# Biologically-inspired algorithms and models

## 4. Nature-inspired mechanisms in evolutionary algorithms

Maciej Komosinski

# Gene duplication: a complex situation in nature

So far we have discussed haploid genotypes. In nature there also exist diploid genotypes, consisting of one or more pairs of chromosomes (there are 23 pairs in humans), and polyploid.

Such a representation requires dedicated operations, such as **segregation** (selecting chromosomes from pairs), **translocation** (moving genes to another chromosome), **duplication** (doubling a gene and inserting a copy into a chromosome) and **deletion** (removing a duplicated copy of a gene).

In addition, there are problems with the interpretation of an individual (due to the redundancy of information in the genotype). In nature, there exists a mechanism of **dominance** that determines which genes will manifest their influence; dominance can also occur not only between chromosomes, but also within them.

Discussion: when can redundancy of information in a genotype be an advantage and help in optimization?

| | | | | |
|---|---|---|---|---|
| $x_1'$ | $x_2'$ | $x_3'$ | $x_4'$ | ... |
| $x_1''$ | $x_2''$ | $x_3''$ | $x_4''$ | ... |
| ... | | | | |

# Gene duplication: what does it offer for optimization?

In EAs, information redundancy means genetic memory: although some traits do not manifest themselves externally, they are stored and passed on. Nature stores solutions that once proved beneficial in the form of latent genes that are dominated. Such genes – under the right conditions in the future – may begin to dominate themselves.

The use of a similar mechanism in an EA is reasonable when the objective function is variable – non-stationary (this corresponds to varying environmental conditions). It has been found experimentally [SW15, Fig. 2, 3] that in such a case, an algorithm with a diploid representation of individuals is able to faster adapt to changes in the environment (especially periodic ones) while finding good solutions.

There are many mechanisms of dominance: fixed, variable, random, deterministic, dominance of the better chromosome, etc. For the last mechanism, converting the traditional haploid algorithm into a polyploid one is very simple: what was a haploid genotype becomes a chromosome, the genotype consists of many chromosomes and its fitness (for selection) is the fitness of the best chromosome, selection operates on complete genotypes, and mutation and crossover operate on chromosomes just as they used to operate on haploid genotypes.

# The gene–phene relationship

In nature there exist:

- redundant genes: they do not affect the phenotype of an individual,
- the pleiotropic effect: a single gene can affect multiple phenotypic traits,
- the polygenic effect: a single phenotypic characteristic of an individual can be determined by the simultaneous interaction of multiple genes.

On the other hand, in a conventional GA: one gene — one trait. You can implement redundancy, pleiotropy and/or polygenicity as long as you are well aware of the consequences, advantages and disadvantages of each of these mechanisms – recall the concept of epistasis discussed earlier.

# Inversion

**Inversion** (reversing the order of a random subsequence of a genotype) allows the discovery of beneficial arrangements and associations of genes (in the sense of their meaning). Inversion involves the cooperation with the crossover operation, which cuts the sequences of genes while also cutting long schemata (recall the schema theorem). Fragments of the genotype nonlinearly related to each other and located far apart from each other will usually be separated in crossover, even though together they may constitute an essential and beneficial coupling.

However, the crossover operation in the presence of inversion is no longer so obvious: since the parents have different orders of gene meanings, after cutting and swapping genotype slices, the offspring usually do not have the full suite of genes (recall why such a problem does not occur in the "shuffle crossover" we discussed, but occurs in the messy GA covered earlier). Therefore, one needs to limit the set of parents that can be crossed over (to those that will yield a valid offspring), or use various complex crossover operations.

# Partial complement

**Partial complement** involves replacing a certain subset of genes in selected individuals with opposite values – complements.[*] The goal is to increase the diversity of individuals and to protect against excessive focusing of the population. Although this goal is achieved, it comes at the cost of a decrease in the speed of convergence of the genetic algorithm.

---

[*] First proposed in [Fra72, p. 70], this operation was called "migration" by the author, because it corresponds to the influx of individuals of the same species from outside of the deme (immigrants) to the deme (natives). https://en.wikipedia.org/wiki/Deme_(biology) Individuals inside and outside of the deme are genetically compatible, but are adapted to different environments, so the immigrants' alleles are not adapted to the conditions of the deme they enter – that is, they are different from the natives' alleles. This results in an increase of genetic diversity in the deme when immigrants are crossed over with natives.

# Additional mechanisms involving operators

- Variable (adaptive) operator probabilities: especially useful for non-stationary (variable) objective functions! [example of a stationary function]
- Adaptive operators: included in the genotypes of individuals and evolve alongside them.
- Crossover with seduction: preferences of an individual stored in its genotype and subject to evolution.

# Genetic redundancy, polygenicity and pleiotropy

Ideas tested in [Kwa97, p. 142]:

- Transition: the transfer of a single copy of a gene from one genotype to another, where it is inserted as a redundant gene.
- Transposition: a redundant gene swaps places with a phenotypic gene (i.e., one that affects the phenotype).
- Recrudescence: a small number of individuals in each generation have an increased likelihood of mutation and transposition. After recrudescence, there are significant phenotypic changes and most individuals are poor, but sometimes "*hopeful monsters*" arise.
- A crisis (a catastrophe): at a random moment in time, a significant reconfiguration of genotypes occurs in the entire population. Consequences – extinction of the population or the possibility of colonizing a new peak in the fitness landscape.
- Different mutation intensity ("macromutations" and "micromutations").

# Inspirations from nature in operators – summary

When imitating nature in implementations, it is important to remember that not all of its mechanisms are understood and explained. There are numerous problems in analyzing natural evolution – groups of researchers have their own divergent theories: *punctuated equilibria*[*], how new species form, how evolution works and at what level (levels?) – a gene, individual, group of individuals/herd, meme, population/species.

---

[*] https://en.wikipedia.org/wiki/Punctuated_equilibrium

# Modifications to the algorithm logic

**Variable population size** (e.g., by an order of magnitude) – for example, how long an individual exists in the population depends on its fitness value. Self-adaptation of the population size; “demographic explosions” – searching for optima.

**Cultural algorithms*** – „evolution in evolution”. Individuals improve their ability to learn through the experience acquired from the evolutionary process (like the cultural evolution of human societies**, which could potentially be confirmed by the super-exponential increase in the level of development of mankind).

---

*https://en.wikipedia.org/wiki/Cultural_algorithm

**https://en.wikipedia.org/wiki/Dual_inheritance_theory

# Nondeterminism revisited

## The indeterminism of evaluation can have many sources

Consider the following situations when we optimize:

1. An ML model evaluated using C-V: very time-consuming evaluation, natural nondeterminism, cannot be removed (unless at huge computational cost – all possible C-V splits).
2. A robot/3D design in simulation: very time-consuming evaluation, determinism of the idealized simulation – we know that it does not reflect reality.
3. A bot for a multiplayer game. Like in the robot example above – you can easily ensure the determinism of the opponents, but what will it result in...
4. A TSP route: nondeterminism of travel times in reality, but if we have average times in the matrix, the evaluation of permutations is instant.

   But what if we don't have average times, instead we have an idealized simulator of a traveling agent?

# Artificially introduced nondeterminism

Whenever in the real environment there is uncertainty, nondeterminism or very many/infinitely many states affecting the evaluation of a solution: when evaluating it, random noise is added in the simulated environment – thus solutions become more *robust*, although their optimization is more difficult because the evaluation is no longer deterministic.

This approach is often used when optimizing robots and in evolutionary design.

Recalling the discussion of the role and types of randomness we carried out when discussing selection, consider and compare the following scenarios: fixed combinations of evaluation parameter values (a regular grid), fixed combinations of evaluation parameter values (irregular, chaotic), nondeterministic combinations (changing with each evaluation).

# Dealing with nondeterministic evaluation

If the evaluation is nondeterministic, then despite the fact that EAs (without elitism) are relatively robust to evaluation inaccuracies (“uncertainty”), in order to make the fitness value more stable, averaging is used – i.e., multiple evaluations of each solution. Selection, unfortunately, promotes “the lucky”, and evaluating each individual multiple times – while radically increasing the computational cost – only reduces the magnitude of the problem. Reduces how many times when evaluating an individual 100 times?

# Dealing with nondeterministic evaluation

If the evaluation is nondeterministic, then despite the fact that EAs (without elitism) are relatively robust to evaluation inaccuracies ("uncertainty"), in order to make the fitness value more stable, averaging is used – i.e., multiple evaluations of each solution. Selection, unfortunately, promotes "the lucky", and evaluating each individual multiple times – while radically increasing the computational cost – only reduces the magnitude of the problem. Reduces how many times when evaluating an individual 100 times?

**Aging of individuals** – for example, removing the oldest individuals to make room for new ones. Discussion: think about what impact this will have on the search process compared to removing the worst individuals and to removing random individuals?

# Dealing with nondeterministic evaluation

If the evaluation is nondeterministic, then despite the fact that EAs (without elitism) are relatively robust to evaluation inaccuracies ("uncertainty"), in order to make the fitness value more stable, averaging is used – i.e., multiple evaluations of each solution. Selection, unfortunately, promotes "the lucky", and evaluating each individual multiple times – while radically increasing the computational cost – only reduces the magnitude of the problem. Reduces how many times when evaluating an individual 100 times?

**Aging of individuals** – for example, removing the oldest individuals to make room for new ones. Discussion: think about what impact this will have on the search process compared to removing the worst individuals and to removing random individuals? Named by the authors "regularized evolution" [Rea+19], it can mitigate problems arising from nondeterministic (noisy) evaluation ("the lucky" and "the unlucky" being subjected to selection) by increasing diversity and promoting exploration [Yin+19].

**Cloning** – another simple method of dealing with nondeterministic (noisy) evaluation. To avoid wasting computational resources on repeatedly evaluating each individual and averaging fitness, you can repeat the evaluation proportionally to the quality of the individual: in addition to mutation and crossover, introduce the cloning operator and average fitness of clones, see [KR01, Fig. 12].

# Specialization and speciation

Discussion: what answer do you know to the question

What is the purpose of mating types (distinct sexes) in nature (or: why do they exist) ?

---

[*] https://en.wikipedia.org/wiki/Evolution_of_sexual_reproduction

# Specialization and speciation

Discussion: what answer do you know to the question

What is the purpose of mating types (distinct sexes) in nature (or: why do they exist) ?

A special case of specialization in nature is the **differentiation of individuals into sexes**. Since so far there is no single, definite explanation of this phenomenon in biology*, let us theoretically consider a very simple model [Gol02, p. 181] to understand the advantages and disadvantages of sexual distinction.

---

*`https://en.wikipedia.org/wiki/Evolution_of_sexual_reproduction`

# Specialization – sex/gender: a simple model

In this model, we assume that the survival $s$ of offspring depends on what proportion the parents spend on two types of activities (for example hunting $h$ and nurturing $n$): $s = n \cdot h$. The time available to the parent is fixed and can be freely divided between these activities. We additionally introduce an optional (adjustable by parameter $a$) time cost associated with the parent "switching" between activities $anh$. The division of the parent's time satisfies equation $n + h + anh = 1$.

When there is no sexual differentiation, it is best for the parent to divide the time equally between the activities, $n = h = \frac{1}{2}$. Increasing $a$ degrades $s$.

In turn, when sexual differentiation is present and a pair of parents takes care of an offspring with $a = 0$, it does not matter how exactly both parents allocate their own times – the maximum $s$ will be achieved by collectively giving their offspring the same time $n$ and $h$. When $a > 0$, the only optimal situations are those when each parent devotes all of their time to a different activity (i.e., maximum specialization).

By eliminating the cost of "switching" between activities, the case of sexual differentiation of parents yields higher $s$ than the corresponding case of no differentiation.*

---

*Sweden: a bonus up to 1 500 € for splitting parental leave equally between parents.

# Speciation

Sexual differentiation is the specialization into two types of individuals. In nature, specialization also occurs in the form of splitting organisms into **species** (**speciation**), each of which has its own place (**niche**) and goals [Gol02, p. 185]. A similar phenomenon can be beneficial in EAs: if a function has several optima, we can expect that at the end of evolution, individuals will not cluster in one of them, but will be distributed equally among the optima (or in proportion to the quality of the local optima). Individuals similar to each other, occupying the neighborhood of a certain extreme, can be referred to as a species.

Discussion: how would you achieve a stable, lasting coverage of local optima with individuals, whose number should be proportional to the quality of the optima?

# Speciation – implementations

The effect of species formation and maintenance can be achieved, for example, through:

- replacing with new individuals the old individuals most similar to them (selection according to the crowding factor model discussed earlier),
- modifying the **fitness** value depending on the **similarity** of individuals.
  new fitness := original fitness/(the sum of similarities to other individuals)
  A popular, simple and effective method!
  More on similar methods in the following presentation.

Maintaining species can eventually lead to the discovery of a better solution than in a standard EA that does not modify fitness values. It is also beneficial for multi-criteria optimization, when we want individuals-solutions to evenly spread over the front of non-dominated solutions.

When using speciation, it may be advisable to limit crossover to individuals of the same species if crossover of very dissimilar individuals yields poor solutions.

# Coevolution

Inspirations from nature are also found in experiments with **co-evolving** systems, in which several populations exist that influence each other. The evaluation of the solutions of one population may depend on the solutions from another population.

Coevolution can be used to determine the optimal strategy, which is dependent on the strategies of other populations – all populations seek to outcompete the others or they cooperate. This situation leads to behaviors typical of parasites, predators, and it introduces an element of competition or cooperation.

We will talk about coevolutionary architectures and pitfalls in coevolution in a separate lecture.

# References I

[Fra72] Daniel Raymond Frantz. “Non-linearities in genetic adaptive search”. PhD thesis. University of Michigan, 1972. URL: https://deepblue.lib.umich.edu/bitstream/handle/2027.42/4950/bac2356.0001.001.pdf.

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

[KR01] Maciej Komosinski and Adam Rotaru-Varga. “Comparison of different genotype encodings for simulated 3D agents”. In: *Artificial Life Journal* 7.4 (Fall 2001), pp. 395–418. DOI: 10.1162/106454601317297022. URL: http://www.framsticks.com/files/common/ComparisonGeneticEncodings3DAgents.pdf.

[Kwa97] Halina Kwaśnicka. “Nadmiarowość genetyczna, poligeniczność i plejotropowość w algorytmach ewolucyjnych”. In: *Proceedings of the 2nd National Conference on Evolutionary Computation and Global Optimization*. Rytro: Oficyna Wydawnicza Politechniki Warszawskiej, Sept. 1997, pp. 137–144.

[Rea+19] Esteban Real et al. “Regularized evolution for image classifier architecture search”. In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 33. 1. 2019, pp. 4780–4789. URL: https://ojs.aaai.org/index.php/AAAI/article/download/4405/4283.

[SW15] Boris Shabash and Kay C. Wiese. “Diploidy in evolutionary algorithms for dynamic optimization problems: A best-chromosome-wins dominance mechanism”. In: *International Journal of Intelligent Computing and Cybernetics* 8.4 (2015), pp. 312–329. URL: https://www.researchgate.net/profile/Kay-Wiese/publication/283648490_Diploidy_in_evolutionary_algorithms_for_dynamic_optimization_problems/links/5eea7a3d92851ce9e7ec6cce/Diploidy-in-evolutionary-algorithms-for-dynamic-optimization-problems.pdf.

# References II

[Yin+19] Chris Ying et al. “NAS-Bench-101: Towards reproducible neural architecture search”. In: *International Conference on Machine Learning*. PMLR. 2019, pp. 7105–7114. URL: `https://proceedings.mlr.press/v97/ying19a/ying19a.pdf`.