

# SEARCHING FOR SURPRISING SEQUENCES WITH A GENETIC ALGORITHM

IT3708 - SUBSYMBOLIC AI METHODS

Aleksander BURKOW

Department of Computer and Information Science  
Norwegian University of Science and Technology

## Contents

1	Genetic encodings	1
2	Fitness functions	1
3	Locally surprising sequences	1
4	Globally surprising sequences	3
5	Problem difficulty ranking	4

## List of Figures

## Listings

## 1 Genetic encodings

The genetic encoding used is the bit vector genome from the One Max Problem, modified to contain a number at each index. The phenotype translation function is therefore the identity function, as the genome can be used for calculations directly

## 2 Fitness functions

Both fitness functions used are punishing, in that they reward genomes with few errors. The fitness is calculated as the inverse of the amount of errors found, where the error count is incremented each time an already found sequence reappears.

The Globally surprising fitness function consists of a nested for-loop, comparing all pairs of characters from left to right. The locally surprising version iterates once, checking neighbors only.

## 3 Locally surprising sequences

---

---

Symbol set	Pop size	Gen	Seq length	Sequence
------------	----------	-----	------------	----------

---

**Table 1:** *Locally surprising sequences. Crossover chance=0.9, per-character-mutation=0.01, generation cutoff=1000. Full generational replacement, tournament selection(size=8, epsilon=0.1), punishing fitness function*



## 4 Globally surprising sequences

Symbol set	Pop size	Gen	Seq length	Sequence
3	150	1	7	[1, 2, 0, 2, 2, 1, 0]
4	150	2	9	[0, 3, 2, 0, 0, 1, 3, 1, 2]
5	150	5	12	[3, 0, 4, 0, 2, 1, 1, 2, 4, 3, 1, 0]
6	150	179	15	[3, 0, 4, 2, 1, 5, 1, 4, 5, 0, 1, 3, 2, 2, 4]
7	150	120	17	[6, 2, 3, 0, 1, 5, 2, 0, 6, 4, 4, 3, 5, 3, 6, 5, 1]
8	150	7	19	[6, 4, 2, 7, 4, 3, 1, 3, 2, 6, 7, 0, 5, 5, 6, 0, 4, 1, 2]
9	15	130	22	[8, 3, 1, 7, 5, 7, 3, 0, 2, 4, 4, 8, 5, 4, 6, 2, 0, 7, 1, 3, 6, 8]
10	150	561	25	[9, 4, 3, 5, 1, 7, 8, 0, 7, 2, 8, 1, 4, 6, 7, 6, 1, 0, 3, 3, 4, 9, 2, 5, 8]
11	150	140	27	[7, 10, 3, 0, 8, 4, 6, 9, 2, 4, 3, 5, 0, 10, 5, 1, 6, 8, 8, 10, 1, 3, 9, 7, 2, 7, 4]
12	150	336	29	[0, 9, 7, 2, 1, 5, 6, 11, 4, 9, 3, 1, 3, 4, 10, 7, 6, 2, 0, 4, 8, 0, 11, 10, 8, 5, 9, 1, 6]
13	150	25	31	[6, 11, 9, 8, 7, 5, 10, 11, 3, 2, 0, 7, 2, 12, 1, 4, 3, 3, 12, 4, 11, 5, 7, 6, 9, 6, 2, 10, 12, 8, 0]
14	150	37	33	[3, 4, 13, 8, 11, 5, 6, 13, 11, 0, 9, 7, 5, 5, 3, 10, 1, 2, 6, 2, 12, 4, 1, 12, 10, 0, 7, 3, 6, 9, 5, 8, 4]
15	150	277	38	[9, 10, 12, 2, 5, 12, 3, 6, 7, 8, 2, 14, 13, 11, 1, 4, 7, 0, 11, 0, 1, 8, 13, 3, 4, 6, 9, 14, 5, 5, 10, 2, 8, 11, 10, 7, 12, 4]
16	150	186	40	[6, 4, 2, 9, 15, 5, 2, 7, 0, 1, 3, 14, 14, 4, 8, 10, 7, 13, 12, 11, 12, 3, 5, 13, 11, 14, 6, 0, 12, 2, 1, 9, 7, 1, 15, 10, 8, 5, 9, 3]
17	150			
18	150			
19	150			
20	150			

**Table 2:** Globally surprising sequences. Crossover chance=0.9, per-character-mutation=0.01, generation cutoff=1000. Full generational replacement, tournament selection(size=8, epsilon=0.1), punishing fitness function

## 5 Problem difficulty ranking

Ranked from easiest to hardest:

1. *One-max problem.* This is a simple hill-climbing problem where local maxima are a non-issue. Anything that exerts selection pressure is incredibly effective.
2. *Local surprising sequence for  $s=9$ .* A harder problem than One-max, though by only caring about neighboring sequences the complexity is kept relatively low. Mutating a single character can at most introduce two new errors.
3. *Global surprising sequence for  $S=9$ .* A single mutation can now introduce multiple errors, leading to higher complexity. Mutations can introduce multiple errors.