

Evolutionäre Verfahren

Sudoku

Marius Huke

TU Ilmenau

1. Allgemeines
2. Algorithmus
 - 2.1 Kodierung
 - 2.2 Initialisierung
 - 2.3 Fitnessfunktion
 - 2.4 Selektionsmethode
 - 2.5 Mutation
 - 2.6 Crossover
 - 2.7 Zusammensetzung
3. Auswertung
4. Ausblick
5. Vorführung

Allgemeines

	6		2		9			
				3			1	
1			6					9
4	2		5				9	
		5	3		2	8	6	
	8	3	1				2	4
8	7		9		6		3	5
3	4			5		2	7	
2		6		7	3			1

- 9x9 Gitter mit Zahlen von 1 bis 9
- leere Felder so ausfüllen, dass jede Zahl in jeder Zeile, jeder Spalte und jedem 3x3-Block genau 1-mal vorkommt
- verschiedene Schwierigkeitsgrade in Abhängigkeit von Anzahl gegebener Felder
- weitere Varianten wie z.B. 25x25
- BruteForce: $9^{41} = 1.3 \cdot 10^{39}$ - $9^{64} = 1,2 \cdot 10^{61}$ Möglichkeiten (abhängig von Schwierigkeit)

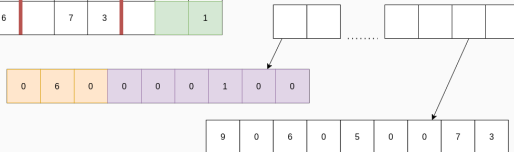
Kodierung

	6		2		9			
				3			1	
1			6					9
4	2		5				9	
		5	3		2	8	6	
	8	3	1				2	4
8	7		9		6		3	5
3	4			5		2	7	
2		6		7	3			1

Reihenrepräsentation:

0	6	0	2	0	0	1
---	---	---	---	---	-------	---	---

Gitterrepräsentation:



- Verwendung von `shared_ptr`
- leere Felder = 0
- Reihenrepräsentation
→ 1D-Array
- Gitterrepräsentation
→ 2D-Array

Initialisierung

	6		2		9			
				3			1	
1			6					9
4	2		5				9	
		5	3		2	8	6	
	8	3	1				2	4
8	7		9		6		3	5
3	4			5		2	7	
2		6		7	3			1



3	6	2
9	8	5
1	4	7



2	3	5
2	7	8
4	9	1

- zufällige Initialisierung leerer Felder
- einfache Methode: keine doppelten in Blöcken
- intelligente Initialisierung: möglichst keine doppelten in Zeilen

Fitnessberechnung

9(2)	6(1)	3(1)	2	1	9	2	4	6
5(1)	2(1)	7(1)	8	3	5	7	1	3
1(0)	8(2)	4(1)	6	4	7	8	5	9
4	2	1	5	4	9	1	9	5
6	7	5	3	6	2	8	6	3
9	8	3	1	7	8	7	2	4
8	7	1	9	8	6	4	3	5
3	4	5	4	5	2	2	7	8
2	9	6	1	7	3	6	9	1

- Berechnen der individuellen Kollisionen
- Anlegen eines "Fitness-Sudokus" zur Speicherung der Werte
→ Mutation (s. Block 0)
- Kollisionszahl als Fitnesswert (=86)
- 0 als Idealwert entspricht gelöstem Sudoku

Elitismus:

- Wahl der besten x-Prozent der Individuen
- Vorteil: schnelles Erreichen des (lokalen) Minimums
- Nachteil: eventuell globales Minimum schwerer zu erreichen

Glücksradauswahl (universelles Stichprobenziehen):

- Skalierung der Fitness, sodass die Summe der Fitnesswerte 1 ergibt
- hintereinander reihen der Individuen mit ihren Fitnesswerten
- Auswahl von Individuen in gleichmäßigen Abständen (angepasst an Selektionsrate)
- Vorteil: höhere Diversität
- Nachteil: langsamere Konvergenz

Mutation

9	6	3	2	1	9	2	4	6
5	2	7	8	3	5	7	1	3
1	8	4	6	6	7	8	5	9
4	2	1	5	4	9	1	9	5
6	7	5	3	6	2	8	6	3
9	8	3	1	7	8	7	2	4
8	7	1	9	8	6	4	3	5
3	4	5	4	5	2	2	7	8
2	9	6	1	7	3	6	9	1

7	4	8
2	1	3
6	5	9
3	9	7
8	6	1
5	2	4
4	3	5
2	7	6
8	9	1

- Ausführung blockweise
 - gegebene Felder werden ignoriert
1. Auswählen von Kollisionszellen (beliebige Chance für andere Zellen)
 2. zufälliges paarweises Austauschen der gewählten Zellen

2-Punkt-Crossover

9	6	3	2	1	9	2	4	6
5	2	7	8	3	5	7	1	3
1	8	4	6	4	7	8	5	9
4	2	1	5	4	9	1	9	5
6	7	5	3	6	2	8	6	3
9	8	3	1	7	8	7	2	4
8	7	1	9	8	6	4	3	5
3	4	5	4	5	2	2	7	8
2	9	6	1	7	3	6	9	1



9	6	3	2	1	9	2	4	6
5	2	7	8	3	5	7	1	3
1	8	4	6	4	7	8	5	9
4	2	1	5	7	6	1	9	5
3	9	5	3	4	2	8	6	3
7	8	6	1	9	8	7	2	4
8	7	1	9	8	6	4	3	5
3	4	5	4	5	2	2	7	8
2	9	6	1	7	3	6	9	1

2	6	9	2	1	9	2	3	4
7	3	5	7	3	5	6	1	7
1	4	8	6	4	6	8	5	9
4	2	1	5	7	6	1	9	5
3	9	5	3	4	2	8	6	3
7	8	6	1	9	8	7	2	4
8	7	1	9	4	6	8	3	5
3	4	5	2	5	8	2	7	9
2	9	6	1	7	3	4	6	1

2	6	9	2	1	9	2	3	4
7	3	5	7	3	5	6	1	7
1	4	8	6	4	6	8	5	9
4	2	1	5	4	9	1	9	5
6	7	5	3	6	2	8	6	3
9	8	3	1	7	8	7	2	4
8	7	1	9	4	6	8	3	5
3	4	5	2	5	8	2	7	9
2	9	6	1	7	3	4	6	1

- Invariante (keine Kollision in Blöcken) bleibt erhalten
- Übernahme der Eltern in neue Population
- "Auffüllen" der Population durch Crossover
- Zeilenkollisionen bleiben erhalten

Diagonales-Crossover

9	6	3	2	1	9	2	4	6
5	2	7	8	3	5	7	1	3
1	8	4	6	4	7	8	5	9
4	2	1	5	4	9	1	9	5
6	7	5	3	6	2	8	6	3
9	8	3	1	7	8	7	2	4
8	7	1	9	8	6	4	3	5
3	4	5	4	5	2	2	7	8
2	9	6	1	7	3	6	9	1

2	6	9	2	1	9	2	3	4
7	3	5	7	3	5	6	1	7
1	4	8	6	4	6	8	5	9
4	2	1	5	7	6	1	9	5
3	9	5	3	4	2	8	6	3
7	8	6	1	9	8	7	2	4
8	7	1	9	4	6	8	3	5
3	4	5	2	5	8	2	7	9
2	9	6	1	7	3	4	6	1

5	6	7	2	1	9	2	8	5
9	2	8	7	3	5	4	1	3
1	4	3	6	8	4	7	6	9
4	2	9	5	8	9	1	9	7
7	1	5	3	4	2	8	6	3
3	8	6	1	7	6	5	2	4
8	7	1	9	8	6	9	3	5
3	4	9	4	5	1	2	7	8
2	5	6	2	7	3	4	6	1

9	6	3	2	1	9	2	4	6
5	2	7	8	3	5	7	1	3
1	8	4	6	4	7	8	5	9
4	2	1	5	7	6	1	9	5
3	9	5	3	4	2	8	6	3
7	8	6	1	9	8	7	2	4
8	7	1	9	8	6	9	3	5
3	4	9	4	5	1	2	7	8
2	5	6	2	7	3	4	6	1

2	6	9	2	1	9	2	3	4
7	3	5	7	3	5	6	1	7
1	4	8	6	4	6	8	5	9
4	2	9	5	8	9	1	9	7
7	1	5	3	4	2	8	6	3
3	8	6	1	7	6	5	2	4
8	7	1	9	8	6	4	3	5
3	4	5	4	5	2	2	7	8
2	9	6	1	7	3	6	9	1

5	6	7	2	1	9	2	8	5
9	2	8	7	3	5	4	1	3
1	4	3	6	8	4	7	6	9
4	2	1	5	4	9	1	9	5
6	7	5	3	6	2	8	6	3
9	8	3	1	7	8	7	2	4
8	7	1	9	4	6	8	3	5
3	4	5	2	5	8	2	7	9
2	9	6	1	7	3	4	6	1

beste Zusammensetzung (empirisch):

1. Standard-Initialisierung (effizienter)
2. Fitnessberechnung
3. Selektion (20%)
4. Mutation ($\frac{1}{9}$ Wahrscheinlichkeit für nicht kollidierende Felder)
5. diagonales Crossover (größere Diversität)

Abbruchkriterium: 25 Generation ohne Verbesserung des Fitnesswertes

Test auf 40 Sudokus (10 aus jeder Schwierigkeitsstufe (Leicht, Mittel, Schwer, Extrem)):

Populationsgröße 100:

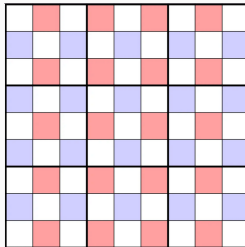
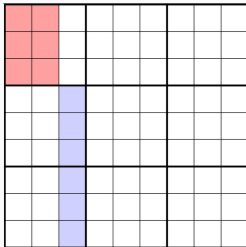
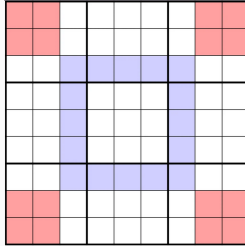
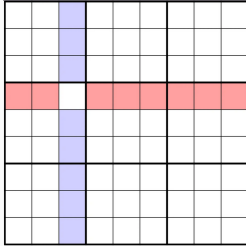
Schwierigkeit	gelöst (%)	durchschnittliche Zeit (ms)	durchschnittliche Generationen	durchschnittliche Zeit(Abbruch) (ms)
Leicht	100	1102	7.2	-
Mittel	60	2721	18.8	6381
Schwer	0	-	32.3	9412
Experte	0	-	33.4	9583

Populationsgröße 500:

Schwierigkeit	gelöst (%)	durchschnittliche Zeit (ms)	durchschnittliche Generationen	durchschnittliche Zeit(Abbruch) (ms)
Leicht	100	4662	5.6	-
Mittel	90	13222	11	35048
Schwer	50	25979	32.4	53385
Experte	0	-	40.1	51216

Populationsgröße 1000:

Schwierigkeit	gelöst (%)	durchschnittliche Zeit (ms)	durchschnittliche Generationen	durchschnittliche Zeit(Abbruch) (ms)
Leicht	100	9208	5.3	-
Mittel	100	24384	14.8	-
Schwer	60	43039	25.5	87460
Experte	20	91698	54.5	123294



- Anwenden von Set AEquivalenz Theorie (z.B. in Fitnessbewertung oder Mutation)
- mehr Exploration durch andere Selektionsverfahren
- Anpassung an andere Sudokuarten

Vorführung



crackingthecryptic.

Set equivalence theory.