Divergence of character and premature convergence: A survey of methodologies for promoting diversity in evolutionary optimization

Original
Divergence of character and premature convergence: A survey of methodologies for promoting diversity in evolutionary optimization / Squillero, Giovanni; Tonda, ALBERTO PAOLO. - In: INFORMATION SCIENCES. - ISSN 0020-0255. - STAMPA. - 329:(2016), pp. 782-799. [10.1016/j.ins.2015.09.056]

Publisher:
Elsevier Science Incorporated

Published
DOI:10.1016/j.ins.2015.09.056

Divergence of Character and Premature Convergence: a Survey of Methodologies for Promoting Diversity in Evolutionary Optimization

G. Squillero[1], A. Tonda[2]

Abstract

In the past decades, different evolutionary optimization methodologies have been proposed by scholars and exploited by practitioners, in a wide range of applications. Each paradigm shows distinctive features, typical advantages, and characteristic disadvantages; however, one single problem is shared by almost all of them: the "lack of speciation". While natural selection favors variations toward greater divergence, in artificial evolution candidate solutions do homologize. Many authors argued that promoting diversity would be beneficial in evolutionary optimization processes, and that it could help avoiding premature convergence on sub-optimal solutions. The paper surveys the research in this area up to mid 2010s, it re-orders and re-interprets different methodologies into a single framework, and proposes a novel three-axis taxonomy. Its goal is to provide the reader with a unifying view of the many contributions in this important corpus, allowing comparisons and informed choices. Characteristics of the different techniques are discussed, and similarities are highlighted; practical ways to measure and promote diversity are also suggested.

Keywords: Evolutionary Optimization; Diversity Preservation

---

1. Introduction

Evolution is the biological theory that animals and plants have their origin in other types, and that the distinguishable differences are due to modifications in successive generations. Natural evolution is not a random process, on the contrary, it is based on random variations, but some are rejected while others preserved according to objective evaluations. Only changes that are beneficial to the individuals are likely to spread into subsequent generations. Darwin called this principle "natural selection" in his *Origin of the Species* [19], a process where random variations simply "afford materials".

When natural selection causes variations to be accumulated in one specific direction the result strikingly resembles a deliberate optimization process. Indeed, such processes only require to assess the effect of random changes, not the ability to design intelligent modifications, inspiring several researchers. *Evolutionary computation* (EC) is the offshoot of computer science focusing on algorithms loosely inspired by the theory of evolution. The definition is deliberately vague since the boundaries of the field are not, and cannot

---

[☆] **This article has been definitely accepted for publication in a future issue of *Information Sciences*, but has not been fully edited. Content may change prior to final publication.**

[☆☆] Authors are listed in alphabetical order.

Email addresses: `giovanni.squillero@polito.it` (G. Squillero), `alberto.tonda@grignon.inra.fr` (A. Tonda)

[1] Politecnico di Torino, DAUIN, Corso Duca degli Abruzzi 24, 10129 Torino, ITALY

[2] UMR 782 GMPA, INRA, 1 Avenue Lucien Brtignres, 78850 Thiverval-Grignon, FRANCE

Preprint submitted to Information Sciences *September 27, 2015*

be, sharply defined. EC is a branch of computational intelligence, and it is also included into the broad framework of bio-inspired heuristics.

Divergence of character is a cornerstone of Darwinian theory: "the principle, which I have designated by this term, is of high importance on my theory, and explains, as I believe, several important facts" [19]. The English biologist favored sympatric speciation, and such principle perfectly rationalizes why breeds diverge in character not only from their common parents, but also from each other; and why differences, at first barely appreciable, steadily increase over generations. Indeed, the principle is "simplicity itself" [51]: the more the co-habitants of an area differ from each other in their ecological requirements, the less they will compete; therefore, in natural evolution, any variation toward greater divergence is likely to be favored.

Differently, artificial evolution in EC is plagued by an endemic lack of diversity: during evolutionary optimization processes, all candidate solutions frequently homologize. This situation has different effects on the different search algorithms, but almost all are quite deleterious. Such a *lack of speciation* has been pointed out by Holland in his seminal works [42], and nowadays is plainly recognized by scholars. The problem is usually labeled with the oxymoron "premature convergence", that is, the tendency of an algorithm to converge towards a point where it was not supposed to converge to in the first place.

EC is based by necessity on oversimplifications of the complex mechanics of nature. Darwinian theory focuses on members of the same species vying for limited resources, and the push for evolutionary diversification is prominent in ecosystems with limited resources [77], while it is not clear whether the competition between different species favors or rather impedes diversification [5]. In artificial evolution, on the other hand, there is no clear distinction between *intraspecific* and *interspecific* competition, because either the struggle is simulated at the level of individuals inside a single species, or at the level of species with no explicitly defined individuals.

Moreover, according to Darwin "the same spot will support more life if occupied by very diverse forms" [19]. But optimization algorithms use a *fitness function* that evaluates the goodness of each candidate solution with respect to a given task – that is, the whole ecosystem is indirectly modeled through its effects, and only few very specific facets are taken into consideration. The general inability to exploit environmental niches noted by Holland could be explained with the absence of such *natural spots* to survive in. Indeed, even the term "environment" is not widely used by Evolutionary Computation (EC) scholars, that favor "fitness landscape".

Over the years, EC has shown the capability to tackle quite difficult problems with very complex fitness landscapes. Evolutionary optimizers have been successfully exploited both in stationary and dynamic situations, and they were demonstrated able to identify either single optimums or whole Pareto sets in multi-objective problems. In such a wide variety of applications, promoting diversity inside the population has often been seen as the common key factor to improve performances (e.g., [73, 84, 13, 4, 15]).

Not surprisingly, scientific literature reports several sharp methodologies to promote diversity, that range from general techniques to problem-dependent heuristics. However, the fragmentation of the field and the difference in terminology led to a general dispersion of this important corpus of knowledge in many small, hard-to-track research lines – and consequently to the risk of neglecting effective solutions already known in similar domain, or re-discovering equivalent solutions in different communities.

The goal of this survey is to re-order and re-interpret the different approaches for promoting diversity into a single comprehensive framework, and to propose a taxonomy that enables the comparison of techniques originally presented for different EAs. For the nature of the topic and the vastness of the field, such a classification will be necessarily coarse-grained. Nevertheless, it could help scholars undertaking these issues, and practitioners tackling new problems.

It is important to stress that these considerations concern EAs applied to optimization tasks, only. There is a considerable amount of research lines that employ evolutionary computation as a means to analyze the dynamics of artificial life, with different problems and objectives, for which the premises of this work may not hold true[3].

In the following, section 2 briefly introduces EC and the notion of diversity, and section 3 introduces the proposed taxonomic scheme. The following sections use the first axis of the taxonomy as main theme to survey and classify the different approaches presented in literature, namely section 4 for lineage-based ones, 5 for genotype-based, and 6 for phenotype-based. Section 7 provides some recommendations for practitioners who approach this topic. Finally, section 8 concludes the paper and shows possible future research directions for the subject.

2. Evolutionary computation and the notion of diversity in evolutionary algorithms

EC does not have a single recognizable origin. Some scholars identify its starting point in 1950, when Alan Turing drew attention to the similarities between learning and evolution [83]. Others pointed out the inspiring ideas that appeared later in the decade, despite the fact that the lack of computational power impaired their diffusion in the broader scientific community [32]. More commonly, the birth of EC is set in the 1960s with the appearance of three independent research lines: John Holland's genetic algorithms (GA) [42]; Lawrence Fogel's evolutionary programming (EP) [33]; Ingo Rechenberg's and Hans-Paul Schwefel's evolution strategies (ES) [9]. The three paradigms monopolized the field until the 1990s, when John Koza entered the arena with genetic programming (GP) [46].

These four main paradigms, together with several variants proposed over the years, have been grouped under the umbrella term of "evolutionary algorithms" (EAs). In the 1990s, Kennedy et al. proposed the new "particle swarm optimization" (PSO), a population-based optimization technique that mimics the principles of social interaction rather than struggling for survival. Despite the apparent differences, PSO and all related *swarm-based* approaches are nowadays usually listed amongst EA techniques [60].

In such algorithms, a single candidate solution is an *individual*; the set of all candidate solutions that exists at a particular time represents the *population*. Evolution proceeds through discrete steps called *generations*. In each of them, the population is first expanded and then collapsed, mimicking the processes of breeding and struggling for survival. Moreover, most evolutionary optimization algorithms proceed by alternating phases of *exploration*, where distant parts of the search space are sampled, with phases of *exploitation*, where small neighborhoods of the best solutions are thoroughly investigated for improvements. Maintaining a set of solutions, EAs are more resilient than other optimization techniques to the attraction of local optima [29]; nevertheless, particularly promising points might induce an algorithm to a premature phase of exploitation, concentrating the whole population around few single points. When all candidate solutions are very close in the solution space, EAs could eventually lose their ability to explore new promising areas.

This phenomenon is a direct consequence of the mechanisms used to generate new individuals, collectively known as *genetic operators*. They can be roughly divided into *recombinations* and *mutations*: the formers mix together the information contained in two or more solutions to create new ones; the latter work by changing the structure of a single solution, and usually perform small adjustments. When all parents are very similar, the potential to *jump* to remote parts of the search space is strongly impaired: thus, "conventional wisdom suggests that increasing diversity should be generally beneficial" [72]. It is, therefore, useful to first discuss the notion of "diversity", the reciprocal concept of "similarity", and how they can be measured.

---

[3] Readers interested in *Artificial Life* as a discipline may find a comprehensive review of its roots, methodological tools, and applications in [6].

Diversity is usually quantified in three different ways: as a distance metric between individuals; as a measurable attribute of the individuals (*individual diversity*); as a characteristic of the population as a whole (*population diversity*). Individual diversity may be measured considering *how far* an individual is from the whole population, or from a subset; population diversity, similarly, may be defined as the average individual diversity. However, in many applications, a distance is not required to sensibly define the concept of "individual diversity", nor individual diversity is required to define an acceptable "population diversity".

When analyzing an evolutionary process, at least three different levels can be recognized: *genotype*, *phenotype*, and *fitness* – and promoting diversity might be pursued in different manners at each level. Considering diversity at the level of phenotype is usually the more effective way to enhance performances, but phenotypic diversity is also the hardest to define and the more demanding to calculate. In practical cases, assessing the diversity at the level of genotype requires acceptable effort, and there is more correlation between genotype and phenotype than between phenotype and fitness. This considerations may explain why the majority of the methodologies for promoting diversity take into account the genotypic diversity.

2.1. Genotype-phenotype distinction

In biology, the distinction between genotype and phenotype is apparent: the former is the genetic constitution of an organism; the latter is the composite of the organism's observable characteristics or traits. Moreover, albeit a precise definition of the term "fitness" led to some discussions [20], all scholars agree on the relationship between individual's fitness and reproductive success[4].

In EC, the genotype is the internal representation of the individual, or, more operatively, the entity that is directly manipulated by genetic operators; the fitness is the measure of how well the candidate solution is able to solve the target problem. While alternative definitions have been proposed in literature, for the purpose of this discussion the term "phenotype" denotes the candidate solution that is encoded in the genotype. That is, whenever the genotype cannot be evaluated directly by the fitness function, but needs to be transformed into something else, the phenotype is the *intermediate form* in which the genotype needs to be transformed into[5]. When the genotype can be directly evaluated, genotype and phenotype coincide. The genotype-phenotype mapping is, by definition, strictly deterministic, as there is no environment that could interfere with the process. Whether random elements need to be considered, scholars classify the fitness as *noisy*, assuming that the same phenotype could be evaluated differently. Such cases are not considered here.

In evolutionary paradigms such as Genetic Programming (GP) [46] and Linear Genetic Programming (LGP) [12] the distinction between genotype, phenotype and fitness is usually clearly visible. For instance, when GP is applied to symbolic regression, the genotype is the tree, the phenotype is the function, and the fitness relates to the difference between experimental data and those generated by the expression itself (Figure 1).

It may be worth noting that the phenotype is the function in its mathematical sense[6], or, equivalently, an expression in a defined canonical form. As a consequence, $x+y+2$, $y+2+x$, and $x+3+xy+y-1-yx$ express the same phenotype, although they are likely to derive from different genotypes. Similar considerations hold true for any other EA adopting complex individual encoding.

On the contrary, in the classic GA, modern EP, ES, PSO, and Differential Evolution (DE) [76] such a difference can be less evident. Individuals are usually represented by arrays of numbers – either binary or

---

[4] The phrase "survival of the fittest" was first used by Herbert Spencer in his *Principles of Biology* [74] to better describe Darwin's idea of "preservation of favoured races in the struggle for life."

[5] The distinction between "structure" and "behavior" found in [13] is also compatible with the above definition.

[6] A function f from S into T is a subset of the product set $S \times T$ with the property that for each element $x \in S$, there exists a unique element $y \in T$ such that $(x, y) \in f$.

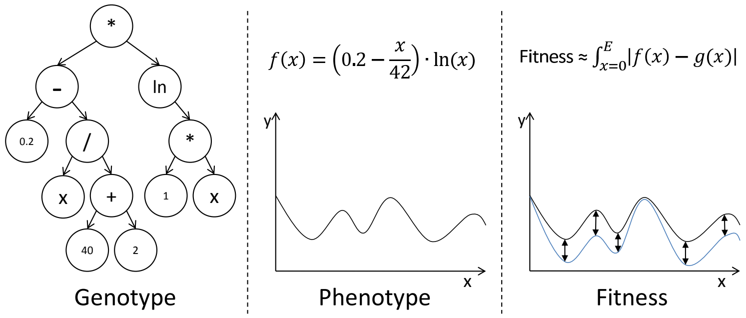

Figure 1: Genotype, phenotype and fitness for GP applied to a symbolic regression problem.

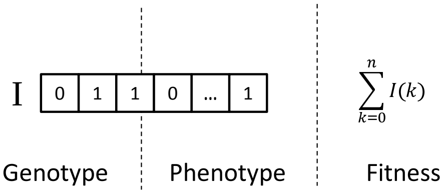

Figure 2: Genotype, phenotype and fitness for GA applied to the one-max test function.

real values – and for opposite reasons the phenotype has been sometimes identified with the array itself, and sometimes with the fitness value. As noted above, here the former option is adopted (Figure 2).

2.2. Measuring diversity at different levels

As a matter of fact, detecting whether two individuals are *clones*, i.e., identical, is often an easy task at any level. As seen in section 2.1, both the transformation of a phenotype into a genotype, and the evaluation of a phenotype are assumed to be fully deterministic processes. Thus, for two individuals x and y, different fitness values imply different phenotypes, and different phenotypes imply different genotypes:

$$\mathcal{F}_x \neq \mathcal{F}_y \Rightarrow \mathcal{P}_x \neq \mathcal{P}_y \Rightarrow \mathcal{G}_x \neq \mathcal{G}_y \tag{1}$$

where $\mathcal{F}$ denotes the fitness, $\mathcal{P}$ the phenotype, and $\mathcal{G}$ the genotype. However, different genotypes are not necessarily translated to different phenotypes, as different behaviors are not necessarily rewarded with different fitness values. Nor should the amount of diversity at one level suggest a similar amount at a different one.

As fitness values are usually real numbers, or vectors in $\mathbb{R}^n$, it may be quite easy to define distance metrics at this level. Such measures have *per se* little significance, however, unless they can be used as a proxy for different distances. When “phenotypic diversity is synonymous with fitness diversity” [18], the lack of difference between genotypes can be sensibly inferred from the fitness values being equal. Similarly, variety in fitness values could be used as a proxy for measuring population diversity at the level of phenotype. Intuitively, phenotype variations could cause more fitness variations when the fitness is a vector in a high-dimensional space $v_f \in \mathbb{R}^n$, as it happens in multi-objective optimization, because obtaining identical fitness values is less probable.

The *locality principle* states that small modifications of the genotype should correspond to small modifications of the phenotype, and small modification of the latter should yield small variations in the fitness (that is, in the typical terminology of ES scholars, the problem exhibits a *strong causality*). In this situation the fitness distance can be used as an effective measure for both genotypic and phenotypic distance. And, in practical terms, such smoothness may be regarded as a lucky foreteller of an easy problem. Indeed, *locality* is not an intrinsic characteristic of the problem, but of the genotype-phenotype-fitness mapping; and a doltish definition of the fitness function could cause different phenotypes to yield the very same value.

Measuring diversity in a population of individuals is a simple task if a distance metric is available. In a classic GA, for example, the genotype distance can be straightforwardly evaluated resorting to the Hamming distance [72]. In other cases, a considerable number of distance metrics can be defined over real-value genotypes, or over real-value fitnesses, a comprehensive analysis of which can be found in [18]. For EAs featuring more complex individual structures, such as GP, the literature presents several solutions. In his seminal work, Koza proposes to take into account the number of different genotypes contained in a population [46], and the idea is later considered as a sufficient upper bound of population diversity [47]. Genotypical diversity in GP can be also be measured using subtrees and their relative frequency in the population [44, 78]. In [53], numerical tags are assigned to each node in the population, in order to track their survival and the changes of context from the initial stages, confirming that populations in the final steps of the EA often descend from one single individual. A thorough analysis of diversity measurements in GP and their performances can be found in [13].

The *edit distance*, also known as "Levenshtein distance", is a well-known string metric used to compute the difference between two sequences [48]. It has been applied to GP genotypes considering single nodes insertion, deletions and substitutions as possible operations [57, 21]; a modified version proposed in [30] also takes into account the cost of replacing a node with one of a different type; and in [11], a similar idea is applied to LGP, considering the coding part of the genome along with a sequence of operators. Interestingly, in many applications the edit distance could be also used to appraise diversity between phenotypes, although, given its computational complexity, it may not be very efficient [34].

It is also possible to define a measure of diversity for a set of individual without relying on distance metrics. Shannon ideas [71] have been applied to fitness values in the population to define *entropy* and *free energy*: absence of changes or monotonic decreases in the grade of disorder in subsequent generations correlate with a fall into local optima [63]. The Shannon entropy is also used in [75]: nodes and pattern of nodes in the genotypes are regarded as symbols in messages. Then, individuals that increase the amount of information carried by the whole population are slightly favored regardless of their fitness value. The idea of bits of information is also applied in [34], where the distance between LGP individuals is computed as the symmetric difference between their sets of symbols.

It is always important to consider the possibility of creating *ad-hoc* distance measures, exploiting specific characteristics of the target problem. For example, [11] proposes alternative distance metrics for a program evolved through LGP, based on the number and type of test cases satisfied.

Interestingly, methodologies in EC often consider the average distance between solutions as a proxy for the global amount of diversity, and resort to a comparison of all possible pairs in a population, with algorithms of complexity $O(n^2)$, where n is the size of the population: in [86], the authors propose a new algorithm that is proved to obtain the same result in time $O(n)$.

3. Proposed taxonomy

It may be maintained that a methodology for promoting diversity alters the selection probability of individuals:

$$\overline{p}_{x|\Psi} = p_{x|\Psi} \cdot \xi(x, \Psi) \tag{2}$$

Table 1: Proposed taxonomic scheme.

Characteristic	**Possible values**
Element considered	Lineage
	Genotype
	Phenotype
Type of selection	Parent
	Survival
	Both
Context dependency	True
	False

where $\overline{p}_{x|\Psi}$ is the selection probability of individual x given that individuals in set Ψ are also chosen (the set Ψ may be empty); $p_{x|\Psi}$ is the same probability without the adopted methodology; and ξ the corrective factor. Such a definition does not imply that a mechanism operates explicitly on the selection operators, but rather the effects on selection probabilities are assessed to classify it. For instance, in the well-known island model $\xi(a, \{b\}) = 0$ when the two individuals a and b are on different islands.

We propose a simple taxonomic scheme, based on three independent categories (Table 1). No distinction is made whether the goal is to *preserve* the existing diversity in the population, or rather to *promote* it. The main classification rests on the individual's relevant characteristics, that is, which element influences most the value of ξ; this text considers *lineage* ($\mathcal{L}$), *phenotype* ($\mathcal{P}$), and *genotype* ($\mathcal{G}$). Then, the different methodologies can be categorized considering the *type of selection* influenced by the methodology: *parent selection*, *survival selection*, or both. Finally, it may be considered whether the probability of choosing an individual is influenced by the choice of another one (*context-dependent* methodologies) or not (*context-independent* methodologies).

The first axis of the taxonomy is the more apparent. A mechanism based on lineage considers the topology of the population, the individual's progenitors, or, more generally, the *conditions* of its birth, for calculating ξ. Mechanisms based on the genotype evaluate the individuals' internal structure, while mechanisms based on the phenotype take into account how individuals behave.

The fitness is not considered here, because we maintain that it is always used as a proxy to measure distance at a different level. Some methodologies consider the fitness of individuals to determine which one should be removed during, or prior to, diversity promotion, and here they are categorized as phenotype-based.

The second categorization is based on the type of selection modified by the diversity preservation mechanism: *parent selection* or *survival selection*. During the former, parents are chosen to generate offspring; while the latter individuals are chosen to survive up to the next generation.

The last distinction is between *context dependent* preservation mechanisms and *context independent* ones. The former methods base the value of the corrective factor ξ on the value of Ψ; the latter methods assume:

$$\forall \Psi : \xi(x, \Psi) = \xi(x, \emptyset) = \xi(x) \tag{3}$$

In other words, a methodology is classified as context dependent if the probability of selection for an individual x (reproduction or survival) is modified in function of the presence of other (possibly specific) individuals in the set Ψ of individuals already selected. On the opposite, a methodology is considered context independent if the probability of selection for an individual x is not influenced by the composition of set Ψ.

If a mechanism operates on parent selection and the algorithm does not implement sexual recombination, it is considered context-independent.

The latter distinction is relevant because it may help better understand how the mechanisms operate. Intuitively, context-dependent methodologies support the exploitation of diversity already inside the population. However, it may be generally less straightforward to add such techniques to an existing algorithm, and they may strongly bias the whole evolutionary process. According to this point of view, *migrants* can be seen as the necessary patch to un-bias the evolution after a division into *islands* enforces a strictly context-dependent scheme.

All the techniques presented in the following, well-known approaches found in literature, are organized using the proposed taxonomy. Table 2 on page 20 shows the complete summary of the classification performed: the first column reports the section of the paper where the technique is discussed in more detail; the *main element* column, $\mathcal{L}$ indicates lineage, $\mathcal{G}$ genotype, and $\mathcal{P}$ phenotype; in the *selection* columns, “P” stands for “parent” and “S” for “survival”; the last column reports whether the considered technique depends on the context.

4. Lineage-based methodologies

In *lineage-based* methodologies the value of ξ does not depend on an individual structure or behavior, but it can be determined considering the *circumstances* of its birth (e.g., time, location). Such techniques can thus be applied to any kind of problem, even in addition to other diversity preservation methods.

Several techniques belonging to this category are based on limiting interactions between individuals, or creating constraints on selection for reproduction or survival. Island models and cellular EAs, which share the basic idea of fractionating the panmictic population of a classical EA into several sub-populations, are sometimes called *decentralized* EAs and are the most popular lineage-based methods in practice.

4.1. Island models

A popular lineage-based diversity method consists of splitting the main population into sub-populations, and greatly limiting the exchange of individuals between the sub-populations. EAs resorting to this technique are often called *island models* [85], or *distributed EAs*. The intuition behind this technique is that, since EAs are stochastic in nature, different populations may explore different parts of the search space. Since a larger population is also proven to be beneficial, many island-model EAs employ a periodic exchange of the best individuals between sub-population that is usually termed *migration*, allowing the recombination operators to mix useful traits emerged separately.

Island models are easy to implement and well-suited for parallel or distributed computing, so they have been and are successfully used in real-world problems [10, 56, 54]. In addition, the computational overhead of managing several islands is almost negligible, and concentrated in the migration function, usually performed once every *epoch* (few generations). An example of the benefits provided by island models is summarized in Figure 3. Belonging to an island influences both the probability of reproduction and survival of an individual. The technique is context-dependent, because individuals in a sub-population can be selected only with others on the same one.

4.2. Segregation

Besides island-model and cellular EAs, other diversity preservation approaches are also based on the notion of limiting interaction between individuals. *Segregation* [1] is a lineage-based diversity technique, where the global population is initially split into *N* sub-populations that evolve independently. Once all sub-population reach *stagnation* (e.g., the best individual has not improved for a user-defined number of

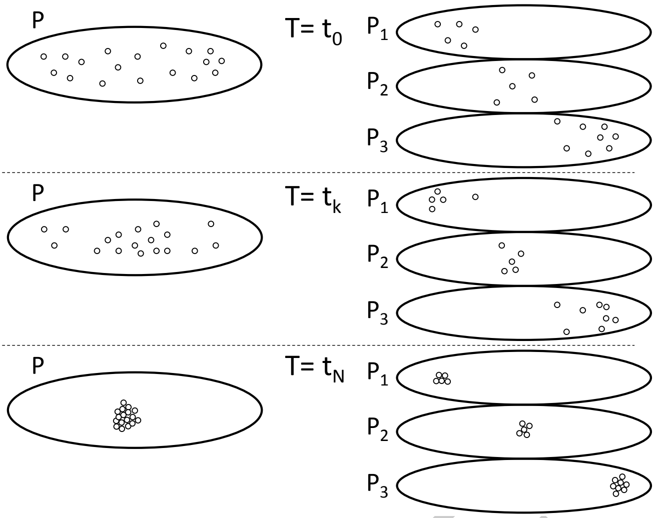

Figure 3: Distribution of solutions over generations, in a single-population EA (**left**) against a island-model EA with three sub-populations (**right**). Even if the global number of individuals is the same, the isolated sub-populations converge on different optima in the search space. Eventually, migration of individuals between sub-populations can help the algorithm to escape local points of attraction and converge on global optima.

generations, or a user-defined limit of iterations is reached), they are merged into $N - 1$ populations, and the evolution is resumed. The process is iterated until a single panmictic population reaches stagnation, see Figure 4 for an example. Like for islands, the rationale of this method is that different sub-populations will explore different areas of the search space: when each segregated sub-population converges on a local optimum, they are merged with the hope of finding better optima by combining solutions in different parts of the search space. Segregation influences both reproduction and survival, and it is context-dependent.

4.3. Cellular EAs

Cellular EAs [62, 2], also known as *lattice models*, limit the interaction between individuals by creating a structure similar to that of cellular automata [81]: candidate solutions evolve in overlapped neighborhoods, often over a bi-dimensional grid of individuals, such as the one presented in Figure 5. When an individual is selected for reproduction, it can only combine with others in the same neighborhood. Newly created solutions replace individuals in the same neighborhood of their parents, usually following a fitness-based criteria. Like cellular automata, cellular EAs can be either *asynchronous* or *synchronous* [70], depending on the strategy of individual replacement and population update.

Since cellular EAs promote the smooth diffusion of good solutions, creating temporary niches, they preserve diversity better than classical EAs with a panmictic population. In fact, cellular EAs present a large number of tightly connected, small sub-populations, while island models maintain a small number of loosely connected, large sub-populations. Still, this approach is generally more complex to implement than islands. This technique has an influence on both parent and survival selection, and it is classified as context-dependent because individuals can be chosen only among the neighborhoods.

4.4. Aging

In *generational approaches*, the parental population is completely replaced by the offspring before applying survival selection: the main advantage of this approach lies in an added resilience against local optima.

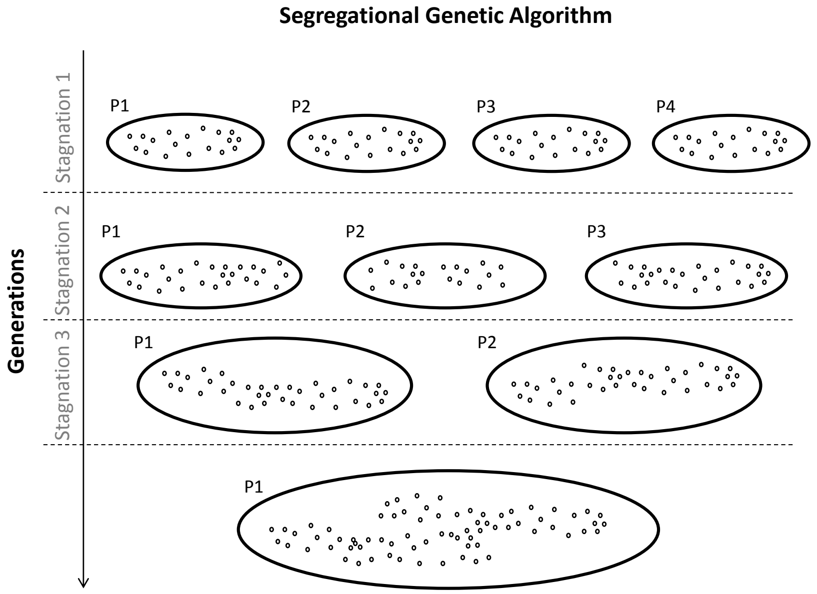

Figure 4: Segregational genetic algorithm, starting with 4 subpopulations. Each time a stagnation condition is reached, all individuals are mixed and merged into $N - 1$ subpopulations, and the process is iterated until a single panmictic population is obtained.

In its basic form, *Aging* can be seen as a softened version of a generational approach: individuals do age, and are discarded after A_{max} generations [14]. Indeed, a full generational EA can be seen as an EA with aging where the maximum age is set to $A_{max} = 1$.

This technique has an influence on both parent and survival selection, and it is classified as context-independent. However, the aging process could be influenced by the performance of the individual or by its fitness [64]. In such cases, the methodology should be considered as based on phenotype.

4.5. *Deterministic crowding*

The basic idea of *Deterministic crowding* [50] is to generate a competition between the children individuals and their parents. This technique resembles *standard crowding* (section 5.3). Every time the offspring is created, if it is fitter than the parents it is inserted in the population and the parent individual is removed; otherwise, the child individual is removed. Differently from standard crowding, deterministic crowding does not require the explicit definition of a distance between individuals, and that is why it is classified under lineage-based techniques, while standard crowding falls into the class of genotype-based methods. Deterministic crowding does not influence parent selection, and it is independent from the context.

4.6. *Allopatric selection*

Closely related to deterministic crowding, *allopatric selection* [82] puts all offspring generated during the same application of certain genetic operators in competition: only the fittest progeny is stored in the population, and then the standard fitness-based survival selection is applied. In a context where several genetic operators may produce a considerable number of children each, this technique is used to avoid early

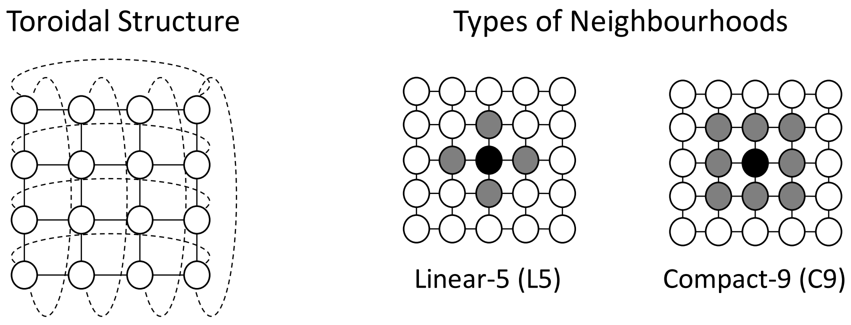

Figure 5: On the left, the classical structure of cellular EAs. Solutions are placed in a toroidal bi-dimensional grid. On the right, examples of neighborhoods. Neighborhoods are user-defined, and usually follow one of the portrayed structures: linear-*n* (covering up to *n* elements in the same row and column) or compact-*n* (considering the $n - 1$ nearest individuals in every direction, including the diagonal). In practice, two of the most used are linear-5 and compact-9.

colonizations of the population by successful lineages. For a comparison with deterministic crowding, see Figure 6. Allopatric selection is a context-independent technique that acts on survival selection only.

4.7. Gender

Another diversity-promoting technique is to add a feature inside the EA, representing the *gender* of an individual: crossovers are then allowed to operate between individuals of different genders, only [3, 35]. Variants of this approach try to exploit the presence of more than two sexes to further promote diversity [49], while other make use of different mutation rates for each sex [65]. In principle, gender could be managed even externally, through a reference table embedded in the algorithm. However, if the gender is inserted inside the genome and it is inheritable, its effect is to add a new trait with the only purpose to ease the evaluation of diversity, thus, the methodology should be classified as genotype-based. Several methodologies that are labeled with the word "gender" use the fitness values as discriminant [61], and should therefore be included into phenotype-based methodologies. All gender-based techniques are context-dependent and influence reproduction alone.

5. Genotype-based methodologies

A considerable number of methods exploit information at genotype level to promote diversity inside the population. Such techniques can be effective especially when it is straightforward to define a distance measure between different individuals: distances are often used to avoid overexploitation of peaks in the fitness landscape and to promote the generation of new solutions very *far* from the most successful ones.

The most prominent techniques in the field belong to the family of *niching methods*, derived from Holland's observations on fitness sharing and the successive refinements of Goldberg, Richardson and Deb [36, 42, 25]. The basic idea is to achieve the emergence of artificial *niches* in the search space, following the paradigm of natural niches: in nature, a niche is defined as a subspace in the environment with a finite amount of physical resources, that can support different types of life. An example is reported in Figure 7.

Niching methods can be further divided into two classes [66]: *explicit neighborhood* methods, that require an explicit definition of the size of a niche through a parameter called *niche radius*; and *implicit*

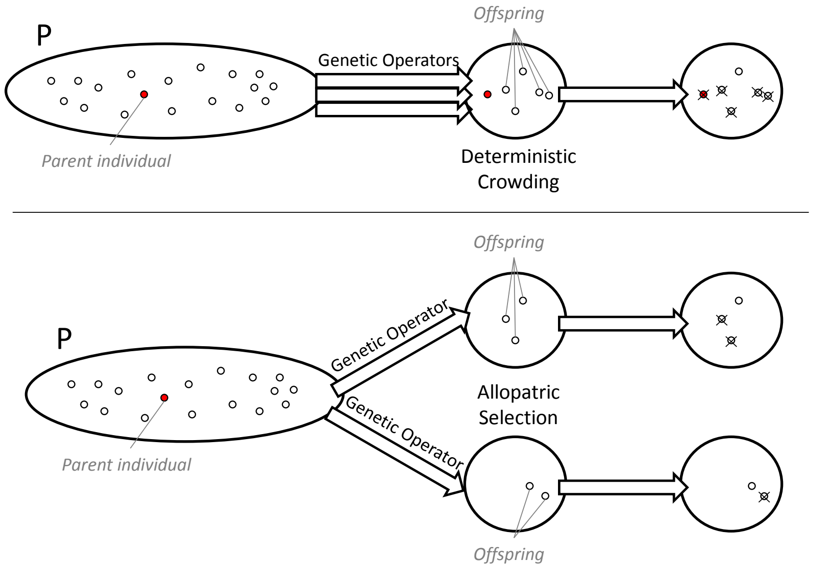

Figure 6: Deterministic crowding (**top**) and allopatric selection (**bottom**) compared. In deterministic crowding, both parents and offspring compete for a place inside the population; in allopatric selection, the competition is between children individuals generated during the same application of a genetic operator, only.

neighborhood methods, where the algorithm requires no information about the search space. It is important to notice that, without a distance metric defined between individuals at genotype or phenotype level, explicit neighborhood techniques cannot operate. All niching methods introduce an overhead in the evolutionary process, but the computational effort required to enforce diversity is usually negligible when compared to fitness evaluation in real-world problems.

Besides niching techniques, many other research lines follow the idea of using information gathered from the genotype of the population to maintain and promote diversity.

5.1. Fitness Sharing

A well-known example of niching technique with an explicit niche dimension is *fitness sharing*. Sharing [52] reduces the attractiveness of densely populated regions of the search space, lowering the fitness value of individuals in the same niche by a value proportional to the number of individuals. Given an individual I_k and its fitness value $f(I_k)$, its new fitness with sharing $f'(I_k)$, can be expressed as follows:

$$f'(I_k) = \frac{f(I_k)}{\sum_{i=0}^{individuals} sh(I_k, I_{i \neq k})} \tag{4}$$

with

$$sh(I_a, I_b) = \begin{cases} 1 - \left(\frac{d(I_a, I_b)}{\sigma_s}\right)^{\alpha} & d(I_a, I_b) < \sigma_s \\ 0 & d(I_a, I_b) \geq \sigma_s \end{cases} \tag{5}$$

where $d(I_a, I_b)$ is the distance function between two individuals I_a and I_b and α is a constant parameter which regulates the shape of the sharing function (commonly $\alpha = 1$). For its characteristics, fitness sharing

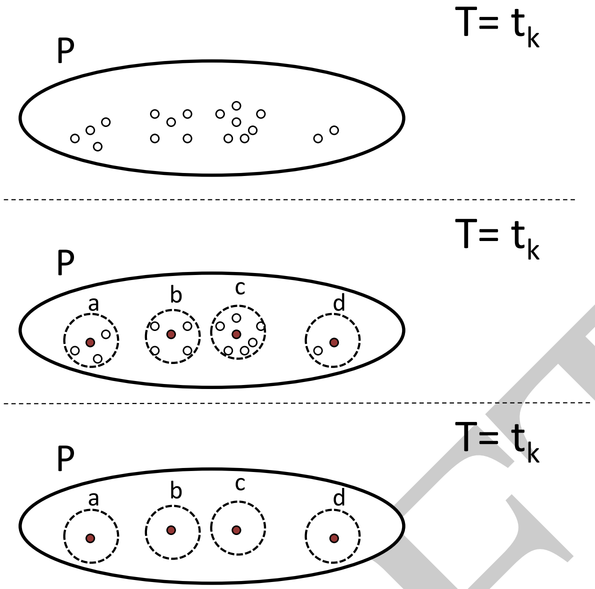

Figure 7: When niching is used, individuals are in competition with other individuals in the same niche, only. This method makes it possible for low-fitness solutions to survive, provided they are in a relatively unexploited area of the search space. For example, all individuals in niche **c** could have a higher fitness value than individuals in niche **d**, and in a classic scenario individuals in **d** would be removed from the population. When using niching, an individual in **d** is nevertheless preserved, since it was in a sparsely explored area of the search space.

influences both the probability of reproduction and survival of an individual, and it is context-independent since the selection of an individual is independent from other individuals already chosen for that purpose.

5.2. Clearing

Clearing [58] is part of the niching techniques with an explicit neighborhood: differently from sharing, however, it relies upon the concept of *dominant individuals* of the niche. Inside each niche, the k best individuals preserve their fitness, while all others have their fitness reset. As in the sharing method, individuals belong to the same niche if their distance in the search space is less than a dissimilarity threshold σ_s, here called *clearing radius*. The complexity of the procedure is $O(qN)$, where q is the number of niches maintained during the search. Clearing is sometimes used along with other diversity-promoting techniques, such as in [8]: it influences the selection probabilities for both reproduction and survival, and it is context-independent.

5.3. Standard Crowding

Standard crowding [22] is a niching technique with an implicit neighborhood: it makes use of a scheme where only part of the population reproduces and dies at each generation. Every time a new individual is created, a sub-population of size CF is randomly drawn, and the offspring replaces the most similar solution inside this sub-population. The *similarity* is measured at the level of genotype, and if the locality principle is not true, the methodology could lead to *replacement errors*, as noted by Deb and Goldberg [25]. This technique influences survival selection, and it is context-independent.

5.4. Restricted Tournament Selection

Among niching techniques with implicit neighborhoods, one of the most successful is *Restricted Tournament Selection* (RTS) proposed by [41]. RTS selects two individuals from the population, to undergo crossover and mutation; then, each offspring is compared with the closest element in a randomly drawn sub-population of size CF; finally, the winners are inserted into the global population, while the losers are discarded. This process is repeated $N/2$ times, where N is the size of the global population. The complexity of RTS is $O(CF \cdot N)$, and it can grow up to $O(N^2)$ if $CF = N$. This technique is context-independent, and it influences the survival probability of individuals.

5.5. Sequential Niching

An interesting variant of niching is the *Sequential Niching* [7], whose basic idea is to alter the fitness of parts of the search space where good solutions have already been found. Differently from other fitness sharing approaches, it performs several iterations of the EA: the most promising points in the search space after each run are altered so to become less interesting in further executions. This method might be more performing than standard fitness sharing on multimodal search spaces, since there is evidence that mating between individuals on different peaks often leads to uninteresting solutions [23]. Sequential niching is a context-independent technique that influences both the probability of reproduction and survival of individuals.

5.6. Reference points partitioning

To maintain diversity during a multi-objective optimization problem with several objective functions, Deb and Jain improved their *Non-Sorting Genetic Algorithm II*[7] (NSGA-II) creating the *Many-objective NSGA-II* (or NSGA-III, as it is sometimes called) [26, 27].

The new tool uses a predefined set of *reference points*: reference points are initially set, either manually or with an automatic procedure; then, during evolution, each individual is dynamically associated to the closest reference point, partitioning the population; eventually, a traditional niching is used on each subset. This methodology is a context-independent technique that influences both the probability of reproduction and survival of individuals.

5.7. Delta entropy and pseudo entropy

In [75], the population is considered as a message, composed of the concatenation of all the individuals, with each gene corresponding to a symbol. Authors calculate the *entropy* associated to the message using Shannon's formula:

$$H = -\sum_{s \in P} f(s) \cdot \log(f(s)) \tag{6}$$

where s is a symbol (i.e., a gene) in the population P, and $f(s)$ is its frequency. The effect on the global population entropy caused by each individual is considered as an indication of the amount of diversity brought by it. Then, with a given probability candidate solutions are compared on their capability to increase the global entropy instead on their fitnesses.

---

[7] Other techniques used by NSGA-II can be found in section "Crowded-comparison operator" (6.3)

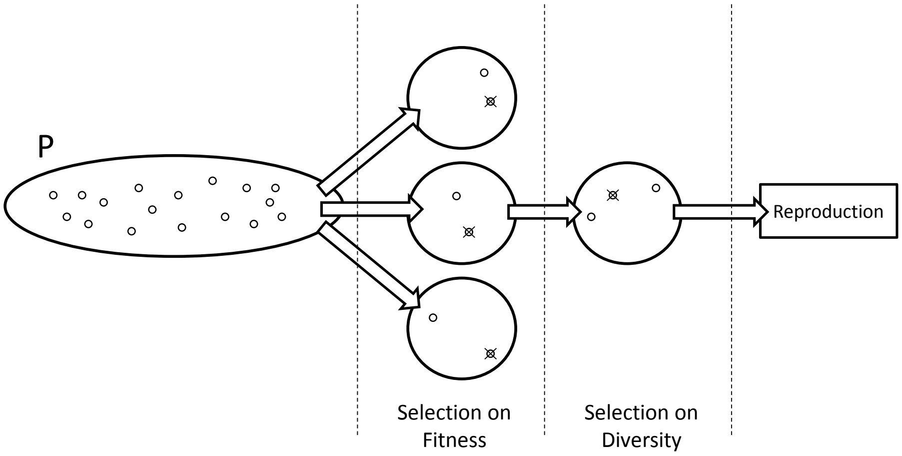

Figure 8: An example of two-level diversity selection. In the first part, a fitness-based tournament selection between two individuals is repeated 3 times. In the second part, the three winners are put in competition with each other, and the two most *diverse* are selected for mating.

The approach was called “pseudo entropy” because authors acknowledged that the computation of the population entropy was not fully correct. This problem was solved in [64], where the concept of *symbol* was further extended, taking into consideration genes and small sequences of genes.

5.8. Two-level diversity selection

When a genotype-level distance between individuals can be defined, diversity can be promoted by putting a selective pressure on diversity as well as fitness. This idea, presented in [11], employs a two-level tournament selection, where three individuals are first chosen, based on their fitness values, and subsequently the two with *maximum distance* are finally selected for reproduction. In this case, the distance is evaluated between the three fittest individuals, only, and the rest of the population is not considered. A scheme of the technique is proposed in Figure 8. The two-level diversity selection is context-dependent, and acts on the selection of the parents.

5.9. Tarpeian method

Named after the Tarpeian rock in Rome, the infamous execution place for traitors and criminals, this method proposes to randomly kill individuals if they do not satisfy a genotype-level metric [59]. In the original paper, the purpose is to limit bloating in GP by lowering the selection probability of programs bigger than a user-defined threshold, but in general the technique can be used to promote diversity by favoring less fit individuals that score well on the considered genotype-level metric. The author argues that this method dynamically and non-deterministically creates *fitness holes* in the fitness landscape, and that it could be superior to just creating static *holes* with other bloat-controlling techniques. The Tarpeian method is context-independent, and it influences both survival and reproduction.

5.10. FOCUS and GDEM

Another interesting approach is to exploit the potential of *Multi-Objective Evolutionary Algorithms* (MOEA) by adding a diversity-related function among the objectives to optimize. The contribution of a single individual to the variety of the population is thus evaluated. This idea is first presented in [21] and applied to GP in order to limit the growth of solutions over time, using a technique called *Find Only and Complete Undominated Sets* (FOCUS), where only non-dominated individuals are preserved in the population.

The *Genetic Diversity Evaluation Method* (GDEM) [80] makes use of a similar idea. Genetic diversity is used again as a second objective in a MOEA, but this time it is managed so that solutions with the same rank with regards to other objectives will not dominate one another, independently from the values of their diversity measure. In other words, individual $u(rank_u, diversity_u)$ dominates individual $v(rank_v, diversity_v)$ if and only if:

$$rank_u > rank_v \wedge diversity_u \geq diversity_v$$

Both these techniques have been applied to MOEAs, but in principle they could be used for single-objective optimization as well; both alter the probability of selecting an individual for survival; and they are context-independent since choosing a specific individual would not alter the probabilities of selecting a second one for the same purpose.

5.11. Diversifiers

A more refined and computationally expensive version of the *random immigrants* strategy (section 6.1), presented in [45], tries to fill *gaps* of under-represented areas in the search space with individuals, called *diversifiers*, created specifically for the task. If a two-dimensional distance metric d can be defined over a population of individuals, a distance space $D_G \subset \mathbb{R}^2_{0+}$ can be used to approximate areas of *relative emptiness*, where few genomes can be found. In particular, there are algorithms able to find the largest-area rectangle R_{max} that is axis-parallel to the x, y axis of D_G, whose opposite vertices are two genomes, in $\mathcal{O}(n \cdot log(n))$ where n is the number of individuals. The part of the genotype space R_{max} can then be used to generate individuals able to fall inside it.

This technique is context-independent and indirectly influences the probability of reproduction and survival of other individuals.

6. Phenotype-based methodologies

All techniques that operate directly in the fitness space, either artificially altering the fitness landscape, or relying upon fitness-level information to promote diversity are grouped under the label *phenotype diversity*. At a first glance, diversity preservation at fitness level might look impractical: if the fitness landscape of the problem is multi-modal, there could be several genotype-level points corresponding to the same fitness value, and enforcing a distinction could thus appear meaningless. Still, especially if a genotype-level distance measure is hard to conceive, even single-objective fitness can be exploited to enforce diversity.

Moreover, the thriving sub-field of MOEAs relies upon a multi-dimensional fitness space. Since MOEAs return a set of non-comparable candidate solutions, it is in the user's interest to obtain very diverse solutions, avoiding the concentration of individuals on some parts of this multi-dimensional fitness landscape, only.

6.1. Random immigrants

A simple but effective technique to maintain diversity, often adopted in problems where the fitness landscape is dynamic, is to periodically add *random immigrants* [38] to the current population. Such individuals are randomly generated, thus offering fresh material for the genetic operators to exploit: usually, however, their fitness value is very small when compared to the other individuals in the population. For this reason, algorithms employing this technique often act on the survival selection, favoring the unfit but useful random immigrants for, at least, some generations [79].

The presence of random immigrants influences the probability of reproduction and survival of other individuals: both these techniques are context-independent.

6.2. Extinction

The difficulty to survive faced by *random immigrants* is mitigated by the *extinction* methodology [37], that operates by periodically removing a significant amount of the population. More in detail, at each generation, a stress factor $\eta(t)$ is generated according to $\eta(t) \sim U(0, 0.96)$. Assuming a minimization problem, for each individual I_i the algorithm scales its fitness $f(I)$ to the interval $[\alpha; 1]$ on the basis of the following formula:

$$f'(i) = \alpha + (1 - \alpha) \cdot \frac{f(I_i) - f(I_{max})}{f(I_{min}) - f(I_{max})} \tag{7}$$

where $f(I_{max})$ and $f(I_{min})$ are the fitness of the worst and best individuals, respectively, and $\alpha \in [0, 1]$ controls the lower bound of the assigned fitness. The individuals with fitness values f' less than the stress factor are removed, and the empty slots are filled with a tournament selection between mutated variants of survived individuals. If no individual is killed, a percentage m of the population will be replaced by mutants, with a process called a *background extinction*.

This context-independent technique acts exclusively on the survival selection mechanism.

6.3. Crowded-comparison operator

Multi-objective optimization addresses all problems that give rise to a set of trade-off optimal solutions (known as *Pareto-optimal solutions*) [24]. A trivial example of a multi-objective problem with two objectives is reported in Figure 9. MOEAs strive to find as many Pareto-optimal points as possible, because any two solutions on the Pareto front represent a trade-off between the objectives: when a large groups of solutions is returned, users are in a better position to make an informed decision.

For the same reason, it is interesting for the user to obtain solutions that are well distributed along the Pareto front, in order to have a more complete picture of the problem. The *crowding* mechanism introduced in the MOEA *Non-Sorting Genetic Algorithm II* (NSGA-II) [28] is designed to tackle this problem, effectively enforcing diversity on the Pareto front through a *Crowded-Comparison Operator* (CCO).

The CCO guides the selection process in various steps of the algorithm, trying to achieve a uniformly spread-out Pareto-optimal front. Every individual I in the population possesses two attributes, a *non-domination rank* I_{rank} and a *crowding distance* $I_{distance}$. The non-domination rank basically identifies the front the individual belongs to, and represents the primary source of comparison. If two individuals have the same non-domination rank, they are then compared on their $I_{distance}$, so that solutions located in less crowded regions are preferred.

NSGA-II uses an estimate of the density of solutions surrounding individual I, by taking the nearest neighbors as vertices of a cuboid, and then calculating the average distance from the vertices along each of the objectives, see again Figure 9. This method only operates in the objectives' space, so it can be

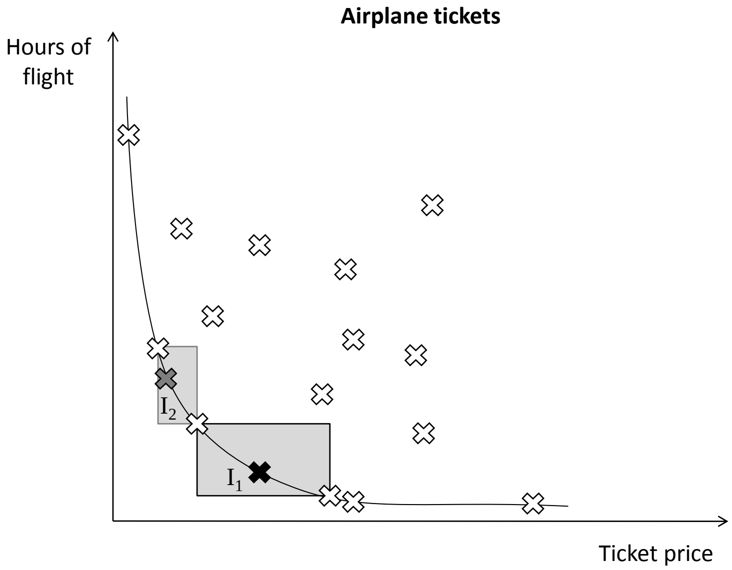

Figure 9: Sample multi-objective problem. Each solution, representing an airplane ticket for the same destination, is a trade-off in terms of price and hours of flight. Optimal, non-dominated solutions are on the line representing the Pareto front. Points in the top left of the curve represent cheap tickets with long flight times; points in bottom right represent expensive tickets with the shortest flights. The highlighted areas represent the cuboids used by NSGA-II to estimate the density of solutions surrounding individuals. The cuboid for individual I_1, in black, is clearly bigger than the cuboid of individual I_2, in gray; thus, individual I_1 will be preferred for selection.

applied independently from the genotype or phenotype representation of the individuals: there are even LGP tools that exploit this technique for multi-objective problems [64]. Furthermore, NSGA-III, the successor of NSGA-II, exploits the concept of ϵ-domination to adaptively discretize the Pareto-optimal front and find a better-distributed set of points [26, 27].

6.4. Hierarchical fair competition

Sub-populations are the focus of *Hierarchical Fair Competition* (HFC) [43]: in this approach each sub-population tries to contain individuals of similar fitness values, promoting the best ones to *upper* sub-populations and demoting the worst ones to *lower* sub-populations (see Figure 10). The technique proves to be quite effective to help new patterns emerge in the gene pool without being immediately overwhelmed by existing already-adapted individuals. HFC can be seen as an enhanced version of the island model (section 4.1) and it shares the assumptions that a multi-start scheme would yield different solutions, however it is based on how individuals behaves rather than where they have been generated.

The main drawback of the methodology is that it leaves several parameters for the user to define, such as the number of sub-populations, the size and the two thresholds for *acceptance* and *upward migration*, for each sub-population. There exist HFC models that self-adapt all parameters, but a good user-defined regulation usually returns better results. Relying on an idea similar to the islands, this technique influences both reproduction and survival, and it is context-dependent.

6.5. Vector evaluated genetic algorithm

Besides NSGA-II's CCO (section 6.3), other strategies are employed by MOEAs to promote diversity inside the population. The *Vector Evaluated Genetic Algorithm* (VEGA) [67], for example, evenly divides

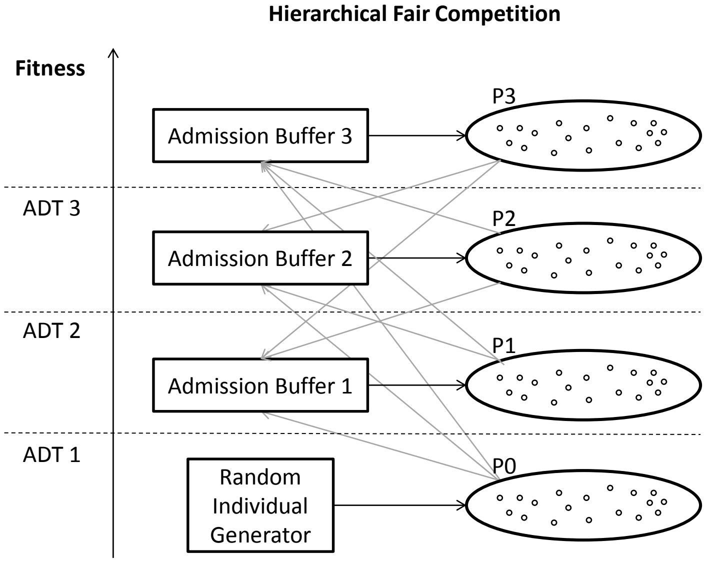

Figure 10: Hierarchical Fair Competition scheme. Depending on their fitness value, individuals are divided into sub-populations $P1 \ldots Pn$, synchronously or asynchronously through admission buffers, following user-defined *Admission Thresholds* (ADT). Fresh genetic material is added through a random individual generator.

the mating pool into a number of parts equal to the number of objectives: each part is filled with individuals selected on a different objective. A derived technique proposed in [39] uses a similar division of the mating pool, considering different trade-offs between objectives as a weighted sum. The weights are encoded in the genotype, and evolved to search for multiple solutions simultaneously. Diversity within weight combinations is promoted by phenotype-level fitness sharing.

6.6. *Strength Pareto*

This approach promotes diversity by using an external secondary population to store the non-dominated solutions [87]. The *strength* of an individual in the secondary population is proportional to the number of other individuals covered by it, while dominated individuals are assigned a fitness based on the strength of individuals that cover them. The secondary population is updated at every generation and pruned by clustering if the number of the non-dominated individuals exceeds a predefined size.

7. Promoting diversity: a hands-on approach

The amount of different methodologies proposed to cope with the lack of diversity can be disconcerting for someone approaching EC. This section presents a few rules of thumb that could help practitioners to assess what technique may be best suited for their particular cases, together with hints and tricks to implement

Table 2: Summary of all techniques analyzed in the paper in alphabetic order, classified following the proposed taxonomy. In the *main element* column, $\mathcal{L}$ indicates lineage, $\mathcal{G}$ genotype, and $\mathcal{P}$ phenotype. In the *selection* columns, "P" stands for "parent" and "S" for "survival".

Sec	Methodology name	Main element	Selection P	Selection S	Context dep.
4.4	Aging	$\mathcal{L}$	yes	yes	no
4.6	Allopatric selection	$\mathcal{L}$	no	yes	no
4.3	Cellular EAs	$\mathcal{L}$	yes	yes	yes
5.2	Clearing	$\mathcal{G}$	yes	yes	no
6.3	Crowded-comparison operator	$\mathcal{P}$	yes	no	no
5.7	Delta/pseudo entropy	$\mathcal{G}$	yes	no	no
4.5	Deterministic crowding	$\mathcal{L}$	no	yes	no
5.11	Diversifiers	$\mathcal{G}$	yes	yes	no
6.2	Extinction	$\mathcal{P}$	no	yes	no
5.10	Find only and complete undominated set	$\mathcal{G}$	no	yes	no
5.1	Fitness sharing	$\mathcal{G}$	yes	yes	no
4.7	Gender	$\mathcal{L}$	yes	no	yes
5.10	Genetic diversity evaluation method	$\mathcal{G}$	no	yes	no
6.4	Hierarchical fair competition	$\mathcal{P}$	yes	yes	yes
4.1	Island model	$\mathcal{L}$	yes	yes	yes
6.1	Random immigrants	$\mathcal{P}$	yes	yes	no
5.6	Reference points partitioning	$\mathcal{G}$	no	yes	no
5.4	Restricted tournament selection	$\mathcal{G}$	no	yes	no
4.2	Segregation	$\mathcal{L}$	yes	yes	yes
5.5	Sequential niching	$\mathcal{G}$	no	yes	no
5.3	Standard crowding	$\mathcal{G}$	no	yes	no
6.6	Strength pareto	$\mathcal{P}$	yes	no	no
5.9	Tarpeian method	$\mathcal{G}$	yes	yes	no
5.8	Two-level diversity selection	$\mathcal{G}$	yes	no	yes
6.5	Vector evaluated genetic algorithm	$\mathcal{P}$	yes	no	yes

it[8]. These spare indications are to be intended as a starting point for further research, and not as absolute rules applicable to every circumstance.

The first step before considering diversity promotion is to understand if the problem at hand requires the effort. It may be assumed that, if a practitioner is questioning how to promote diversity, the evolutionary optimizer already got stuck in suboptimal solutions several times. However, if repeated experiments yield very different results, but the optimizer is able to find at least few acceptable solutions, then the fitness landscape is probably very rough, and the task is probably going to be hard. Nevertheless, in that situation lack of diversity would not represent the most taxing problem: we would rather suggest to revise the encoding

---

[8] Differently, scholars willing to evaluate and compare on their own the listed methodologies could need to implement them starting from their descriptions, as source code is not generally available. A good framework that already implements islands and provides a convenient platform is *Open BEAGLE*, available under *GNU Lesser GPL* from `https://code.google.com/p/beagle/`. Moreover, the simplistic implementations and didactic videos showing the effects of few methodologies are available under Creative Commons *Attribution-NonCommercial-ShareAlike 4.0 International License* from `https://bitbucket.org/atonda/eatutorial/src/`.

and the fitness function. On the contrary, when repeated experiments yield very similar sub-optimal results, premature convergence may be the issue.

Indeed, it is not required to run the full-fledged evolutionary optimizer for performing this first check. A *random mutation hill climber* [55] is probably faster and equally effective to this end. Moreover, the hill climber could be easily built using the same mutation operators already present in the core of the original optimizer, by stripping down the evolutionary process. Unfortunately, for many real-world industrial problems the time required to run such experiments would make them effectively useless. Thus, in several practical applications, the best option is to include some methodologies for promoting diversity as a default. We recommend resorting to well established methodologies (section 7.1).

Even when dealing with quite complex representations where a mathematical analysis of the fitness landscape is impractical or too computationally expensive, it may be possible to obtain useful problem-specific knowledge that can be exploited inside future diversity-preservation mechanisms. In such cases practitioners could also try to prepare their own methodology for promoting diversity (section 7.2).

7.1. Out-of-the box methodologies

In absence of any domain-specific information on the problem, *Extinction* (section 6.2) is a methodology able to return sensible results with minimal effort. Additionally, it features few parameters to be tuned, namely, how to trigger an extinction, and the percentage of the population to be replaced. Several implementations of popular optimization algorithms make use of extinction or similar mechanisms to periodically remove a considerable part of the population.

Island models (section 4.1) represent another quick and simple, yet effective, way to promote diversity even when information about the target problem is scarce. This technique only needs few parameters to be set by the user, namely the size of each island and the modalities of migration, and can provide a first assessment of how adding diversity preservation can significantly improve the results. Additionally, it requires only minor modification to the original algorithm: for this reason, island models are often adopted even by commercial evolutionary software [68].

Both methodologies can be added to an existing algorithm with limited effort, although the latter could be slightly more complex to implement but easier to parallelize. To be fully effective, both rely on the crossover operator. After an *extinction*, good, fresh traits of new individuals may be merged with the genome of already good solutions, allowing to escape from the local optima. In *island models*, after migration the crossover operator could merge solutions exploiting different local optima, allowing to explore new regions of the search space.

If information about the problem is reliable, we would suggest adding *niching* to the optimizer. Techniques that use an implicit neighborhood, such as *restricted tournament selection* (section 5.4) or even *standard crowding* (section 5.3), can be easily adopted. Both techniques require to evaluate the distance between genotypes, but dos not rely on assumptions on the local optima distribution. However, if also a good estimation for the distance between local optima in the fitness landscape is available, we recommend niching techniques that operate with an explicit neighborhood, such as *fitness sharing* (section 5.1) or *clearing* (section 5.2), can be applied to the problem.

7.2. Hybrid methodologies

As a general consideration, tackling diversity at the level of genotype is simpler, but operating at the level of phenotype is likely to be more effective. Fortunately, in several problems, genotypes and phenotypes are highly correlated. Two simple tests could help a practitioner to detect this situation: count the collisions and check whether the locality principle is fulfilled.

Collisions happen whenever two or more individuals that are different at the level of genotype are given exactly the same fitness value. A large number of collisions could seriously impair the effectiveness of artificial evolution, because the principle of *differential survival* is a cornerstone of the whole process: different individuals should have different chances to survive, and this is not true if they share the same fitness value. In general, a high number of collisions is a sign that the fitness function should be redesigned and improved.

To check if the locality principle is fulfilled, we suggest to generate a set **P** of random individuals. Then, for $i = 1...n$ generate the sets $\mathbf{O_i}$ containing the offspring obtained applying exactly i mutation operators to each parent in P. Finally, examine the difference in the fitness values $\Delta(I_1, I_2)$ between individuals in **P** and individuals in $\mathbf{O_i}$ with respect to i. As a rule of thumb, the locality principle is satisfied if there is a correlation between i and $\Delta(I_1, I_2)$ with $I_1 \in \mathbf{P}, I_2 \in \mathbf{O_i}$

If it is possible to assign a *diversity bonus* to each individual, for instance, by measuring the average distance between itself and the whole population, or by measuring population diversity with and without it, then diversity can also be pursued as an explicit goal using multi-objective algorithms [21, 80]. The consequence of searching for a Pareto-front with some highly-fit and some highly-diverse solutions is to promote diversity during the evolutionary process.

Alternatively, it may be possible to promote diversity by tweaking selection probabilities. The *real* (i.e., the original) fitness could be scaled, or in any other way modified, according to a diversity measure. The main advantage of this technique is that it can be implemented with limited coding effort, because, except for the fitness calculation, the algorithm does not need to be modified. However, the amount of scaling needs to be carefully considered. It is also possible to tweak the selection by changing how individuals are compared, creating artificial *holes* in the fitness landscape. Literature reports several successful stories: in [31], such holes are created by taking into account the size of individuals, trying to favor smaller, less effective GP trees over bigger and more performing ones; in [59] the fitness of a certain proportion of the offspring is radically zeroed; while in [17] original fitness values are used with probability h, while with probability $1 - h$ an individual able to diversity of the population is preferred.

These techniques do not require the definition of a distance between all pairs of individuals, although such a metric could be easily used to implement them. Even basic information on the problem, that might not be directly exploited to measure the difference between two candidate solution, may be nevertheless capitalized on to introduce randomness in individual selection and indirectly prevent premature convergence. For example, the *tarpeian method* (section 5.9) was originally used to limit bloating in experiments with individuals of non-fixed size, but the possibility to create non-deterministic *fitness holes* in the fitness landscape should not be underestimated. If a metric other than fitness value is available to compare two individuals, it could be employed to promote diversity.

8. Conclusions

As the *divergence of character* is an essential element in natural evolution, the *lack of divergence of character* is an endemic problem in evolutionary optimization. This paper surveys notable methods to overcome this problem, promoting diversity. The algorithms are classified on the basis of the elements considered (lineage, phenotype or genotype), the type of selection influenced (reproduction and/or survival) and their dependency from the context.

Looking at the resulting compendium, it is easy to identify the recurring ideas that have been exploited through different domains: restrict the interaction of individuals to random sub-groups for mating (e.g. *gender*), survival (e.g. *niching*) or both (e.g. *islands*, *segregation*); strengthen the competition between solutions that are closely related (e.g. *allopatric selection*, *deterministic crowding*); artificially alter the fitness values of individuals, taking into account their contribution to population's diversity as well as their goodness with regards to the problem's objectives (e.g. *FOCUS*, *genetic diversity evaluation method*, *fitness holes*, *Tarpeian method*); and regularly add entirely new genetic material to the population (e.g. *random immigrants*, *diversifiers*, *hierachical fair competition*).

In fact, it must be remarked that it is often hard to quantify the improvement that a diversity-promoting technique can add to an evolutionary optimization process. While several extremely challenging test functions are available, modern EAs often feature a set of mechanisms aimed at increasing their efficiency, and separating the contribution of each to the final result can be quite complex. We feel that there is still a lack of benchmarks able to consistently evaluate diversity preservation in evolutionary optimization, and especially finding a test case that could be generalized to all different paradigms in EC could constitute an interesting and challenging research line.

Several well-known EAs with proved effectiveness include indirect mechanisms for promoting diversity. However, such techniques are so deeply embedded – or have so many side-effects – that they cannot be easily extrapolated from the original algorithm and analyzed separately. Paradigmatic examples are *generational approaches* [29] (or *comma strategies* in ES), the core of CMA-ES [40], *competitive co-evolution*, used for example in [69], or the effect of increasing the *crossover rate* in DE [16].

From a practitioner's perspective, it is interesting to notice how the great majority of techniques try to exploit information at genotype-level, the easiest to deal with; that several diversity-promoting techniques can be applied to the same algorithm at the same time, acting at different levels (lineage and genotype, for example); and that not necessarily methods that are more complex to implement return better results.

A considerable number of EAs have been adopted by practitioners of other domains as effective means to find reasonable solutions for problems that could not be tackled with classical optimization techniques. Looking at the most popular software, it is striking to notice how almost all provide some default value for the parameters, so that the users can try an approach *out of the box*, without having to tweak the population size or the activation probabilities for the genetic operators.

While diversity preservation is essential, the main challenge for scholars is devising general methodologies that could be applied seamlessly, trying to limit the number of parameters the user has to set, or providing a few default values that work in most cases.

Acknowledgments

The authors would like to thank Evelyne Lutton for her invaluable suggestions and insightful comments.

References

[1] Affenzeller, M., 2001. A new approach to evolutionary computation: segregative genetic algorithms (SEGA). In: Connectionist Models of Neurons, Learning Processes, and Artificial Intelligence. Springer, pp. 594–601.

[2] Alba, E., Dorronsoro, B., 2008. Cellular genetic algorithms. Vol. 42. Springer.

[3] Allenson, R., 1992. Genetic algorithms with gender for multi-function optimisation. Tech. Rep. EPCC-SS92-01, Edinburgh Parallel Computing Centre, Edinburgh, Scotland,.

[4] Badran, K., Rockett, P. I., 2007. The roles of diversity preservation and mutation in preventing population collapse in multiobjective genetic programming. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). ACM, pp. 1551–1558.

[5] Bailey, S. F., Dettman, J. R., Rainey, P. B., Kassen, R., Jul 2013. Competition both drives and impedes diversification in a model adaptive radiation. Proceedings of the Royal Society B: Biological Sciences 280 (1766), 20131253–20131253.

[6] Banzhaf, W., McMullin, B., 2012. Artificial Life. Springer.

[7] Beasley, D., Bull, D. R., Martin, R. R., 1993. A sequential niche technique for multimodal function optimization. Evolutionary computation 1 (2), 101–125.

[8] Ben Hamida, S., Schoenauer, M., 2002. Aschea: new results using adaptive segregational constraint handling. In: Evolutionary Computation, 2002. CEC'02. Proceedings of the 2002 Congress on. Vol. 1. IEEE, pp. 884–889.

[9] Beyer, H.-G., Schwefel, H.-P., 2002. Evolution Strategies – A comprehensive introduction. Natural computing 1 (1), 3–52.

[10] Brameier, M., Banzhaf, W., 2001. A comparison of linear genetic programming and neural networks in medical data mining. Evolutionary Computation, IEEE Transactions on 5 (1), 17–26.

[11] Brameier, M., Banzhaf, W., 2002. Explicit control of diversity and effective variation distance in linear genetic programming. Springer.

[12] Brameier, M. F., Banzhaf, W., 2007. Linear genetic programming. Springer.

[13] Burke, E. K., Gustafson, S., Kendall, G., 2004. Diversity in genetic programming: An analysis of measures and correlation with fitness. Evolutionary Computation, IEEE Transactions on 8 (1), 47–62.

[14] Castrogiovanni, M., Nicosia, G., Rascunà, R., 2007. Experimental analysis of the aging operator for static and dynamic optimisation problems. In: Knowledge-Based Intelligent Information and Engineering Systems. Springer, pp. 804–811.

[15] Chen, G., Low, C. P., Yang, Z., 2009. Preserving and exploiting genetic diversity in evolutionary programming algorithms. Evolutionary Computation, IEEE Transactions on 13 (3), 661–673.

[16] Coelho, L. d. S., Bora, T. C., Mariani, V. C., 2014. Differential evolution based on truncated lévy-type flights and population diversity measure to solve economic load dispatch problems. International Journal of Electrical Power & Energy Systems 57, 178–188.

[17] Corno, F., Sánchez, E., Squillero, G., 2005. Evolving assembly programs: how games help microprocessor validation. Evolutionary Computation, IEEE Transactions on 9 (6), 695–706.

[18] Corriveau, G., Guilbault, R., Tahan, A., Sabourin, R., 2012. Review and study of genotypic diversity measures for real-coded representations. IEEE transactions on evolutionary computation 16 (5), 695–710.

[19] Darwin, C., 1859. On the Origin of the Species by Means of Natural Selection: Or, The Preservation of Favoured Races in the Struggle for Life. John Murray.
URL `http://books.google.it/books?id=jTZbAAAAQAAJ`

[20] Dawkins, R., 1999. The extended phenotype: the long reach of the gene. Popular Science. Oxford University Press.

[21] De Jong, E., Watson, R., Pollack, J., July 2001. Reducing bloat and promoting diversity using multi-objective methods. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). L. Spector et al. Eds.

[22] De Jong, K. A., 1975. Analysis of the behavior of a class of genetic adaptive systems. Ph.D. thesis, University of Michigan, Ann Arbor.

[23] Deb, K., 1989. Genetic algorithms in multimodal function optimization. Ph.D. thesis, Clearinghouse for Genetic Algorithms, Department of Engineering Mechanics, University of Alabama.

[24] Deb, K., 2005. Multi-objective optimization. In: Burke, E. K., Kendall, G. (Eds.), Search Methodologies. Springer US, pp. 273–316.

[25] Deb, K., Goldberg, D. E., 1989. An investigation of niche and species formation in genetic function optimization. In: Proceedings of the 3rd International Conference on Genetic Algorithms. Morgan Kaufmann Publishers Inc., pp. 42–50.

[26] Deb, K., Jain, H., 2013. An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part I: Solving problems with box constraints. IEEE Transactions on Evolutionary ComputationEarly Access.

[27] Deb, K., Jain, H., 2013. An evolutionary many-objective optimization algorithm using reference-point based non-dominated sorting approach, Part II: Handling constraints and extending to an adaptive approach. IEEE Transactions on Evolutionary ComputationEarly Access.

[28] Deb, K., Pratap, A., Agarwal, S., Meyarivan, T., 2002. A fast and elitist multiobjective genetic algorithm: NSGA-II. Evolutionary Computation, IEEE Transactions on 6 (2), 182–197.

[29] Eiben, A. E., Smith, J. E., 2010. Introduction to evolutionary computing. Vol. 2. Springer Berlin.

[30] Ekárt, A., Németh, S. Z., 2000. A metric for genetic programs and fitness sharing. In: Genetic Programming. Springer, pp. 259–270.

[31] Ekárt, A., Nemeth, S. Z., 2001. Selection based on the pareto nondomination criterion for controlling code growth in genetic programming. Genetic Programming and Evolvable Machines 2 (1), 61–73.

[32] Fogel, D. B., 1998. Evolutionary computation: the fossil record. Wiley-IEEE Press.

[33] Fogel, L. J., 1962. Autonomous automata. Industrial Research 4, 14–19.

[34] Gaudesi, M., Squillero, G., Tonda, A., 2013. An efficient distance metric for linear genetic programming. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). ACM, pp. 925–932.

[35] Goh, K. S., Lim, A., Rodrigues, B., 2003. Sexual selection for genetic algorithms. Artificial Intelligence Review 19 (2), 123–152.

[36] Goldberg, D. E., Richardson, J., 1987. Genetic algorithms with sharing for multimodal function optimization. In: Proceedings of the 2nd International Conference on Genetic Algorithms. pp. 41–49.

[37] Greewood, G., Fogel, G. B., Ciobanu, M., 1999. Emphasizing extinction in evolutionary programming. In: Evolutionary Computation, 1999. CEC 99. Proceedings of the 1999 Congress on. Vol. 1. IEEE.

[38] Grefenstette, J. J., 1992. Genetic algorithms for changing environments. In: Parallel Problem Solving from Nature (PPSN). Vol. 2. pp. 137–144.

[39] Hajela, P., Lin, C.-Y., 1992. Genetic search strategies in multicriterion optimal design. Structural optimization 4 (2), 99–107.

[40] Hansen, N., 2006. The cma evolution strategy: a comparing review. In: Towards a new evolutionary computation. Springer, pp. 75–102.

[41] Harik, G. R., 1995. Finding multimodal solutions using restricted tournament selection. In: Proceedings of the Sixth International Conference on Genetic Algorithms. San Francisco, CA, pp. 24–31.

[42] Holland, J. H., 1975. Adaptation in Natural and Artificial Systems. University of Michigan Press.

[43] Hu, J., Goodman, E., Seo, K., Fan, Z., Rosenberg, R., Jan. 2005. The hierarchical fair competition (HFC) framework for sustainable evolutionary algorithms. Evolutionary computation 13 (2), 241–77.

[44] Keijzer, M., 1996. Efficiently representing populations in genetic programming. In: Advances in genetic programming. MIT Press, pp. 259–278.

[45] Keller, R., Banzhaf, W., 1995. Explicit maintenance of genetic diversity on genospaces. Tech. rep., University of Dortmund.

[46] Koza, J. R., 1992. Genetic Programming: vol. 1, On the programming of computers by means of natural selection. Vol. 1. MIT press.

[47] Langdon, W. B., 1998. Genetic programming and data structures: genetic programming+ data structures= automatic programming! Vol. 1. Springer.

[48] Levenshtein, V. I., 1966. Binary codes capable of correcting deletions, insertions and reversals. In: Soviet physics doklady. Vol. 10. p. 707.

[49] Lis, J., Eiben, A., 1997. A multi-sexual genetic algorithm for multiobjective optimization. In: Evolutionary Computation, 1997., IEEE International Conference on. pp. 59–64.

[50] Mahfoud, S. W., 1995. Niching methods for genetic algorithms. Ph.D. thesis, University of Illinois, Urbana-Champaign.

[51] Mayr, E., 1992. Darwin's principle of divergence. Journal of the History of Biology 25 (3), 343–359.

[52] McKay, R. I., 2000. Fitness sharing in genetic programming. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). pp. 435–442.

[53] McPhee, N. F., Hopper, N. J., 1999. Analysis of genetic diversity through population history. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). Vol. 2. Citeseer, pp. 1112–1120.

[54] Merelo-Guervós, J. J., García-Sánchez, P., 2015. Modeling browser-based distributed evolutionary computation systems. arXiv preprint arXiv:1503.06424.

[55] Mitchell, M., Holland, J. H., Forrest, S., et al., 1993. When will a genetic algorithm outperform hill climbing? In: NIPS. pp. 51–58.

[56] Ooka, R., Komamura, K., 2009. Optimal design method for building energy systems using genetic algorithms. Building and Environment 44 (7), 1538–1544.

[57] O'Reilly, U.-M., 1997. Using a distance metric on genetic programs to understand genetic operators. In: Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation., 1997 IEEE International Conference on. Vol. 5. IEEE, pp. 4092–4097.

[58] Pétrowski, A., 1996. A clearing procedure as a niching method for genetic algorithms. In: Evolutionary Computation, 1996., Proceedings of IEEE International Conference on. IEEE, pp. 798–803.

[59] Poli, R., 2003. A simple but theoretically-motivated method to control bloat in genetic programming. In: Genetic Programming. Springer, pp. 204–217.

[60] Poli, R., Kennedy, J., Blackwell, T., 2007. Particle swarm optimization. Swarm intelligence 1 (1), 33–57.

[61] Rejeb, J., AbuElhaij, M., 2000. New gender genetic algorithm for solving graph partitioning problems. In: Circuits and Systems, 2000. Proceedings of the 43rd IEEE Midwest Symposium on. Vol. 1. IEEE, pp. 444–446.

[62] Robertson, G. G., 1987. Parallel implementation of genetic algorithms in a classifier system. In: Proceedings of the Second International Conference on Genetic Algorithms on Genetic algorithms and their application. L. Erlbaum Associates Inc., pp. 140–147.

[63] Rosca, J. P., 1995. Entropy-driven adaptive representation. In: Proceedings of the workshop on genetic programming: From theory to real-world applications. Vol. 9. Citeseer, pp. 23–32.

[64] Sanchez, E., Schillaci, M., Squillero, G., 2011. Evolutionary Optimization: the μGP toolkit. Springer.

[65] Sánchez-Velazco, J., Bullinaria, J., 2003. Sexual selection with competitive/co-operative operators for genetic algorithms. In: Neural Networks and Computational Intelligence.

[66] Sareni, B., Krahenbuhl, L., 1998. Fitness sharing and niching methods revisited. Evolutionary Computation, IEEE Transactions on 2 (3), 97–106.

[67] Schaffer, J. D., 1985. Multiple objective optimization with vector evaluated genetic algorithms. In: Proceedings of the 1st international Conference on Genetic Algorithms. L. Erlbaum Associates Inc., pp. 93–100.

[68] Schmidt, M., Lipson, H., 2009. Distilling free-form natural laws from experimental data. science 324 (5923), 81–85.

[69] Schmidt, M. D., Lipson, H., 2008. Coevolution of fitness predictors. Evolutionary Computation, IEEE Transactions on 12 (6), 736–749.

[70] Schönfisch, B., de Roos, A., 1998. Synchronous and asynchronous updating in cellular automata. In: Cellular Automata: Research Towards Industry. Springer, pp. 42–46.

[71] Shannon, C. E., 1948. A mathematical theory of communication. Bell System Technical Journal 27 (3), 379?423.

[72] Shimodaira, H., 1997. DCGA: A diversity control oriented genetic algorithm. In: Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on. IEEE, pp. 367–374.

[73] Shimodaira, H., 2001. A diversity-control-oriented genetic algorithm (DCGA): Performance in function optimization. In: Evolutionary Computation, 2001. Proceedings of the 2001 Congress on. Vol. 1. IEEE, pp. 44–51.

[74] Spencer, H., 1864. The Principles of Biology. No. v. 1 in A system of synthetic philosophy. vol. II-[III]. Williams and Norgate.

[75] Squillero, G., Tonda, A. P., 2008. A novel methodology for diversity preservation in evolutionary algorithms. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). ACM, pp. 2223–2226.

[76] Storn, R., Price, K., 1997. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of global optimization 11 (4), 341–359.

[77] Svanbäck, R., Bolnick, D. I., 2007. Intraspecific competition drives increased resource use diversity within a natural population. Proceedings of the Royal Society B: Biological Sciences 274 (1611), 839–844.

[78] Tackett, W. A., 1994. Recombination, selection, and the genetic construction of computer programs. Ph.D. thesis, University of Southern California.

[79] Tinós, R., Yang, S., 2007. A self-organizing random immigrants genetic algorithm for dynamic optimization problems. Genetic Programming and Evolvable Machines 8 (3), 255–286.

[80] Toffolo, A., Benini, E., 2003. Genetic diversity as an objective in multi-objective evolutionary algorithms. Evolutionary Computation 11 (2), 151–167.

[81] Tomassini, M., 2005. Spatially structured evolutionary algorithms: Artificial evolution in space and time. Natural Computing.

[82] Tonda, A., Lutton, E., Squillero, G., 2012. A benchmark for cooperative coevolution. Memetic Computing 4 (4), 263–277.

[83] Turing, A. M., 1950. Computing machinery and intelligence. Mind, 433–460.

[84] Ursem, R. K., 2002. Diversity-guided evolutionary algorithms. In: Parallel Problem Solving from Nature (PPSN). Springer, pp. 462–471.

[85] Whitley, D., Rana, S., Heckendorn, R. B., 1999. The island model genetic algorithm: On separability, population size and convergence. Journal of Computing and Information Technology 7, 33–48.

[86] Wineberg, M., Oppacher, F., 2003. The underlying similarity of diversity measures used in evolutionary computation. In: Proceedings of the Genetic and Evolutionary Computation Conference (GECCO). Springer, pp. 1493–1504.

[87] Zitzler, E., Thiele, L., 1999. Multiobjective evolutionary algorithms: A comparative case study and the strength pareto approach. Evolutionary Computation, IEEE Transactions on 3 (4), 257–271.