d	i	e	h
Kind 1:	c	b	d	a	j	f	g	h	e	i
Kind 2:	a	f	c	g	e	b	d	i	h	j

- Absolute Positionen der Städte werden vererbt
- Geringer Rechenaufwand
- mäßige Ergebnisse für TSP

Order Crossover-Operator (OX)

- Zufällig Wahl zweier Crossover-Punkte, Austausch der Chromosomteile, die zwischen den Crossover-Punkten liegen
- Ordnung der Städte des jeweils anderen Elternteils wird so gut wie möglich erhalten

Beispiel:

Elternteil 1:	a	b	c	d	e	f	g	h	i	j
Elternteil 2:	c	f	g	a	h	b	d	i	e	j
Kind 1:	c	a	b	d	i	f	g	h	e	j
Kind 2:	c	e	f	g	h	b	d	i	j	a

- Relative Ordnung der Gene wird weitgehend vererbt
- Absolute Genpositionen bleiben kaum erhalten
- Geringer Rechenaufwand
- Gute Ergebnisse für TSP

Cyclic Crossover-Operator (CX)

Beispiel:

Elternteil 1: a b c d e f g h i j

Elternteil 2: c f g a h b d i e j

Kind 1: a f c d h b g i e j

Kind 2: c b g a e f d h i j

- Absolute Positionen der Gene bleiben erhalten
- Ordnung von Genen wird weniger gut vererbt
- Etwas größerer Rechenaufwand
- Weniger gute Ergebnisse für TSP