Biologically-inspired algorithms and models

2. Theory of evolutionary algorithms

Maciej Komosinski

Reminder from earlier studies (and, possibly, a supplement)

- What does Holland's *schema theorem* talk about?
- What does the *building-block hypothesis* state?
- What does the *"No Free Lunch" theorem* tell us?

In order to construct a problem that will cause difficulty for the genetic algorithm, let's negate the building-block hypothesis: good building blocks, when combined, should constitute an unprofitable structure.

The simplest such case can be produced for genotypes of length 2 (this is a deceptive problem of order two).

Let's build a deceptive problem of order two

Let's assume that there exist four schemata:

```
*** 0 *** 0 ***
*** 0 *** 1 ***
*** 1 *** 0 ***
*** 1 *** 1 ***
```

Asterisks correspond to any number of non-determined bits (but the positions of determined bits are the same in all schemata). Let's denote the average fitness of schemata by f_{00}, f_{01}, f_{10}, f_{11} and the schema with two ones is the global optimum (f_{11}). For a problem to be deceptive, we want schemata with one determined '0' to be better than the corresponding schemata with one determined '1' – that is, at least one of the inequalities should be satisfied:

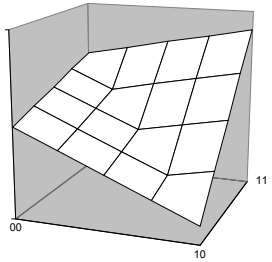
$f_{0*} > f_{1*}$ $\qquad$ $f_{0*} = (f_{00} + f_{01})/2$, etc.
$f_{*0} > f_{*1}$

The fitness landscape of the minimal deceptive problem

Such a problem is called the *minimal deceptive problem, MDP*, because a deceptive problem of order one does not exist.

There are two types of deceptive problems of order two (the plot on the right shows an example of a deceptive problem of order two, type I):

Type I: $f_{01} > f_{00}$
Type II: $f_{00} \geq f_{01}$



In *MDP*, the fitness function *cannot* be expressed as a linear combination of individual alleles, that is, in the form of $f(x_1, x_2) = b + \sum_{i=1}^{2} a_i x_i$.

The fact that a problem is deceptive does not mean that a genetic algorithm will not find the optimum. It means though that the fitness function (shown on the vertical axis) cannot be expressed as a linear combination of individual bits, and so the phenomenon of epistasis (nonlinearity) occurs. Usually, however, such a problem does not turn out to be GA-hard, that is, the genetic algorithm can find the optimal solution.

The behavior of the algorithm, though, depends on a number of factors – such as, for example, the initial existence of schemata in the population of individuals. In particular, if all four schemata occur in the initial population, the deceptive problem of order two, type I is not GA-hard. For type II problems, the performance of the algorithm depends on the distribution of schemata in the population: if the 00 schema prevails, the algorithm may converge to a suboptimal solution (although such a situation is rare). A more detailed analysis of deceptive problems can be found in [Gol02, pp. 46–48, 377].

Epistasis

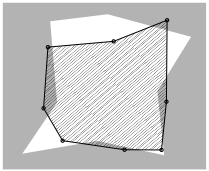
This is a property of a representation (and potentially, operators that act on it) – the degree of interdependence between different genes in a chromosome [Dav90]. If a given representation exhibits high epistasis, the phenotypic effect of certain genes depends on the alleles (values) of other genes (i.e., polygenicity).

Zero epistasis: each gene independently influences the value of the objective function (and then there is no merit in using EAs). For the effectiveness of EAs, the less epistasis the better. For some definitions of the objective function, when designing the representation and operators, it may be beneficial to accept a slight increase in epistasis if we gain a more favorable relationship between the topology of the search space and the fitness landscape.

Estimate roughly the epistasis for the following objective functions $f(x_1, x_2)$, where x are numbers – values of genes: $x_1 + x_2$, $x_1 - x_2$, $x_1 \cdot x_2$, $x_1 + x_2 \cdot x_2$, $x_1 + x_1 \cdot x_2$, $\frac{x_1}{x_2}$, $x_2 + \frac{x_1}{x_2}$.

Epistasis – an example

We are optimizing the shape of a 2D geometry (or a cross-section of a 3D object); genes represent the position of vertices in space (coordinates x_i, y_i).

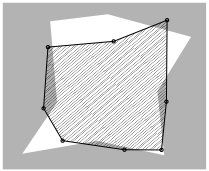


Compare the epistasis in the following scenarios:

1. Just the x_i, y_i genes and the mutation operator that moves a random vertex.
2. We introduce two additional genes: rotation and scale. The mutation of these individual genes changes the orientation (rotation) and size of the entire geometry.
3. Instead of these two genes, we introduce dedicated rotation-mutation and scaling-mutation operators.

Epistasis – an example

We are optimizing the shape of a 2D geometry (or a cross-section of a 3D object); genes represent the position of vertices in space (coordinates x_i, y_i).



Compare the epistasis in the following scenarios:

1. Just the x_i, y_i genes and the mutation operator that moves a random vertex.
2. We introduce two additional genes: rotation and scale. The mutation of these individual genes changes the orientation (rotation) and size of the entire geometry.
3. Instead of these two genes, we introduce dedicated rotation-mutation and scaling-mutation operators.

In scenario (1), adjusting the orientation and the size of the geometry is only possible by very many independent mutations of the vertex coordinates. However, in some problems (for example in online optimization, where the target shape keeps changing), the ability to quickly rotate and scale the geometry can accelerate convergence and improve the quality of obtained solutions. In scenario (2), consider the impact of a particular implementation (mathematical operations transforming genotype $\rightarrow$ phenotype) – think about the effects of combining mutations (1) and (2).

Epistasis – a formal description

Epistasis as the degree of interaction between genes [RW95; NK00]:

$$
\begin{aligned}
f(s) = \text{constant} \\
&+ \sum_{i=0}^{l-1} \text{effect of } s_i \\
&+ \sum_{i=0}^{l-2} \sum_{j=i+1}^{l-1} \text{interaction between } s_i \text{ i } s_j \\
&+ \dots \\
&+ \text{interaction between } s_0, s_1, \dots, s_{l-1} \\
&+ \text{random noise.}
\end{aligned}
$$

s_i – i-th gene (or allele at i-th gene, if s is a specific solution),
l – the number of genes (indexed starting from 0).

The interactions can be positive and negative: if, for example, the interaction between s_i and s_j has the same sign as the effects of s_i and s_j, then such an interaction amplifies the effect of these two genes and may be desirable.

Epistasis and mutation

The epistasis of genes has consequences when they are mutated: explore this plot.

Evaluating the efficiency and behavior of EAs can be performed either theoretically or experimentally (empirically).

Theoretical analysis results in well-established, definite knowledge, but often only very simple models (i.e., models with many assumptions) can be investigated in this way.

Empirical analysis of EAs results in less definite knowledge, and it is more difficult to make generalizations, but it is always possible to perform such analysis. Apart from testing the efficiency of the algorithm on the given optimization problem, well-known test problems* are often used, or parameterized models of problems that allow to control the degree of epistasis – such as the NK model** and its variations with neutrality: NKP (probabilistic) and NKQ (quantised).

*https://en.wikipedia.org/wiki/Test_functions_for_optimization
**https://en.wikipedia.org/wiki/NK_model

Consequences of neutrality in fitness landscapes [Gea+02]

- naively/traditionally, fitness landscape is thought to be "composed of hills", and mutations cause a change in fitness

Consequences of neutrality in fitness landscapes [Gea+02]

- naively/traditionally, fitness landscape is thought to be "composed of hills", and mutations cause a change in fitness
- this leads to the concept of "local optima" where solutions or their populations can be trapped

Consequences of neutrality in fitness landscapes [Gea+02]

- naively/traditionally, fitness landscape is thought to be "composed of hills", and mutations cause a change in fitness
- this leads to the concept of "local optima" where solutions or their populations can be trapped
- molecular research on RNA folding landscapes suggests that a large proportion of mutations at the molecular level are selectively neutral

Consequences of neutrality in fitness landscapes [Gea+02]

- naively/traditionally, fitness landscape is thought to be "composed of hills", and mutations cause a change in fitness
- this leads to the concept of "local optima" where solutions or their populations can be trapped
- molecular research on RNA folding landscapes suggests that a large proportion of mutations at the molecular level are selectively neutral
- neutrality is also present in many real-world optimization problems (popular "plateaus" or same-quality neighbors)

Consequences of neutrality in fitness landscapes [Gea+02]

- naively/traditionally, fitness landscape is thought to be "composed of hills", and mutations cause a change in fitness
- this leads to the concept of "local optima" where solutions or their populations can be trapped
- molecular research on RNA folding landscapes suggests that a large proportion of mutations at the molecular level are selectively neutral
- neutrality is also present in many real-world optimization problems (popular "plateaus" or same-quality neighbors)
- thus, many genotypes [$\rightarrow$ single phenotype] $\rightarrow$ single fitness

Consequences of neutrality in fitness landscapes [Gea+02]

- naively/traditionally, fitness landscape is thought to be "composed of hills", and mutations cause a change in fitness
- this leads to the concept of "local optima" where solutions or their populations can be trapped
- molecular research on RNA folding landscapes suggests that a large proportion of mutations at the molecular level are selectively neutral
- neutrality is also present in many real-world optimization problems (popular "plateaus" or same-quality neighbors)
- thus, many genotypes [$\rightarrow$ single phenotype] $\rightarrow$ single fitness
- neutrality: one of the reasons of punctuated equilibria

Consequences of neutrality in fitness landscapes [Gea+02]

- naively/traditionally, fitness landscape is thought to be "composed of hills", and mutations cause a change in fitness
- this leads to the concept of "local optima" where solutions or their populations can be trapped
- molecular research on RNA folding landscapes suggests that a large proportion of mutations at the molecular level are selectively neutral
- neutrality is also present in many real-world optimization problems (popular "plateaus" or same-quality neighbors)
- thus, many genotypes [$\rightarrow$ single phenotype] $\rightarrow$ single fitness
- neutrality: one of the reasons of punctuated equilibria
- if neutrality is frequent in the landscape, the risk of an EA population being trapped in local optima is low

Consequences of neutrality in fitness landscapes [Gea+02]

- naively/traditionally, fitness landscape is thought to be "composed of hills", and mutations cause a change in fitness
- this leads to the concept of "local optima" where solutions or their populations can be trapped
- molecular research on RNA folding landscapes suggests that a large proportion of mutations at the molecular level are selectively neutral
- neutrality is also present in many real-world optimization problems (popular "plateaus" or same-quality neighbors)
- thus, many genotypes [$\rightarrow$ single phenotype] $\rightarrow$ single fitness
- neutrality: one of the reasons of punctuated equilibria
- if neutrality is frequent in the landscape, the risk of an EA population being trapped in local optima is low
- accordingly, the role of the characteristics of reconfiguration operators (mutation/neighborhood, crossover) and the role of the genetic drift is increased

Consequences of neutrality in fitness landscapes [Gea+02]

- naively/traditionally, fitness landscape is thought to be "composed of hills", and mutations cause a change in fitness
- this leads to the concept of "local optima" where solutions or their populations can be trapped
- molecular research on RNA folding landscapes suggests that a large proportion of mutations at the molecular level are selectively neutral
- neutrality is also present in many real-world optimization problems (popular "plateaus" or same-quality neighbors)
- thus, many genotypes [$\rightarrow$ single phenotype] $\rightarrow$ single fitness
- neutrality: one of the reasons of punctuated equilibria
- if neutrality is frequent in the landscape, the risk of an EA population being trapped in local optima is low
- accordingly, the role of the characteristics of reconfiguration operators (mutation/neighborhood, crossover) and the role of the genetic drift is increased
- key role of the strict vs. non-strict inequality in local search implementations we discussed earlier (strict $\rightarrow$ all neutral moves are potential dead ends)

The main driving forces of evolution

- what is driving evolution? influencing the trajectory of the population?

The main driving forces of evolution

- what is driving evolution? influencing the trajectory of the population?
- fitness landscape vs. reconfiguration operators

The main driving forces of evolution

- what is driving evolution? influencing the trajectory of the population?
- fitness landscape vs. reconfiguration operators
- what happens when all or most mutations are detrimental and old individuals are not preserved?

The main driving forces of evolution

- what is driving evolution? influencing the trajectory of the population?
- fitness landscape vs. reconfiguration operators
- what happens when all or most mutations are detrimental and old individuals are not preserved?
- "bias" / "neutrality" of fitness landscape vs. bias/neutrality of reconfiguration operators

The main driving forces of evolution

- what is driving evolution? influencing the trajectory of the population?
- fitness landscape vs. reconfiguration operators
- what happens when all or most mutations are detrimental and old individuals are not preserved?
- "bias" / "neutrality" of fitness landscape vs. bias/neutrality of reconfiguration operators
- reconfiguration operators: changes in allele frequency, but usually no intended bias!

The main driving forces of evolution

- what is driving evolution? influencing the trajectory of the population?
- fitness landscape vs. reconfiguration operators
- what happens when all or most mutations are detrimental and old individuals are not preserved?
- "bias" / "neutrality" of fitness landscape vs. bias/neutrality of reconfiguration operators
- reconfiguration operators: changes in allele frequency, but usually no intended bias!
- you see a population of individuals with blue, green, and brown eyes; after some generations, everybody has green eyes. Why?

Genetic drift

- let's turn off selective pressure, mutation and crossover

Genetic drift

- let's turn off selective pressure, mutation and crossover
- marbles-in-a-jar example

Genetic drift

- let's turn off selective pressure, mutation and crossover
- marbles-in-a-jar example
- the discrete nature of populations (made of discrete individuals), so a perfectly equal distribution of alleles is not possible (e.g., 1/64 among 5000 individuals)

Genetic drift

- let's turn off selective pressure, mutation and crossover
- marbles-in-a-jar example
- the discrete nature of populations (made of discrete individuals), so a perfectly equal distribution of alleles is not possible (e.g., 1/64 among 5000 individuals)
- bottlenecks (population temporarily contracts to a very small size)

Genetic drift

- let's turn off selective pressure, mutation and crossover
- marbles-in-a-jar example
- the discrete nature of populations (made of discrete individuals), so a perfectly equal distribution of alleles is not possible (e.g., 1/64 among 5000 individuals)
- bottlenecks (population temporarily contracts to a very small size)
  - the founder effect

Genetic drift

- let's turn off selective pressure, mutation and crossover
- marbles-in-a-jar example
- the discrete nature of populations (made of discrete individuals), so a perfectly equal distribution of alleles is not possible (e.g., 1/64 among 5000 individuals)
- bottlenecks (population temporarily contracts to a very small size)
  - the founder effect
- the influence of trait distribution and population size

Genetic drift

- let's turn off selective pressure, mutation and crossover
- marbles-in-a-jar example
- the discrete nature of populations (made of discrete individuals), so a perfectly equal distribution of alleles is not possible (e.g., 1/64 among 5000 individuals)
- bottlenecks (population temporarily contracts to a very small size)
  - the founder effect
- the influence of trait distribution and population size
  - the probability that a trait will ultimately dominate the entire population is its frequency in the population

Genetic drift

- let's turn off selective pressure, mutation and crossover
- marbles-in-a-jar example
- the discrete nature of populations (made of discrete individuals), so a perfectly equal distribution of alleles is not possible (e.g., 1/64 among 5000 individuals)
- bottlenecks (population temporarily contracts to a very small size)
  - the founder effect
- the influence of trait distribution and population size
  - the probability that a trait will ultimately dominate the entire population is its frequency in the population
  - the expected number of generations for the total domination to occur is proportional to population size

Genetic drift

- let's turn off selective pressure, mutation and crossover
- marbles-in-a-jar example
- the discrete nature of populations (made of discrete individuals), so a perfectly equal distribution of alleles is not possible (e.g., 1/64 among 5000 individuals)
- bottlenecks (population temporarily contracts to a very small size)
  - the founder effect
- the influence of trait distribution and population size
  - the probability that a trait will ultimately dominate the entire population is its frequency in the population
  - the expected number of generations for the total domination to occur is proportional to population size
- the influence of randomness (unrestricted) vs. "fairness" – reminder: our earlier detailed discussion

Genetic drift

- let's turn off selective pressure, mutation and crossover
- marbles-in-a-jar example
- the discrete nature of populations (made of discrete individuals), so a perfectly equal distribution of alleles is not possible (e.g., 1/64 among 5000 individuals)
- bottlenecks (population temporarily contracts to a very small size)
  - the founder effect
- the influence of trait distribution and population size
  - the probability that a trait will ultimately dominate the entire population is its frequency in the population
  - the expected number of generations for the total domination to occur is proportional to population size
- the influence of randomness (unrestricted) vs. "fairness" – reminder: our earlier detailed discussion

Genetic drift

- let's turn off selective pressure, mutation and crossover
- marbles-in-a-jar example
- the discrete nature of populations (made of discrete individuals), so a perfectly equal distribution of alleles is not possible (e.g., 1/64 among 5000 individuals)
- bottlenecks (population temporarily contracts to a very small size)
  - the founder effect
- the influence of trait distribution and population size
  - the probability that a trait will ultimately dominate the entire population is its frequency in the population
  - the expected number of generations for the total domination to occur is proportional to population size
- the influence of randomness (unrestricted) vs. "fairness" – reminder: our earlier detailed discussion

---

- consequences can be counterintuitive, misinterpreted and misattributed if you don't know about these phenomena!
- the interaction of the fitness landscape (selection), reconfiguration operators, and genetic drift

An example of a theoretical analysis of a GA

Let's imagine a genetic algorithm (binary representation) with roulette selection, crossover, but no mutation.

What will happen after the algorithm has been running for a long time?

An example of a theoretical analysis of a GA

Let's imagine a genetic algorithm (binary representation) with roulette selection, crossover, but no mutation.

What will happen after the algorithm has been running for a long time?

How to prove it?

Let us consider a population with m individuals of length k. How many states the population can be in?

*https://en.wikipedia.org/wiki/Stochastic_matrix
**https://www.youtube.com/watch?v=nnssRe5DewE
***https://en.wikipedia.org/wiki/Absorbing_Markov_chain

Let us consider a population with m individuals of length k. How many states the population can be in? How many states are attractors?

*https://en.wikipedia.org/wiki/Stochastic_matrix
**https://www.youtube.com/watch?v=nnssRe5DewE
***https://en.wikipedia.org/wiki/Absorbing_Markov_chain

Let us consider a population with m individuals of length k. How many states the population can be in? How many states are attractors?

We will employ Markov chains; let π be a state, and P be a transition matrix containing probabilities of transitions between states*. There are 2^{mk} states. The distribution of probabilities of being in each state after n transitions** starting from state π is the π-th row of the matrix P^n, i.e., πP^n. Since there is no mutation, some states are absorbing (there is no transition out of them) – all individuals are identical; there are $a = 2^k$ such states. Consequently, we can describe the matrix P as***

$$P = \begin{bmatrix} I_a & 0 \\ R & Q \end{bmatrix}$$

where I_a is an $a \times a$ matrix with ones on the main diagonal and zeros elsewhere (the identity matrix), R is a $t \times a$ submatrix describing transitions to an absorbing state, and Q is a $t \times t$ submatrix describing transitions to transient (non-absorbing) states; $t = 2^{mk} - a$.

*https://en.wikipedia.org/wiki/Stochastic_matrix
**https://www.youtube.com/watch?v=nnssRe5DewE
***https://en.wikipedia.org/wiki/Absorbing_Markov_chain

For n transitions,

$$P^n = \begin{bmatrix} I_a & 0 \\ N_n R & Q^n \end{bmatrix}$$

where $N_n = I_t + Q + Q^2 + Q^3 + \ldots + Q^{n-1}$, and I_t is the identity matrix of size $t \times t$.

As n approaches infinity,

$$\lim_{n \to \infty} P^n = \begin{bmatrix} I_a & 0 \\ (I_t - Q)^{-1} R & 0 \end{bmatrix}.$$

So, as one can easily see by multiplying sample matrices in numpy or even in a spreadsheet, our algorithm starting from a non-absorbing state (probabilities $(I_t - Q)^{-1}R$) is bound to end up in some absorbing state and stay there (I_a). Let's calculate the probability of reaching an absorbing state [Fog00, p. 105]. Let $\Gamma = \{0, 1\}$. After n steps, our algorithm will end up in a state γ, $\gamma \in (\Gamma^k)^m$:

$$\Pr(\gamma \in A) = \sum_{i=1}^{a} (\pi^* P^n)_i = \sum_{i=1}^{a} \left(\pi^* \begin{bmatrix} I_a \\ N_n R \end{bmatrix} \right)_i$$

where $(\cdot)_i$ denotes the i-th element of the row vector, A is the set of all absorbing states, and π^* is a row vector that contains the probabilities of starting the algorithm with each state of the population. At the limit, the probability of absorption

$$\lim_{n \to \infty} \sum_{i=1}^{a} \left(\pi^* \begin{bmatrix} I_a \\ N_n R \end{bmatrix} \right)_i = \sum_{i=1}^{a} \left(\pi^* \begin{bmatrix} I_a \\ (I - Q)^{-1} R \end{bmatrix} \right)_i = 1.$$

An optional exercise

We use a generational evolutionary algorithm, a population of n individuals, g of n individuals are good, $n - g$ are bad. We use tournament selection (tournament size k) with replacement, in which if a good individual meets a bad one, the good one wins.

Assume that crossover and mutation do not change the quality of an individual (good remains good, bad remains bad).

1. Assuming that there are half of the good individuals ($g = \frac{n}{2}$) in the population, how will this proportion change in the next generation (i.e., after one selection)?
2. Assume that all individuals are subject to mutation. Mutation never improves a bad individual, however, it degrades a good individual with probability m. What is the minimum g to prevent the number of good individuals in the population from decreasing?

References I

[Dav90] Yuval Davidor. "Epistasis variance: Suitability of a representation to genetic algorithms". In: *Complex Systems* 4.4 (1990), pp. 369–383. URL: http://www.complex-systems.com/pdf/04-4-1.pdf.

[Fog00] David B. Fogel. *Evolutionary computation: toward a new philosophy of machine intelligence*. 2nd ed. IEEE Press, 2000.

[Gea+02] Nicholas Geard et al. "A comparison of neutral landscapes – NK, NKp and NKq". In: *Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02*. Vol. 1. IEEE. 2002, pp. 205–210. URL: https://eprints.soton.ac.uk/264208/1/cec1-comp.pdf.

[Gol02] David Edward Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, 2002.

[NK00] B. Naudts and L. Kallel. "A comparison of predictive measures of problem difficulty in evolutionary algorithms". In: *IEEE Transactions on Evolutionary Computation* 4.1 (2000), pp. 1–15. DOI: 10.1109/4235.843491.

[RW95] Colin R. Reeves and Christine C. Wright. "Epistasis in Genetic Algorithms: An Experimental Design Perspective". In: *ICGA*. 1995, pp. 217–224. URL: https://www.researchgate.net/profile/Christine_Wright5/publication/2504851_Epistasis_in_Genetic_Algorithms_An_Experimental_Design_Perspective/links/0912f50bf2d3d75799000000.pdf.