

# EA.01: Modellierung von GA

## Aufgabe: 8-Queens-Problem und Landkarten-Färbeproblem

Für beide Probleme sollen eine geeignete **Kodierung der Individuen**, passende **Operatoren (Crossover, Mutation)** und eine **Fitnessfunktion** angegeben werden, sodass die Probleme mit einem Genetischen Algorithmus (GA) lösbar sind.

### 1 Landkarten-Färbeproblem

#### Problem:

Färbe jede Region einer Landkarte mit einer Farbe, sodass benachbarte Regionen **nicht dieselbe Farbe** haben. Ziel ist eine konfliktfreie Einfärbung mit möglichst **wenig unterschiedlichen Farben**. Wir starten mit fünf verfügbaren Farben.

#### Kodierung der Individuen:

Jedes Individuum ist eine Liste der Farbzweisungen an die Regionen.

$$g = (A=\text{rot}, B=\text{blau}, C=\text{grün}, D=\text{gelb}, E=\text{rot}, F=\text{blau})$$

**Erklärung:** Jedes Gen entspricht einer Region. Der Wert ist die Farbe. Damit wird der gesamte Zustand vollständig beschrieben.

Region	A	B	C	D	E	F
Farbe	rot	blau	grün	gelb	rot	blau

#### Crossover:

Ein-Punkt-Crossover: Zwei Eltern werden an einer zufälligen Position geteilt und die hinteren Teile getauscht. Man muss beide an die gleiche Stelle schneiden (Hier habe wir in der Mitte geschnitten).

Eltern 1:  $[rot, blau, grün, gelb, rot, blau]$   
Eltern 2:  $[grün, rot, rot, blau, gelb, grün]$   
 $\Rightarrow$   $[rot, blau, grün, \text{blau, gelb, grün}], [grün, rot, rot, \text{gelb, rot, blau}]$

**Begründung:** Durch die Mischung der Farbzweisungen können Kombinationen entstehen, die weniger Konflikte verursachen.

## Mutation:

Jedes Gen (Region) hat mit kleiner Wahrscheinlichkeit  $p_{mut}$  eine Farbänderung:

$$[rot, blau, grün, gelb, rot, blau] \rightarrow [rot, blau, grün, \textcolor{red}{grün}, rot, blau]$$

## Fitnessfunktion:

Bewertet wird die Einfärbung nach zwei Kriterien:

- Anzahl der Konflikte (gleiche Farbe bei benachbarten Regionen)
- Anzahl der verwendeten Farben

$$\Phi(\mathbf{g}) = -(\text{Konflikte}) - \alpha \cdot (\text{verwendete Farben})$$

mit  $\alpha$  als kleinem Gewichtungsfaktor.

- Die Fitness steigt, wenn Konflikte abnehmen und weniger Farben genutzt werden.

## 2 8-Queens-Problem

### Was ist das überhaupt?:

Es sollen acht Königinnen auf einem  $8 \times 8$  Schachbrett so platziert werden, dass sich keine gegenseitig schlagen können. Das bedeutet: keine zwei Königinnen dürfen in derselben Zeile, Spalte oder Diagonale stehen.

### Kodierung der Individuen:

Jede Spalte des Schachbretts enthält genau eine Königin. Der Wert im Gen zeigt an, in welcher Zeile sie steht.

$$\mathbf{g} = (1, 5, 8, 6, 3, 7, 2, 4)$$

Das bedeutet:

- In Spalte 1 steht eine Königin in Zeile 1.
- In Spalte 2 steht eine Königin in Zeile 5.
- In Spalte 3 steht eine Königin in Zeile 8.
- In Spalte 4 steht eine Königin in Zeile 6.
- In Spalte 5 steht eine Königin in Zeile 3.
- In Spalte 6 steht eine Königin in Zeile 7.
- In Spalte 7 steht eine Königin in Zeile 2.
- In Spalte 8 steht eine Königin in Zeile 4.

## Darstellung der Eltern auf dem Schachbrett:

**Eltern 1:** (1, 5, 8, 6, 3, 7, 2, 4)

8			○					
7						○		
6				○				
5		○						
4								○
3					○			
2							○	
1	○							

Spalten: a-h (links nach rechts)    Zeilen: 1-8 (oben nach unten)

**Eltern 2:** (4, 6, 1, 5, 8, 2, 7, 3)

8					○			
7							○	
6		○						
5				○				
4	○							
3								○
2						○		
1			○					

Spalten: a-h (links nach rechts)    Zeilen: 1-8 (oben nach unten)

## Crossover:

Beim Ein-Punkt-Crossover werden zwei Eltern an einer zufälligen Stelle getrennt und die hinteren Teile getauscht. Bei beiden muss außerdem an die gleiche Stelle geschnitten werden (Hier haben wir in der Mitte geschnitten)

Eltern 1:                    [1, 5, 8, 6, 3, 7, 2, 4]

Eltern 2:                    [4, 6, 1, 5, 8, 2, 7, 3]

⇒                    [1, 5, 8, 6, 8, 2, 7, 3], [4, 6, 1, 5, 3, 7, 2, 4]

**Erklärung:** So entstehen neue Kombinationen aus den Genen beider Eltern.

## Mutation:

Ein einzelnes Gen (eine Spalte) wird zufällig geändert:

[1, 5, 8, 6, 3, 7, 2, 4] → [1, 5, 8, **2**, 3, 7, 2, 4]

**Erklärung:** Die Mutation sorgt für kleine Veränderungen und verhindert, dass der Algorithmus stecken bleibt.

## Fitnessfunktion:

Die Fitness bewertet, wie viele Konflikte es gibt:

$$\Phi(\mathbf{g}) = \frac{1}{1 + \text{Anzahl der Konflikte}}$$

**Erklärung:** Je weniger Konflikte, desto besser. Eine perfekte Lösung ohne Konflikte hat den Wert  $\Phi = 1$ .

## Simulated Annealing

Um die beiden Probleme (8-Queens und Landkarten-Färben) mit **Simulated Annealing** zu lösen, braucht man folgende Sachen:

- **Startzustand:** Eine zufällige Anfangslösung zum Beispiel zufällige Damenpositionen bei der 8-Queen-Problem oder zufällige Farben bei der Landkarten Färben-Problem).
- **Nachbarschaftsfunktion:** Kleine Änderungen der aktuellen Lösung  
- eine Dame in eine andere Zeile oder eine Region bekommt eine neue Farbe.
- **Kostenfunktion:** Misst, wie gut eine Lösung ist zum Beispiel man misst die Anzahl der Konflikte, die man hat.
- **Temperatur  $T$ :** Beginnt hoch und wird langsam kleiner (Abkühlungsplan, z. B.  $T_k = 0.95 \cdot T_{k-1}$ ).

### Wie man eigentlich vorgeht und was die Idee davon ist?:

Am Anfang dürfen auch schlechtere Lösungen angenommen werden. Mit sinkender Temperatur werden nur noch Verbesserungen übernommen – so kann das Verfahren aus lokalen Minima entkommen und bessere Ergebnisse finden.