



**NOVA**

**IMS**

Information  
Management  
School

# MMAA

---

**Mestrado em Métodos Analíticos Avançados**

Master Program in Advanced Analytics

## NON-GEOMETRIC PULSE

An adaptive geometricity approach for Genetic Algorithms

**José Pedro Ferreira**

Dissertation presented as partial requirement for obtaining  
the Master's degree in Data Science and Advanced Analytics

**NOVA Information Management School**  
**Instituto Superior de Estatística e Gestão da Informação**  
Universidade Nova de Lisboa



**NOVA Information Management School**  
**Instituto Superior de Estatística e Gestão de Informação**  
Universidade NOVA de Lisboa

## **NON-GEOMETRIC PULSE**

by

José Pedro Ferreira

Dissertation presented as partial requirement for obtaining the Master's  
degree in Data Science and Advanced Analytics

**Advisor:** Mauro Castelli

**Co-advisor:** Luca Manzoni

⟨August⟩, ⟨2022⟩



## **Non-Geometric Pulse**

Copyright © José Pedro Ferreira, NOVA Information Management School, NOVA University Lisbon.

The NOVA Information Management School and the NOVA University Lisbon have the right, perpetual and without geographical boundaries, to file and publish this dissertation through printed copies reproduced on paper or on digital form, or by any other means known or that may be invented, and to disseminate through scientific repositories and admit its copying and distribution for non-commercial, educational or research purposes, as long as credit is given to the author and editor.



## ACKNOWLEDGEMENTS

Taking one's guidance originated determination. I would like to thank my pleasant supervisors, Prof. Mauro Castelli and Prof. Luca Manzoni for their crucial work in correctly orienting me towards a successful delivery.





*“The a priori is greatly neglected.” (Kurt Gödel)*



# ABSTRACT

Evolutionary algorithms (EAs) are a family of algorithms inspired by the Darwinian theory of evolution. Mathematics, particularly geometry and topology, allows for the possibility of developing a general geometrical framework and consequently generating deeper insights that may be shared among the different EAs.

Genetic Algorithm (GA), inspired by Darwin's theory of natural selection, is a popular algorithm among EAs. This is a population-based, fitness-oriented algorithm that performs a convex heuristic search to optimize a plethora of problems.

Common limitations of GA as well as other EAs have geometrical origins like premature convergence, where the final population's convex-hull might not include the best solution, called Global Optima. Population diversity maintenance is a key idea that tries to tackle this problem but is often performed through geometrical methods that constantly diminish the search space's area.

In this work, a self-adaptive geometricity approach will be presented. In particular, the non-geometric crossover is strategically employed in a symbiotic relation with geometric crossover, maintaining diversity in a logical way from a geometric/topological grammar standpoint.

A comparison with well-known diversity maintenance methods is provided, using common benchmarks that serve as general testing ground for the considered techniques.

**Keywords:** Convex Search, Evolutionary Algorithms, Genetic Algorithms, Geometric semantic operators, Diversity maintenance



## RESUMO

Algoritmos evolutivos (AEs) são uma família de algoritmos inspirados na teoria da evolução de Darwin. Matemática, especificamente os ramos de geometria e topologia, permitiu o desenvolvimento de uma framework geométrica que consequentemente é capaz de gerar percepções e interpretações detalhadas que são comuns aos diferentes AEs.

Algoritmo genético (AG), inspirado na teoria de seleção natural avançada por Darwin, é um algoritmo popular entre os AEs. Através da criação de uma população e reconhecimento do fitness dos seus indivíduos, este algoritmo efetua uma procura heurística convexa de modo a otimizar inúmeros problemas.

As limitações comuns ao AG, também presentes em outros AEs, têm origens geométricas como o problema de convergência prematura onde o invólucro convexo da população final pode não incluir a melhor solução possível, denominada de Ótimo Global. A manutenção da diversidade da população é um fator chave que tenta corrigir este problema, apesar da sua execução ocorrer maioritariamente através do uso de métodos geométricos que diminuem continuamente o espaço de procura.

Neste trabalho será apresentada uma abordagem de geometricidade auto-adaptativa. A recombinação não geométrica é utilizada estrategicamente numa relação simbiótica para com a recombinação geométrica, mantendo logicamente a diversidade da população dentro do âmbito gramatical geométrico/topológico.

É apresentada uma comparação com renomadas metodologias prévias referentes ao problema de manutenção de diversidade, utilizando desafios comuns como banco de ensaio para o teste das técnicas consideradas.

**Palavras-chave:** Procura Convexa, Algoritmos Evolutivos, Algoritmos Genéticos, Operadores semânticos geométricos, Manutenção de diversidade



# CONTENTS

<b>List of Figures</b>	<b>xvii</b>
<b>List of Tables</b>	<b>xix</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Motivations . . . . .	2
1.2 Research Questions . . . . .	3
<b>2 Methods</b>	<b>5</b>
2.1 Structure of genetic algorithms . . . . .	5
2.2 Search operators geometricity . . . . .	7
2.2.1 Crossover . . . . .	8
2.2.2 Mutation . . . . .	11
2.2.3 Convex combination, convex hull and convex search . . . . .	12
2.3 Grammar taxonomy: Biology vs Mathematics . . . . .	14
2.4 Adaptive framework and non-geometric operator placement . . . . .	15
2.4.1 Constant diversity maintenance . . . . .	15
2.4.2 Dynamic diversity maintenance . . . . .	16
2.4.3 Non-geometric pulse - dynamic geometricity rationale . . . . .	18
<b>3 Experiment</b>	<b>21</b>
3.1 Algorithm collection . . . . .	21
3.1.1 DCGA1 and DCGA2 distinctions . . . . .	21
3.1.2 P, P' and SA distinctions . . . . .	22
3.1.3 Interpretation of findings . . . . .	23
3.2 Benchmark functions . . . . .	24
3.3 Benchmark results . . . . .	26
3.4 Results overview . . . . .	30
<b>4 Conclusions and future work</b>	<b>33</b>

## CONTENTS

---

4.1	Conclusions . . . . .	33
4.2	Future work . . . . .	34
	<b>Bibliography</b>	<b>35</b>
	<b>Annexes</b>	
I	<b>General ABF boxplots</b>	<b>37</b>



## LIST OF FIGURES

2.1	Genetic Algorithm course of action. . . . .	5
2.2	Distance visualization of 2-dimensional different space types. . . . .	8
2.3	2-dimentional spatial visualization of both crossovers in action. . . . .	10
2.4	Parents representation recombination sequence of steps in both crossovers.	10
2.5	Bitflip mutation outcome $P'_1$ spatial view. . . . .	11
2.6	Convex set on the left and two non-convex sets on the right. . . . .	12
2.7	Convex hull reduction through a generation. Triangles are the offspring created by their parents, the dots. . . . .	13
2.8	Convex hull expansion through a generation with adaptive geometricity. Stars are the offspring created by their parents, the triangles. . . . .	19
3.1	DCGA1 portrayed as the blue curve, DCGA2 as the black curve. . . . .	22
3.2	ABF comparison over each of the 30 functions with Dimension = 10, calcu- lated by averaging each generation of 30 different independent runs. . . .	26
3.3	P-value results of the Mann–Whitney U test with the Bonferroni correction over each of the 30 functions with Dimension = 10. . . . .	27
3.4	ABF comparison over each of the 30 functions with Dimension = 30, calcu- lated by averaging each generation of 30 different independent runs. . . .	28
3.5	P-value results of the Mann–Whitney U test with the Bonferroni correction over each of the 30 functions with Dimension = 30. . . . .	29
I.1	Boxplots of the ABFs over each of the 30 functions with Dimension = 10.	37
I.2	Boxplots of the ABFs over each of the 30 functions with Dimension = 30.	38



## LIST OF TABLES

3.1	Definitions and optimum values of the CEC 2017 benchmark functions. . .	24
3.2	General testing parameters for all algorithms. . . . .	25



# INTRODUCTION

Intelligence, in the context of computer science, can be defined as the ability to deal with computational limitations by avoiding combinatorial explosions.

Time is an obstacle that humans and machines face when tackling any problem. To solve challenges and achieve solutions, intelligence is used to figure out a way of not exploring the entire search space, that could be, in some cases, infinite.

Two fundamental variables in problem solving are: *problem formulation*, the skill that reduces complexity by wisely designing one's problem; *relevance realization*, the ability to redirect attention into valuable directions [1]. Primarily, by creating algorithms, humans try to transfer their relevance realization ability to machines with the intention of scaling it as complexity emerges.

Evolutionary computation, a subfield of artificial intelligence, is inspired by biological evolution. It leverages computing power to model global optimization strategies that are present in nature and can be applied to almost every other field.

Genetic algorithms, a prime example of evolutionary computation, are metaheuristic<sup>1</sup> algorithms inspired by Darwinian evolution [2], which is defined as "descent with modification", meaning that species change over time/generations to fulfil their environmental challenges. This definition is also a valid abstraction for many existing Artificial Intelligence (AI) methodologies.

GAs are stochastic optimisers, minimizing or maximizing an objective function, possessing effective relevance realization abilities while going through the underlying search space. This space can be better understood if one abandons the popular biology framework and proceeds to adopt a mathematical one, creating a geometrical and topological view of the overall evolutionary process.

Language independently, when tackling any problem, there are a number of input arguments or independent variables that, when combined, form a set of cartesian coordinates that serve as a candidate solution. From this, one can infer the problem's search space dimensionality. Each of these arguments can be bounded by a certain domain or

---

<sup>1</sup>One could think of a metaheuristic algorithmic framework as being high-level and problem independent.

not, representing the length of each dimension. If the arguments are bounded, we can imagine our search space as a hypercube with a volume equal to the number of possible solutions to the problem. These solutions can be described as points in a geometric space, within a dynamical system expressed as the whole Genetic Algorithm process, its steps, that will change the state/move these points in space as time (generations) goes through.

In very simplistic terms, depending on the effect they produce in the underlying space (also called fitness landscape), there are two main categories of search operators present in these algorithms, while acting on a point/solution:

- Geometric operators, that result in a convex-search.
- Non-geometric operators, that result in a non-convex search.

Convex search will contract the volume of the hypercube that represents the whole observable fitness landscape and non-convex search has a non-zero probability of expanding it. To converge to a solution, we need convex search to be dominant, and this might be a reason for the overlooking of non-geometric crossover.

This work revolves around the possibility of including non-geometric operators, specifically crossover, in a self-adaptive<sup>2</sup> diversity maintenance framework that is able to compete not only with the default implementation of GA, but also with other popular self-adaptive or even static implementations. If correctly applied, non-geometric crossover can foster diversity in an optimized and rational manner.

## 1.1 Motivations

Genetic Algorithms revolve around the creation of a general process with the capability to deal with different complex problems. These algorithms are extremely elegant, but their full potential has not yet been achieved due to the computational power of traditional machines that might be tackled with future quantum computing developments. In the meantime, more optimization is the call to action.

General optimization is often semantically blocked by the No Free Lunch Theorem, which states that search performance is conserved<sup>3</sup>[3]. The theorem teaches us that there is no master algorithm and therefore, no short cut to global performance superiority. Clearly, this is not enough reason to stop looking for new ideas, especially for problems that might have an underlying structure that can be more efficiently exploited than uniformly distributed spaces.

Usually GAs work with the same search operators throughout the whole optimization process but this might not always be profitable in our search. Therefore, creating a

---

<sup>2</sup>Relative to the underlying landscape topology.

<sup>3</sup>Conservation law is used as an analogy applied to algorithms in order to predict symmetry or equilibrium in their results, meaning that any two algorithms should be equivalent when measured across the infinite amount of problems.

dynamical framework that responds to the search space needs with different operators might help to further augment their potential for a successful search.

When faced with infinity, one might argue that the main problem at hand is the impossibility of knowing the direction towards the solution. When this takes place, one often forgets that there might be additional information that can create a higher probability of positive outcome when rolling the dice. An example of this type of information is simply knowing where *not* to go.

To optimize is essentially to create order. Order often comes from boundaries and these boundaries are abstractions for where not to go. The objective of adapting the geometricity of the search operators is to leverage mathematical hints that point away from a certain direction<sup>4</sup>. This translates into the continuous change of the probability of either geometric or non-geometric crossover being pursued in different phases of the search process characterized by different space topology conditions.

This dynamical geometricity creates a framework of diversity maintenance that evolves through the generations and creates a path for non-geometric methods to be implemented as effective diversity enhancers.

When complexity arises, every geometric good practice creates an edge. Like in a heuristic process, the more educated guesses we create, the higher the probability of achieving a new best candidate solution.

## 1.2 Research Questions

These are the following questions this research aims to answer:

1. Why should we frame our Genetic Algorithm's grammar, meaning the overall structure of this algorithm, in a geometrical manner?
2. What are the prepositions that make non-geometric crossover an efficient catalyst for diversity maintenance?
3. How can we introduce non-convexity without ruining the overall convex process, i.e. turning it into a random search?
4. How does this framework of adaptive geometricity perform against previous work?

---

<sup>4</sup>Inwardly or outwardly.





## METHODS

### 2.1 Structure of genetic algorithms

Genetic algorithms follow a set of ordered operations that act upon the individuals of a randomly initialized population (Figure 2.1).

According to Darwinian evolution, within a population, the genetic information of those who successfully mate is inherited by their offsprings. Inherited along are mutations in their genome that create minor changes to the original genetic information. After this process, the offsprings will (hopefully) hold increased chances of surviving within their environment due to their updated information, resisting its pressure.

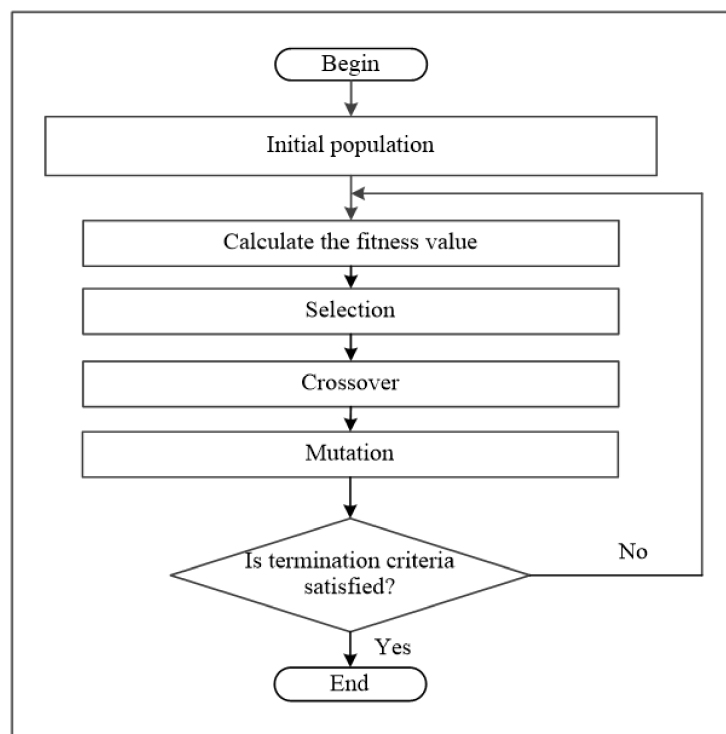


Figure 2.1: Genetic Algorithm course of action.

The selection for mating follows the main concepts of natural selection, therefore the chance of an individual spreading its genetic material is correlated with its fitness.

Examples of popular selection methods are:

- Tournament selection [4]
- Roulette Wheel [5]
- Rank selection [6]

These methods do not transform the state of our population, meaning they are not moving our points in space, yet are choosing pairs that have a certain distance between each other and will be used as arguments in the next phase, called recombination or crossover.

In this recombination phase, the previously selected pair of solutions will combine their genomes and generate new offspring. It is possible to approach this step with different mixing techniques that end up being part of two broad groups of operators: geometric or non-geometric. These operators will be discussed in more detail in later sections of this thesis. Usually, there are high rates of crossover as the point of this process is to evolve our solutions and not being stuck with old information.

Mutation will be the last step of the new individual's formation, where one or more of its genes, part of its genotypical representation, will be randomly altered, slightly<sup>1</sup> contributing to maintaining diversity. The rate of mutation is often kept low during the implementation of this algorithm. If a certain technique of mutation impacted a high percentage of the solution's representation while also having a high rate of incidence, it would be easy to understand that this would undermine the crossover's purpose while also weighting on the scale of randomness of this algorithm [7].

Crossover and mutation are the steps moving our solutions in space, with starting points previously defined by the selection step.

After arriving at a new creation, if one wants to describe an individual solution, one can characterize it genotypically, by studying its raw genetic information, or phenotypically, by studying its fitness, meaning the general ability to perform the problem in hand.

The merging of these two dimensions together establishes a *fitness landscape* - a landscape in which all the possibilities within our genotypical domain and consequent fitness outcomes can be envisioned.

Mathematically speaking, this space<sup>2</sup> can be thought as a hypercube with its volume being equal to the product of all its domain bounded dimensions. The volume of our solution space represents the set of all possible individual's genetic combinations solved for our objective, the problem's fitness function.

---

<sup>1</sup>Because mutation methods, besides their low rates, do not necessarily impact a big percentage of the solution's representation.

<sup>2</sup>This space is often bounded by the dimension of the individual's genetic material, its features represented in its DNA. Each of these dimensions has its own domain and sometimes these domains can be infinite, meaning our search space can also consequently be infinite.

Without any assumptions on a landscape, one would need to go through every point that makes part of this volume to reach the conclusion of achieving the best possible solution, the global optima, when solving any problem.

It is important to understand and develop the idea of a topology associated with a problem. Using an algorithm to solve a problem is essentially applying a strategy for searching that topology in an (hopefully) optimized fashion - this reduces the number of steps mentioned in the previous paragraph<sup>3</sup>. This means that the operators that are part of this strategy must be deeply understood from a geometrical perspective, perfectly interpreting how they are transforming any point in the underlying space.

Finally, if the termination criteria is satisfied, be it a fitness threshold that is reached, an upper limit to the number of generations or an a priori defined maximum number of fitness evaluations (MaxFES), which is commonly used in evolutionary algorithm competitions, the process ends and we will hopefully be in possession of an acceptable solution, if not the global optima.

## 2.2 Search operators geometricity

To properly understand why a given operator is either geometric or not, it is imperative to explore some key ideas developed in the geometric framework that unified various evolutionary algorithms. This framework explored the depths of the space where these operators act as well as their inner workings with a mathematical perspective [8]. The parents that will perform crossover are two points in space and their cartesian coordinates are a representation of their genome. These two points hold a distance between them that is often referred as their space metric:

- Hamming distance can be referred to when our information is coded as symbol strings, often binary strings, meaning each position of our genes would either be 0 or 1 (Figure 2.2(a)).

$$d(p, q) = \# \{i : p_i \neq q_i, i = 1, \dots, n\}$$

- Euclidian distance can be referred to in continuous real-coded representations (Figure 2.2(b)).

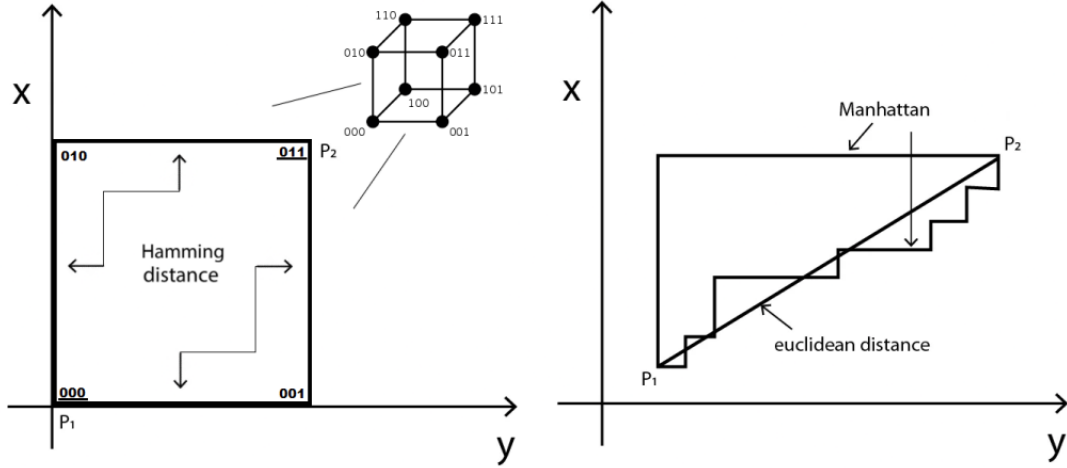
$$d(p, q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2}$$

- Manhattan distance, also called Taxicab or City Block distance, can be referred to in discrete real-coded representations (Figure 2.2(b)).

$$d(p, q) = \sum_{i=1}^n |p_i - q_i|$$

---

<sup>3</sup>It would still be necessary to go through space's entirety in order to claim a global optima in unsolved problems, but one is augmenting the chances of landing on that solution without its prior knowledge.



(a) Hamming distance through the lens of one of the faces of our 3-dimensional hamming cube. (b) Euclidean and Manhattan distances as well as their spatial differences.

Figure 2.2: Distance visualization of 2-dimensional different space types.

If a binary string of length  $n$  is viewed as a vector in  $\mathbb{R}^n$ , that information forms the vertices of the hypercube and the Hamming distance is equivalent to the Manhattan distance.

One's problem formulation abilities creates the possibility for some problems to be expressed with different distance metrics, if preferred. For example, Euclidian spaces can be transformed into Hamming spaces by converting our information into binary strings that represent the previous continuous real numbers.

So far, the following topics have been covered: 1) the different types of mathematical spaces, 2) correspondent distances and how malleable they can be. The focus now will be to geometrically understand the search operators' transformations when applied in our algorithms.

### 2.2.1 Crossover

Let  $S$  be the space of all possible solution representations and the image set  $\text{Im}[OP]$  of a genetic operator  $OP$  be the set of all possible offspring produced by  $OP$  with non-zero probability. For a recombination technique to be considered geometric or not, it must abide by the following definitions [8]:

- geometric crossover class  $\mathcal{G}$ : a recombination  $OP$  belongs to this class if there exists at least a distance  $d$  under which such a recombination is geometric:  

$$OP \in \mathcal{G} \iff \exists d : \forall p_1, p_2 \in S : \text{Im}[OP(p_1, p_2)] \subseteq [p_1, p_2]_d$$
- non-geometric crossover class  $\overline{\mathcal{G}}$ : a recombination  $OP$  belongs to  $\overline{\mathcal{G}}$  if there is no distance  $d$  under which such a recombination is geometric:  

$$OP \in \overline{\mathcal{G}} \iff \forall d : \exists p_1, p_2 \in S : \text{Im}[OP(p_1, p_2)] \setminus [p_1, p_2]_d \neq \emptyset$$

The fundamental metric axiom of geometric crossover is that the crossover of two parents leads to creation of offspring in between them, in the segment that connects them in space. Contingent upon that and assuming all possible outcomes are comprised within that segment<sup>4</sup>, if this operation produces offspring with equal probability in any point of that same segment, one would refer to it as Uniform Geometric Crossover *UX*.

There are three inbreeding properties [9] that arise from using a recombinator operator that is considered geometric:

1. *Property of Purity* states that the recombination of one parent with itself can only reproduce the parent itself.
2. *Property of Convergence* states that the recombination of one parent with its offspring cannot produce the other parent of that offspring unless the offspring and the second parent coincide.
3. *Property of Partition* states that the two recombinations, the first of parent *a* with a child *c* of *a* and *b*, and the second of parent *b* with the same child *c*, cannot produce a common grandchild *e* other than *c*.

If any crossover operator fails to meet any of these properties, it is, by definition, considered non-geometric.

In this work there are two different recombination techniques that are used: *one-point crossover* (geometric) and *extension ray crossover* (non-geometric). Both represent a lower end of complexity regarding the vast amount of crossovers that can be used to further introduce dynamic geometricity. These provide a level of simplicity that is useful when creating the idea of such a framework. This choice also mitigates the causal relationship between the usage of more specific complex types of crossover and general results of one's algorithm, maintaining the focus on the essence of the strategy employed in this work.

One-point crossover [10] is a mask-based crossover for binary strings. It uniformly produces offspring in the segment between the two parents, respecting all our previous properties, thus being referred as a uniform geometric crossover as represented in Figure 2.3(a). This crossover works by choosing a random number in range of the parent's representation length, then cutting and swapping tails (Figure 2.4(a)).

Extension ray crossover [11] extends the segment passing through both parents, originating in a point and passing through a second point, producing offspring outside this connected segment (Figure 2.3(b)).

---

<sup>4</sup>This is termed a *complete* geometric crossover.

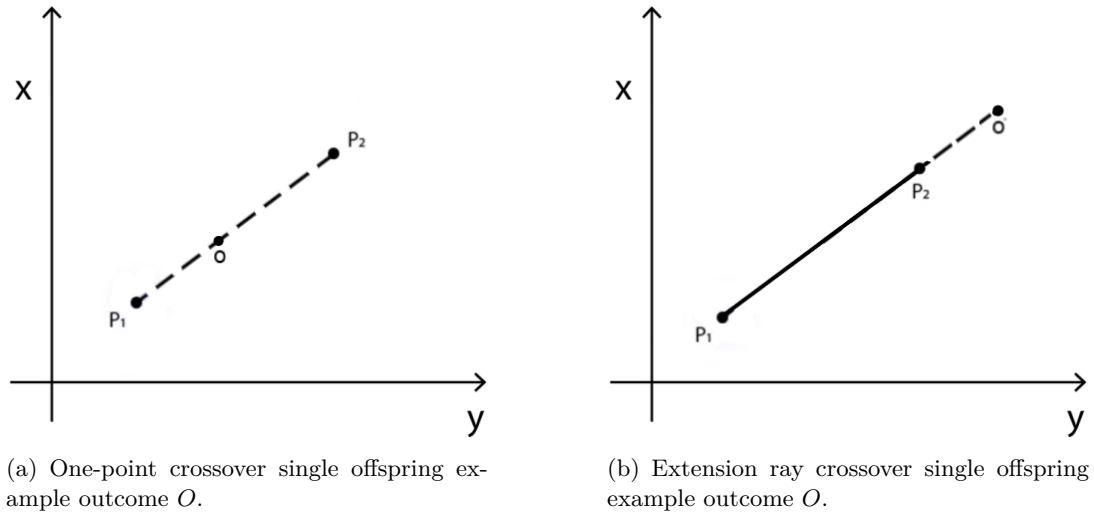


Figure 2.3: 2-dimentional spatial visualization of both crossovers in action.

Still in a hypothetical binary string environment, this method works by starting  $O$  as a copy of  $p_2$ , then choosing the positions of the bit commonality between  $p_1$  and  $p_2$ , the order of their schema, and finally changing the position of those in  $O$  into their complement<sup>5</sup> (Figure 2.4(b)). This process increases, instead of decreases, the hamming distance between  $p_1$  and  $O$ , comparing to one-point or any other geometric crossover.

This technique does not respect the property of purity nor convergence as it can indeed produce  $p_1$  by recombining offspring  $o$  with  $p_2$ , previously originated from the crossover of  $p_1$  and  $p_2$ .

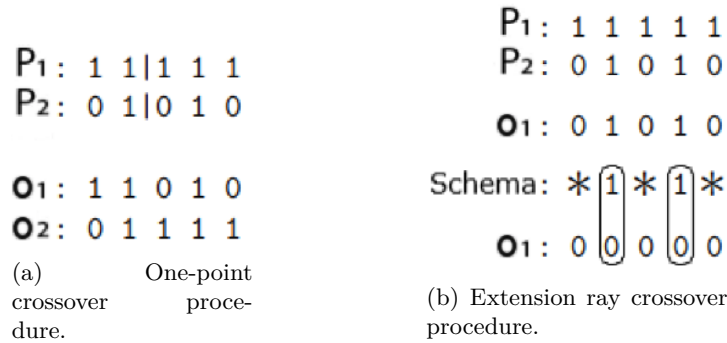


Figure 2.4: Parents representation recombination sequence of steps in both crossovers.

<sup>5</sup>There is another way of implementing this by adding weights to each segment,  $[p_1, p_2]$  and  $[p_2, o]$ , creating a probability that will dynamically control the level of extension of the  $[p_2, o]$  segment instead of always maximizing it by complementing every previous schema order position in  $o$ , as represented in 2.3(b).

### 2.2.2 Mutation

Mutation is a transformation of a point inflicted on itself, a unary operator.

This is a geometric  $\varepsilon$ -mutation operator on the search space  $(C, d)$ , if:

$$\text{Im}[M(p)] \subseteq B_d(p; \varepsilon),$$

where  $\varepsilon$  is the smallest real number for which this condition holds true. The maximum possible distance from its previous state is  $\varepsilon$ , thus all mutations are geometric for a certain value of  $\varepsilon$ .

In particular, the *bitflip mutation* [12] is an example on the lower end of the complexity spectrum of the vast amount of unary operators, in this case applied to binary strings. When applied, each bit of our solution's representation has a predefined probability of being changed into its complement. In this case,  $\varepsilon$  is equal to the length of our representation; it represents the maximum distance achieved by collapsing the previous probability on every bit in the same boolean state that represents a bitflip incidence. This does not mean the distance will be always the maximum. In fact, this distance,  $K$ , will tend to the predefined probability  $p$  multiplied by our representation length  $L$ , but that is not enough to satisfy the definition of a deterministic constant  $\varepsilon$ . Note that  $K$  is always smaller than  $\varepsilon$ , unless  $p$  is 1, in which case it should not even be called a mutation due to its usual definition, but a total spatial symmetric transformation.

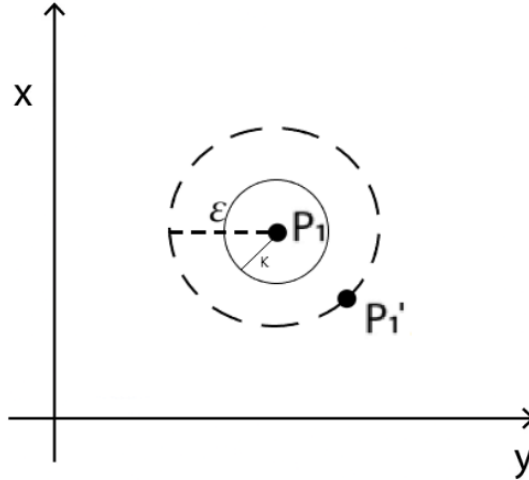


Figure 2.5: Bitflip mutation outcome  $P'_1$  spatial view.

In figure 2.5,  $P'_1$  is a rare example of a possible outcome as it could have landed anywhere in-between the circle of radius  $\varepsilon$ . But even if  $K$  is below 50% (like abstractly portrayed by the smaller circle), the flipping is still an independent event in each position in the representation of  $P_1$ .

It is imperative to understand that the concept of diversity is connected to the notion of distance created by operators acting on previous solutions. Let point  $Z$  be the initial point before any unary operation and in the case of binary operators, let

this point be the exact middle point between both parents. Intuitively, to compare the spatial effect of these transformations in terms of distance created to the origin  $Z$ , it would be correct to reason that mutation is generally a weak-diversity enhancer operator and non-geometric crossover is often a strong-diversity<sup>6</sup> enhancer. In contrast, geometric crossover is always a diversity reducer.

The next section will explore the same intuition with new more adequate concepts.

### 2.2.3 Convex combination, convex hull and convex search

Understanding convexity facilitates reaching the final abstract conclusions on how these dynamical systems are evolving while solving problems, space-wise.

Three examples of 2-dimensional sets, convex and non-convex, are presented in Figure 2.6.

*Convex combination* is a combination of vectors where all coefficients, their multiplicative factors, are non-negative and sum up to 1. If one has a set  $S$  closed under convex combinations, it is referred as a convex set so that any  $a, b \in S$  implies  $\overline{ab} \subseteq S$ . The *convex hull* of this set is the boundary of its convex closure, also called a convex *polytope*, i.e. platonic solids.

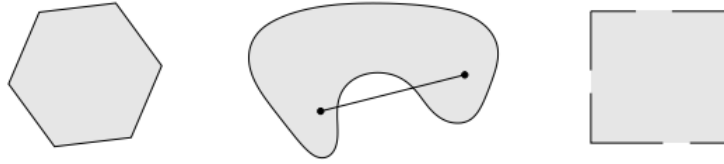


Figure 2.6: Convex set on the left and two non-convex sets on the right.

Geometric crossover properties lead to convex outcomes, reducing the global closure by locally reducing the convex hull of the present generation's pairs of parents. In other words, the distance between two offsprings will be smaller than the distance between their parents.

Intuitively, this formulates a concept of convex search given that over time, executing selection, crossover and mutation multiple times over any number of generations will lead to a search space reduction [8].

Mutation is locally non-convex due to the convex hull of a point being the point itself, thus always creating a new point outside of the set  $S$  - itself. However, this will extremely rarely counter the convexity outcomes carried by geometric crossover. This is because mutation generally does not possess a sufficient radius of action to influence our global, population level, convex hull.

Figure 2.7 illustrates an hypothetical spatial evolution from one generation to the next one, capturing the global convex hull update produced by the usage of selection, geometric crossover and mutation in both populations as well as the local pairings'

<sup>6</sup> $K$  is generally smaller than the distance between  $P'_1$  and  $Z$  originated by extension ray crossover.



convex hulls. It conceptually accounts for the possible effect of mutation that may be observed by the expansion of the simple parent connected line (which represents the convex hull of geometric crossover) into a larger stadium geometric shape. In this geometric shape, the radius is equal to  $K$ . This evolution is depicted in a phenotype agnostic environment<sup>7</sup> in order to preserve the simplicity and objective of this visualization.

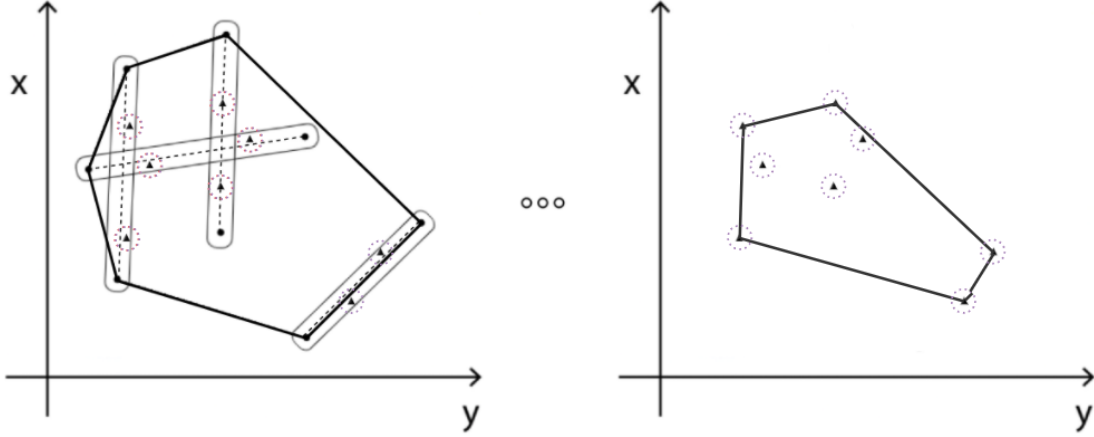


Figure 2.7: Convex hull reduction through a generation. Triangles are the offspring created by their parents, the dots.

The fact that a certain search strategy is considered a convex search is the reason one is able to explore gradients found in geometrical space, to pursue them, doing what one previously called *relevance realization*. This is also a fundamental challenge researchers encounter while creating new strategies. The premature convergence problem exists where space exploration leads to final stages where neither the global optima nor acceptable local optima are present or will be present because, while diminishing the volume of our population's convex hull, those were let out.

An equilibrium that is hard to achieve is present when tackling any problem, especially if one is trying to stay problem agnostic by not using a priori information about it.

Diversity maintenance strategies will often focus on creating or maintaining distance between members of our population, but this will only have effect relatively to our continuous volume decreasing convex polytope, meaning there will be diminishing returns when it comes to these methods.

These strategies will, abstracting from the vast number of different implementations, create ways that artificially increase the chances of future genetic material propagation of certain individuals in the name of diversity. This is achieved by creating

<sup>7</sup>Hence the random selection of pairings/lack of selection rationale.

methods in which genotypically remote individuals have their phenotype (fitness) artificially augmented and/or artificially reducing the fitness of genotypically close individuals in centralized subpopulations. Phenotype agnostic methods also exist, focusing only on promoting or demoting certain individuals with the help of genotypic distances. Each method or their combination leads to the same outcome - the increase of targeted individuals' probabilities regarding mating/prevaling in the population. Cross-generational Probabilistic Survival Selection (CPSS) reduction [13, 14] and fitness sharing [15], while using convex methods, are examples of these strategies.

Other examples of strategies that will disrupt this pattern of diminishing returns have origins in random search abstractions. These include: 1) the spawning of new random solutions, as they have a non-zero probability of landing outside our global population's convex hull, or 2) non-geometric crossover, since this is a local non-convex transformation that will also have a non-zero probability of creating offspring outside our global convex hull.

### 2.3 Grammar taxonomy: Biology vs Mathematics

The grammar of any language, in this specific abstract case the language of evolutionary algorithms, is the underlying structure and systematic description of its features.

Not only in genetic algorithms but also in other biology-inspired algorithms, one should not aspire to discover different mechanisms of random species to coin a new algorithm. Instead, one should focus in acquiring a priori understanding of the mathematical properties of this algorithm's operators. This enables the capacity for directly pointing at tangible differences that can indeed make it a new algorithm instead of an offshoot of a previous one.

Biological differences can clearly be present i.e. being inspired in different species that have no connection whatsoever. However, if two algorithms hold in their operators the same geometrical foundations, there will be a harder time justifying their uniqueness as well as their results' differences, such that when being compared, they might lie on the different search space properties leaning towards a slightly easier search process in one of them. This would not be due to their main rational pillar's discrepancies, but due to smaller discrepancies between both, creating a harder path to draw causality from.

As commented in Castelli et al. (2022) [16], exploiting a natural metaphor observed in the external world is evermore incomplete and can easily lead to unreliability or misinterpretation of results, as bias that might not be comprehended/caught by readers can be present. Framing one's external methodology findings or internal ideation in a mathematical manner is extremely important to bring structure and create order in such an extensive field of research. It is also intuitively helpful to understand how a certain strategy works and might unfold on more general terms as in other types of problems that might not be tested in a specific work.

Framing these algorithms in geometrical and topological terms, paying attention to how the operators transform points in the underlying space can also be the difference between white-box and black-box ideation.

There is no escaping geometry, statements like 1) 'This operator maintains diversity' and 2) 'This operator produces offspring outside/inside of the convex hull of its parents' do not hold the same impact even though they can be used to answer similar questions on how to deal with genetic drift. Without knowing what one means (space-wise) by maintaining diversity or what it is relative to, how could one perceive it in a deeper way that is so necessary to interpret the operator or measure<sup>8</sup> this effect? On the other hand, the second statement offers more in-depth, insightful information. It also produces less possible points of inconclusion, being less dependent on further information. Ideally both statements would complement each other, reflecting both the abstract and precise degrees of expression.

Geometrical claims will often complement and in some cases substitute the biological context of our operators in an instrumental way. To put it in simple terms, translating the biological grammar into geometrical grammar augments the explaining ability, permits a deeper analysis of components and opens a path towards the general development of the field instead of keeping these methodologies semantically scattered.

Finally, the following analogy will be explored: The creation of a successful algorithm is in itself an optimization problem. Importing external metaphors and continuously testing them will be closer to a definition of random search than acknowledging what was already extracted into mathematical terms and having a better insight on how to maneuver between different operators found in those metaphors. This facilitates a better relevance realization ability, opening the possibility of intuitive interior<sup>9</sup> design, taking it a step further away from random search. This is to be taken with a grain of salt as the No Free Lunch theorem tends to agree that random search is far from useless, matter of fact, as useful as any other strategy.

## 2.4 Adaptive framework and non-geometric operator placement

### 2.4.1 Constant diversity maintenance

Usually, when implementing genetic algorithms, one decides on static arguments such as type of selection, crossover and mutation as well as the probability of each one occurring in any given generation. For example, by deciding on using one-point crossover, as previously explained, the convex hull of one's population will decrease in size, reducing the genotypical diversity between members. For this reason and for the sake of not converging without properly exploring a non-minimal part of our space, measures that

---

<sup>8</sup>Most processes are probabilistic but one can expect most outcomes closer to the center of their distributions, creating realistic expectations on a certain operator.

<sup>9</sup>Exporting one's creativity instead of exclusively importing it from the outside world, by developing rational methodologies.

address this problem must also be considered. These measures consist in deciding which techniques to implement and their degree of magnitude. In the CPSS reduction of DCGA [13], there is an a priori choice of the shape coefficient  $c$  and the exponent  $a$ , creating a curve that directly influences the probability of an individual getting selected by its genotypical distance to the best current solution, changing the course of the natural elitism that is inclined to greater phenotypes. During fitness sharing [15], a niching technique, the variable  $\sigma$ -share depicts how distant two individuals must be for them to suffer a fitness decrease, also indirectly taking an a priori assumption of the landscape ruggedness, specifically when referring to genotypic fitness sharing. These are examples of a priori static choices.

If one has any information on the details of the topology of our problem, there can be utilised variable values that promote greater or lesser diversity, but the point of the best possible heuristics is to perform while being the most problem agnostic possible.

In a problem agnostic framework, choosing a diversity inducing static magnitude will always be a blind trade-off, even though considered a good practice.

It is almost impossible to fully justify these actions that are applied to every member of one's population, for every generation. Would humans have a bird's eye view of the progress of an algorithm, we would know when it would be justified to activate and deactivate diversity measures, although static diversity methodologies are nevertheless extremely useful and practical way of navigating this genetic drift problem, especially in rugged spaces<sup>10</sup>, proving their superiority against other algorithms that do not take these diversity measures i.e. a simple Hill Climber.

## 2.4.2 Dynamic diversity maintenance

Dynamic diversity denotes the application of one's diversity maintenance methods, whatever they may be, in different instances of the search process. It must follow some sort of rationale that causally links to either these method's usage or to variation in their diversity-affecting parameters, like the magnitude of a certain operator. This implies there is the introduction of a justification for using certain methods at selected times (generations) that will affect a group or the entirety of the individuals in the population along the lifespan of the search process.

Jassadapakorn & Chongstitvatana (2011) [17] presents a fair example of an approach where diversity is highly connected to the continuously varying necessities of the underlying space. There are two important concepts that must be understood before diving into the full process:

- *Preference type* - an individual bias towards recombining with different degrees of genotypically distanced mates.

---

<sup>10</sup>A priori solution space knowledge is present.

- *Contribution* - a record for each preference type performance after a new population is created, meaning the degree of positive outcomes found with each preference type.

The preference type represents a discrete scale from 0 to  $\tau_{max}$ , where  $\tau_{max}$  dictates in how many pieces the genotypical bias spectrum will be sliced. This spectrum goes from genotypically close to remote, therefore the higher/closer preference type  $\tau$  is to  $\tau_{max}$ , the stronger the bias towards farther individuals in space, influencing the following difference function  $D$ :

$$\begin{cases} D(\tau, d_i) = 0.5 + \frac{\tau}{\tau_{max}} (d_i - 0.5), & \text{for maximization problems.} \\ D(\tau, d_i) = 0.5 + \frac{\tau}{\tau_{max}} (0.5 - d_i), & \text{for minimization problems.} \end{cases}$$

where  $d_i$ , in this case, is simply the relative hamming distance between  $x_1$  and its candidate mate  $x_2$ .

This function  $D$  will then be used to choose the second parent  $x_2$  that mates with the previously naturally<sup>11</sup> selected first parent  $x_1$  in a tournament  $S_t$ , where its size, the number of candidates, is previously chosen by the algorithm designer:

$$x_2 = \operatorname{argmax}_{i \in S_t} [f(y_i) \cdot D(\tau, d_i)]$$

For instance, in a minimization problem, if  $\tau$  is high, the fitness  $f$  of a distant candidate mate  $x_{2a}$  will be significantly reduced while the fitness of a closer candidate  $x_{2b}$  will be negligibly reduced due to their difference function  $D$  dissimilar results.

After a generation  $t$  ends, the contribution of each preference type is calculated:

$$\text{Contribution}(\tau, t) = \frac{\# \text{Successful Cross}(\tau, t)}{\# \text{Cross}(\tau, t)}$$

These contributions represent the next generation's  $t + 1$  pool of probabilities attributed to each of our preferences  $\tau$ . These probabilities, after normalized, will be drafted in a proportional manner into our individuals<sup>12</sup> as the selection method takes place multiple times until the end of the generation. It is also important to refer that in the initial generation of this algorithm, the contributions are equally set to  $1/\tau_{max}$ , where  $\tau_{max} = 4$  during this exercise.

This work will consider a successful crossover one that produces at least one offspring with a superior or equal fitness than both parents if not equal in representation to any of both, instead of superior only as was portrayed in [17]. This detailed difference arrives from the assumption that in the rare event of an offspring possessing the same fitness as its best parent, besides the possibility of that offspring and a parent coinciding, the creation of a solution in a new spatial position that is not inferior in fitness is a positive

<sup>11</sup>Meaning with any type of selection method, contrary to  $x_2$ .

<sup>12</sup>Only the first selected parents are assigned one preference type that will be used to find their mates.

event. One can intuitively see how this could be important in the presence of a plane sector of our fitness landscape<sup>13</sup>.

To maintain every preference type alive throughout the search process, instead of occasionally having certain contributions nullified due to their failure in creating positive outcomes in a specific generation or search space zone, there is a minimum threshold of 10% for each contribution at every generation. The contributions are still competing against themselves for a total of 60%, instead of 100%, to conserve this dynamism during the entire search. This is of specific importance in the framework presented in the next section, due to the low probability of successful outcome of a specific employed recombination method, non-geometric crossover.

An important detail regarding these contribution fluctuations is in the case that every recombination within a given generation produces non-successful outcomes, the contributions will not all drop to 0 but instead be reset to  $1/\tau_{max}$ .

The present process can be outlined as following:

1. Randomly generate the population of individuals.
2. Evaluate each individual by a fitness function.
3. Set the contribution equally for each preference type for the first time.
4. Select an individual and its partner with the proposed mating procedure. The probability of choosing a preference type is proportional to its contribution.
5. Reproduce two new individuals for the next generation by crossover.
6. Repeat step 4 and 5 for the whole population.
7. Evaluate each new individual by the fitness function.
8. Compare the fitness value of the new individuals and their parents. Calculate the contribution of each preference type.

Repeat step 4-8 until the final generation is reached.

### 2.4.3 Non-geometric pulse - dynamic geometricity rationale

The magnitude of spatial transformation brought by local geometric crossover operations are proportional to the global population's convex hull volume.

Non-geometric crossover can be a powerful diversity inducing method, yet in a non-dynamic environment lacks direction. Its limitation is that it does not point at anything, not following any gradient in space unlike geometric crossover and again, a blind trade-off that can ruin a locally constructive convex search while weighting on the randomness magnitude of one's algorithm.

As previously discussed, non-geometric crossover (especially applied in a framework like mentioned in 2.4.2) can be a very important component of the success of a strategy by heuristically playing with the notion of usefulness of contraction or expansion by listening to the contributions of the previous generations.

---

<sup>13</sup>A closure in space in which all points have the same fitness score.

Complementing this geometrical idea, one can use the idea of episodes of con-span<sup>14</sup>, material contraction during space expansion, as metaphor for understanding what may occur within this framework. This material contraction alludes to the fact that a majority of members will become closer in space due to the convexity of geometric crossover present in most preference types, but the non-convexity of non-geometric crossover present in the highest preference level(s) also allows the boundaries of this space and its correspondent volume to be expanded when new offspring are located outside our global convex-hull.

In step 5. of the previous process depicted in 2.4.2, when applying crossover, one will employ non-geometric crossover if parent  $x_1$  carries a preference  $\tau = \tau_{max}$ , shaping the largest possible bias towards the farthest possible mates into a non-convex combination requirement. In contrast, lower preference types will employ geometric crossover, remodelling the definition of preference into a geometricity regulator throughout the optimization process of this genetic algorithm.

Figure 2.8 illustrates an hypothetical spatial evolution from one generation to the next one, this time also using non-geometric crossover in a couple of pairings, following the latter transformed population in Figure 2.7.

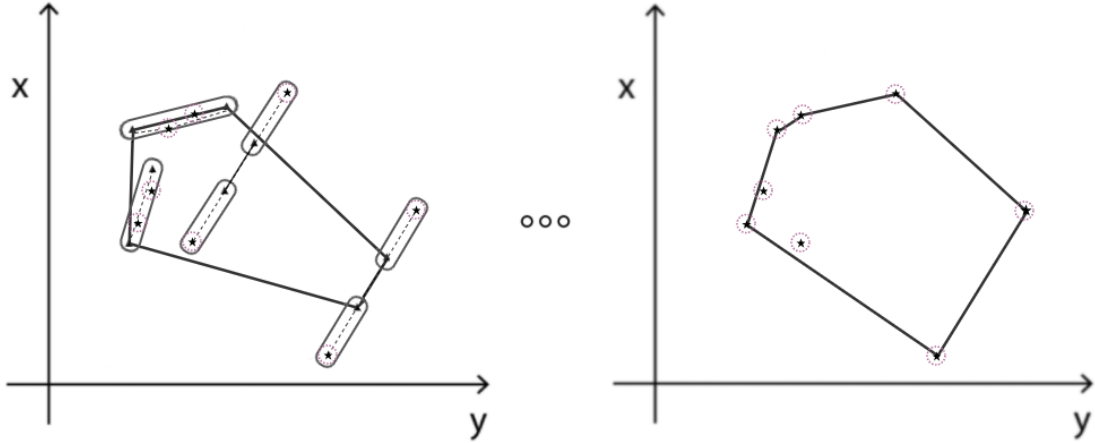


Figure 2.8: Convex hull expansion through a generation with adaptive geometricity. Stars are the offspring created by their parents, the triangles.

In this example, two pairs of parents performed one-point crossover while the other two pairs performed extension ray crossover. Notice that the expansion in the global convex hull is not something that constantly occurs. Firstly, non-geometric crossover can occur but its contribution fluctuation will dictate the augmenting/reducing of the probability of a new individual receiving it. Even if preference  $\tau_{max}$  possesses a 15% contribution, there is always a case where this probability is not manifested/collapsed during the selection step. Secondly and more important, the offspring produced by non-geometric crossover are not necessarily outside the global convex hull, as observed in one

<sup>14</sup>Assuming that during the entire process or the totality of recombinations, geometrical crossover is used a majority of times by our selected pairings.

offspring created by this method in Figure 2.8. On the other hand, material contraction will only occur when geometric crossover superimposes non-geometric crossover, with respect to the number of occurrences and their spatial outcomes in a generation. If this is the case, in the event that non-geometric crossover does not produce any solution outside of the global convex hull, we can equate this material contraction with a search space reduction.

Ideally, the moment convex search starts having mostly negative effects on our phenotypes, the contributions of the preferences that employ geometric crossover decrease. Non-convex search will have an increased share of occurrences as the contribution of the preference that employs non-geometric crossover takes a larger slice of our pool of contributions. Likewise, there will also be moments where non-convex search will have a negligible share in our generational process, creating an interesting diversity maintenance dynamic between both types of crossover usage.

As referred in 2.2.1, because this work's version of extension ray crossover maximizes the possible distance in our extended line that originates in one parent  $x_1$  and crosses another  $x_2$  (instead of uniformly creating offspring in the segment from  $x_2$  until the maximum travelled distance, its schema order), it can be thought as a convex hull pulse, metaphorically speaking. This is because it augments the probability of creating offspring outside the global convex hull - expanding it for the next generation.

As a final implementation detail, in order to create two offsprings during the non-geometric crossover *ER* instead of just one, this method is applied in both directions  $ER[x_1, x_2]$  and  $ER[x_2, x_1]$ , also boosting this promotion of diversity.

Finally, it is important to express that there is not only one way of employing this framework due to the existence of more strategies that hold dynamic diversity maintenance properties. A plethora of methodologies can be created from this main idea of trying to logically map non-geometric crossover into the continuously revised search space necessities.



## EXPERIMENT

### 3.1 Algorithm collection

Before proceeding to the testing bed used in this work, the collection of algorithms, their correspondent implementation details, and the parameters used, will be introduced. The following are the six genetic algorithm variants used:

- Genetic algorithm, *GA*.
- Two variants of DCGA, *DCGA1* and *DCGA2*.
- Self-Adaptive genetic algorithm, *SA* referred in 2.4.2.
- Two variants of non-geometric pulse, or self-adaptive geometricity genetic algorithm, *P* and *P'* referred in 2.4.3.

This is a relatively broad group of techniques that ranges from the simplest implementation of the genetic algorithm, to the *DCGA*, an algorithm that uses a static strategy to maintain diversity, and finally to both types of self adaptive GAs, the *SA* possessing static geometricity and *P*, implementing adaptive geometricity.

#### 3.1.1 DCGA1 and DCGA2 distinctions

As exemplified in Shimodaira (2001) [13], one can create different curves that externally control the selective pressure and population diversity during the CPSS reduction step. This is based on the assumption that the algorithm can't differentiate populated regions being local optima or global optima. In this work, we use the following parameters:

- DCGA1:  $c = 0.01$ ,  $a = 0.19$
- DCGA2:  $c = 0.234$ ,  $a = 0.5$

These are the parameters exemplified in that work. As previously explained, these curves portray the probability of an individual making part of the next generation's population as a function of the relative distance to the current best phenotype.

The difference between these curves can be seen in the Figure 3.1:

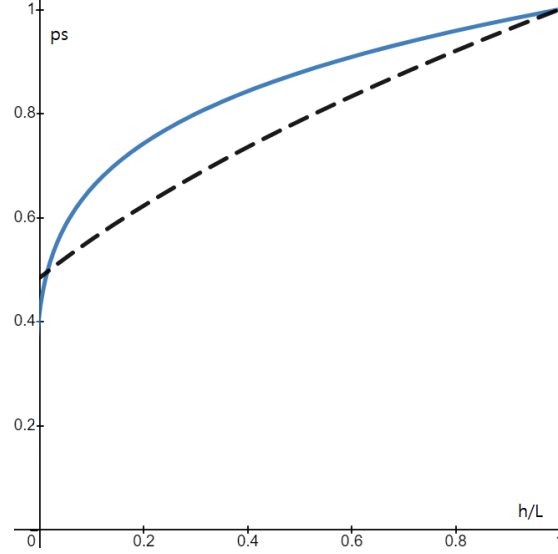


Figure 3.1: DCGA1 portrayed as the blue curve, DCGA2 as the black curve.

### 3.1.2 P, P' and SA distinctions

The possible adjustments outside of the main framework skeleton<sup>1</sup>, meaning the pivotal methods that represent the main structure of these versions of genetic algorithms, are innumerable. One should try to capture the questions that are considered important when implementing the methods that account for these non-pivotal details. This is contrary to the ordinary trial and error method implementation that aims for better results across one's benchmarks, while inadvertently losing generalization ability. Doing so also increases the probability of getting structurally lost without the capacity to causally link certain adjustments on the algorithm to the latest outcomes.

Although part of the same framework, P and P' have two main differences. These study the outcomes of the considered discrepancies that were deemed important to pursue.

The first main difference is the selection technique.  $P'$ , after the selection process of the first parent, will use a global population difference function tournament to find its most preferred candidate in the entire population, instead of in a randomly selected limited set of candidates. This change impacts the  $P'$  performance when compared with  $P$ , but answers the question: "Is it significant for the first parent to pursue the most precise representation of its preference<sup>2</sup> - highest possible difference function scoring individual in the population?". During this process, both parents  $x_1$  and  $x_2$  are deleted from the population. Thus, none of the genetic material of our entire population is left unused nor it is reused, studying the diminishing power of the schema theorem (possibly weaker/slower convergence) as the best phenotypes will not, in a single generation,

<sup>1</sup>Adaptive selection, adaptive crossover and possible mutation.

<sup>2</sup>Acknowledging that the worse the fitness, the lower the probability of having a broad set to choose a preferred individual from.

reproduce with more than one partner.

The second main difference is that on  $P'$  there is also a similar implementation of part of a technique that is used in *DCGA*, the CPSS reduction. Given that we control diversity with the preference assignments and correspondent crossover type, which are brought by the contribution updates every generation, only the elitist portion of this process is used, as well as the elimination of duplicates. Thus, the parent population will compete with their offspring for a spot in the next generation's initial population in a purely fitness-wise competition. This studies the enhancement of elitism and the suppression of identical genetic material usage in a generation (in the case of duplicates). We are exploiting the ability of this framework to maintain or create the needed amount of diversity at every moment, possibly reducing the negative effect that elitism commonly carries, i.e. premature convergence.

***SA/P selection skeleton:***

1. Tournament selection, size=3
2. Difference function tournament, size=3
3. Return pair  $(x_1, x_2)$

***P' selection skeleton:***

1. Tournament selection, size=3
2. Eliminate winner  $x_1$  from population
3. Difference function tournament,  
size=length(population)
4. Eliminate winner  $x_2$  from population
5. Return pair  $(x_1, x_2)$

$P$ , as portrayed in the previous skeleton, goes hand in hand with  $SA$ , which is important to provide a fair comparison and evaluate the main hypothesis. In particular, the only difference between these two is the main concept of this work: the presence of the underlying space necessity swaying the usage of non-convex methods, more specifically, non-geometric crossover.

There are 4 ( $\tau \in \{0, \dots, 3\}$ ) preference levels used in this work for the algorithms that use this notion,  $P$ ,  $SA$  and  $P'$ . In both Pulses, the last level of preference was reserved for the usage of extension ray crossover, the others to one-point crossover. In  $SA$ , one-point crossover was used in every preference level.

### 3.1.3 Interpretation of findings

Conceptually, there should be two different levels of comparison when analysing further results. The first one being the performance comparison, leaving aside the plausibility of the hypothesis rationale of  $P$ , between  $SA$  with  $P$  since both are antagonistic in their theoretical foundations.

In a second level of analysis, even if successful in the first level, we should also understand how this framework ( $P + P'$ ) competes with all other variants focusing more directly in general results across the considered benchmark.

### 3.2 Benchmark functions

The problem set used in this work is the *CEC 2017* function suite [18] for single-objective real-parameter numerical optimization.

These functions provide a complex test bed for various kinds of novel optimization algorithms. The suite is composed by unimodal, multimodal, hybrid and composition functions that are shifted, rotated and non-separable.

There are 30 functions and all of them possess search space boundaries  $[-100, 100]^D$ , where  $D$  represents the dimensions of these functions.

Typology	No.	Function name	Opt.
<i>Unimodal functions</i>	1	Shifted and Rotated Bent Cigar	100
	2	Shifted and Rotated Sum of Different Power	200
	3	Shifted and Rotated Zakharov	300
<i>Simple multimodal functions</i>	4	Shifted and Rotated Rosenbrock	400
	5	Shifted and Rotated Rastrigin	500
	6	Shifted and Rotated Expanded Schaffer F6	600
	7	Shifted and Rotated Lunacek Bi-Rastrigin	700
	8	Shifted and Rotated Non-Continuous Rastrigin	800
	9	Shifted and Rotated Levy	900
	10	Shifted and Rotated Schwefel	1000
<i>Hybrid functions</i>	11	Zakharov; Rosenbrock; Rastrigin	1100
	12	High-conditioned Elliptic; Modified Schwefel; Bent Cigar	1200
	13	Bent Cigar; Rosenbrock; Lunacek bi-Rastrigin	1300
	14	High-conditioned Elliptic; Ackley; Schaffer F7; Rastrigin	1400
	15	Bent Cigar; HGBat; Rastrigin; Rosenbrock	1500
	16	Expanded Schaffer F6; HGBat; Rosenbrock; Modified Schwefel	1600
	17	Katsuura; Ackley; Expanded Griewank plus Rosenbrock; Schwefel; Rastrigin	1700
	18	High-conditioned Elliptic; Ackley; Rastrigin; HGBat; Discus	1800
	19	Bent Cigar; Rastrigin; Griewank plus Rosenbrock; Weierstrass; Expanded Schaffer F6	1900
	20	HappyCat; Katsuura; Ackley; Rastrigin; Modified Schwefel; Schaffer F7	2000
<i>Composition functions</i>	21	Rosenbrock; High-conditioned Elliptic; Rastrigin	2100
	22	Rastrigin; Griewank; Modified Schwefel	2200
	23	Rosenbrock; Ackley; Modified Schwefel; Rastrigin	2300
	24	Ackley; High-conditioned Elliptic; Griewank; Rastrigin	2400
	25	Rastrigin; HappyCat; Ackley; Discus; Rosenbrock	2500
	26	Expanded Schaffer F6; Modified Schwefel; Griewank; Rosenbrock; Rastrigin	2600
	27	HGBat; Rastrigin; Modified Schwefel; Bent Cigar; High-conditioned Elliptic; Expanded Schaffer F6	2700
	28	Ackley; Griewank; Discus; Rosenbrock; HappyCat; Expanded Schaffer F6	2800
	29	15; 16; 17	2900
	30	15; 18; 19	3000

Table 3.1: Definitions and optimum values of the CEC 2017 benchmark functions.

In this experimental phase, dimensions  $D=\{10,30\}$  were used.

The maximum number of evaluations used was  $\text{MaxFES} = 80000$ , in both dimensions. This can be calculated through the multiplication of the number of generations (200) and the population size (400).

The ABF (average best fitness) in each of the 30 functions was calculated across every generation for each of our 30 independent runs.

An important detail that emerged from averaging these generations is that in a small number of functions, about 3 functions on 10 dimensions and only 1 function on 30 dimensions, these averages and their graphical representations are only useful up to a certain point. In this specific framework, specifically in  $P$ , the usage of non-geometric crossover will decrease the coupling between the passing of generations and fitness evolution. It is simple to understand why this will happen in some functions, but not necessarily why it doesn't happen in all of them, although the main hypothesis for this might have to do with the ruggedness level present in each function's fitness

landscape. Hence, the assumption that the best solution is most likely found in the last generation(s) of our runs is not correct, or at least not as correct as in all the other algorithms inside our collection: *GA*, *DCGA* and *SA*. Because non-geometric crossover produces episodes of impactful transformation on the underlying space and the standard convex gradient, the best solution might be found in different generations throughout our independent runs. This may possibly create a misleading mean ABF evolution interpretation due to the assimilation of this best solution’s fitness by other fitnesses present on other seeds at that exact generation during that calculation.

It is also important to notice that the reason this does not affect  $P'$  is simply its usage of the elitist section of CPSS reduction, that maintains the best-found phenotype in the population until it is surpassed, re-coupling the duality of generations unfolding and fitness evolution.

In conclusion, these averages are only useful up to the point that they do *not* modify the *genuine* final ABF ranking order - calculated by averaging the best solution in 30 separate runs, independently of its location in the 200 generations of each of those seeds.

Even though these are rare events, we will exemplify in the next section where these averages fail to represent the results of  $P$ , to correctly depict the overall experiment findings.

The final summary of the general parameters is showcased in the following table:

<i>Parameter</i>	<i>Value</i>
Population size	400
Length of chromosome	{200,600} bits, D={10,30}
Number of generation	200
Number of independent run	30
Crossover probability ( $P_c$ )	100%
Mutation rate ( $P_m$ )	0%
Tournament size	3

Table 3.2: General testing parameters for all algorithms.

### 3.3 Benchmark results

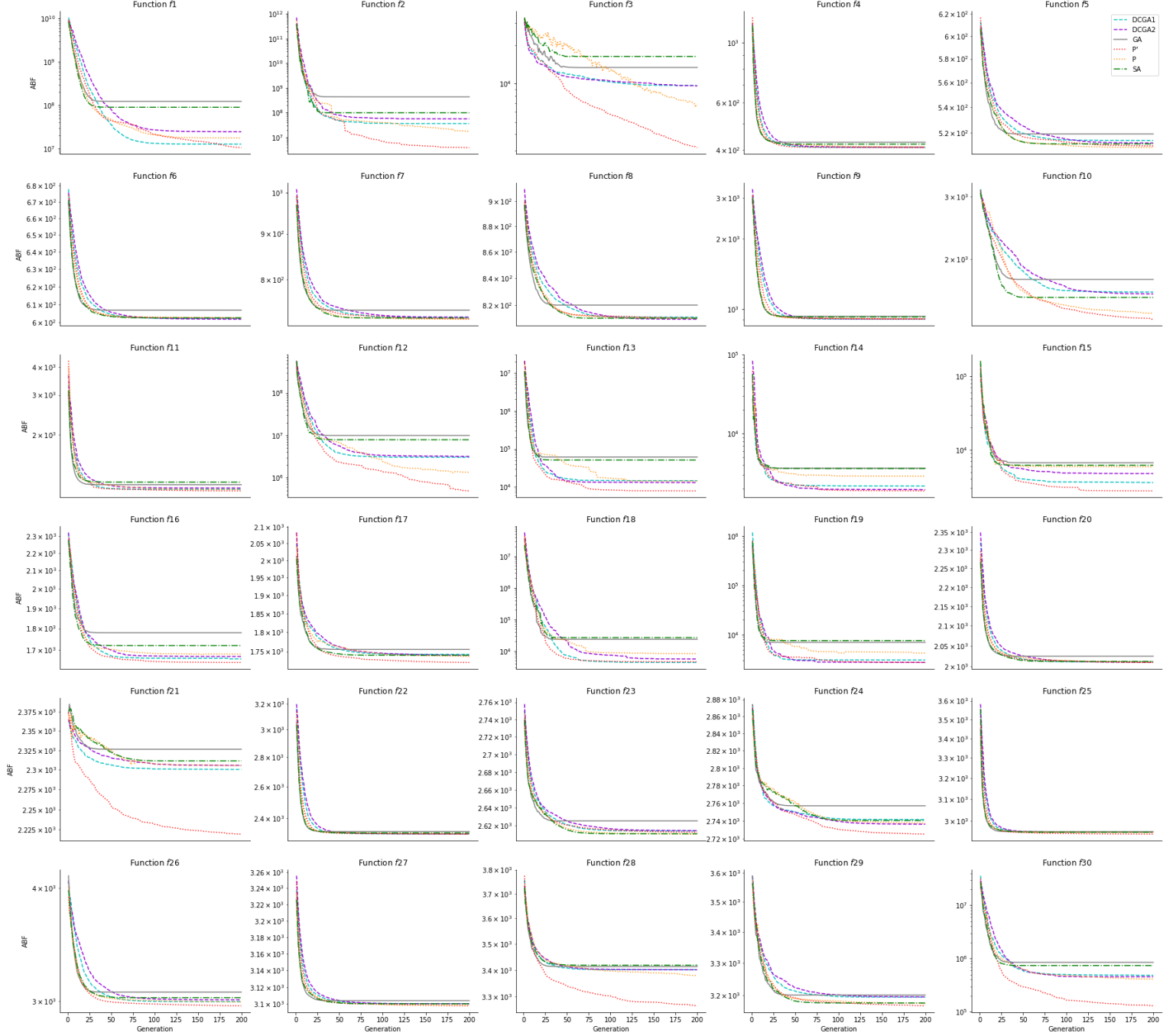


Figure 3.2: ABF comparison over each of the 30 functions with Dimension = 10, calculated by averaging each generation of 30 different independent runs.

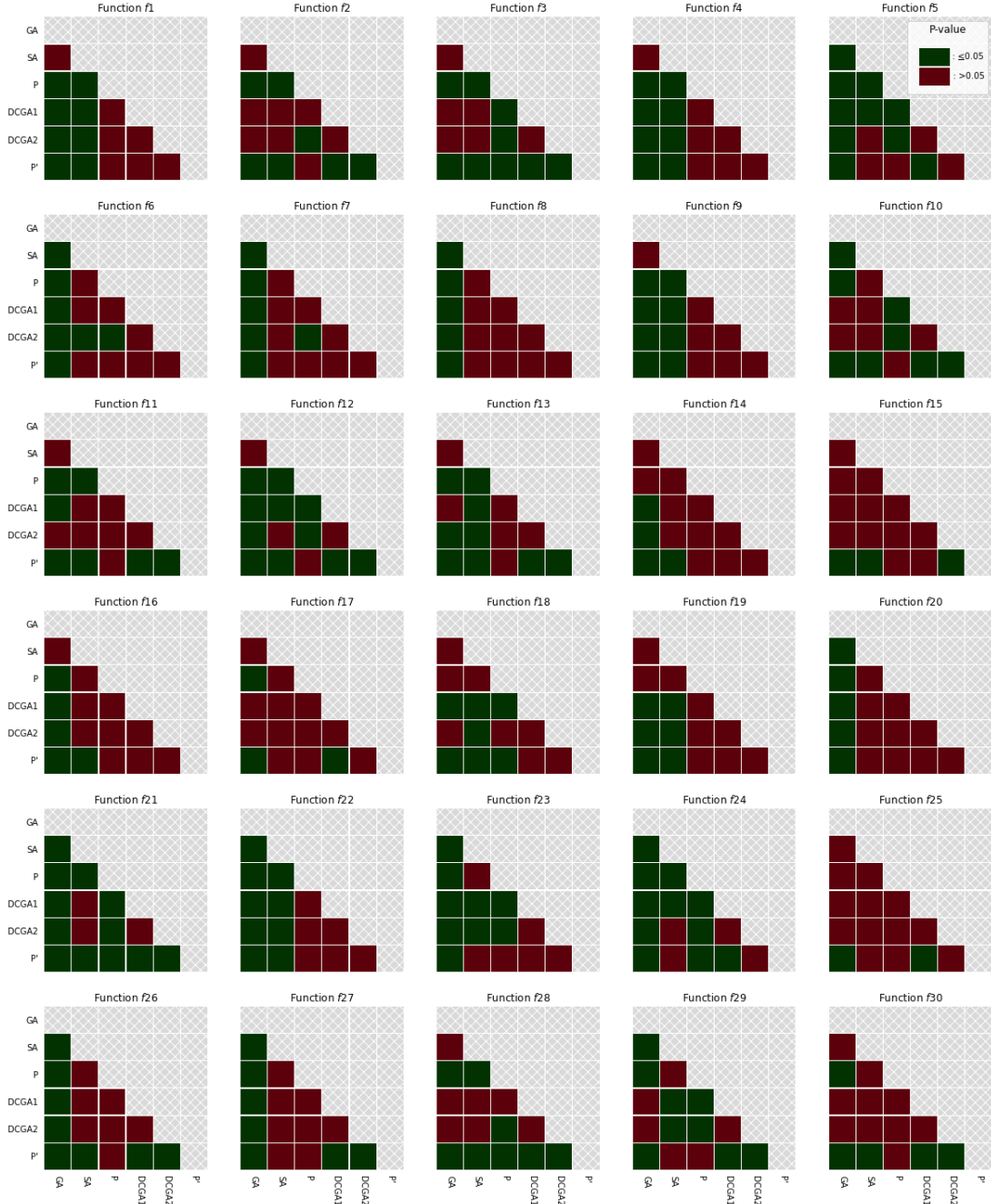


Figure 3.3: P-value results of the Mann–Whitney U test with the Bonferroni correction over each of the 30 functions with Dimension = 10.

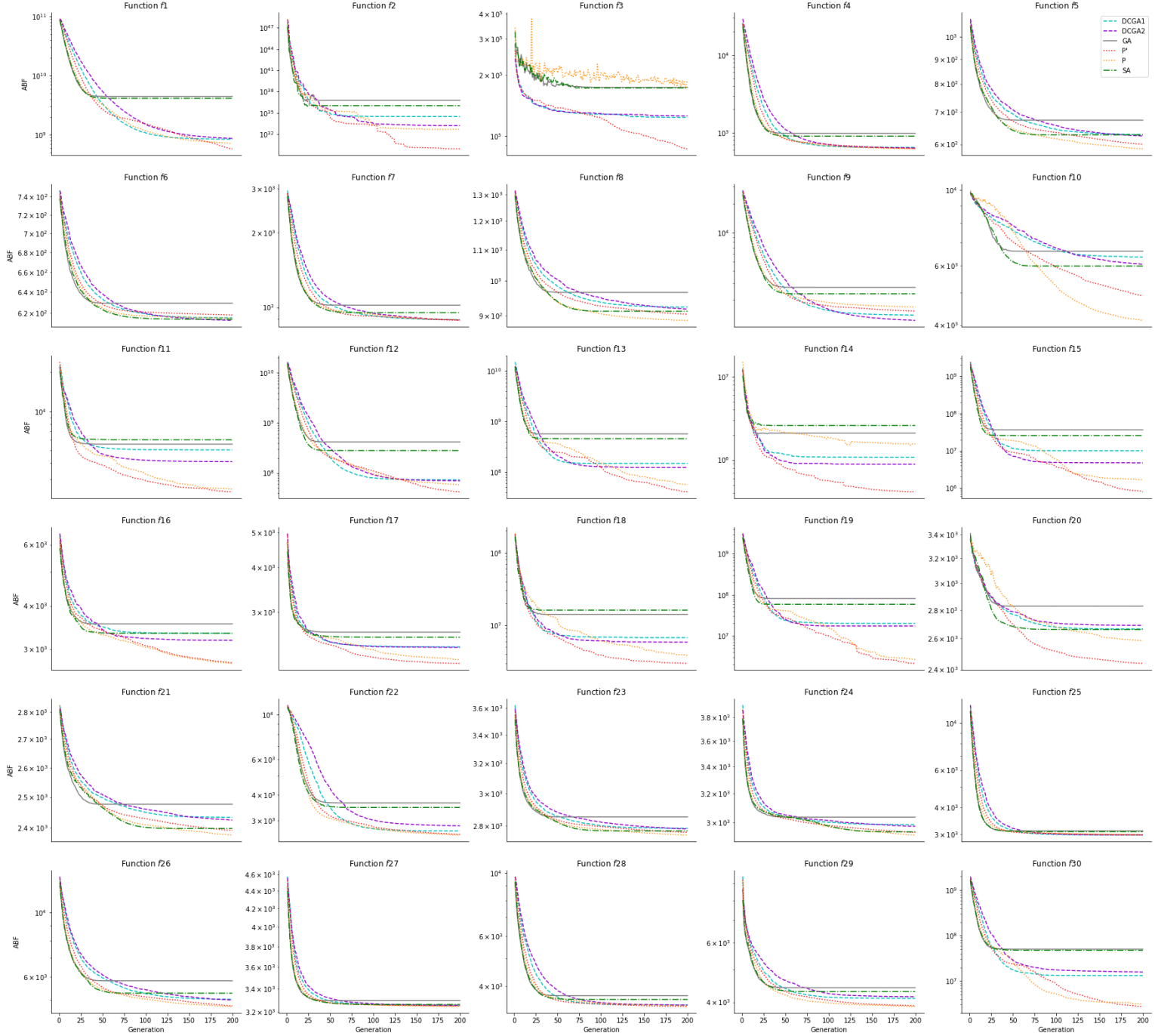


Figure 3.4: ABF comparison over each of the 30 functions with Dimension = 30, calculated by averaging each generation of 30 different independent runs.



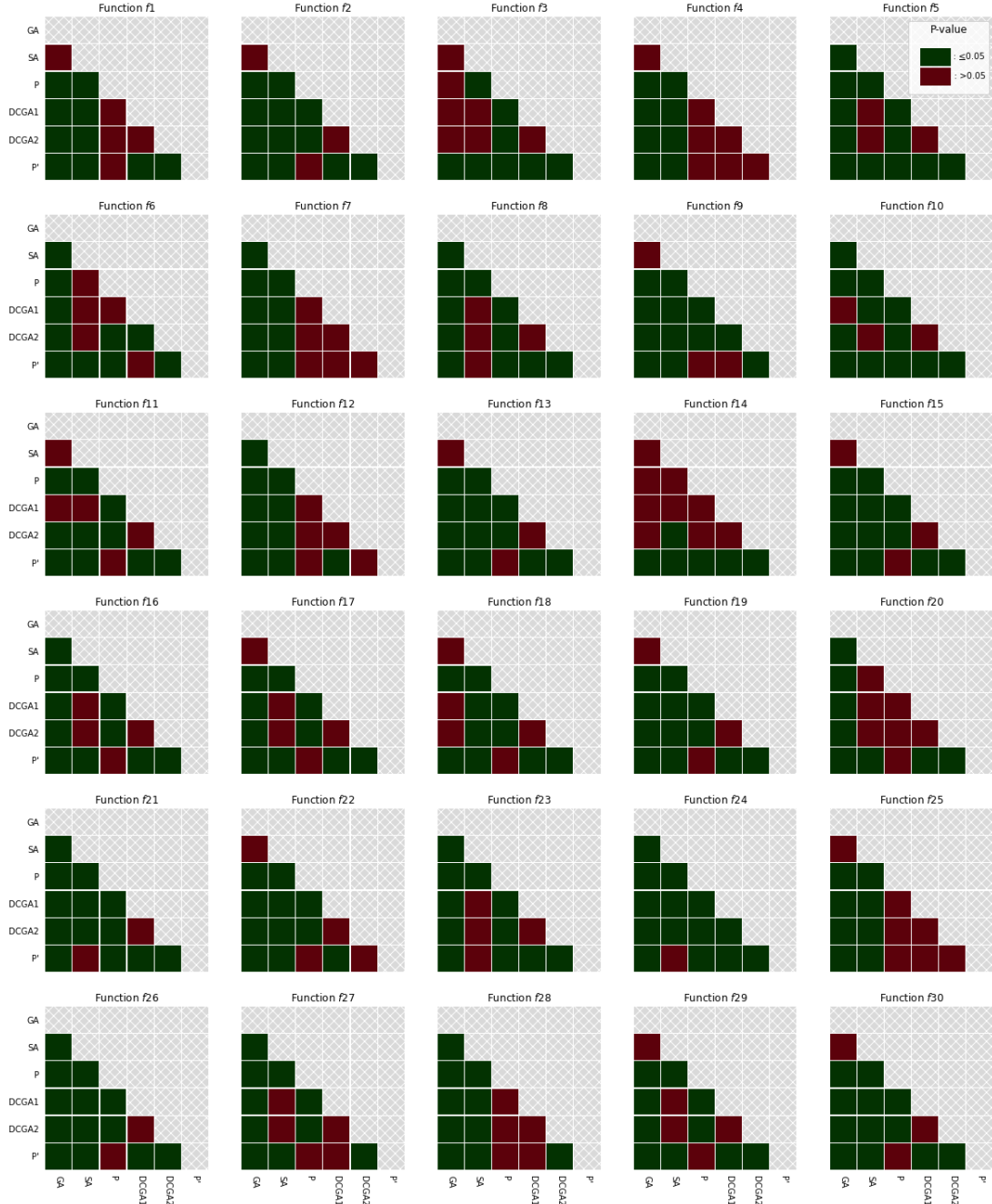


Figure 3.5: P-value results of the Mann-Whitney U test with the Bonferroni correction over each of the 30 functions with Dimension = 30.

### 3.4 Results overview

From the first level of analysis present in 3.1.3, there was only 1/30 function,  $f6$ , in both dimensions, that favored  $SA$ 's average best fitness (ABF) against  $P$ .

In a second level of analysis, the general results revealed a superiority of this framework's algorithms,  $P$  or  $P'$ , concerning the ABF for 25/30 functions using the 10-dimensional test and 27/30 using the 30-dimensional test. The functions in which none of the previous algorithms had the best ABF were functions where  $DCGA1$  or  $DCGA2$  were the best performers.

The Mann-Whitney U statistical test, that evaluates the distribution differences of the fitnesses<sup>3</sup> in each function, allows quantifying the significance of the results. Even if algorithm  $x$  performs better than another  $y$ , if there is no statistical difference present in the Mann-Whitney U test (calculated with the Bonferroni correction [19]), one can only conclude that  $x$  is able to compete with  $y$ . Only with that statistical difference present can one conclude that  $x$  outperforms  $y$ .

Regarding both dimensions  $D=\{10,30\}$ , although the result differences are generally backed by the statistical results, they are augmented in the 30-dimensional test.

It is noteworthy to explain that, even though  $P$ ,  $SA$  and  $GA$  do not necessarily conserve their population's best individual throughout generations due to the absence of any elitist method, we can still see that in most cases their average fitness evolution curves are *monotonically decreasing* (Figures 3.2 and 3.4). One would argue this is a product of the natural convexity bias of the overall search process. The reason this is important to notice, specifically in the algorithm  $P$ , is for the 3rd research question in 1.2 "How can we introduce non-convexity *without* ruining the overall convex process?". The fact that these monotonically decreasing curves are present in  $P$ , despite every seed's distinguished initialization, along with the presence of non-convex methods, indicates that the overall convex process is still dominant. Otherwise, one would be presented with something closer to an horizontal line as the average of the 30 independent runs would not converge into a negative slope.

An interesting detail on this analysis is that considering the algorithm  $P$ , on function  $f3$  in the 30-dimensional test (Figure 3.4), we can connect all these concepts by assuming a greater presence of non-geometric crossover in the search process. In this case, the contribution of preference 4 ( $\tau_{max}$ ) noticeably increased its share when compared to other functions<sup>4</sup>. This lead to a non-monotonic fitness curve due to the generational ABF calculation having to account for the increased presence of non-convexity. These events preceded good solution fitnesses being in separate generations across the 30 seeds.

---

<sup>3</sup>The best fitnesses in each seed, in each function, that are used to calculate the average best fitnesses - genuine ABF.

<sup>4</sup>Both in percentage as well as in number of generations where this increase took place. The contribution of preference 4 was in the 25-35% range throughout the process in all independent runs although it is usually closer to the its floor 10-15% while only having local spikes.

Finally, the *genuine* ABF - not the last generation's ABF - put  $P$  in second place, behind  $P'$ , which is the correct reading for  $f3$ .

This should again enlighten why in this specific framework, even though important, these curves might not tell the full story and downgrade the result of one's algorithm in certain functions. This was a rare event in this work, hence the general interpretability of both figures' (3.2 and 3.4) last generation results is consistent with further analysis of their genuine ABF ranking order. However, it is important to know that, unless one is using elitist methods that conserve the best solution throughout generations (like used in  $P'$ ), which is totally plausible, there might be a necessity of complementing the previous analysis with new strategies that paint the real picture. In the annex I, the boxplots of the genuine ABF results across the function set are provided.

As already explained, the statistical test did not have any problem in this respect because the genuine ABF could be and was used instead of the last generation's ABF.

Apropos of the contributions of each preference type, their behaviour will be different relative to each problem. Different variables such as population initialization and, mainly, the complexity of the fitness landscape topology will be the main predictors of preference fluctuations as well as their distributions. This complexity level can be relative to the entirety of a landscape or to subsections within it.

The implications of this adaptive usage of convex and non-convex methods throughout different topologies are very positive. On a theoretical and practical level, there will be no a priori bias towards any problem, either manifested by forceful diversity measures on harder problems or forceful elitist measures on easier problems. This will, in theory, augment the generalization ability of this algorithm, as it maneuvers through problems with its own guidance, diminishing the amount of weak points generated by static a priori defined variables.

Non-geometric crossover brings forth the ability to create offspring outside of the population convex-hull. The implications that originate from this are also crucial to the generality of the good results across this benchmark, especially when comparing to other dynamic approaches (in this case, using only geometric methods), like the  $SA$  algorithm. There is a clear problem that is being tackled by both  $P$  and  $P'$  - whether through static or dynamic strategies, other algorithms that only use geometric methods will suffer from diminishing returns on their diversity measures. There will come to a point in these algorithms where the diversity measures will create offspring inside populated niches that we are trying to avoid, as they simply do not account for the space reduction that occurs along generations of geometric crossover usage. One can see how all other algorithms that exclusively employ one-point crossover are, in general, reaching a fitness plateau faster than both Pulse algorithms (Figures 3.2 and 3.4). Although slow in the initial generations<sup>5</sup>, they show less inclination to plateau, even in latter generations.

---

<sup>5</sup>Due to the fact that non-convex methods will incidentally slow the convergence process.

The principles that make up this framework - on both  $P$  and  $P'$  - showed consistency across the problem set and are transversal to other evolutionary algorithms.

## CONCLUSIONS AND FUTURE WORK

### 4.1 Conclusions

In this thesis a fully mathematical approach to understanding every step that composes the genetic algorithm, as well as the solution space in which it takes place, was explored.

The process of 1) undergoing our search space, going through different metric distances and their possible shaping, its evolution into a solution space or fitness landscape with different topologies through the appliance of the fitness function, to 2) the understanding of geometric and non-geometric properties of the search operators, exemplifying these different methods and their correspondent explanations while focusing on how they navigate the problem in hand, space-wise. 3) The general outcomes that are produced by these methods, convex and non-convex search. Finally, 4) we took that information into account in order to link the position of the individuals that inhabit a given generation and their positive or negative origin in relationship to their parents' phenotypes to further regulate geometric and non-geometric crossover usage in future recombinations.

The objective was to maintain diversity not by holding a priori assumptions of any problem, but letting the population evolution dictate its necessity through the created bridges, creating this idea of adequate diversity stimulation. Simultaneously, it was acknowledging that, without the usage of non-geometric crossover, this diversity maintenance measure would only produce results relatively to the permanently decreasing global convex hull instead of the entire domain of search.

The overall theoretical hypothesis that looks into the notion of diversity and applies methods that shape it in relationship with current topological necessities while also diminishing its dependence to the current area of possible search (termed global population convex-hull) was successfully pursued in this work. Practical results showed how this framework of ideas can compete in what is currently one of the best benchmark test suite available that is held under IEEE congress on Evolutionary Computation.

Non-geometric crossover, employed in this type of framework, can be treated as a rational option, flipping its usual interpretation on its head. Just like any other

method, it will serve an algorithm designer if one knows how to set it in a favourable way. Finally, proving the superiority of the algorithms is not the objective nor would serve as a conclusion due to the problem set limitation, but transmitting the rationale behind them might open new doors to new ideas and criticisms.

## 4.2 Future work

The pursued line of research is an interestingly open one where, due to the generality of the concept, many other diversity maintenance strategies can integrate these ideas. Furthermore, an analysis on specific types of space topologies can be created in order to test these integrations (new algorithms) until one finds the correct equilibrium for this duality composed by geometric and non-geometric crossover. This can be achieved by acknowledging that on extremely rugged spaces, non-geometric crossover should have a higher rate of usage than in smooth spaces. The use of the NK model can help defining the magnitude of these space properties -  $N$  being the length of the population and  $K$  representing the landscape ruggedness. Using these tools could extract a measure that correlates with this vital equilibrium, helping in qualifying these integrations. In this case, there is also a possibility of creating a new measure of diversity, one that looks into the context of its existence and is able to quantify itself by its necessity. It can be argued that looking at a diversity measure with the intuition that the higher it gets, the better, has its own nuances as we approach this topic understanding that, given a certain population state, augmenting diversity or not is a positive event depending only on the underlying space situation, not on general assumptions i.e maximizing entropy being always positive.

An example of a possible future integration is by using the famed approach of genotypic fitness sharing. This approach would exploit the sharing coefficient of an individual to choose a future type of crossover or any other method that promotes convexity or non-convexity, instead of directly changing individual's fitnesses.

Acknowledging that mutation is also an important piece of the puzzle and can be perfectly described within the geometrical framework might also open creative strategic employments of this method that can also possess an adaptive property.

Studying the long-term effects<sup>1</sup> of this framework is also important as it will not just converge and waste the remaining pre-defined number of generations. Instead, it will keep pulsing out of those convergences as generations go by, making this approach interesting for problems that make use of extensive search and would otherwise need any type of reset, like a new random initialization of the algorithm.

Finally, cross-studies with other evolutionary algorithms that do not belong in the genetic algorithm category can prove beneficial to the overall geometrical unification mission that was introduced years ago.

---

<sup>1</sup>Regarding the number of generations in the search process.

## BIBLIOGRAPHY

- [1] J. Vervaeke, T. Lillicrap, and B. Richards. “Relevance Realization and the Emerging Framework in Cognitive Science”. In: *J. Log. Comput.* 22 (Feb. 2012), pp. 79–99. DOI: [10.1093/logcom/exp067](https://doi.org/10.1093/logcom/exp067) (cit. on p. 1).
- [2] D. Goldberg and J. Holland. *Genetic algorithms and machine learning*. 3 (2): 95–99. 1988 (cit. on p. 1).
- [3] C. Schaffer. “A Conservation Law for Generalization Performance”. In: *Machine Learning Proceedings 1994*. Ed. by W. W. Cohen and H. Hirsh. San Francisco (CA): Morgan Kaufmann, 1994, pp. 259–265. ISBN: 978-1-55860-335-6. DOI: <https://doi.org/10.1016/B978-1-55860-335-6.50039-8>. URL: <https://www.sciencedirect.com/science/article/pii/B9781558603356500398> (cit. on p. 2).
- [4] B. L. Miller, D. E. Goldberg, et al. “Genetic algorithms, tournament selection, and the effects of noise”. In: *Complex systems* 9.3 (1995), pp. 193–212 (cit. on p. 6).
- [5] R. Kumar et al. “Blending roulette wheel selection & rank selection in genetic algorithms”. In: *International Journal of Machine Learning and Computing* 2.4 (2012), pp. 365–370 (cit. on p. 6).
- [6] D. E. Goldberg and K. Deb. “A comparative analysis of selection schemes used in genetic algorithms”. In: *Foundations of genetic algorithms*. Vol. 1. Elsevier, 1991, pp. 69–93 (cit. on p. 6).
- [7] S. Mirjalili. “Genetic Algorithm”. In: *Evolutionary Algorithms and Neural Networks: Theory and Applications*. Cham: Springer International Publishing, 2019, pp. 43–55. ISBN: 978-3-319-93025-1. DOI: [10.1007/978-3-319-93025-1\\_4](https://doi.org/10.1007/978-3-319-93025-1_4). URL: [https://doi.org/10.1007/978-3-319-93025-1\\_4](https://doi.org/10.1007/978-3-319-93025-1_4) (cit. on p. 6).
- [8] A. Moraglio. “Towards a Geometric Unification of Evolutionary”. In: (2007) (cit. on pp. 7, 8, 12).

- [9] A. Moraglio and R. Poli. “Inbreeding properties of geometric crossover and non-geometric recombinations”. In: *International Workshop on Foundations of Genetic Algorithms*. Springer. 2007, pp. 1–14 (cit. on p. 9).
- [10] M. Srinivas and L. M. Patnaik. “Genetic algorithms: A survey”. In: *computer* 27.6 (1994), pp. 17–26 (cit. on p. 9).
- [11] A. Moraglio, J. Togelius, and S. Silva. “Geometric differential evolution for combinatorial and programs spaces”. In: *Evolutionary computation* 21.4 (2013), pp. 591–624 (cit. on p. 9).
- [12] F. Chicano et al. “Fitness probability distribution of bit-flip mutation”. In: *Evolutionary Computation* 23.2 (2015), pp. 217–248 (cit. on p. 11).
- [13] H. Shimodaira. “A diversity-control-oriented genetic algorithm (DCGA): performance in function optimization”. In: *Proceedings of the 2001 Congress on Evolutionary Computation (IEEE Cat. No.01TH8546)*. Vol. 1. 2001, 44–51 vol. 1. DOI: [10.1109/CEC.2001.934369](https://doi.org/10.1109/CEC.2001.934369) (cit. on pp. 14, 16, 21).
- [14] H. Shimodaira. “DCGA: a diversity control oriented genetic algorithm”. In: *Second International Conference On Genetic Algorithms In Engineering Systems: Innovations And Applications*. 1997, pp. 444–449. DOI: [10.1049/cp:19971221](https://doi.org/10.1049/cp:19971221) (cit. on p. 14).
- [15] D. E. Goldberg, J. Richardson, et al. “Genetic algorithms with sharing for multimodal function optimization”. In: *Genetic algorithms and their applications: Proceedings of the Second International Conference on Genetic Algorithms*. Vol. 4149. Hillsdale, NJ: Lawrence Erlbaum. 1987 (cit. on pp. 14, 16).
- [16] M. Castelli et al. “Salp Swarm Optimization: A critical review”. In: *Expert Systems with Applications* 189 (2022), p. 116029 (cit. on p. 14).
- [17] C. Jassadapakorn and P. Chongstitvatana. “Self-Adaptation Mechanism to Control the Diversity of the Population in Genetic Algorithm”. In: *CoRR* abs/1109.0085 (2011). arXiv: [1109.0085](https://arxiv.org/abs/1109.0085). URL: <http://arxiv.org/abs/1109.0085> (cit. on pp. 16, 17).
- [18] N. Awad et al. “Evaluation criteria for the CEC 2017 special session and competition on single objective real-parameter numerical optimization”. In: *Tech. Rep.* (2016) (cit. on p. 24).
- [19] R. A. Armstrong. “When to use the Bonferroni correction”. In: *Ophthalmic and Physiological Optics* 34.5 (2014), pp. 502–508 (cit. on p. 30).



# GENERAL ABF BOXPLOTS

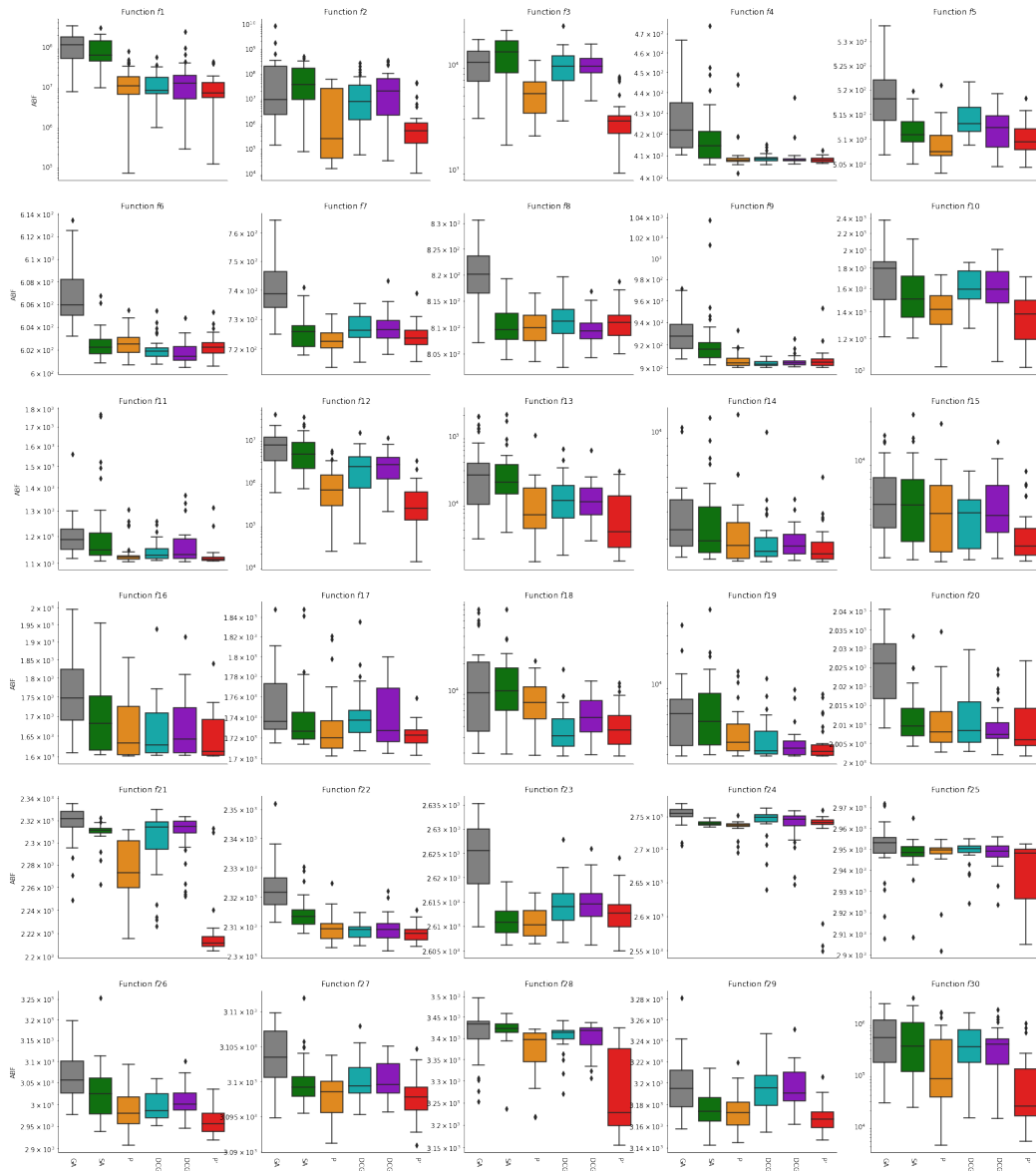


Figure I.1: Boxplots of the ABFs over each of the 30 functions with Dimension = 10.

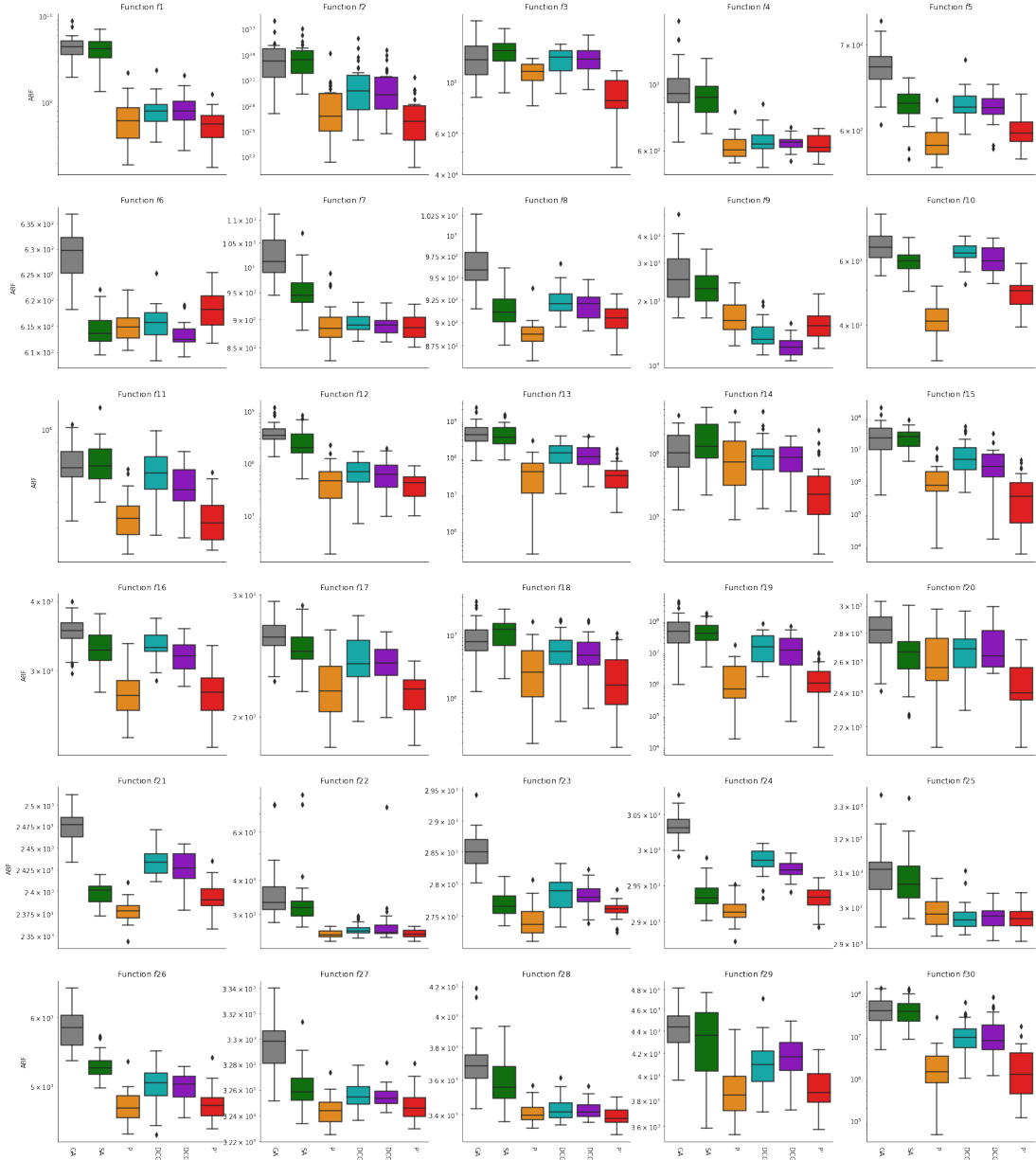


Figure I.2: Boxplots of the ABFs over each of the 30 functions with Dimension = 30.



