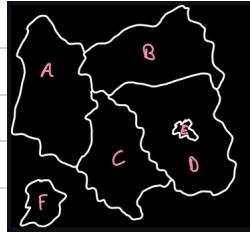


Nr. 1

Landkartenfärbeproblem

Ziel konfliktfreie Einführung mit einer minimalen Anzahl an Farben sein

Variablen = Länder



k = Anzahl Konflikte I = Individuen Fd = Anzahl Farben

Start population: A B C D E F $(g_1, \dots, g_m) \in \{0,1\}^m$

Individuum 1: $y = (0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0)$ (red, green, blue, black, red, yellow)

Individuum 2: $g = (1, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0)$ (green, green, green, black, blue, red)

Individuum 3: $g = (0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0)$ (red, green, blue, red, green, red)

Individual 4: g(0,0,0, 0,0,0 ,0,1,0 ,0,0,1, 1,0,0, 0,0,0) (red, red, blue, black, green, red)

Problemvariablen belegen

$\{ A = \text{red}, B = \text{green}, C = \text{blue}, D = \text{black}, F = \text{yellow} \}$

10 14 12 11 16

$$I_1: k=0 \text{ FA} = 5$$

$$I_2 : k=2 \quad F_A = 4$$

$$I_3 : k=0 \quad FA = 3$$

$$I_4 : k=1 \quad FA=4$$

(Reduzierter Vektor)

Dekodierungsfunktion: $\Gamma : \{0,1\}^m \rightarrow \mathbb{R}^n$ (Ergebnis Tugel bestehend aus reellen Zahlen) $\underbrace{(1,1,0)}_{\substack{\uparrow \\ \text{nimm gesamten} \\ \text{Wert}}} \quad (\text{z.B. hier } (6,1,1,1,1,1)) \quad \underbrace{''}_{6}$

Fitwurfunktion: $\Theta(g_i) = F(\tau(g_i)) - \text{(Anzahl der verwendeten Farben} \cdot 5)$

Zielfunktion: $F(v(g_i)) = 100 - \text{Anzahl der Konflikte} \cdot 10$

Maximal 6 Konflikte sind möglich und basierend auf der Anzahl der Konflikte wird erstmal gewennt, wie nah ich am Ziel bin. Hierbei gilt, je weniger Konflikte, desto besser und die werden hier am stärksten gewichtet. Hierzu kommt, dass ich die Anzahl der verwendeten Farbe subtrahiere, dies aber in einem sehr engemten Rahmen, damit das Ergebnis positiv bleibt und die Anzahl der verwendeten Farben berücksichtigt wird.

Constraints

$$C = \{C_1, C_2, C_3, C_4, C_5, C_6\} \quad (\text{Kante} = \text{Constraint})$$

$$C_1 = A \neq B \quad C_5 = C \neq D \quad (\text{Variablen} = \text{Knoten})$$

$$C_2 = A \neq C \quad C_6 = D \neq E$$

$$C_3 = B \neq C$$

$$C_4 = B \neq D$$

Domänen (Wertebereich)

$$D = \{0, 1, 2, 3\}$$

Roulette Wheel Selektion

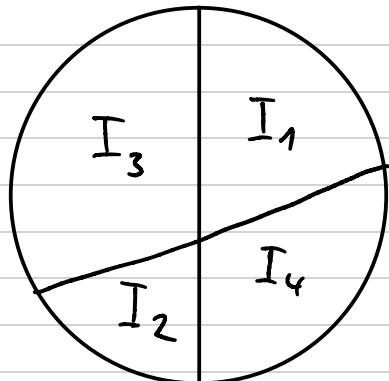
$$F(I_1) = (100 - 0 \cdot 20) - (5 \cdot 5) = 75 \quad (28\%)$$

$$F(I_2) = (100 - 2 \cdot 20) - (4 \cdot 5) = 40 \quad (15\%)$$

$$F(I_3) = (100 - 0 \cdot 20) - (3 \cdot 5) = 85 \quad (32\%)$$

$$F(I_4) = (100 - 1 \cdot 20) - (4 \cdot 5) = 60 \quad (23\%)$$

WK: Fitnes / Summe aller Fitnesse Anzahl wahlbereichsfertigkeit



Crossover (Rekombination)

1. Selektion: I_1 und I_2

2. $p_{crossover} = 0.6$ (Ergebnis vom Zufallsexperiment / Zufällig bestimmt)

Festlegung WK des übertragen gekreuzt wird

Unter 60% kein Crossover

Größer gleich 60% \rightarrow Crossover

$$I_1 = [0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 1, 0]$$

$$I_2 = [1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0]$$

(Kinder)

$$I_3 = [0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 1, 0, 0]$$

$$I_4 = [1, 0, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0, 0] \quad (\text{Vereinigte Individuen})$$

(Habe jetzt in der Mitte zerschritten, obwohl ich Variablen zerschneiden könnte, da ich aber die Mitte schneiden wollte und es sich um eine gerade Anzahl an Elementen handelt, wurde keine Variable zerschritten)

(Als zufällige Stelle $i = 9$ als Schnittpunkt für angemessene Rekombination)

Mutation

- Mutationswahrscheinlichkeit: $p_{mut} = 0.01$
- Zufallsexperiment (Zenschrift zufälligen Zahl für jeden Eintrag im I und oben)
$$g_i^{(t+1)} = \begin{cases} 1 & g_i^{(t)} \text{ falls } X_i \leq p_{mut} \text{ (Bit negieren)} \\ g_i^{(t)} & \text{sonst} \end{cases}$$

(Nicht negieren)

$$I_5 = [0,0,0,1,0,0,0,1,0,0,1,0,1,0,0,0,0]$$

$$I_6 = [1,0,0,1,0,0,1,0,0,1,0,0,0,1,0,0,1]$$

Haben jetzt eine optimierte konfliktfreie Lösung in der wir 3 anstatt 4 farben nutzen.

Hier habe ich durch Mutation ein Konflikt weniger.

- Geflippte Bits

8-Queens Problem

I = Individuum

K = Anzahl Konflikte I = Individuum

Startpopulation:

Individuum 1: $g = (A, B, C, D, E, F, G, H)$

Individuum 2: $g = (0, 0, 1, 0, 0, 0, 1, 0)$

Individuum 3: $g = (1, 0, 0, 0, 1, 0, 0, 1)$

Individuum 4: $g = (0, 0, 1, 0, 0, 0, 1, 0)$

Problemvariablenbelegung

$\{ A=0, B=1, C=2, D=3, E=4, F=5, G=6, H=7 \}$ Spalte = Variable = X-Koordinate
Bitwert = y-Koordinate

Dekodierungsfunktion: $\Gamma : \{0, 1\}^m \rightarrow \mathbb{R}^n$

Fitnesfunktion: $\Theta(g_i) = F(\Gamma(g_i))$

Zielfunktion: $F(v(g_i)) = 100 - \text{Anzahl der Konflikte} \cdot 3$

(Hier optimieren wir nicht zusätzlich nach Farben, das heißt, die Positionierung mit den wenigensten Konflikten gewinnt. Mehr brauchen wir nicht.)

Constraints (A und B hätten einen Abstand von 1)

$$C = \{C_1, C_2, C_3\}$$

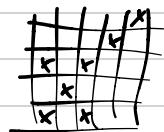
$$C_1 = v(g_i) \neq v(g_j)$$

$$C_2 = v(g_i) \neq v(g_j) + \text{Variablenabstand}$$

$$C_3 = v(g_i) \neq v(g_j) - \text{Variablenabstand}$$

Die X-Koordinate variiert immer

} Konflikte können öfter auftreten
Max. 28



Domänen = {0, 1, 2, 3, 4, 5, 6, 7}

Crossover (Rekombination)

1. Selektion: I_1 und I_3 (Roulette Wheel Selektion)

2. $p_{cross} = 0.6$

$$I_1: g = (1,0,0, 0,0,0, 1,1,1, 0,1,1, | 1,0,1, 1,0,0, 1,1,1, 1,1,0,)$$

$$I_2: g = (0,0,1, 0,0,0, 0,1,1, 0,1,0, | 0,0,1, 1,1,0, 0,1,0, 1,0,0,)$$

$i = 12$ (Schnittstelle)

$$I_5 = (1,0,0, 0,0,0, 1,1,1, 0,1,1, | 0,0,1, 1,1,0, 0,1,0, 1,0,0,)$$

$$I_6 = (0,0,1, 0,0,0, 0,1,1, 0,1,0, | 1,0,1, 1,0,0, 1,1,1, 1,1,0,)$$

Mutation

- Mutationswahrscheinlichkeit: $p_{mut} = 0.01$

- Zufallsexperiment (Auswahl zufälligen Zahl für jeden Eintrag im I und over)

$$- g_i^{(t+1)} = \begin{cases} \text{7 } g_i^{(t)} & \text{falls } X_i \leq p_{mut} \text{ (BH begieren)} \\ g_i^{(t)} & \text{sonst} \end{cases} \quad \text{(Nicht begieren)}$$

$$I_5 = (1,0,0, 0,0,0, 1,1,1, 0,1,1, | 0,0,1, 1,1,0, 0,1,0, 1,0,0,)$$

1

Dies entspricht jetzt einer Lösung durch die zufällige Mutation.

$$I_6 = (0,0,1, 0,0,0, 0,1,1, 0,1,0, | 1,0,1, 1,0,0, 1,1,1, 1,1,0,)$$

Die Mutation entfernt einen einzigen Konflikt jetzt.

Was würden Sie noch benötigen, um die obigen Probleme mit Simulated Annealing lösen zu können?

Es fehlt noch ein Temperatur Parameter, der im Laufe der Suche immer kleiner wird. Dies um zu verhindern, dass der Algorithmus stecken bleibt, während der Suche. Dies könnte man sehr gut in Form einer dynamischen Gewichtung darstellen.

Nr 2

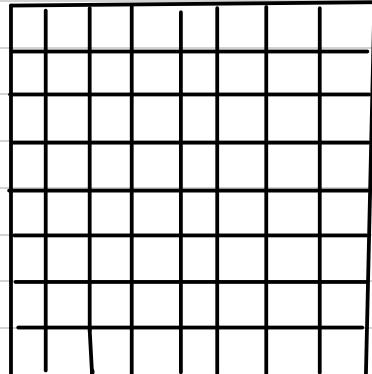
Auswertung 8-Queens - Problem

Population: 100

Mutationswahrscheinlichkeit: 0.1

Generationen: 100

Crossover Stelle: Zufällig



Ergebnisse:

[2, 4, 6, 0, 0, 1, 3, 1] x 3 Konflikte

[6, 1, 1, 5, 3, 0, 3, 0] x 2 Konflikte

[6, 1, 2, 7, 1, 3, 5, 0, 4] x 1 Konflikt

[1, 4, 0, 3, 0, 1, 7, 5, 2] x 1 Konflikt

[0, 1, 5, 1, 1, 6, 0, 3, 0] x 4 Konflikte

Population: 500

Mutationswahrscheinlichkeit: 0.2

Generationen: 100

Crossover Stelle: Zufällig

[5, 2, 0, 1, 7, 0, 1, 7, 1, 4] 2 Konflikte

[1, 1, 1, 7, 5, 1, 7, 0, 1, 4, 6] 2 Konflikte

Population: 300

Mutationswahrscheinlichkeit: 0.2

Generationen: 100

Crossover Stelle: Zufällig

[1, 6, 2, 7, 4, 1, 1, 3, 5, 0] 2 Konflikte

[1, 4, 6, 0, 3, 5, 7, 1, 3, 1] 1 Konflikt

[0, 1, 5, 1, 6, 0, 2, 4, 1] 2 Konflikte

[1, 6, 1, 3, 5, 1, 0, 1, 7, 0, 1, 4, 1] 2 Konflikte

[1, 4, 0, 1, 7, 5, 1, 2, 6, 1, 1, 3] 0 Konflikte ✓

Population: 200

Mutationswahrscheinlichkeit: 0.2

Generationen: 100

Crossover Stelle: Zufällig

[1, 3, 7, 0, 1, 4, 1, 1, 6, 1, 4] 3 Konflikte

[1, 0, 1, 2, 7, 1, 1, 3, 0, 1, 6] 2 Konflikte

[1, 6, 1, 2, 7, 1, 1, 3, 1, 5, 0, 1, 4] 0 Konflikte ✓

Bei der Auswertung hat sich herausgestellt, dass die Populationssgröße von 200 am schnellsten eine Lösung für das 8-Queen Problem gefunden hat, wobei eine zu große Populationssgröße wie 500 oder zu kleine wie 100 keine optimale Lösung gefunden hat. Der Rechenaufwand war auch erheblich größer bei einer Population von 500. Eine Mutationswahrscheinlichkeit von 0.2 hat zusätzlich für bessere Werte gesorgt.

Landkartenfarbierproblem

Population: 100

Mutationswahrscheinlichkeit: 0.2

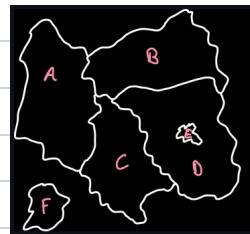
Generationen: 100

Crossover Stelle: Zufällig

Ergebnisse:

[0,0,0, 1,0,0 0,1,0 , 0,1,0 1,0,0 1,0,0 ,0,0,0] Optimalste Lösung

[0,0,1 0,1,0 ,0,0,0, 0,0,1, 0,0,0 ,0,0,1] Konfliktfrei 4 Farben



red green blue black red yellow
 0,0,0,1,1,0,0,0,1,1,0,0,0,1,0,0,0,1,1,1,1)

Population: 200

Mutationswahrscheinlichkeit: 0.2

Generationen: 100

Crossover Stelle: Zufällig

Ergebnisse: (Stagniert)

[0,0,0, 0,0,0, 0,0,0, 0,0,0, 0,0,0, 0,0,0] Fehlgeschlagen Max Konflikte

[0,0,0, 0,0,0, 0,0,0, 0,0,0, 0,0,0, 0,0,0] Fehlgeschlagen Max Konflikte

[0,0,0, 1,1,0, 0,1,0, 0,0,0 ,0,1,0, 0,0,0] Konfliktfreie Lösung mit 4 Farben

Population: 100

Mutationswahrscheinlichkeit: 0.2

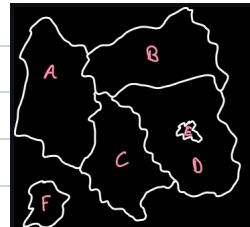
Generationen: 500

Crossover Stelle: Zufällig

Ergebnisse:

[0,0,1, 0,1,0, 1,1,0, 0,0,1, 0,1,0, 0,0,1] Optimale Lösung

[0,1,1, 1,1,0, 0,1,0, 1,1,0, 1,0,0 1,1,0] 1 Konflikt



Population: 100

Mutationswahrscheinlichkeit: 0.2

Generationen: 100

Crossover Stelle: 3

[0,1,0, 1,0,1, 0,0,0, 0,1,0, 0,0,0, 0,0,0] Optimale Lösung

[1,0,0, 0,0,0, 0,1,0, 1,0,0, 0,1,0, 0,0,0] Optimale Lösung

[0,0,0, 0,0,0, 0,0,0, 0,0,0, 0,0,0, 0,0,0] Fehlgeadlungen

Anhand der Ergebnisse kann festgestellt werden, dass der Algorithmus bei einer Populationsgröße von 100, 100 Nachkommen, Mutationswahrscheinlichkeit von 0.2 und zufälligem Crossover die besten Lösungen geliefert hat. Zudem ist es besonders abhängig von der Problemgröße, wann, welche Werte für die jeweiligen Operatoren eine Rolle spielen. Zudem sorgt eine höhere Mutationswahrscheinlichkeit für bessere Individuen.

Nr. 3

Search Strategie: Where is Waldo

- Visualisierung von Waldos Standorte anhand eines Diagramms und Analyse dieser
- Nutzen Positionsmarker (68-Waldo-Koordinaten)
- Keine dynamische Suche, er kennt bereits alle Positionen
- Nutzt eine sogenannte Shuffle Mutation, die einen zufälligen Abschnitt des Pfads nimmt und ihn an eine andere Position innerhalb des Pfades verschiebt
- Individuen stellen Routen dar durch alle 68 Koordinaten (Tupel aus Beendungszeit der Punkte) (siehe Element 1.)
- Keine Nutzung von Crossover
- Nimmt immer nur die 10% besten Nachkommen bzw. die 10% besten Pfade und erzeugt von jedem Nachkommen
- Fitnessfunktion berechnet die Gesamtlänge des Pfades um durch alle Punkte zu kommen.
(je kleiner die Fitness desto besser)
- Berechnet die euklidische Distanz zwischen zwei Punkten, um herauszufinden, welche Beendefolge den kürzesten Gesamtweg ergibt

"Evolution Simulator"

- Visualisierte Auswahl von Kreaturen (Individuen)
- Darstellung der Fitness durch Hovern und freie Auswahl
- Sortierung der Kreaturen nach Geschwindigkeit
- Misst die Fitness über die Schnelligkeit der Kreaturen (Misst die erreichte Distanz der Kreatur nach ¹⁵ Sekunden) in Meter
- Eliminiert die Hälfte der Individuen basierend auf ihre Schnelligkeit Top 50% überleben
- Erzeugt durch Mutation neue Nachkommen
- Sortierung der schnellsten Individuen mit Quicksort und killt dann die untere Hälfte, also die werden selektiert
- Kein Crossover, sondern Mutation
- Keine Roulette Wheel Selektion, sondern standardmäßig die besten 50% überleben
- Evolutionäres Algorithmen

american fuzzy lop

- Einsatz von GAs um automatisch Testfälle zu entdecken für Zielprogramme / Software
- Synthetisiert komplexe Dateisammlungen
- Fuzzing: Erzeugung von massenhaften Eingabedaten, die kleine zufällige Fehler enthalten und testet eine Software damit, um zu prüfen, ob die getestete Software unerwünschtes Verhalten zeigt
- Eingabedateien die das Zielprogramm liest gelten als Individuen
- Anwendung zufälliger Mutationen zur Erzeugung neuer Testdateien
- Selektion: Wenn eine mutierte Eingabedatei neue Tupel enthält werden diese behalten
- Fitnessfunktion: Generierte Eingabedatei wird behalten, wenn sie neue Tupel erzeugt

In welchen anderen Anwendungen werden evolutionäre Algorithmen eingesetzt? Wie und wofür werden EA/GAs genutzt?

- Entwicklung von
 - ↪ Sensornetze
 - ↪ Aktionsmarktoracle
 - ↪ RNA-Struktur-Vorhersage
- Künstliche Intelligenz
- Ingenieurwesen: Optimierung von Designen, Steuerungssysteme und Produktionsprozessen
- Finanzwesen: Portfoliooptimierung, Risikomanagement und Betrugsunterstreichung
- Logistik: Routenplanung, Lagerverwaltung und Lieferkettenoptimierung
- Informatik: Maschinelles Lernen, Mustererkennung und Software-Engineering
- Biologie: Proteinfaltung, Wirkstoffsuchung und Genexpressionsanalyse
- Scheduling Probleme