Linking the Dynamics of Genetic Algorithms to the Encoding of Information

Henrik Åhl
Supervised by Carl Troein and Adriaan Merlevede

June 2, 2016

Presentation overview

An introduction to genetic algorithms

Algorithms inspired by evolution

Genetic algorithms evolve *data sequences* by using the concepts of *mutation*, *reproduction* and *selection*

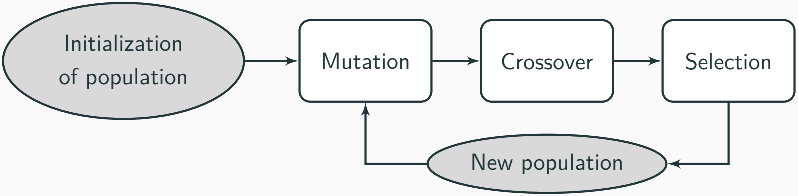
... for optimisation purposes

... for solving hard combinatorial problems

... for studying evolutionary dynamics

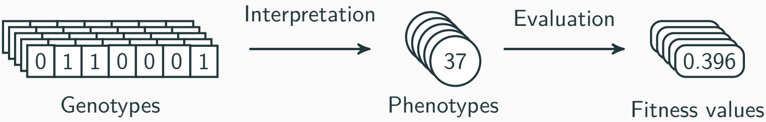
General structure

- Genetic algorithms maintain a pool of *candidate solutions* and modify them using *mutation* and *crossover* (recombination) operators
- A *selection* operator determines which solutions are mutated, recombined or picked for survival



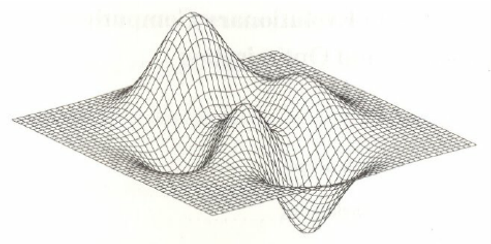
General structure

- Information (*phenotype*) is encoded as a data sequence (*genotype*)
- Phenotypes are evaluated by a *fitness*, *cost* or *objective function*



The search space

- The algorithms traverse a *fitness landscape*, or *search space*



- This is done by mutating and recombining the current data structures to produce new sample points

Genetic operators

- The **mutation** operator changes the genome such that a new candidate solution is produced

  A common choice is the *point mutation* operator

Before mutation	0	1	1	1	1	0	0	0	1
After mutation	0	1	0	1	0	0	0	1	1

- The **selection** operator determines diversity and bias by picking individuals for survival or reproduction

  A common choice is *tournament selection*

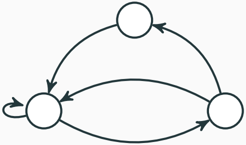
The encoding and decoding is problem specific

1	1	0	1	1	0	0	0	1

37

1.1	1.4	0.4	1.9	6.1	5.1	2.2	0.0	1.0

`true`	`false`	`false`	`false`	`false`



Genotypes

Phenotypes

Question: What are the differences between encodings which code for the same phenotype?

Encoding integers

Integers are normally encoded with a **Binary** encoding scheme

Genotype:	0	1	1	0	1	1	0	0	0	1

Phenotype: $1 \times 2^8 + 1 \times 2^7 + 1 \times 2^5 + 1 \times 2^4 + 1 \times 2^0 = 433$

Problem: adjacent phenotypes are not adjacent genotypes (e.g. 0111 [7] and 1000 [8])

Solution: **Gray code** – all adjacent phenotypes are also adjacent genotypes

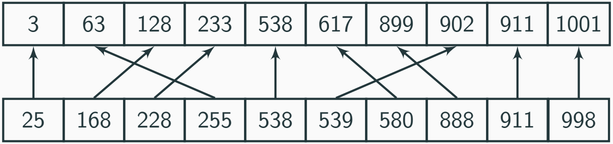
Experiment setup

Algorithm settings

- Population of 40 bitstrings
- All bitstrings are duplicated every generation
- Duplicates undergo point mutation ($p = 1/\text{genome length}$)
- 40 new bitstrings are selected for survival by tournament selection
- No crossover operator

Algorithm objective

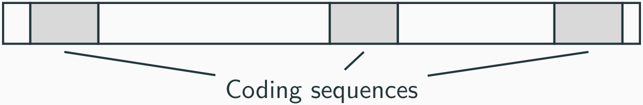
- Find a set of 10 random integers, $I_i \in [0, 1023]$
- The cost is measured as the sum of pairwise differences



- Smallest pairs are prioritised

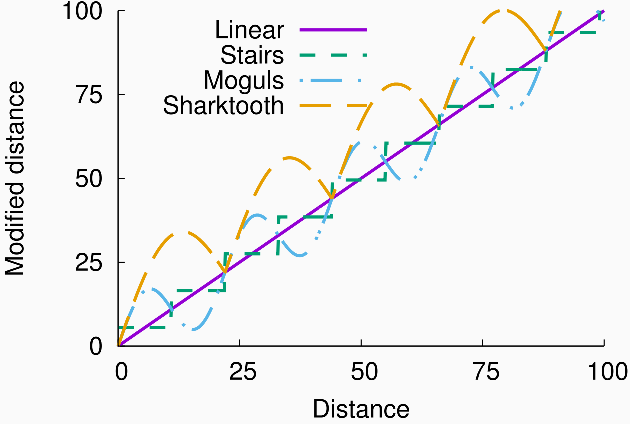
Encodings used

- Binary
- Gray code
- *Consensus encodings*: “Dead code” with coding segments



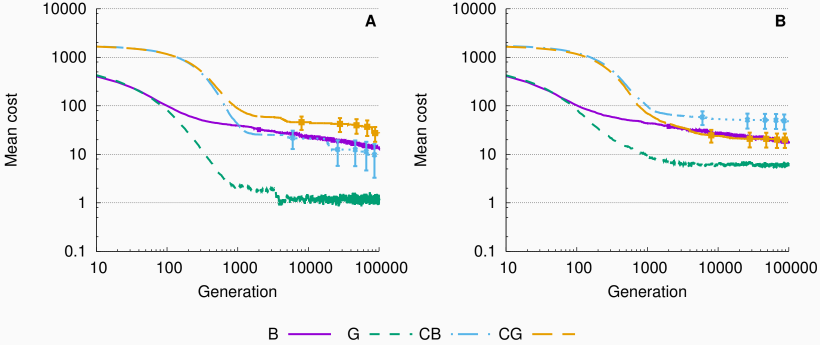
- Coding parts are signified by a start sequence of six bits: `110011`

Search spaces



Results

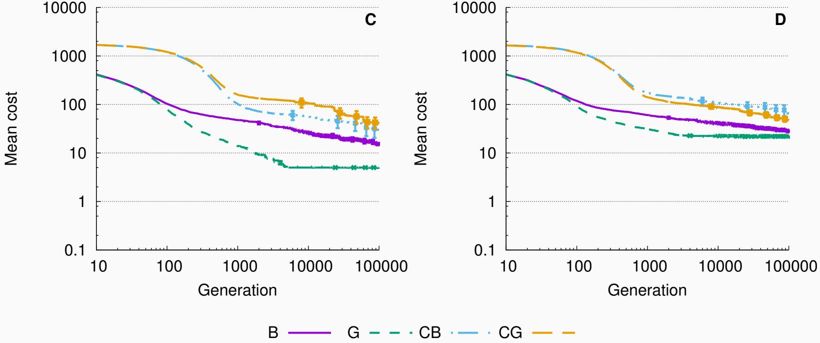
Performance



A: Linear, B: Stairs

- Gray code performs the best. Binary is grouped with consensus encodings.
- Binary performance is unaffected by complexity changes

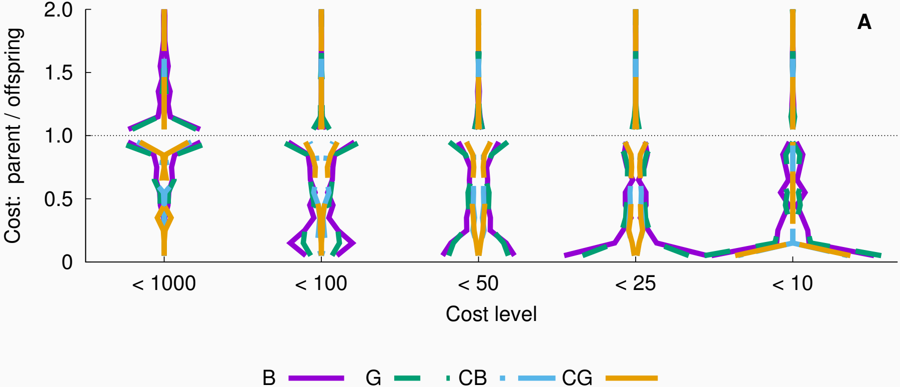
Performance



C: Moguls, D: Sharktooth

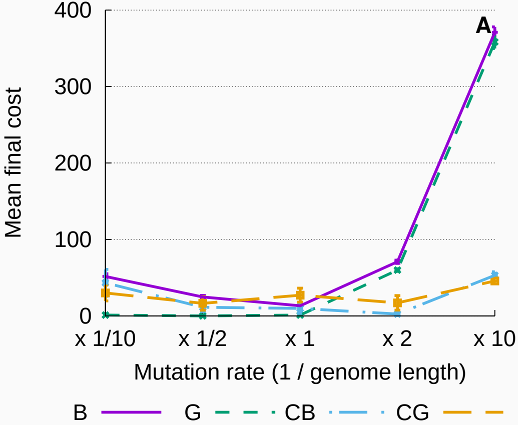
- With high enough complexity, Gray code loses its benefits

Distribution of cost effects – Linear transformation



- Binary and Gray are similar. Consensus encodings are similar.
- Gray code has consistent positive mutations
- Consensus encodings evolve by adding and removing random numbers

Robustness – Linear transformation



- Consensus encodings are more robust, bijective encodings sensitive to high mutation rates
- Gray code evolves mainly by single bit flips, Binary by multiple

Concluding remarks

Implication of the results

- The different modes of evolution give rise to different overall evolutionary dynamics
- Bijectivity is not everything – also directly available states and the connectivity between states matter
- Non-bijectivity make for more flexible genomes

Thank you!

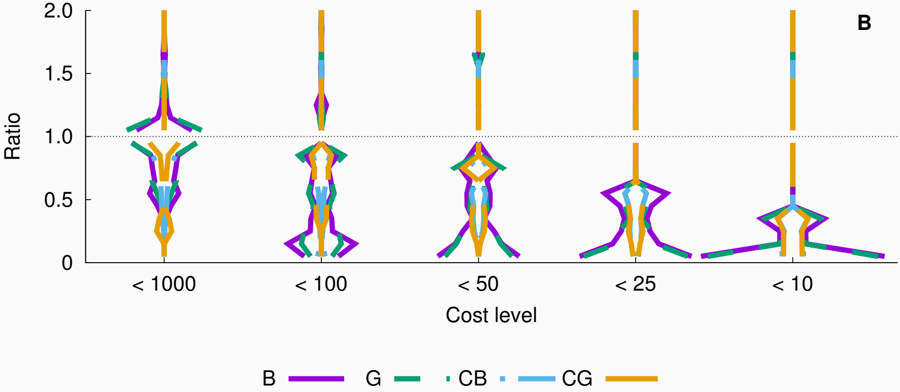
Binary–Gray code table

Integer	Binary	Gray	Gray as integer
0	000	000	0
1	001	001	1
2	010	011	3
3	011	010	2
4	100	110	6
5	101	111	7
6	110	101	5
7	111	100	4

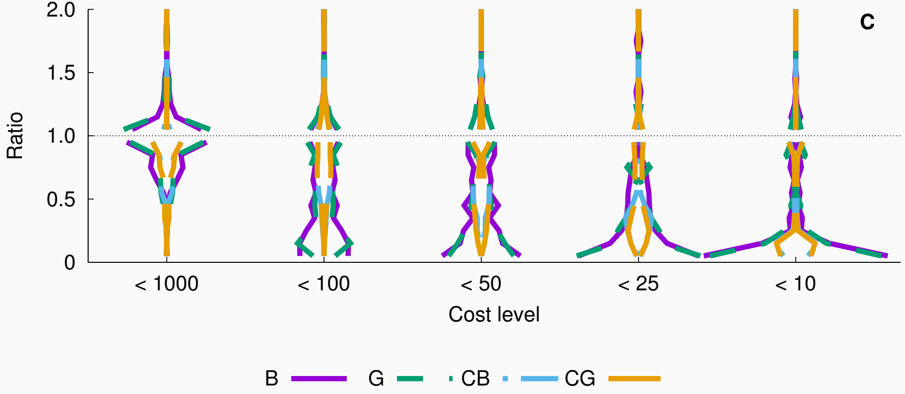
What about the biology?

- Biological genetic strands carry large amounts of ”dead code”
- Natural evolution is a product of itself – what are the alternatives?
- Evolutionary mechanisms are complex and not well understood

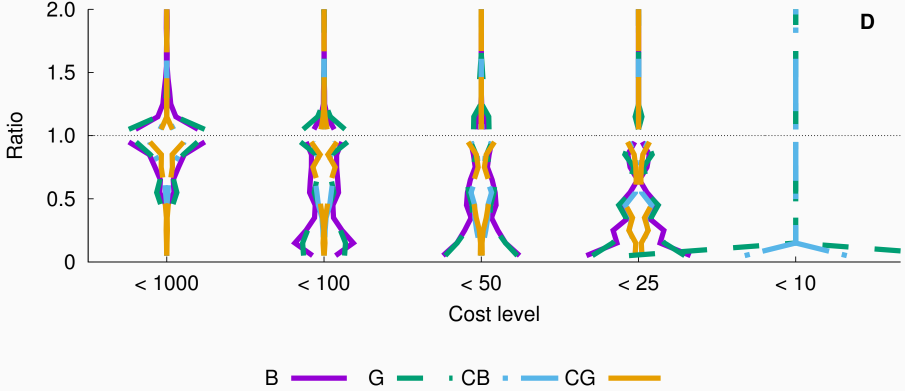
Distribution of cost effects – Stairs transformation



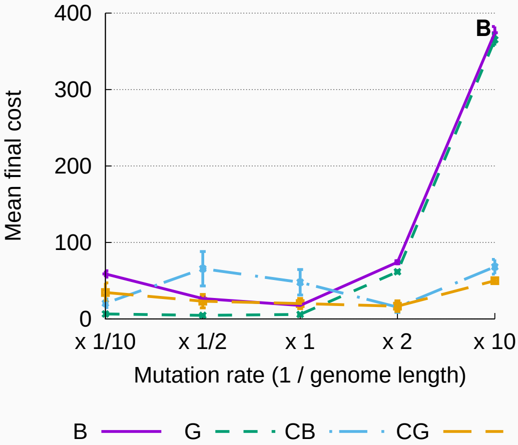
Distribution of cost effects – Moguls transformation



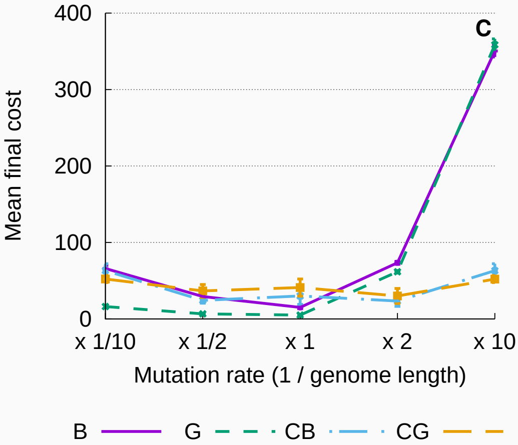
Distribution of cost effects – Sharktooth transformation



Robustness – Stairs transformation



Robustness – Moguls transformation



Robustness – Sharktooth transformation