A Self-Adaptive Approach to Exploit Topological Properties of Different GAs' Crossover Operators

José Ferreira[1], Mauro Castelli[1], Luca Manzoni[2], and Gloria Pietropolli[2(✉)]

[1] NOVA Information Management School (NOVA IMS), Universidade Nova de Lisboa, Campus de Campolide, 1070-312 Lisboa, Portugal
{m20180422,mcastelli}@novaims.unl.pt

[2] Dipartimento di Matematica e Geoscienze, Università degli Studi di Trieste, Via Alfonso Valerio 12/1, 34127 Trieste, Italy
lmanzoni@units.it, gloria.pietropolli@phd.units.it

Abstract. Evolutionary algorithms (EAs) are a family of optimization algorithms inspired by the Darwinian theory of evolution, and Genetic Algorithm (GA) is a popular technique among EAs. Similar to other EAs, common limitations of GAs have geometrical origins, like premature convergence, where the final population's convex hull might not include the global optimum. Population diversity maintenance is a central idea to tackle this problem but is often performed through methods that constantly diminish the search space's area. This work presents a self-adaptive approach, where the non-geometric crossover is strategically employed with geometric crossover to maintain diversity from a geometrical/topological perspective. To evaluate the performance of the proposed method, the experimental phase compares it against well-known diversity maintenance methods over well-known benchmarks. Experimental results clearly demonstrate the suitability of the proposed self-adaptive approach and the possibility of applying it to different types of crossover and EAs.

1 Introduction

Evolutionary computation (EC) [2], a subfield of artificial intelligence, leverages computing power to model global optimization strategies that mimic natural evolution and can be applied to several domains [13,19,24]. Genetic algorithms [8] belong to the family of EC and handle a population of candidate solutions represented as a sequence of genes. GAs are stochastic optimizers, minimizing or maximizing an objective function while exploring the underlying search space. This space can be better understood considering a geometrical and topological view of the evolutionary process. In particular, candidate solutions can be described as points in a geometric space within a dynamical system (expressed as the whole GA search process) that changes these points as time (generations) goes through. To describe a candidate solution, one can characterize it genotypically by studying its genetic information or phenotypically by studying its fitness, meaning its ability to address the optimization problem at hand. The merging of these two dimensions establishes a fitness landscape in which all the possibilities within the genotypical domain, and consequent fitness outcomes, can be envisioned. Depending on the effect they produce in the underlying space, it is possible to identify two main categories of search operators used by a GA to act on a point/solution: (1) geometric operators (that result in a convex search), and (2) non-geometric operators (that result in a non-convex search) [16]. A convex search will contract the hypervolume of the hypercube that represents the fitness landscape observable by the search process, while a non-convex search has, usually, a non-zero probability of expanding it. In this context, it is crucial to understand the idea of a topology associated with a problem. Using an algorithm to solve a problem means essentially applying a strategy for searching that topology in a (hopefully) optimized fashion. Existing literature proposed different search operators in the context of GAs [11]. Each genetic operator produces a specific effect on the search process, and it is challenging to determine which operators are more effective in addressing a specific problem and which operator is more effective in a given phase of the search process. As a consequence, existing works proposed to dynamically modify the probability of using genetic operators based on some criteria [4,12,22]. In this work, we propose a self-adaptive approach to exploit the properties of geometric and non-geometric crossovers to achieve a more effective search. We expect this method can help overcome (or at least reduce) the problem of premature convergence of the population, one of the main limitations of GAs. In particular, we rely on a self-adaptive technique which, based on the current stage of the search process, decides whether to use the non-geometric crossover to (possibly) increase the population's convex hull. This work differs from the existing methods that, in the majority of the cases, simply modify the probabilities of crossover and mutation based on the status of the search process but without considering topological information concerning the genetic operators. In particular, by adapting the search operators used in GAs, we will leverage the continuous need to apply either geometric or non-geometric crossover in different phases of the search process characterized by specific space topology conditions [3].

The remaining part of the manuscript is organized as follows: Sect. 2 reviews some concepts concerning the geometric properties of genetic operators and convex search; Sect. 3 links this study to the existing literature; Sect. 4 presents the proposed self-adaptive approach; Sect. 5 outlines the experimental settings; Sect. 6 analyzes the results achieved, while Sect. 7 summarizes the main findings of the paper and suggests future research avenues.

2 Fundamental Concepts

This section presents important concepts and definitions related to the geometrical properties of genetic operators. To frame the discussion, it is essential to recall some ideas developed in the geometric framework that unified various EAs [15]. This framework analyzes the working principles of the genetic operators from a mathematical perspective.

2.1 Crossover

Let S be the space of all possible solutions and the image set $\mathrm{Im}[OP]$ the set of all possible offspring produced by a recombination operator OP with non-zero probability.

A recombination operator OP belongs to the geometric crossover class $\mathcal{G}$ [15] if there exists at least a distance d under which such a recombination is geometric:

$$OP \in \mathcal{G} \Longleftrightarrow \exists d : \forall p_1, p_2 \in S : \mathrm{Im}\left[OP\left(p_1, p_2\right)\right] \subseteq \left[p_1, p_2\right]_d .$$

On the other hand, a recombination operator OP belongs to the non-geometric crossover class $\overline{\mathcal{G}}$ if there is no distance d under which such a recombination is geometric:

$$OP \in \overline{\mathcal{G}} \Longleftrightarrow \forall d : \exists p_1, p_2 \in S : \mathrm{Im}\left[OP\left(p_1, p_2\right)\right] \setminus \left[p_1, p_2\right]_d \neq \emptyset.$$

The geometric crossover leads to the creation of offspring lying on the segment that connects the parent individuals in the space. As pointed out by Moraglio, there are three properties [17] that arise from using a geometric recombination operator:

- *Property of Homology.* It states that the recombination of one parent with itself can only produce the parent itself.
- *Property of Convergence.* It 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.
- *Property of Partition.* It states that 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, non-geometric.

In this paper, we use two recombination operators: *one-point crossover* (geometric) and *extension ray crossover* (non-geometric). Among the (several) existing crossover operators, we decided to rely on these simple operators for presenting the proposed self-adaptive approach. This choice mitigates the causal relationship between the use of more complex types of crossover operators and the results achieved in the experimental phase. One-point crossover [23] is a mask-based crossover for binary strings that produces offspring in the segment between the two parents. On the other hand, extension ray crossover [18] extends the segment passing through both parents, thus producing offspring outside this segment.

The other operator typically adopted in the GA framework is the mutation. Anyway, we will not discuss its property in further detail, as the study of this paper is focused on the crossover operator (we do not use the mutation in the search process).

2.2 Convex Combination, Convex Hull, and Convex Search

This section reports the concepts necessary to fully understand the role of geometric and non-geometric operators in the search process and discusses diversity maintenance strategies in convex search. A *convex combination* is a combination of vectors where all coefficients, their multiplicative factors, are non-negative and sum up to 1. A set S closed under convex combinations is a convex set. In this case, 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*.

Geometric crossover leads to convex outcomes and reduces the convex hull of the present generation's pairs of parents. In other words, the distance between two children will be smaller than the one between their parents. Intuitively, this formulates a concept of convex search given that executing selection and crossover multiple times over any number of generations will lead to a search space reduction [15]. Figure 1 illustrates a hypothetical spatial evolution from one generation to the subsequent, showing the modification of the global convex hull produced by the usage of selection and geometric crossover.

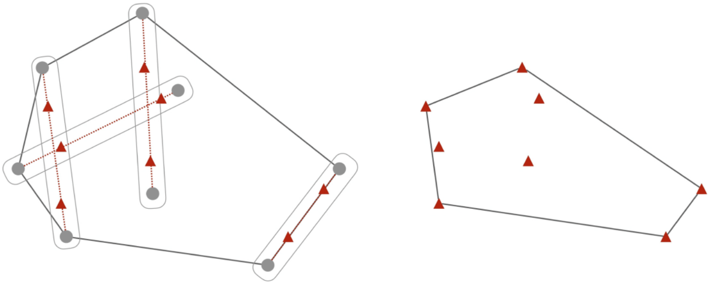

Fig. 1. Convex hull reduction through a generation. Gray dots are the vertex of the polytope which represents the solution space. Red triangles are the individuals created through geometric crossover. (Color figure online)

A fundamental problem researchers encounter while creating new search strategies is the one of premature convergence. It 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 the population's convex hull, these solutions are left out. Diversity maintenance strategies often focus on creating or maintaining distance between members of the population, and they represent an attempt to counteract premature convergence. However, these strategies will only have an effect relative to the continuous volume-decreasing convex polytope. As a consequence, there will be diminishing returns when it comes to these strategies. In particular, these methods (abstracting from the vast number of different implementations) artificially increase the chances of future genetic material propagation of specific individuals in the name of diversity. This is achieved by augmenting the fitness of genotypically remote individuals and/or by artificially reducing the fitness of genotypically close individuals in subpopulations. Phenotype-agnostic methods also exist, focusing only on promoting or demoting certain individuals with the help of genotypic distances. Each method (or a combination of methods) leads to the same outcome - the increase of targeted individuals' probabilities regarding mating/survival in the population. Cross-generational Probabilistic Survival Selection (CPSS) reduction [20,21] and fitness sharing [6] are examples of these strategies.

Examples of strategies that will disrupt this pattern of diminishing returns include: 1) the spawning of new random solutions, as they have a non-zero probability of landing outside the global population's convex hull, or 2) non-geometric crossover since it is a local non-convex transformation with a non-zero probability of creating offspring outside the global convex hull [9].

3 Related Works

Maintaining the diversity of individuals in the population received greater attention in recent years. In particular, a wide variety of methods to enhance diversity have been developed, and for a detailed review of these techniques, the reader can refer to [7]. In this section, we recall the most commonly-used methods, including Diversity Control oriented Genetic Algorithm (DCGA) and the Self-adaption Genetic Algorithm (SA). In the experimental phase, the performance of these techniques will be compared against our proposed method to assess its usefulness in improving the GA search process.

Fitness sharing is the most frequently used technique in literature. Here, population diversity is maintained via introducing a diversity function, which ensures the mitigation of unbridled head-to-head competition between widely disparate points in the solution space [6]. Another popular method is *deterministic crowding*, where the diversity issue is solved by forcing every offspring to compete with one of its parents and eventually replace it if the offspring is not worse [5]. Most existing diversity-maintenance mechanisms – as the two examples aforementioned – require problem-specific knowledge to set up specific parameters properly. A clear example is DCGA [21]. In particular, in this method, the selection criterion exploits the distance between a candidate individual and the best-performance individual and uses it (based on a probabilistic function) to produce a higher selection probability for a candidate solution with a larger distance. Hence, to ensure the effectiveness of the method, the probability function must be defined properly.

In [10], authors proposed SA to control the diversity of the population without explicit parameter setting. A self-adaptation mechanism is proposed: for controlling diversity, two measures are introduced: the *difference function*, which computes the degree of dissimilarity, and the *contribution*, which monitors the effect of the recombination.

4 Methodology

This section discusses the proposed method. Firstly, Sect. 4.1 provides details on how to perform diversity maintenance dynamically. Subsequently, Sect. 4.2 provides an in-depth description of how to couple such dynamic diversity maintenance with a combination of geometric and non-geometric crossovers.

4.1 Dynamic Diversity Maintenance

As stated above, one of the principal weaknesses of GA is the premature convergence to a solution, sometimes causing the consequent stuck of the algorithm in a local optimum. One of the recurrent ideas to overcome this issue is the maintenance of a certain level of diversity among candidate solutions. This work proposes a *self-adaptation mechanism* to control and guarantee diversity in the population and, simultaneously, to avoid the time-consuming task of setting specific hyperparameters of the GA.

As proposed in [10], a successful diversity maintenance strategy consists of looking at the population as a society divided in multiple groups. In this case, a *group* is represented by candidate solutions that share similar chromosomes. This approach promotes recombination between parents of different group and, in the meantime, disincentives crossover among individuals belonging to the same group. To formalize this concept, two quantities must be introduced: a *preference type*, which affects the characteristics of diversity in mating, and *contribution*, which measures the merit of each preference type. In this work, we rely on the same idea to determine the most suitable crossover operator to be used in the different generations of the search process.

Mating. The preference type τ is a parameter that indicates the preference of an individual to recombine with another one based on their degree of diversity. It is a positive quantity ranging between $0 \leq \tau \leq \tau_{max}$, where τ_{max} is the maximum preference type. Higher values of τ will lead, intuitively, to offspring which differ from the parents, thus encouraging diversity among members of the population.

τ is used to compute the *difference function* $\mathcal{D}$ as follows:

$$\mathcal{D}(\tau, d_i) = 0.5 + \frac{\tau}{\tau_{max}}(d_i - 0.5) \tag{1}$$

where d_i is the difference between the first selected individual x_1 and a candidate mate y_i, calculated as follows:

$$d_i = \frac{h(y_i, x_1)}{\ell} \tag{2}$$

where h is the Hamming distance between two individuals, and ℓ is the length of a chromosome.

At this point, we have all the ingredients to define how the mating is performed (i.e., for a maximization problem). Once the first individual x_1 is provided, then its recombination mate x_2 is selected in the following way:

$$x_2 = \arg\max_{i \in s_t}[f(y_i)\mathcal{D}(\tau, d_i)] \tag{3}$$

where f is the fitness function and s_t is the tournament size. Thus, a candidate who has a higher value of $\mathcal{D}$ has more chance to be selected as the second parent. Let us remark that when $\tau = 0$, the probability of selection does not depend on d, and the mating is just a fitness-based selection.

Diversity Control. Considering that the degree of diversity of the population is controlled by the preference type, it is necessary to define a procedure that correctly updates the value of τ according to the population's needs. The idea is to associate a parameter at each possible value of τ, called *contribution*, which quantifies how solutions with a given preference type produce better quality offspring. Contribution depends on t (the training epoch) and τ, and it is defined as the ratio between successful and total crossover:

$$Contribution(\tau, t) = \frac{\#SuccCross(\tau, t)}{\#Cross(\tau, t)} \tag{4}$$

At generation $t + 1$, the probability of choosing τ will be equal to its contribution at generation t. Thus, the more a preference type is associated with the creation of good-quality individuals, the higher will be the probability for it to be reused. In this work, we defined a crossover to be successful if produces at least an offspring with fitness equal to or better than both parents. This is different with respect to the approach described in [10], where authors defined as positive crossover a recombination in which the fitness of the resulting offspring is superior to the one of both the parents. In fact, in the (rare) scenario where the offspring and its best parent share the same fitness value and do not coincide, the introduction of a new solution may produce a positive effect on the search process. An example is the presence of a plateau – a part of the space in which all points have the same fitness score – in the fitness landscape.

More precisely, the diversity control procedure works as follows:

1. Randomly generates the initial population of individuals and evaluates their fitness.
2. Initialize the contribution equally for each preference type.
3. Select an individual and its partner with the aforementioned mating procedure. Recall that the probability of choosing τ is equal to its contribution.
4. Create two new individuals by crossover and evaluate their fitness.
5. Repeat step **4** and **5** for the whole population.
6. Compare the fitness of the offspring with the one of their parents. Update the contribution values consequently.
7. Repeat step **3**–**6** until a termination criterion is satisfied.

In this work, to maintain all the preference type values throughout the search process, we impose a minimum threshold of 10% for each contribution.

4.2 Self-adaptive Crossover

In Sect. 2.1, we introduced two different definitions for crossover:

- *Geometric crossover*, which, if employed alone, decreases the size of the population convex hull, thus being a diversity reducer.
- *Non-geometric crossover*, which has the ability to behave as a diversity enhancer method.

By applying the mating routine described in Sect. 4.1, our objective is to obtain a method that allows us to self-adapt the choice of crossover operators in the GA algorithm depending on the specific stage of the search process. The idea is that most recombination will still be geometric, but occasionally non-geometric crossover will be applied to avoid the situation of premature convergence where the global optima may not be contained in the convex hull. Intuitively, this combination can be described as a phenomenon of *conspansion*, i.e., material contraction during space expansion. Contraction is a consequence of the geometric crossover, while expansion is a consequence of the non-geometric one.

As mentioned above, the procedure described in Sect. 4.1 can be adequately modified for the selection of the most performing type of crossover. Specifically, non-geometric crossover will be chosen if the parent x_1 has preference type $\tau = \tau_{max}$, otherwise geometric one will be used. In fact, a high preference type indicates the need to augment the diversity in the population. Therefore, when τ assumes the highest possible value, the crossover technique that is a diversity enhancer must be selected.

To summarize, when the convex search starts leading to negative effects on population phenotype, the contribution parameter values associated with preference types related to geometric crossover start to decrease. Non-convex search, as a consequence, will be selected with more probability, as the share of contribution of the preference type τ_{max} (i.e., the one linked to non-geometric crossover) will increase. On the other hand, if the non-geometric crossover causes a worsening in the individuals' fitness, the geometric crossover will be preferred by reducing non-geometric crossover contribution.

Let us emphasize the fact that the expansion of the global convex hull is not ensured at each step. Firstly, as the choice of which crossover to use is non-deterministic, it can simply not occur during a generation. Secondly – and more important – the individuals produced by non-geometric crossover are not necessarily outside the global convex hull.

We propose and investigate two variants for the self-adaptive crossover introduced, namely *P* and *P'*.

P'

1. Tournament selection, size= 3
2. Eliminate tournament winner x_1 from population
3. Difference function tournament, size=length of the population
4. Eliminate winner x_2 from population
5. Return children (y_1, y_2)

P

1. Tournament selection, size= 3
2. Difference function tournament, size= 3
3. Return children (y_1, y_2)

The main difference between these two variants lies in the selection technique. In particular, we want to study how different approaches in the choice of the second parent affect the algorithm. In fact, once the first parent x_1 is fixed, P randomly selects only a limited set of candidates and computes the difference function over this set. On the other hand, P' computes the difference function for all the individuals of the population to find the perfect fit x_2.

5 Experimental Settings

Table 1. Definitions and optimum values (minimum) of the CEC 2017 benchmark functions.

	No.	Functions	Opt.
Unimodal functions	1	Shifted and Rotated Bent Cigar	100
	2	Shifted and Rotated Sum of Different Power	200
	3	Shifted and Rotated Zakharov	300
Simple multimodal functions	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
Hybrid functions	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
Composition functions	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

The set of functions used, described in Table 1, is the *CEC 2017* function suite [1] for single-objective real-parameter numerical optimization. The suite is composed of unimodal, multi-modal, hybrid, and composition functions that are shifted, rotated, and non-separable. Their characteristics of noise and ruggedness make them excellent candidates for studying the effectiveness of the proposed approach, as they require different degrees of diversity in the population to be solved efficiently. The search space is $[-100, 100]^D$. $D = 10, 30$ are investigated in this experimental phase.

Table 2. Experimental settings. All the values of the hyperparameters coincide for $D = 10, 30$, except for the length of the chromosome, which is 200 in the former case and 600 in the second one.

Parameter	*Value*
Population size	400
Length of chromosome	{200,600} bits
Number of generation	200
Number of independent run	30
Crossover probability (Pc)	100%
Mutation rate (Pm)	0%
Tournament size	3

To assess the performance of the proposed method, we considered a GA with a population size equal to 400 and a search process that runs for 200 generations. Thus, the total number of fitness evaluations for each experiment is equal to $MaxFES = 80000$, i.e., the product between these two quantities. In this experimental phase, we decided to concentrate our attention only on crossover operators, whereas mutation is not allowed, to fully understand how the introduction of non-geometric crossover can impact the overall performance of the algorithm. As the algorithm is stochastic, 30 runs have been performed for each benchmark function. Further details concerning the implementation of the GA are reported in Table 2.

The results obtained by the two variants *P* and *P'* of the self-adaptive crossover are compared with:

- The vanilla GA.
- Two variants of DCGA: DCGA1 and DCGA2. In particular, we considered the following DCGA parameters [21] values: for DCGA1, $c = 0.01$, and $a = 0.19$, while for DCGA2, $c = 0.234$, $a = 0.5$.
- SA, the self-adaptive GA outlined in Sect. 4.1.

This is a relatively broad group of techniques as it considers vanilla GA, variants of DCGA (which uses a static strategy to maintain diversity), and finally self-adaptive algorithms: SA (where the crossover operator is fixed) and *P* and *P'* (where there is a choice between geometric and non-geometric operators).

Regarding the self-adaption mechanism, we use 4 – $(\tau \in 0, \ldots, 3)$ – preference levels both for SA, *P*, and *P'* (Figs. 4 and 5).

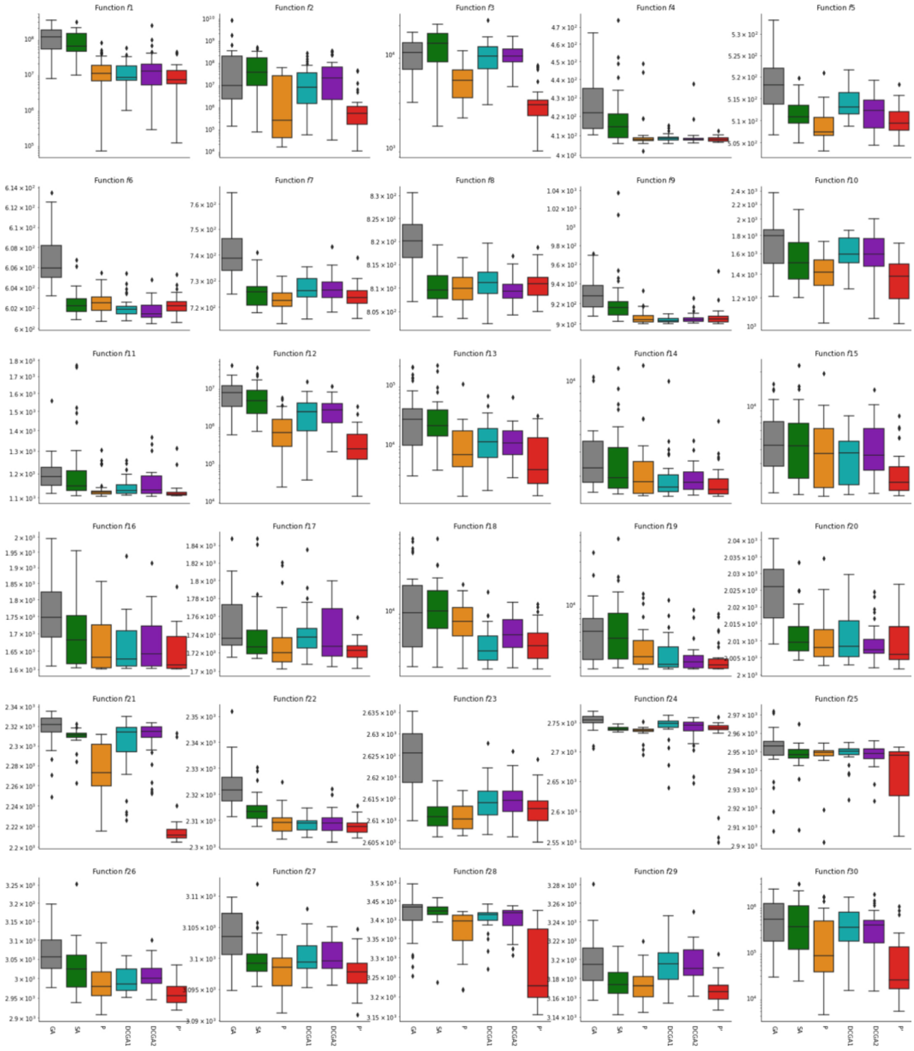

Fig. 2. Median of the fitness over the 30 independent runs for the considered benchmark problems – $D = 10$.

6 Experimental Results

As stated in Sect. 4.2, the goal of this study is to compare the performance of our algorithm against a wide range of well-known GA-based methods. The experimental results, computed over 30 independent runs, are reported through box-plot for $D \in \{10, 30\}$ in Fig. 2 and Fig. 3. Experimental results show that the proposed method generally outperforms the other competitors in the vast majority of the benchmark functions. Specifically, at least one algorithm between

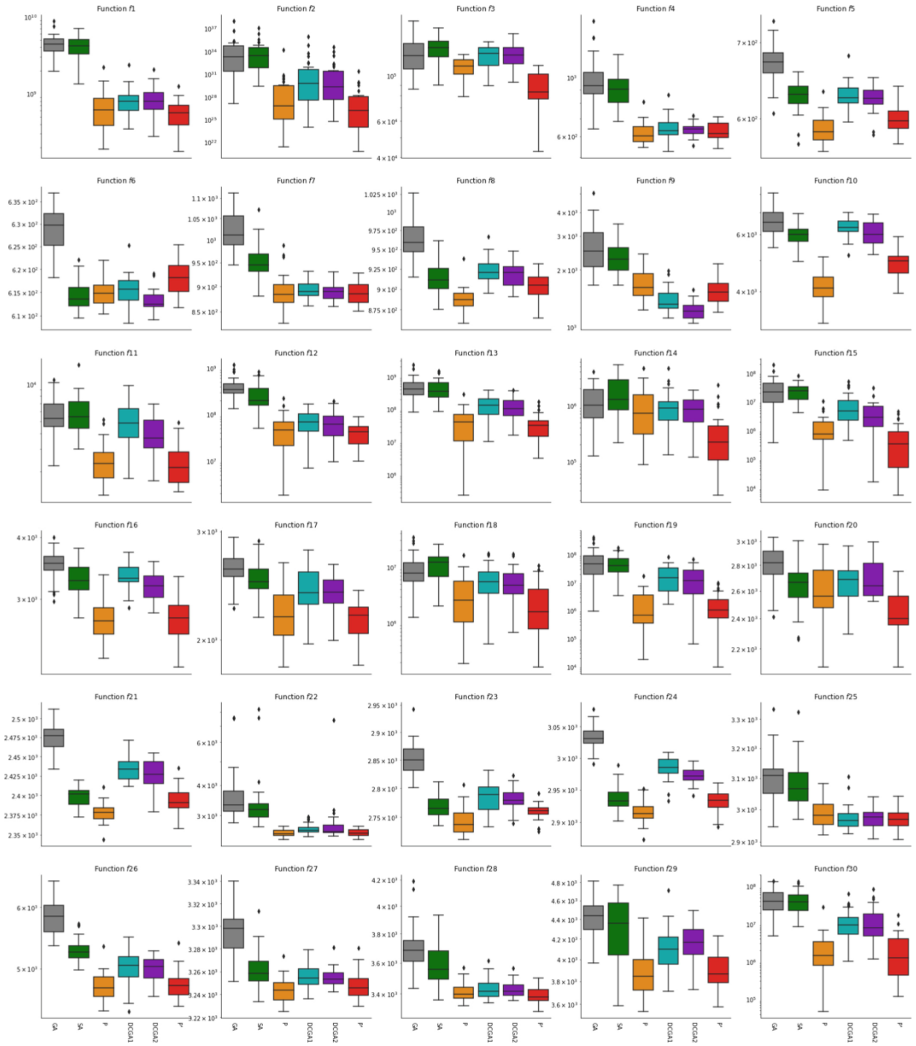

Fig. 3. Median of the fitness over the 30 independent runs for the considered benchmark problems – $D = 30$

P and *P'* leads to an improvement of the fitness in 25 functions of the 10-th dimensional case and in 27 functions of the 30-th dimensional one. The fitness gap between our techniques and the other methods taken into account increases together with the dimension of the problems, suggesting that our algorithm is particularly suitable for solving challenging optimization problems in higher dimensions. On the other hand, when the results of *P* and *P'* are compared, we obtain that for $D = 10$ *P'* seems to achieve better fitness values, while for $D = 30$ the two methods show comparable performances.

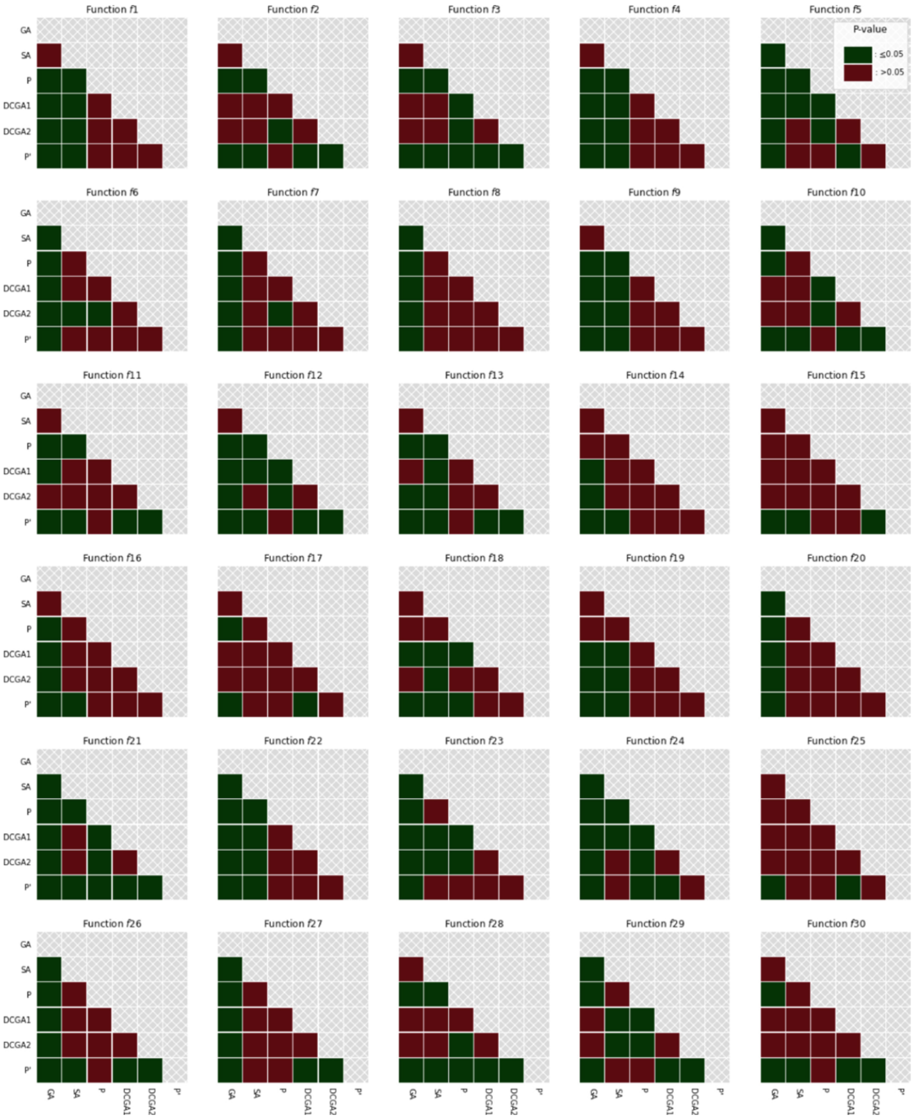

Fig. 4. P-values returned by the Mann-Whitney U test with the Bonferroni correction for each of the 30 functions ($D = 10$). Green denotes a p-value for which the alternative hypothesis cannot be rejected. Red denotes a p-value for which the null hypothesis (i.e., equal median) cannot be rejected. (Color figure online)

To investigate whether our method significantly outperforms the others, the Mann-Whitney U statistical test (computed considering a significance level $\alpha = 0.05$ and the Bonferroni correction [14]) results are displayed in Fig. 2 for $D = 10$, and in Fig. 3 for $D = 30$. Based on these results, it is possible to confirm that the

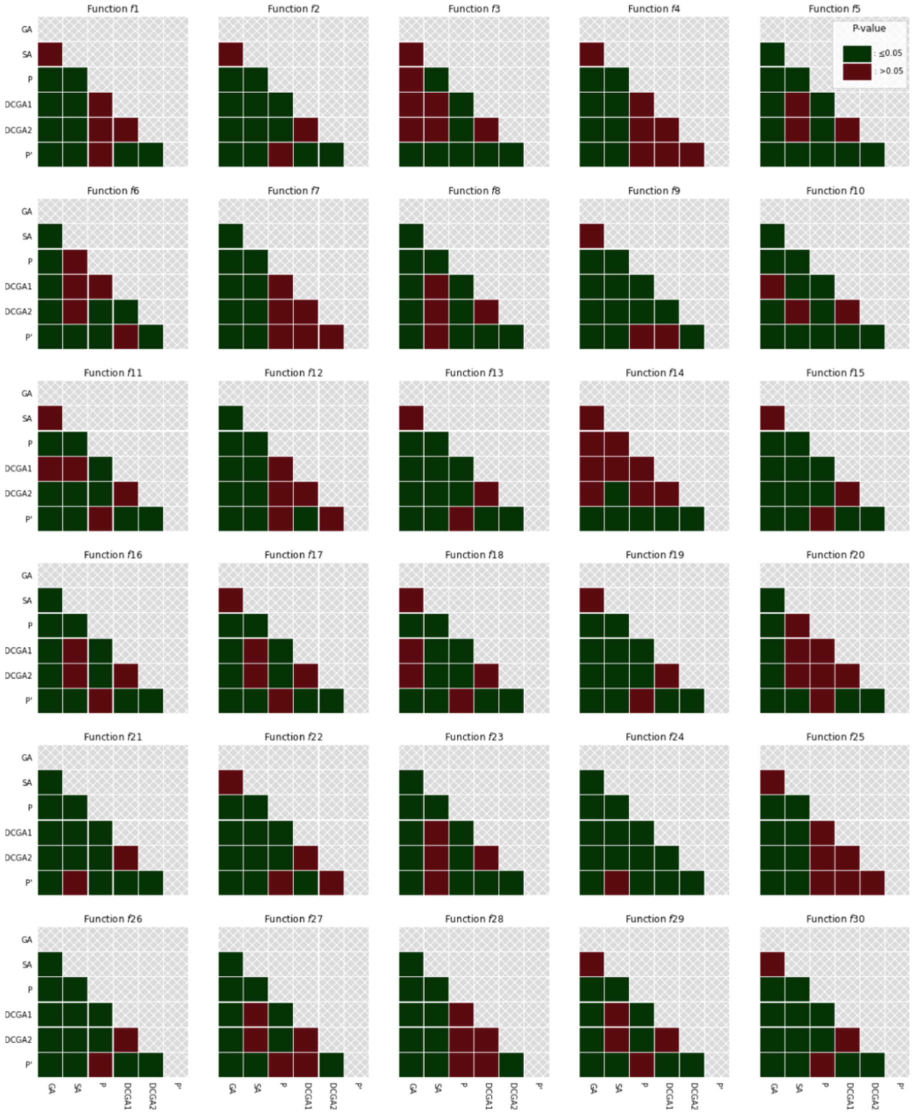

Fig. 5. P-values returned by the Mann-Whitney U test with the Bonferroni correction for each of the 30 functions ($D = 30$). Green denotes a p-value for which the alternative hypothesis cannot be rejected. Red denotes a p-value for which the null hypothesis (i.e., equal median) cannot be rejected. (Color figure online)

proposed technique can produce better performance with respect to the other competitors – and with a statistical significance – in the vast majority of the considered benchmarks.

Again, Mann-Whitney U statistical tests confirm that we obtain better results in $D = 30$ w.r.t. $D = 10$, confirming the hypothesis that the proposed self-adaptive method improves the search for the optimal value when the dimension – and thus, the difficulty of the problem – increases.

7 Conclusions

GAs are a popular technique in the EAs family. Despite their success in different domains, they suffer from a premature convergence problem, where the final population's convex hull might not include the global optimum. Population diversity maintenance strategies are fundamental to counteract this problem, but the typical GA convex search still reduces the population's convex hull over the generations. Thus, the idea of this work is the definition of a self-adaptive method for counteracting the reduction of the convex hull produced by the application of geometric crossover and, at the same time, preserving population diversity. The paper proposed a self-adaptive method for using geometric and non-geometric crossover operators in different stages of the search process based on the information provided by the current candidate solutions and accommodating the ever-evolving necessities of the underlying search space/problem topology. To assess the performance of the proposed approach, an extensive experimental phase was performed considering the CEC 2017 benchmark suite and comparing our proposal against the vanilla GA and popular diversity maintenance techniques. Experimental results clearly show the superior performance of the proposed method and the advantage provided by using both geometric and non-geometric crossover in the search process. In the future, we plan to test the proposed approach with different EAs and crossover operators.

References

1. Awad, N., Ali, M., Liang, J., Qu, B., Suganthan, P., Definitions, P.: Evaluation criteria for the cec 2017 special session and competition on single objective real-parameter numerical optimization. Technical report (2016)
2. Bäck, T., Fogel, D.B., Michalewicz, Z.: Handbook of evolutionary computation. Release **97**(1), B1 (1997)
3. Castelli, M., Manzoni, L., Gonçalves, I., Vanneschi, L., Trujillo, L., Silva, S.: An analysis of geometric semantic crossover: a computational geometry approach. In: IJCCI (ECTA), pp. 201–208 (2016)
4. Castelli, M., Manzoni, L., Vanneschi, L., Silva, S., Popovič, A.: Self-tuning geometric semantic genetic programming. Genet. Program. Evolvable Mach. **17**(1), 55–74 (2016)
5. Friedrich, T., Oliveto, P.S., Sudholt, D., Witt, C.: Analysis of diversity-preserving mechanisms for global exploration. Evol. Comput. **17**(4), 455–476 (2009)
6. Goldberg, D.E., Richardson, J., 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. Lawrence Erlbaum, Hillsdale, NJ (1987)

7. Gupta, D., Ghafir, S.: An overview of methods maintaining diversity in genetic algorithms. Int. J. Emerg. Technol. Adv. Eng. **2**(5), 56–60 (2012)
8. Holland, J.H.: Genetic algorithms. Sci. Am. **267**(1), 66–73 (1992)
9. Ishibuchi, H., Tsukamoto, N., Nojima, Y.: Diversity improvement by non-geometric binary crossover in evolutionary multiobjective optimization. IEEE Trans. Evol. Comput. **14**(6), 985–998 (2010)
10. Jassadapakorn, C., Chongstitvatana, P.: Self-adaptation mechanism to control the diversity of the population in genetic algorithm. arXiv preprint arXiv:1109.0085 (2011)
11. Lim, S.M., Sultan, A.B.M., Sulaiman, M.N., Mustapha, A., Leong, K.Y.: Crossover and mutation operators of genetic algorithms. Int. J. Mach. Learn. Comput. **7**(1), 9–12 (2017)
12. Lin, W.Y., Lee, W.Y., Hong, T.P.: Adapting crossover and mutation rates in genetic algorithms. J. Inf. Sci. Eng. **19**(5), 889–903 (2003)
13. Maan, V., Malik, A.: Genetic algorithm application on 3D pipe routing: a review. Recent Innov. Comput. 139–148 (2022)
14. McKnight, P.E., Najab, J.: Mann-Whitney u test. Corsini Encycl. Psychol. 1 (2010)
15. Moraglio, A.: Towards a geometric unification of evolutionary algorithms. Ph.D. thesis, Department of Computer Science, University of Essex, UK (2007)
16. Moraglio, A.: Abstract convex evolutionary search. In: Proceedings of the 11th Workshop Proceedings on Foundations of Genetic Algorithms, pp. 151–162 (2011)
17. Moraglio, A., Poli, R.: Inbreeding properties of geometric crossover and non-geometric recombinations. In: Stephens, C.R., Toussaint, M., Whitley, D., Stadler, P.F. (eds.) FOGA 2007. LNCS, vol. 4436, pp. 1–14. Springer, Heidelberg (2007). https://doi.org/10.1007/978-3-540-73482-6_1
18. Moraglio, A., Togelius, J., Silva, S.: Geometric differential evolution for combinatorial and programs spaces. Evol. Comput. **21**(4), 591–624 (2013)
19. Sharma, S., Kumar, V.: Application of genetic algorithms in healthcare: a review. Next Generation Healthcare Informatics, pp. 75–86 (2022)
20. Shimodaira, H.: DCGA: a diversity control oriented genetic algorithm. In: Second International Conference On Genetic Algorithms in Engineering Systems: Innovations and Applications, pp. 444–449 (1997). https://doi.org/10.1049/cp:19971221
21. Shimodaira, H.: 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, pp. 44–51 (2001). https://doi.org/10.1109/CEC.2001.934369
22. Srinivas, M., Patnaik, L.M.: Adaptive probabilities of crossover and mutation in genetic algorithms. IEEE Trans. Syst. Man Cybern. **24**(4), 656–667 (1994)
23. Srinivas, M., Patnaik, L.M.: Genetic algorithms: a survey. Computer **27**(6), 17–26 (1994)
24. Vanneschi, L., Henriques, R., Castelli, M.: Multi-objective genetic algorithm with variable neighbourhood search for the electoral redistricting problem. Swarm Evol. Comput. **36**, 37–51 (2017)