

Article

An Improved Local Search Genetic Algorithm with a New Mapped Adaptive Operator Applied to Pseudo-Coloring Problem [†]

Monique Simplicio Viana ^{1,*}, Orides Morandin Junior ¹ and Rodrigo Colnago Contreras ²

¹ Department of Computing, Federal University of São Carlos, São Carlos SP 13565-905, Brazil; orides@ufscar.br

² Department of Applied Mathematics and Statistics, University of São Paulo, São Carlos SP 13566-590, Brazil; contreras@usp.br

* Correspondence: monique.viana@ufscar.br

[†] This paper is an extended version of our preliminary work “Contreras, et al. 2020”.

Received: 29 August 2020; Accepted: 30 September 2020; Published: 14 October 2020



Abstract: In many situations, an expert must visually analyze an image arranged in grey levels. However, the human eye has strong difficulty in detecting details in this type of image, making it necessary to use artificial coloring techniques. The pseudo-coloring problem (PsCP) consists of assigning to a grey-level image, pre-segmented in K sub-regions, a set of K colors that are as dissimilar as possible. This problem is part of the well-known class of NP-Hard problems and, therefore, does not present an exact solution for all instances. Thus, meta-heuristics has been widely used to overcome this problem. In particular, genetic algorithm (GA) is one of those techniques that stands out in the literature and has already been used in PsCP. In this work, we present a new method that consists of an improvement of the GA specialized in solving the PsCP. In addition, we propose the addition of local search operators and rules for adapting parameters based on symmetric mapping functions to avoid common problems in this type of technique such as premature convergence and inadequate exploration in the search space. Our method is evaluated in three different case studies: the first consisting of the pseudo-colorization of real-world images on the RGB color space; the second consisting of the pseudo-colorization in RGB color space considering synthetic and abstract images in which its sub-regions are fully-connected; and the third consisting of the pseudo-colorization in the Munsell atlas color set. In all scenarios, our method is compared with other state-of-the-art techniques and presents superior results. Specifically, the use of mapped automatic adjustment operators proved to be powerful in boosting the proposed meta-heuristic to obtain more robust results in all evaluated instances of PsCP in all the considered case studies.

Keywords: adaptive operator; genetic algorithm; local search; pseudo-coloring problem; visualization

1. Introduction

There are situations of visual data analysis in which an expert must extract complex considerations from images originally arranged in gray levels. In these situations, it is common to use the most diverse tools, but the visual analysis is almost always present. For example, we can highlight some recent challenges to detecting edges in sonar images [1], to detecting weapons in baggage [2], to visually analyze medical images [3–7], to conduct visual routines in remote sensing [8,9], to detect crashes in wire ropes [10], among other situations. Still, the human eye is not able to detect many details in an image colored with different shades of the same color [11,12] because the human brain has difficulty in the task of recognizing signals defined by similar frequencies. To solve this issue, one of the techniques

most commonly used in specialized literature is the artificial attribution of colors in order to obtain a visual enhancement in certain regions of the image.

In this case, it is necessary to define a set of colors for different regions of the original grey-level image, so that these colors are as visually dissimilar as possible and make the task of visually differentiating neighboring regions in an image easier. This problem is known in the specialized literature as pseudo-coloring problem (PsCP), or as the definition of color sets of high visual contrast [13]. This is a problem in which it is necessary to establish a finite set of elements belonging to a given color space, such that these elements are as far distant as possible according to some measure of distance. In this way, we can see that PsCP is an optimization problem. In addition, if the set of colors to be calculated is defined by a few elements, then PsCP can be solved exactly by greedy algorithms; however, if it is necessary to define a reasonably large amount of colors, the problem may not be solved. This is because PsCP is a combinatorial problem of the well-known class of NP-Hard problems. For this reason, many authors have proposed to solve it with the use of inexact methodologies; that is, with the use of meta-heuristics. As an example, we can highlight the work of Radlak and Smolka [14], who solve PsCP with the use of a GA. However, it is known that this meta-heuristic can have difficulty in solving complex combinatorial problems since it is common to be stuck in local optima and present premature convergence [15].

In the literature, we find strategies that can be used in GA to make it more efficient and less susceptible to the difficulties mentioned. For example, Asadzadeh [16] proposed the use of local search strategies across all standard GA operators. In addition, the author also defined an operator specialized in massive local search, the “elite local search procedure”. The methodology is used in the job shop scheduling problem, which is also an NP-Hard combinatorial problem. These strategies proved to be effective and surpassed the other evolutionary techniques considered. Therefore, one of the proposals of our work is to use these strategies in GA and adapt them to be able to solve PsCP. In addition, we also propose to add in GA a step of auto adjustment of parameters [17,18], since the use of this type of technique in the meta-heuristic in question proved to be useful when used to solve other NP-Hard problems [19,20].

This paper is an extended version of our preliminary work [21]. In this text, we generalize the adaptive rules of our technique using mapping functions that automatically update the parameterization of the meta-heuristic. We also added a section of related works that address the different types of pseudo-colorization in the specialized literature and a section that details the methodology followed for the resolution of PsCP by the proposed method. This text is richer in detail since we add descriptions of all the routines of operators of the method through flowcharts and algorithms with comments. The computational experiments were also updated with more comparisons taking into account more recent works.

The paper is organized into 7 sections. Specifically, in Section 2, we discuss the different types of approaches that involve pseudo-colorization methods available in the current literature. In Section 3, we describe the mathematical formulation of PsCP. In Section 4, we present the methodology used in this work to solve the PsCP. The formulation of the proposed algorithm and the details of all its operators are presented in Section 5. The proposed method is evaluated in three test scenarios and the experimental results obtained are presented in Section 6. The manuscript ends with conclusions and proposals for future work in Section 7.

2. Related Works

Associating different colors for partitions of an image is a problem from the well-known four-color theorem [22], which establishes that, using four distinct colors, there is at least one possible color configuration in which the vertices of any planar graph can be colored so that no pair of neighboring vertices are colored with the same color. From this result, the graph coloring theory was created, which concentrates around 200 problems still unsolved [23]. One of these problems is the

pseudo-coloring problem (PsCP), which belongs to the class of NP-Hard problems [24] and that we focus on this work.

PsCP can also be interpreted to keep semantic information during the association of colors in regions of images. In this case, it is considered that the objective of the problem is to convert an image available only in grey-level to a color version so that the color associations make sense and preserve the original meaning of the represented components in the image [25]. One example in this research field is [26], where the PsCP was addressed with the use of an architecture of multiple Generative Adversarial Networks (GANs). The method, called Multi-GANs, has obtained results with acceptable colors, presenting more realistic characteristics and meaning for the images generated. On the other hand, Fang, et al. [27] addresses the problem of image colorization by exploring measures of self-similarity and consistency constraints in the resolution of superpixels. More recently, Wan, et al. [28] proposed a fully automatic method using neural networks and optimization to treat image colorization. For a more comprehensive overview, we recommend the work of Li, et al. [29], who present the most updated and deep review in this specific field.

In this work, our advances are contained in a sub-area of PsCP known as color coding [30]. In this type of research, the problem of associating artificial colors to sub-regions of a grey-level image is a problem of combinatorial optimization defined in a graph [31], in which it is necessary to establish a color for each sub-region considering a fixed set of colors and in such a way that the colors chosen are as visually dissimilar as possible [32]. In other words, the objective of these problems is to determine a high contrast set of colors [33]. In this field, one of the most recognized works is the paper by Carter and Carter [13], which was the first work to propose a heuristic search that aimed to explore the RGB color space for a set of colors that was as dissimilar as possible taking into account distances in CIELUV space. In this case, the algorithm starts with several points in RGB, defined through the Grassman's laws [34], and in each iteration, one of these points was moved away from the others as much as possible, so that, at the end of the method, all points were spread out in space quasi-optimally. Glasbey, et al. [35] show an extension to this heuristic formulated by sequential sets. In detail, the authors present a method of greedy behavior that begins with a set formed only by the white color and to this set is added the most dissimilar color available in the entire color space, giving rise to another set of two colors and so on until the set has a color for each sub-region to be colored. In addition, to determine high contrast sets and taking into account multiple illuminant configurations, Bianco and Citrolo [36] proposed a heuristic that performs local searches in the neighborhood of possible PsCP solutions until no configuration that is better than the previous configuration is found. The authors compare the performance of their technique with two basic meta-heuristics also proposed in their work: simulated annealing (SA) [37] and genetic algorithm (GA) [38]. A generalization of this local search heuristic is proposed in the form of an unsupervised high-contrast coloring (HCC) algorithm [39].

Bioinspired meta-heuristics have also achieved advances in PsCP by determining high contrast sets of colors. For example, Radlak and Smolka [14] proposes a GA that performs searches in the RGB space in order to artificially color a pre-segmented grey-level image. In detail, the authors divide the population of the algorithm into three sub-populations: one specific for the component R; another for component G; and another for component B. In the same work, a new quantitative measure of performance comparison is also proposed, which is described in details in the next sections of our work. More recently, we proposed in our preliminary work [21] a new GA with local search and adaptive operators specialized in solving the PsCP. In this text, we present the expansion of this technique, in which the adaptive operator makes use of mapping functions to update the mutation and crossover rates during the method iterations.

In the next section, we present in detail the mathematical formulation of PsCP treated in this work.

3. Mathematical Formulation of Pseudo-Coloring Problem

To define PsCP, we will use in this work a model similar to that used by Radlak and Smolka [14]. For this, we will consider that we have as input a grey-level image I , pre-segmented in K disjunct sub-regions, and we will obtain as output a set of K colors that are as visually dissimilar as possible in neighboring regions. Mathematically, let I_1, I_2, \dots, I_K be the pre-segmented sub-regions of I , and let $\Delta = (\delta_{i,j})_{i,j=1}^K$ be the neighborhood matrix representing the connections between those sub-regions whose coordinates $\delta_{i,j}$ are defined as follows in the Equation (1):

$$\delta_{i,j} = \begin{cases} 1, & \text{if } I_i \text{ is neighbor of } I_j \text{ in } I, \\ 0, & \text{if } I_i \text{ is not neighbor of } I_j \text{ in } I \text{ or if } i = j. \end{cases} \quad (1)$$

In other words, the neighborhood matrix Δ describes the arrangement of all sub-regions in I and can be interpreted as the adjacency matrix of a graph that represents the connections between the sub-regions. This information is very important to solve the PsCP, since sub-regions that are neighbors must be colored with the most different colors possible, while sub-regions that are not connected can be colored with similar colors without much loss to the visualization. In Figure 1, we present a schematic with a grey-level image I pre-segmented in $K = 4$ sub-regions, the neighborhood matrix Δ defined by the positioning of these sub-regions and the graph that this matrix defines.

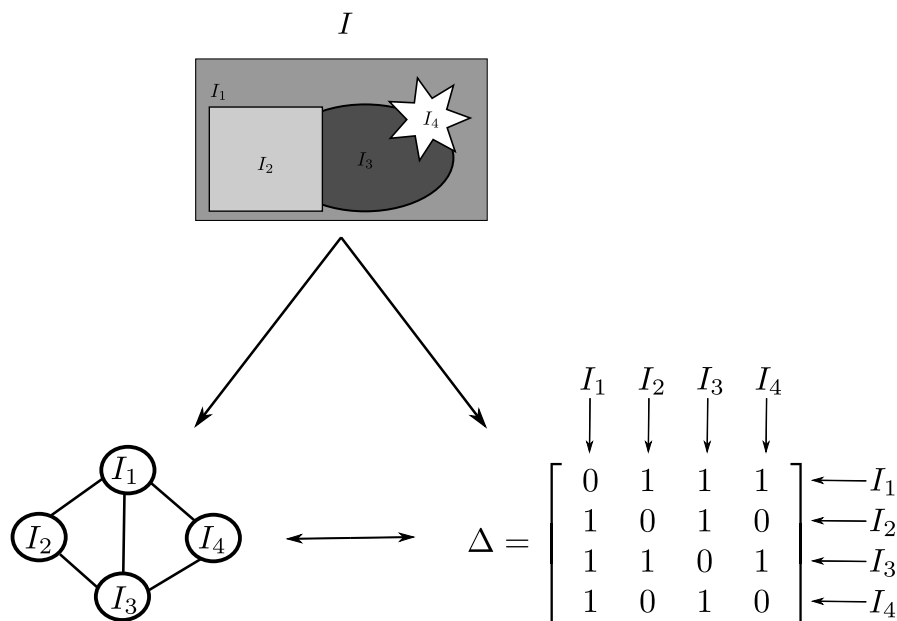


Figure 1. A grey-level image I ; its sub-regions I_1, I_2, I_3 and I_4 ; the graph representing the connections between the sub-regions; and its associated neighborhood matrix Δ .

Thus, to solve the PsCP, we have to determine a set of K colors $\mathcal{C} = \{c_1, c_2, \dots, c_K\}$ to pseudo-coloring all the sub-regions of I , with neighborhood matrix $\Delta = (\delta_{i,j})_{i,j=1}^K$, such that the value of the function F_Δ of Equation (2), which can be interpreted as the lowest distance between the colors of all neighbor sub-regions of I , is the greatest possible value, configuring this version of the PsCP in a max–min optimization problem.

$$F_\Delta(\mathcal{C}) := \min \{d(c_i, c_j) \mid \delta_{i,j} \neq 0, \forall c_i, c_j \in \mathcal{C}\}, \quad (2)$$

in which $d(\cdot, \cdot)$ is a distance function in some color space. Even though $d(\cdot, \cdot)$ can be defined as any distance between two given colors, it is common to define the distance function $d(\cdot, \cdot)$ to be the Euclidean distance between colors in the perceptually uniform color space CIELab (Commission Internationale de l'Eclairage) [40] with illuminant D50, since this space is favorable for performing

tasks of visual distinction of colors [41]. In this case, mathematically, the distance measurement between two colors c_i and c_j is defined in Equation (3).

$$d : \mathbb{Y} \times \mathbb{Y} \rightarrow \mathbb{R}_+ \\ (c_i, c_j) \mapsto \sqrt{(L_i - L_j)^2 + (a_i - a_j)^2 + (b_i - b_j)^2}, \quad (3)$$

in which \mathbb{Y} is some color space, and (L_i, a_i, b_i) and (L_j, a_j, b_j) are the CIELab components of the colors c_i and c_j , respectively.

We are defining the PsCP without any specification regarding the color space since our methodology does not restrict the F_Δ domain. In this way, each color c_i of the problem can be interpreted as belonging to any color space. For example, if the color space considered is the $\mathbb{Y} = \text{sRGB}$ space, then each color c_i must be represented by the three components R_i, G_i and B_i belonging to the set $\{0, 1, 2, \dots, 255\}$, so whereas these three components represent, respectively, the intensity of red, the intensity of green and the intensity of blue of the color c_i . Therefore, the sub-region I_i must be colored with the color (R_i, G_i, B_i) . As a consequence of this, the set of all K colors \mathcal{C} would be a subset of $\{0, 1, \dots, 255\}^{3 \cdot K}$.

Finally, the PsCP approached in this work consists of finding the set of colors \mathcal{C}^* with the greatest possible dissimilarity between its elements c_1, c_2, \dots, c_K and that, therefore, is a solution to the optimization problem presented in Equation (4).

$$\mathcal{C}^* = \underset{\mathcal{C}}{\operatorname{argmax}} F_\Delta(\mathcal{C}) \\ \text{subject to } \mathcal{C} \subset \mathbb{Y}^K. \quad (4)$$

4. Methodology

To solve PsCP, we propose an approach that is configured mainly by four steps:

M_1 Domain definition: This phase consists of defining the image in grey-levels that must be pseudo-colored by our technique.

$M_{1.1}$ However, all results can be easily extended to consider classes of images that have not passed any segmentation process, given the advances achieved in recent years in the area of image segmentation [42–44]. This type of consideration is outside the scope of the paper, so we will assume in this work that the method must work on an image in grey levels I pre-segmented into K sub-regions.

$M_{1.2}$ Once we have the pre-segmented image I , we need to define the neighborhood matrix Δ of this image so that the algorithm can perform the optimization having the information of which subregions of I are neighbors to each other and, consequently, they are assigned different colors.

M_2 Proposed optimization method: In this step of the methodology, we use the proposed meta-heuristic, which consists of a genetic algorithm that was specifically designed to search in a color space for an optimal solution for PsCP. Our main contributions to the method are given in the form of two operators:

$M_{2.1}$ Local search strategies: As in our preliminary work [21], we will also use local search strategies in our GA in two operators: the mutation operator and an exclusive operator dedicated to conducting a local search with massive behavior.

$M_{2.2}$ Adaptive rules: In this text, we generalize the use of the rules for self-adaptation of parameters proposed in our preliminary work [21] having as inspiration ideas that have been successfully used in other classes of combinatorial optimization problems [20]. In this

case, we make use of mapping functions to automate the way that mutation and crossing rates are updated in the course of GA iterations.

M₃ Algorithm responses: After the optimization process carried out by our technique, we obtain two types of responses: a numerical and vector solution in the K -dimensional color space ($M_{3,1}$) and the pseudo-colored image ($M_{3,2}$). In the first case, we have numerical and quantitative information on how different are the colors attributed to I , calculated from the measures of Radlak and Smolka [14]. In the second case, the method must establish the pseudo-colored version of I from the obtained vector solution.

M₄ Validation scenarios: In our methodology, we will validate the solutions found by our meta-heuristics in a quantitative ($M_{4,1}$) and in a qualitative ($M_{4,2}$) ways through three different case studies (CS):

- **CS I:** In the first CS, we will compare the quantitative and qualitative results obtained by our method that represent information on how distant the colors attributed to the I regions are in the CIELAB space and compare the results with the most recent color-coding techniques available in the specialized literature.
- **CS II:** In the second CS, we will compare the quantitative results obtained by our method in 24 synthetic and abstract unreal images in comparison to the other existing techniques in this field of study.
- **CS III:** In the third CS, we we will compare the quantitative results obtained by our method considering the Munsell atlas color space in comparison to the other existing techniques in the specialized literature.

In Figure 2, we present a pipeline of the proposed methodology to solve PsCP. So that the steps of domain definition are colored blue, the steps of the proposed optimization method are colored green, the steps of algorithm response are colored yellow and the steps of validation scenarios are colored red.

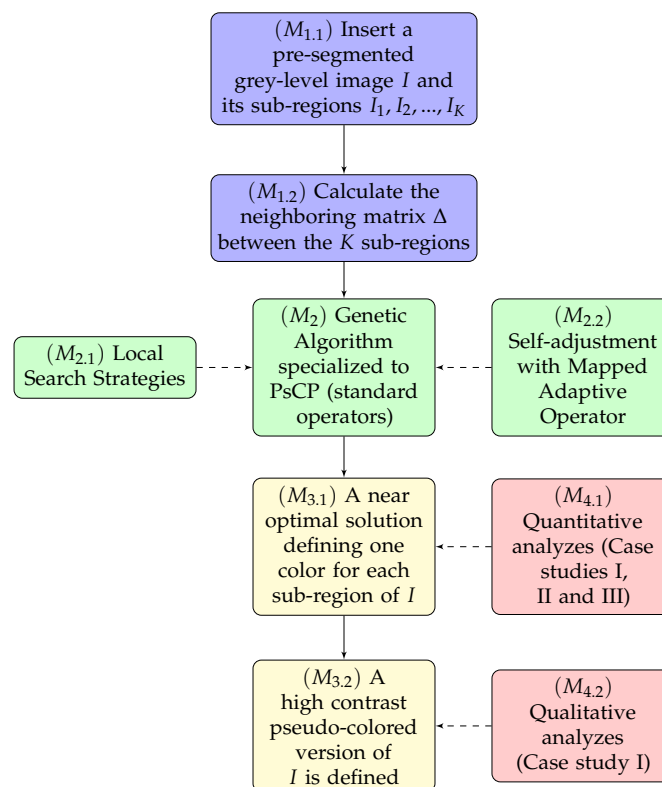


Figure 2. Pipeline of the proposed methodology for solve the pseudo-coloring problem (PsCP).

5. Mapped Local Search Adaptive Genetic Algorithm for PsCP

In this section, we present the explanation of our method, which was developed specifically to solve PsCP. All operators that make up the proposed meta-heuristic are discussed in detail. In addition, to facilitate the reproduction of the technique, pseudo-codes and operating schemes of all operators are presented throughout the manuscript. We highlight as the main innovations of our approach the use of local search strategies in GA; in particular, the definition of a massive local search operator, and the use of adaptive rules in order to perform parameter tuning. To the best of our knowledge, this type of strategy, although widely used in other combinatorial optimization problems, has not yet been employed in the PsCP solution. In addition, in this extended version of our work, we generalize the adaptive operator through antisymmetric mapping functions. Thus, the main contributions of our work are contained in the following topics:

1. A new meta-heuristic specialized in solving the PsCP with local search strategies in all operators. Inspired by the works of Asadzadeh [16] and Viana, et al. [45], we developed a massive local search operator that explores the best individual neighborhood.
2. A new self-parameter adjustment operator [17] to avoid well-known problems in GA such as premature convergence and getting stuck in local optima. In addition, we present the generalization of our preliminary adaptive strategy from [21] using mapping functions, which proved to be favorable in studies related to other types of combinatorial optimization [19,20].
3. The advancement of experimental results in benchmarks that define the state-of-the-art.

5.1. Chromosome Decoding

A possible solution for the PsCP is defined by a set of K colors to be associated with the regions of a pre-segmented gray-level image I . To make this computationally efficient, the chromosome must be the vectorized version of the color set $\mathcal{C} = \{c_1, c_2, \dots, c_K\} \subset \mathbb{Y}^K$. In this way, each gene on this chromosome is formed by a color c_i that must be used to coloring the sub-region I_i . Mathematically, the chromosome decoding adopted in this work that should represent the set \mathcal{C} is the vector C defined in Equation (5).

$$C = [c_1, c_2, \dots, c_K], \quad (5)$$

in which $c_i \in \mathbb{Y}$ is the color to be associated with the region I_i .

As an example, if the considered color space is the sRGB, then the chromosome will be in the form: $C = [(R_1, G_1, B_1), (R_2, G_2, B_2), \dots, (R_K, G_K, B_K)]$, with $R_i, G_i, B_i \in \{0, 1, 2, 3, \dots, 255\}$ for all i .

In general, the chromosome used in this work is a vector in the space $\mathbb{R}^{n \cdot K}$, with n being the dimension of the space \mathbb{Y} . Therefore, there is an ordering between the C coordinates that must be taken into account to coloring the I sub-regions, since the first n coordinates of the chromosome C (first gene) represent the components of the color c_1 that should be used to coloring the I_1 subregion, the following n coordinates of C (second gene) represent the components of the color c_2 that should be used to coloring the subregion I_2 , and so on. In Figure 3, we present an example that outlines, according to the proposed decoding, the relationship between the genes of a chromosome C , with $\mathbb{Y} = \text{sRGB}$, and the pseudo-coloring process in a pre-segmented grey-level image I in $K = 4$ sub-regions.

$$C = [(0, 255, 0), (255, 255, 0), (0, 0, 255), (255, 0, 0)]$$

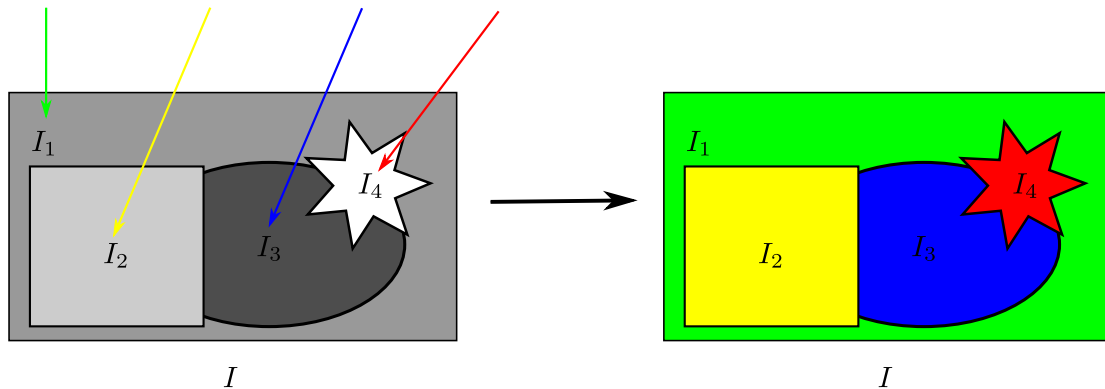


Figure 3. Example of pseudo-coloring of a grey-level image I , pre-segmented in $K = 4$ regions, according to a chromosome C in the color space $\mathbb{Y} = \text{sRGB}$.

5.2. Fitness Function

According to the decoding proposed in Section 5.1, we can naturally define the fitness value associated with a C chromosome as an isomorphism of the function F (Equation (2)), computationally efficient, of its representation as a set of colors \mathcal{C} . Mathematically, given a grey-level image I with neighborhood matrix Δ , we define the fitness value of a chromosome $C = [c_1, c_2, \dots, c_K] \in \mathbb{Y}^K$ the function \bar{F}_Δ , presented in Equation (6).

$$\bar{F}_\Delta : \begin{matrix} \mathbb{Y}^K & \longrightarrow & \mathbb{R}_+ \\ (c_1, c_2, \dots, c_K) & \longmapsto & \bar{F}_\Delta(c_1, c_2, \dots, c_K) := F_\Delta(\{c_1, c_2, \dots, c_K\}) \end{matrix} \quad (6)$$

It is worth highlighting some considerations regarding the function \bar{F}_Δ :

- Both \bar{F}_Δ and F_Δ are defined taking into account the neighborhood matrix Δ . That is, both functions are defined from the image structure that we intend to color.
- The main difference between \bar{F}_Δ and F_Δ is that the first one takes into account the order of the coordinates in the chromosome C , while the second one acts on a set \mathcal{C} . Didactically, it is simpler to reproduce \bar{F}_Δ and, therefore, we define the fitness function in this way.
- The meta-heuristic that finds the greatest values assumed by \bar{F}_Δ will find possible solutions to the problem presented in Equation (4), since these two situations consist of determining K colors with the greatest possible visual dissimilarity. In this way, the feasibility and the equivalence of the problem are maintained.

5.3. Selection Process

The selection step in an evolutionary algorithm like GA is very important since it is from this routine that we simulate the Darwinian phenomenon of “natural selection” [38], making good solutions more likely to perpetuate their genes. In the case of GA, selection strategies are used to define which chromosomes will pass through the crossover operator and which chromosomes will survive in the stage of creating a new population. This step of the method takes into account the numerical fitness value of each chromosome, and since the PsCP is a maximization problem, the higher the fitness value of a chromosome, the greater the chance of being selected. In detail, we will use in this work the strategy known as roulette wheel selection [46] in the case of the choice of individuals for the crossover operator, and the roulette wheel selection retaining the best individual of the population [47] in the case of the generation of new population.

5.4. Crossover Operator

It is at this stage of the method that we will carry out the genetic combination method between the chromosomes selected in the previous section. In this case, the parent chromosomes must be combined

two by two and this process should result in two child chromosomes. In order that no generated chromosome is outside the limits that define the \mathbb{Y} color space, we propose to use an adaptation of the well-known convex crossover [48], in which the convex combination of two parent-chromosomes, Parent_1 and Parent_2 , is made, resulting in two offspring-chromosomes, Kid_1 and Kid_2 , according to Equation (7) given below.

$$\text{Kid}_1 = \text{proj}_{\mathbb{Y}^K} \left(\alpha^T \cdot \text{Parent}_1 + (\vec{1} - \alpha)^T \cdot \text{Parent}_2 \right), \tag{7a}$$

$$\text{Kid}_2 = \text{proj}_{\mathbb{Y}^K} \left((\vec{1} - \alpha)^T \cdot \text{Parent}_1 + \alpha^T \cdot \text{Parent}_2 \right), \tag{7b}$$

where $\vec{1}$ is the $\mathbb{R}^{n \cdot K \times 1}$ vector whose coordinates are all equal to 1 and α is an $\mathbb{R}^{n \cdot K \times 1}$ vector whose coordinates are taken randomly between $[0, 1]$.

It is worth mentioning that we are using a projection function to ensure that Kid_1 and Kid_2 are, in fact, feasible values of colors in the space \mathbb{Y} . In general, this function projects the results obtained by the convex combination between Parent_1 and Parent_2 of the $\mathbb{R}^{n \cdot K}$ space for the respective closest vectors in the \mathbb{Y}^K space, with n being the dimension of \mathbb{Y} . This projection, defined in Equation (8), determines the generated offsprings: Kid_1 and Kid_2 .

$$\text{proj}_{\mathbb{Y}^K} (\cdot) : \begin{matrix} \mathbb{R}^{nK} & \longrightarrow & \mathbb{Y}^K \\ (x_1, x_2, \dots, x_{nK}) & \longmapsto & \arg \min_{(c_1, c_2, \dots, c_K) \in \mathbb{Y}^K} \left\{ \|(x_1, x_2, \dots, x_{nK}) - (c_1, c_2, \dots, c_K)\|^2 \right\} \end{matrix} \cdot \tag{8}$$

Note that in most cases, the projection function $\text{proj}_{\mathbb{Y}^K} (\cdot)$ has an explicit and very simple form. For example, in the special case where $\mathbb{Y} = \text{sRGB}$, then $\text{proj}_{\mathbb{Y}^K} (\cdot) = \text{round} (\cdot)$, in which $\text{round} (\cdot)$ is the well-known approximation function to the nearest integer.

A schematic of our crossover technique is presented in Figure 4.

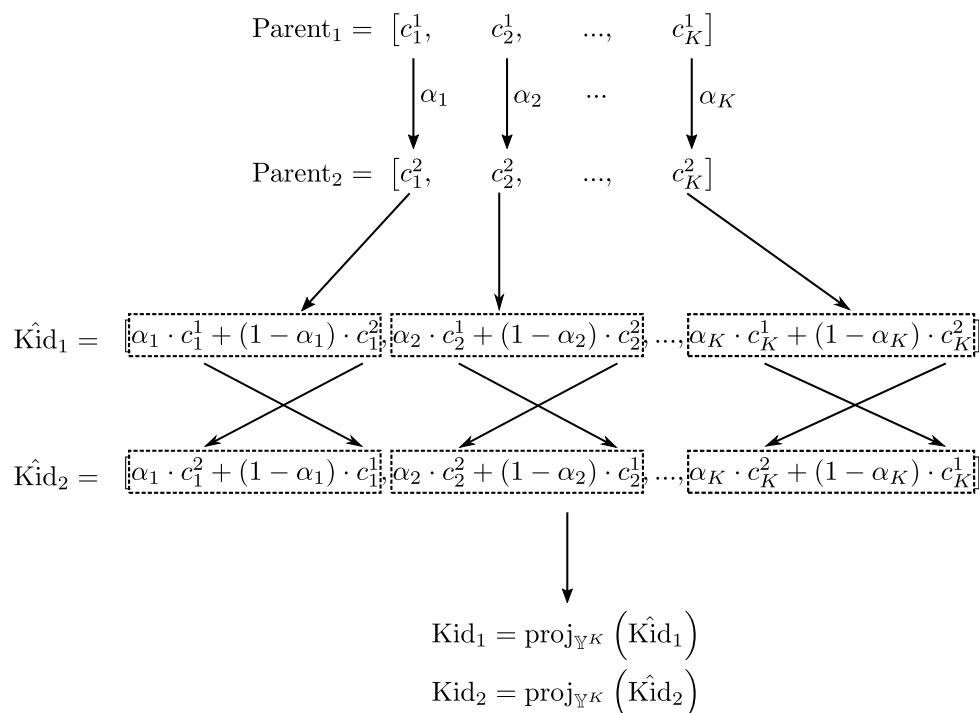


Figure 4. Schematic of the proposed crossover operator. Such that α_i is a random value taken in $[0, 1]$ for all i .

5.5. Mutation Operator

The mutation operator proposed in this work uses inspiration from the local search strategies presented by Asadzadeh [16] and Ombuki and Ventresca [49]. In fact, we propose an improvement based on the union between the techniques of these two works. In this case, Asadzadeh [16] proposes that successive local searches be carried out on the individuals in the mutation operator, while Ombuki and Ventresca [49] suggests that, in some cases, successive local searches are carried out on the individual and in other cases, only a simple mutation is applied on the individual. In this way, our mutation operator is divided into two subroutines:

- Mut₁: In this routine, a mutation function is randomly taken into a set of mutation functions, and, using this function, N_{Mut_1} perturbations are performed on the chromosome, so that the good perturbations are maintained and the bad ones are ignored.
- Mut₂: In this routine, a simple Gaussian mutation is applied to the chromosome.

An individual who is generated on the crossover operator has a p_{mut} chance of passing through the mutation operator. In addition, to control the use of the Mut₁ and Mut₂ subroutines, all individuals selected for the mutation will be perturbed according to only one of these subroutines. Thus, these individuals have a p_{LS} chance of being perturbed according to the Mut₁ subroutine and, consequently, these individuals have a chance $1 - p_{LS}$ of being perturbed according to the Mut₂ subroutine.

To model this operation, a group of mutation functions given in the form of $f_{mut}(\cdot, \cdot)$ is used in Equation (9).

$$f_{mut} : \mathbb{Y} \times \{1, 2, \dots, n \cdot K\}^2 \longrightarrow \mathbb{Y} \tag{9}$$

$$(C, (i, j)) \longmapsto f_{mut}(C, (i, j))'$$

where n is the dimension of \mathbb{Y} .

The definition of the Mut₁ subroutine establishes that it is necessary to randomly choose a mutation function from a group of mutation functions. Thus, it is necessary to define a set formed by possible mutation functions. In this work, as we are going to solve the NP-Hard combinatorial problem PsCP, we are going to use mutation functions widely used in solving problems of this type. In this case, we will use the functions [50]: $f_{swap}(\cdot, \cdot)$, $f_{invert}(\cdot, \cdot)$ and $f_{insert}(\cdot, \cdot)$. These functions affect perturbations on the chromosome in order to shift some of its coordinates. In Figure 5, we can see the result of the disturbance made by all these functions on the same individual considering the $\mathbb{Y} = sRGB$ color space and considering that the image to be pseudo-colored is pre-segmented in $K = 3$ sub-regions. Certainly, feasibility is maintained in cubic spaces defined by intervals of the same range, as is the case of sRGB space. However, if at the end of the entire mutation procedure, the generated individual is outside the color space, then it is necessary to make use of the projection function on the mutant individual.

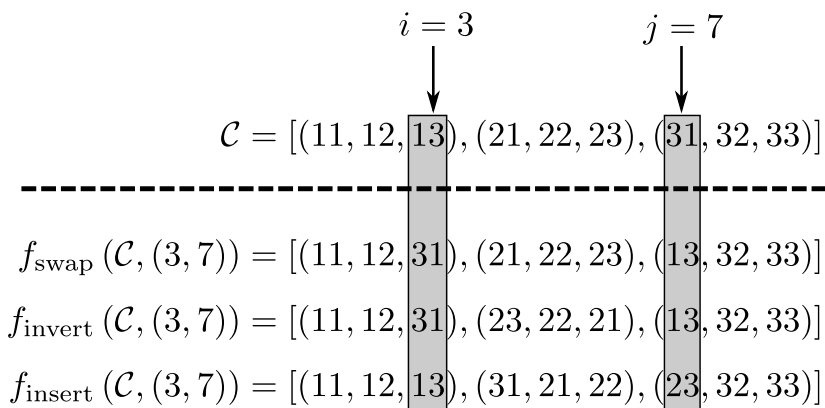


Figure 5. Schematic diagram of the functioning of three mutation functions on the same chromosome C .

In Mut_2 , we propose to use only one mutation function. In this case, we apply a simple Gaussian perturbation [51] on the color adding a Gaussian noise in each color component. In this way, all genes in the chromosome receives a slightly random increase or decrease.

Specifically, we will suppose that the \mathbb{Y} can be represented or approximated to a cube in $Y \subset \mathbb{R}^n$, such that $Y := [l_1, L_1] \times [l_2, L_2] \times \dots \times [l_n, L_n]$. In this way, if one color c_i is in \mathbb{Y} , then $c_i = (c_{i,1}, c_{i,2}, \dots, c_{i,n})$ and $l_j \leq c_{i,j} \leq L_j$ for all $j \in \{1, 2, \dots, n\}$. For example, if we choose \mathbb{Y} as being the sRGB color space, then all the colors would have three components ($n = 3$) and all ranges would be defined by the following lower and upper bounds: $l_1 = l_2 = l_3 = 0$ and $L_1 = L_2 = L_3 = 255$.

Finally, we can define the mutation function used for this situation as the function $f_{Gauss}(\cdot) : \mathbb{Y}^K \rightarrow \mathbb{Y}^K$ defined in Equation (10).

$$f_{Gauss} \left(\begin{bmatrix} c_1 \\ c_2 \\ \vdots \\ c_K \end{bmatrix} \right)^T = f_{Gauss} \begin{pmatrix} \begin{bmatrix} c_{1,1} \\ c_{1,2} \\ \vdots \\ c_{1,n} \end{bmatrix}^T \\ c_{2,1} \\ c_{2,2} \\ \vdots \\ c_{2,n} \\ \vdots \\ c_{K,1} \\ c_{K,2} \\ \vdots \\ c_{K,n} \end{pmatrix} := \text{proj}_{\mathbb{Y}^K} \begin{pmatrix} \begin{bmatrix} c_{1,1} + \beta_1 \\ c_{1,2} + \beta_2 \\ \vdots \\ c_{1,n} + \beta_n \end{bmatrix}^T \\ c_{2,1} + \beta_{n+1} \\ c_{2,2} + \beta_{n+2} \\ \vdots \\ c_{2,n} + \beta_{n+n} \\ \vdots \\ c_{K,1} + \beta_{(K-1) \cdot n+1} \\ c_{K,2} + \beta_{(K-1) \cdot n+2} \\ \vdots \\ c_{K,n} + \beta_{(K-1) \cdot n+n} \end{pmatrix} \right)^T, \quad (10)$$

in which $\beta_{(r-1) \cdot n+j}$ is a random number obtained from a normal distribution with mean 0 and standard deviation equals to $\left| \frac{L_j - l_j}{10} \right|$, for all $r \in \{1, 2, \dots, K\}$ and $j \in \{1, 2, \dots, n\}$. We chose this configuration, as it is a simple and well-established strategy in the specialized literature on gaussian disturbances in GAs, as initially proposed by Hinterding [52]. Note that we are using the projection function in the final disturbance of the chromosome since the addition of gaussian noise can remove the chromosome from the feasible color space.

In the Algorithm 1, the proposed mutation procedure is summarized.

Algorithm 1 Proposed mutation operator.

	P_{kids}	Offsprings generated in crossover
	\bar{F}_{Δ}	Fitness function
Input:	p_{mut}	Probability of mutation
	p_{LS}	Probability of Mut ₁
	N_{Mut_1}	Number of mutation function applications for Mut ₁

```

1:  $f_{\text{mut}} := \text{rand\_take}(\{f_{\text{swap}}, f_{\text{invert}}, f_{\text{insert}}\})$     ▷ The function  $\text{rand\_take}(\cdot)$  randomly returns an
   element from a set.
2:  $P_{\text{mut}} := \{\}$ 
3: for  $C \in P_{\text{kids}}$  do
4:   if  $\text{rand}([0, 1]) \geq p_{\text{mut}}$  then                                ▷ Apply mutation in  $p_{\text{mut}}$  percent of kids.
5:     break
6:   else if  $\text{rand}([0, 1]) \leq p_{\text{LS}}$  then                                ▷ Apply Mut1 in  $p_{\text{LS}}$  percent of cases.
7:      $F_C := \bar{F}_{\Delta}(C)$ 
8:     for  $k := 1$  to  $N_{\text{Mut}_1}$  do                                        ▷ Apply  $f_{\text{mut}}$   $N_{\text{Mut}_1}$  times.
9:        $i := \text{rand\_take}(\{1, 2, \dots, 3K\})$ 
10:       $j := \text{rand\_take}(\{1, 2, \dots, i - 1, i + 1, \dots, 3K\})$ 
11:       $\hat{C} := f_{\text{mut}}(C, (i, j))$ 
12:       $F_{\hat{C}} := \bar{F}_{\Delta}(\hat{C})$ 
13:      if  $F_{\hat{C}} \geq F_C$  then                                        ▷ If the perturbation is beneficial, then it must be maintained.
14:         $C := \hat{C}$ 
15:         $F_C := F_{\hat{C}}$ 
16:      end if
17:    end for
18:   else                                                                ▷ (Mut2) Apply  $f_{\text{Gauss}}$  only once.
19:      $C := f_{\text{Gauss}}(C)$ 
20:   end if
21:    $P_{\text{mut}} := P_{\text{mut}} \cup \{C\}$ 
22: end for

```

Output: P_{mut} Population of mutated individuals

5.6. Massive Local Search Operator

The main objective of a massive local search operator [45] is to perform a more elaborate and systematic search around good solutions. Already used to solve other combinatorial works, the idea behind this type of operator is that we are more inclined to find better solutions when examining the neighborhood of chromosomes with better fitness than when investigating randomly in space. For example, Asadzadeh [16] proposes to apply this strategy to the best individual in the population. Specifically, the author performs, in a controlled manner, successive disturbances with the function $f_{\text{swap}}(\cdot, \cdot)$ so that all the coordinates of the chromosome can be swapped if convenient. In this work, we propose a new massive local search operator that specializes in disturbing, one by one, the colors represented by the genes of the best individual in a population. In detail, we will consider that $C_{\text{Best}} = (c_1, c_2, \dots, c_K)$ is the best individual in a population and, in sequence, for each color c_i represented by the genes of C_{Best} we will perform the following four steps:

- **Step 1:** The color c_i receives a random increase, making it lighter.
- **Step 2:** The perturbation is maintained only if it is beneficial, increasing the fitness value of C_{Best} .
- **Step 3:** If c_i was not modified in the previous step, then a random decrease in c_i is applied, making it darker.
- **Step 4:** The perturbation should be maintained if it is beneficial.

Therefore, in the Algorithm 2, we can see details about our proposed massive local search operator to solve the PsCP.

Algorithm 2 Proposed massive search operator.

Input: $C_{\text{Best}} = (c_1, c_2, \dots, c_K)$ Best chromosome in the population
 \bar{F}_Δ Fitness function

1: **for** $i = 1$ to K **do**
 2: $\hat{c}_i := c_i$ ▷ Initially, \hat{c}_i is equal to c_i .
 3: $\gamma := \text{rand}((0, 1])$
 4: $c'_i := (1 + \gamma)c_i$ ▷ **(Step 1)** Random increase.
 5: $C' := \text{proj}_{\mathbb{Y}^K}(c_1, c_2, \dots, c_{i-1}, c'_i, c_{i+1}, \dots, c_K)$
 6: **if** $\bar{F}_\Delta(C') > \bar{F}_\Delta(C_{\text{Best}})$ **then** ▷ **(Step 2)** If the perturbation is beneficial, then update \hat{c}_i to c'_i .
 7: $\hat{c}_i := c'_i$
 8: **else** ▷ If the increase wasn't beneficial, then try apply a decrease on c_i .
 9: $c'_i := (1 - \gamma)c_i$ ▷ **(Step 3)** Random decrease.
 10: $C' := \text{proj}_{\mathbb{Y}^K}(c_1, c_2, \dots, c_{i-1}, c'_i, c_{i+1}, \dots, c_K)$
 11: **if** $\bar{F}_\Delta(C') > \bar{F}_\Delta(C_{\text{Best}})$ **then** ▷ **(Step 4)** If the perturbation is beneficial, then update \hat{c}_i to c'_i .
 12: $\hat{c}_i := c'_i$
 13: **end if**
 14: **end if**
 15: **end for**
 16: $\hat{C} := \text{proj}_{\mathbb{Y}^K}((\hat{c}_1, \hat{c}_2, \dots, \hat{c}_K))$ ▷ The new chromosome is updated considering only the perturbations that were beneficial and contributed to improving the fitness value of C_{Best} .

Output: \hat{C} Best individual in the neighborhood of C_{Best}

5.7. Mapped Adaptive Rules

Adaptive operators are defined as self-tuning strategies of parameters that are useful to assist the meta-heuristic to avoid problems that can compromise its performance such as the phenomena of premature convergence and getting stuck in local optima. It is common that, during the method iterations, parameters that represent the percentage of individuals selected for the crossover (p_x) and the percentage of individuals selected for the mutation (p_{mut}) are updated according to the genetic variability present in the populations of individuals in GA. In this work, we propose to update these parameters according to the fitness values of two consecutive populations. More specifically, we will use the concept of “improvement”, which represents a measure that models the degree of growth considering the fitness values of one population to another. For this, we will consider that the proposed meta-heuristic works on populations with N_{pop} individuals and, we will define the vectors $\mathcal{F}_{\text{it}}, \mathcal{F}_{\text{it}+1} \in \mathbb{R}_+^{N_{\text{pop}}}$ so that its coordinates are the fitness values of two consecutive populations, i.e., considering the it -th and $(\text{it} + 1)$ -th iterations of the method, respectively. To numerically measure how much a population has improved in relation to the previous population, we will use the function $f_{\text{improvement}}(\cdot, \cdot)$, considering as a prerequisite that all functions used in this section are continuous, structured in Equation (11), which aims to measure how much the fitness of the population of the iteration “it+ 1” ($\mathcal{F}_{\text{it}+1}$) has improved considering the fitness of the previous population (\mathcal{F}_{it}). Thus, the greater the value of $f_{\text{improvement}}(\mathcal{F}_{\text{it}}, \mathcal{F}_{\text{it}+1})$ the greater the improvement of the values of $\mathcal{F}_{\text{it}+1}$ in relation to the values of \mathcal{F}_{it} . Likewise, if $\mathcal{F}_{\text{it}+1}$ has the same fitness values as \mathcal{F}_{it} , then $f_{\text{improvement}}(\mathcal{F}_{\text{it}}, \mathcal{F}_{\text{it}+1})$ must be equal to 0.

$$f_{\text{improvement}}(\cdot, \cdot) : \mathbb{R}_+^{N_{\text{pop}}} \times \mathbb{R}_+^{N_{\text{pop}}} \longrightarrow \mathbb{R}_+ \quad (11)$$

$$(\mathcal{F}_{\text{it}}, \mathcal{F}_{\text{it}+1}) \longmapsto f_{\text{improvement}}(\mathcal{F}_{\text{it}}, \mathcal{F}_{\text{it}+1}).$$

Specifically, we proposed to measure the fitness improvement considering two consecutive populations using an average between the absolute value of the difference of four well-established statistical concepts [53] of those population fitness: standard deviation ($\text{STD}(\cdot)$), the average ($\text{AVG}(\cdot)$), the best fitness ($\max\{\cdot\}$) and the worst fitness ($\min\{\cdot\}$). These measures are directly associated with fitness variability and with the improvement of fitness values from one population to another. Mathematically, we define in the Equation (12) the function $f_{\text{improvement}}(\cdot, \cdot)$, which represents the value of improvement considering two consecutive populations of the technique.

$$\begin{aligned}
 f_{\text{improvement}}(\mathcal{F}_{it}, \mathcal{F}_{it+1}) := & \frac{1}{4} \left| \text{STD} \left(\frac{1}{M} \mathcal{F}_{it} \right) - \text{STD} \left(\frac{1}{M} \mathcal{F}_{it+1} \right) \right| + \\
 & + \frac{1}{4} \left| \text{AVG} \left(\frac{1}{M} \mathcal{F}_{it} \right) - \text{AVG} \left(\frac{1}{M} \mathcal{F}_{it+1} \right) \right| + \\
 & + \frac{1}{4} \left| \max \left\{ \frac{1}{M} \mathcal{F}_{it} \right\} - \max \left\{ \frac{1}{M} \mathcal{F}_{it+1} \right\} \right| + \\
 & + \frac{1}{4} \left| \min \left\{ \frac{1}{M} \mathcal{F}_{it} \right\} - \min \left\{ \frac{1}{M} \mathcal{F}_{it+1} \right\} \right|,
 \end{aligned} \tag{12}$$

in which $M := \max\{\max\{\mathcal{F}_{it}\}, \max\{\mathcal{F}_{it+1}\}\}$ is the maximum fitness value considering all the coordinates of \mathcal{F}_{it} and \mathcal{F}_{it+1} .

It is worth to point out some facts about the function $f_{\text{improvement}}(\cdot, \cdot)$ defined in Equation (12):

- **Fact 1:** The statistical measures used are calculated using the vectors $\frac{1}{M} \mathcal{F}_{it}$ and $\frac{1}{M} \mathcal{F}_{it+1}$. Since the coordinates of these vectors are contained in the interval $[0, 1]$ due to the normalization caused by the multiplication of the factor $\frac{1}{M}$, then all the statistical measures are contained in the interval $[0, 1]$. In this way, each one of the four addends of $f_{\text{improvement}}(\cdot, \cdot)$ can account for a maximum of $\frac{1}{4} |0 - 1| = 0.25$ and, therefore, the values assumed by the function $f_{\text{improvement}}(\cdot, \cdot)$ are contained in the range $[0, 1]$.
- **Fact 2:** If there are no changes between two consecutive generations, then there will be no differences between \mathcal{F}_{it} and \mathcal{F}_{it+1} . Thus, $\mathcal{F}_{it} = \mathcal{F}_{it+1}$ and, therefore, $f_{\text{improvement}}(\mathcal{F}_{it}, \mathcal{F}_{it+1}) = 0$.
- **Fact 3:** From the two previous facts, we can easily deduce that the greater the differences between the fitness of two consecutive generations, the closer to 1 will be the value of $f_{\text{improvement}}(\cdot, \cdot)$ associated with these values of fitness. At the same time, the more similar the fitness values of two consecutive generations, the closer to zero will be the respective value of $f_{\text{improvement}}(\cdot, \cdot)$. This means that the function defined in Equation (12) is able to establish a good numerical correspondence between changes in fitness values between two consecutive populations and can be used as an indicator of population stagnation.

In this text, we propose to adjust the parameters using mapping functions. These functions should be responsible for updating the mutation and crossover rates in all iterations of the method according to the improvement measure provided by the function $f_{\text{improvement}}(\cdot, \cdot)$ taking into account the fitness values of two consecutive populations. In this way, we will proceed in the same way as Xing, et al. [20] and Lin [17], in the sense that if the consecutive populations considered show a great amount of similarities, then the mutation and crossing rates should be changed according to the degree of stagnation established between these two populations. In fact, we will follow the same line of reasoning as we have done in our preliminary work [21], defined by the following two situations:

- **Situation 1:** There are signs of stagnation in relation to the populations \mathcal{F}_{it} and \mathcal{F}_{it+1} : In this case, we will have $f_{\text{improvement}}(\mathcal{F}_{it}, \mathcal{F}_{it+1}) \rightarrow 0$ and, to reverse this situation we are going to increase the mutation rate and decrease the crossover rate.
- **Situation 2:** There is a significant improvement in the population \mathcal{F}_{it+1} compared to the population \mathcal{F}_{it} . In this case, we will have $f_{\text{improvement}}(\mathcal{F}_{it}, \mathcal{F}_{it+1}) \rightarrow 1$ and, therefore, we can start

to return the mutation and crossover rates to their initially defined values. For that, we must decrease the mutation rate and increase the crossover rate.

In order to satisfy both described situations, we will use a symmetric (translated-odd) mapping function ($f_{\text{mapping}}(\cdot)$) that must act according to the improvement measure of two consecutive populations. In detail, if, in an iteration of the method, we come across **Situation 1**, then our mapping function must represent a negative value and if we come across **Situation 2**, then the mapping function must represent a positive value. Following these requirements, a possible structure for the mapping functions used for adjusting the crossover rate and for adjusting the mutation rate is presented in Equation (13).

$$f_{\text{mapping}}(\cdot) : [0, 1] \rightarrow [-1, 1] \\ x \mapsto f_{\text{mapping}}(x). \quad (13)$$

In addition, for the function $f_{\text{mapping}}(\cdot)$ to satisfy the proposed modeling, it is necessary that the following equalities are valid: $f_{\text{mapping}}(0) = 1$ and $f_{\text{mapping}}(1) = -1$. Therefore, if the improvement of two consecutive populations \mathcal{F}_{it} and \mathcal{F}_{it+1} is close to 0, then we have $f_{\text{mapping}}(f_{\text{improvement}}(\mathcal{F}_{it}, \mathcal{F}_{it+1})) \rightarrow f_{\text{mapping}}(0) = 1$. Likewise, if the improvement of two consecutive populations is close to 1, then $f_{\text{mapping}}(f_{\text{improvement}}(\mathcal{F}_{it}, \mathcal{F}_{it+1})) \rightarrow f_{\text{mapping}}(1) = -1$. This symmetric behavior is useful to update the mutation rate values taking into account the improvement measure between two consecutive populations. Finally, this mapping function must update the mutation rate (p_{mut}) following the procedure presented in Equation (14):

$$p_{\text{mut}}^{\text{it}+1} := \min \left\{ p_{\text{mut}}^{\text{max}}, \max \left\{ p_{\text{mut}}^{\text{min}}, \left[p_{\text{mut}}^{\text{it}} + \eta_{\text{mut}} \cdot f_{\text{mapping}} \left(f_{\text{improvement}}(\mathcal{F}_{it}, \mathcal{F}_{it+1}) \right) \right] \right\} \right\}, \quad (14)$$

in which, $p_{\text{mut}}^{\text{min}}$ and $p_{\text{mut}}^{\text{max}}$ are the minimum and maximum acceptable values for the mutation rate and $\eta_{\text{mut}} > 0$ is a fixed numeric parameter that controls the update value that the mutation rate receives in each iteration. Note that the terms “max” and “min” act only to keep the mutation rate within the range defined as feasible for the method.

Additionally, note that according to Equation (14), if two consecutive populations are similar, then the mutation rate will receive an increase and, likewise, if two consecutive populations are significantly different, then the mutation rate will receive a decrease.

Proceeding similarly, we defined an update process for the crossover rate in Equation (15):

$$p_{\times}^{\text{it}+1} := \min \left\{ p_{\times}^{\text{max}}, \max \left\{ p_{\times}^{\text{min}}, \left[p_{\times}^{\text{it}} - \eta_{\times} \cdot f_{\text{mapping}} \left(f_{\text{improvement}}(\mathcal{F}_{it}, \mathcal{F}_{it+1}) \right) \right] \right\} \right\}, \quad (15)$$

in which, p_{\times}^{min} and p_{\times}^{max} are the values that define an acceptable range for the crossover rate and $\eta_{\times} > 0$ is the control parameter that determines the step size used to update this rate. Note that the crossover rate update defined in this case occurs in the opposite way compared to what occurs in the case of the mutation rate update. That is, if two consecutive populations are similar, then the crossover rate will receive a decrease and, likewise, if two consecutive populations are significantly different, then the crossover rate will receive an increase.

For the experiments, we will consider three mapping functions: a linear function (Equation (16a)); a sigmoid function (Equation (16b)); and a cubic function (Equation (16c)).

$$f_{\text{mapping}}^{\text{lin}}(x) := 1 - 2x, \quad (16a)$$

$$f_{\text{mapping}}^{\text{sig}}(x) := \frac{2}{1 + e^{100\left(\frac{1}{2} - x\right)}} - 1, \quad (16b)$$

$$f_{\text{mapping}}^{\text{cub}}(x) := (1 - 2x)^3. \quad (16c)$$

In Figure 6, we present the graph of all mappings defined in Equation (16). In all the cases, we can see that the range $[0, 1]$ is mapped origin-symmetrically to the range $[-1, 1]$.

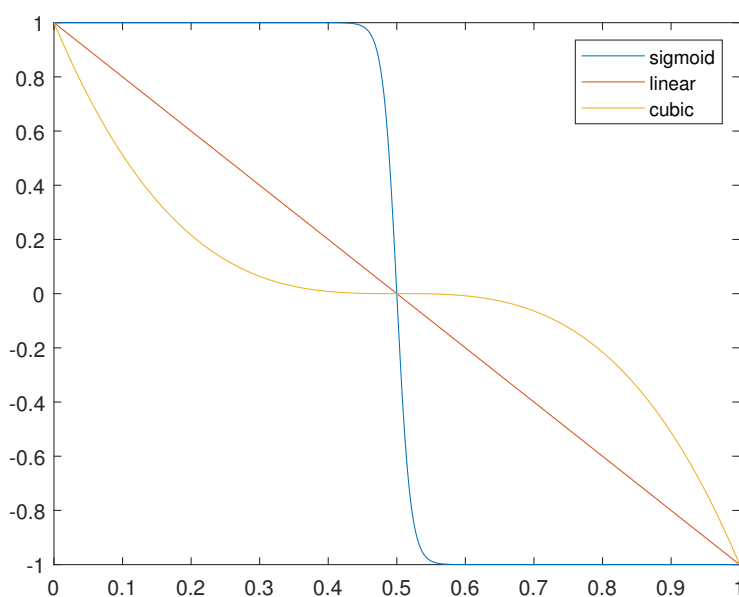


Figure 6. The plot of the graph of the mapping functions considered for the proposed mapped adaptive operator.

We can easily see that the adaptive rules presented in our preliminary work [21] make up only a special case of the adaptation strategy through mapping functions discussed in this section.

5.8. Proposed Algorithm

The proposed method consists of the sequential use of all the operators discussed. As input, we must present the neighborhood matrix defined by the K sub-regions of a pre-segmented gray-level image I to the meta-heuristic. In this case, the following sequence of steps must be performed in order to obtain an artificial coloring with a high contrast set of colors for the image I :

- Step 1 A population of chromosomes must be generated, each of which must be defined by K colors taken randomly in space \mathbb{Y} ;
- Step 2 Select individuals taking into account their fitness values for the crossover operator;
- Step 3 Evaluate these individuals;
- Step 4 Randomly select individuals to undergo mutation;
- Step 5 Select the best individual among all those considered so far;
- Step 6 Massively search its neighborhood for individuals with a better fitness value;

- Step 7 Adjust the method parameters with the adaptation operator;
 Step 8 Create a new population and evaluate it;
 Step 9 If the maximum number of iterations has not been reached, then you need to go back to step 2.

This sequence of steps is detailed in the flowchart of the method, shown in Figure 7.

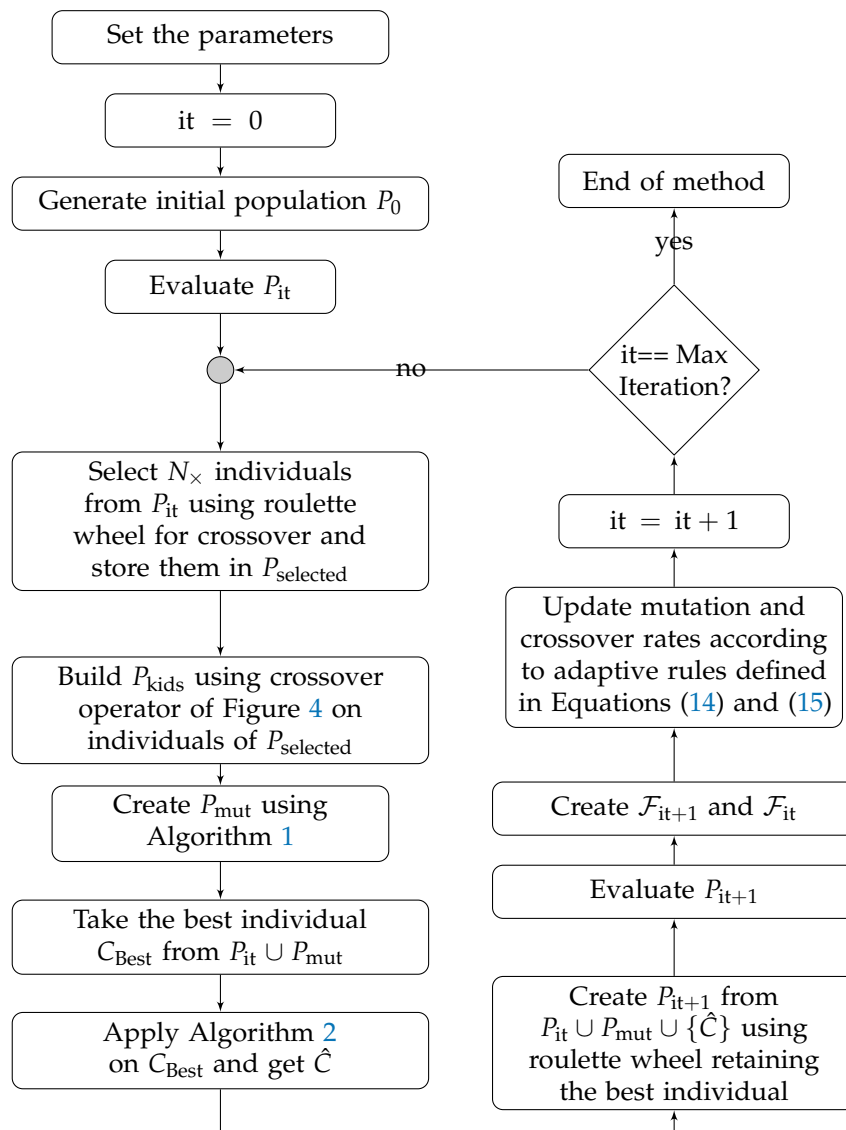


Figure 7. Flow chart of our proposed mapped local search adaptive genetic algorithm.

6. Experiments and Results

To evaluate the proposed methodology, we defined three distinct case studies (CS). In the first CS, we conduct a quantitative and a qualitative analysis of the proposed method considering a set of real world images. In the second CS, we compare the proposed method considering an extension of a well-established benchmark [54] of 24 synthetic and abstract unreal images in which its sub-regions are fully-connected. In the third CS, we conduct quantitative experiments considering the Munsell atlas color space. In addition, the experiments must confirm that the proposed method obtains the best performance in comparison with the current state-of-the-art techniques.

6.1. Setup and Implementation

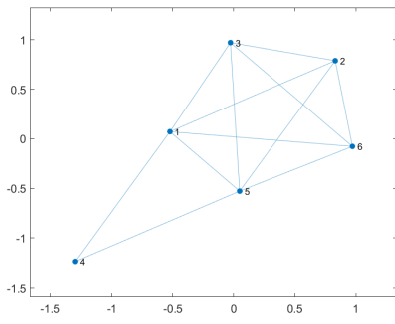
The authors Radlak and Smolka [14] shared with us their GA code so that we could conduct computational analysis and comparisons fairly. In this way, our methodology is configured in order to keep the parameterization as similar as possible with their GA. In detail, we use as population size $N_{\text{pop}} = 30$; the initial probabilities of mutation and crossover are defined as $p_{\text{mut}} = 0.1$ and $p_{\times} = 0.85$, respectively; half of the individuals selected to mutation operator must go through the subroutine Mut_1 and the other half must go through the subroutine Mut_2 , that is $p_{\text{LS}} = 0.5$; the step-sizes η_{mut} and η_{\times} from Equations (14) and (15), respectively, are both considered equals to 0.01, in order to avoid abrupt changes in mutation and crossover rates; and the method runs for 10^4 iterations (generations). The computational experiments were performed at MATLAB environment on an i5-4460 PC with 8GB of ram.

6.2. Case Study I: Qualitative and Quantitative Analysis Considering Real World Images and $\mathbb{Y} = \text{sRGB}$

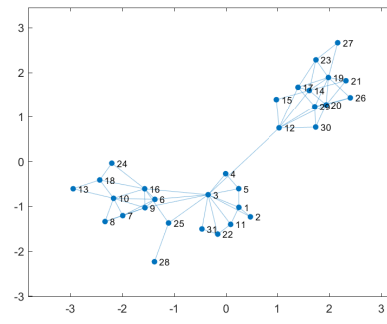
In this case study, we intend to evaluate the performance of the developed material taking into account the sRGB color space. That is, the PsCP solution defined by each image must be into $\mathbb{Y} = \text{sRGB}$. For this, the proposed technique is evaluated on a set of eight images: four images used for experiments in the work of Radlak and Smolka [14] and in our preliminary work [21]; and the pre-segmented grey-level version of four images of the public dataset presented in the Pascal VOC 2012 segmentation challenge [55,56]. In Table 1, we present some structural details about the considered images, such as: the number (K) of sub-regions (I_1, I_2, \dots, I_K) of the image; the number of nonzero values in the neighborhood matrix (Δ) of the image; and the number of connections on the graph defined by the neighborhood matrix. We also present in Figure 8 the graph that describes the connections between the sub-regions of each image considered for this CS. In detail, the presented images that make up the evaluation benchmark for this CS are: “Brains”, in Figure 9a; “Two Brains”, in Figure 10a; “Maps”, in Figure 11a; “Mosaic”, in Figure 12a; “Bicycle”, in Figure 13a (File name in VOC 2012 dataset: “2007_000129.png”); “Horses”, in Figure 14a (File name in VOC 2012 dataset: “2008_004892.png”); “Motorcycle”, in Figure 15a (File name in VOC 2012 dataset: “2007_005173.png”); and “Lunch”, in Figure 16a (File name in VOC 2012 dataset: “2011_003066.png”).

Table 1. Structural details about the used images.

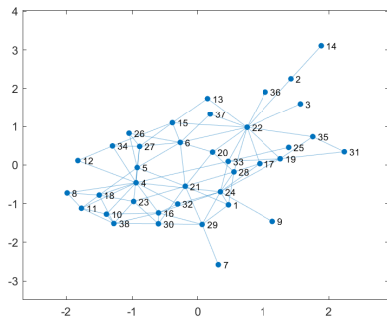
Image	Number of Subregions (K)	Nonzero Values in Δ	Number of Connections in the Graph
Brains	6	24	12
Two Brains	31	132	66
Maps	38	174	87
Mosaic	100	502	251
Bicycle	7	22	11
Horses	13	58	29
Motorcycle	18	96	48
Lunch	20	110	55



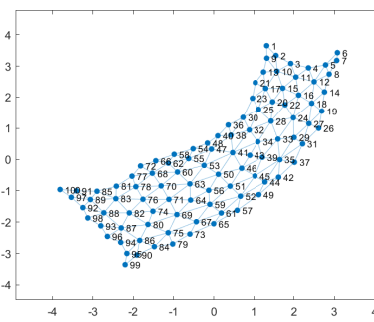
(a) Brains.



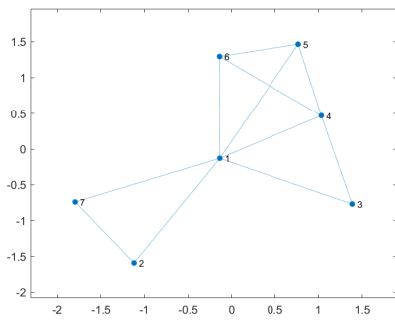
(b) Two Brains.



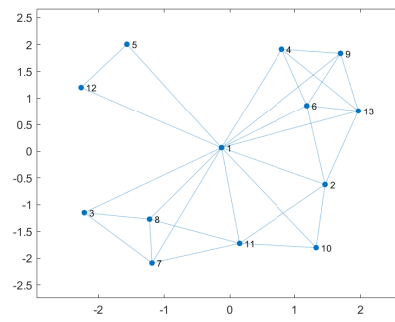
(c) Maps.



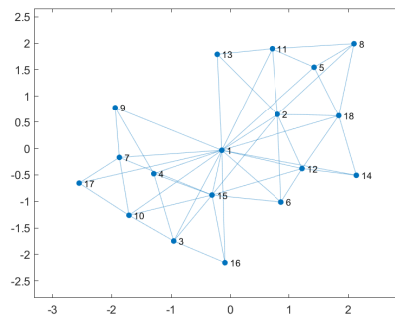
(d) Mosaic.



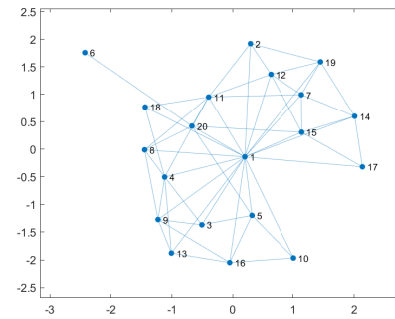
(e) Bicycle.



(f) Horses.



(g) Motorcycle.



(h) Lunch.

Figure 8. Graph of the neighborhood matrix between the sub-regions of each grey-level image. The number in each node is the associated index of the sub-region.

GA [14] was selected for comparison because it is the most recent method that was published in this field and it is the most similar approach of our proposed method. We also selected for comparison the material that were developed in our previous work [21]: a method without the addition of adaptive rules, entitled local search genetic algorithm (LSGA); and another method with the addition of basic adaptive rules, entitled local search adaptive genetic algorithm (LSAGA). In addition, we evaluated three different versions of our method using the adaptive rules described in Section 5.7: the lin-LSAGA, the sig-LSAGA and the cub-LSAGA. For this, we will evaluate the LSAGA considering in the adaptive operator, respectively, the functions $f_{\text{mapping}}^{\text{lin}}(\cdot, \cdot)$, $f_{\text{mapping}}^{\text{sig}}(\cdot, \cdot)$ and $f_{\text{mapping}}^{\text{cub}}(\cdot, \cdot)$, defined in Equation (16), as the mapping function. In Table 2, we present a summary of the compared methods and some of its respective specifications.

Table 2. Methods used for comparison. The symbol “-” means that the method does not use a mapping function.

Method	Local Search	Simple Adaptive Rules	Mapped Adaptive Rules	Mapping Function
GA	No	No	No	-
LSGA	Yes	No	No	-
LSAGA	Yes	Yes	No	-
lin-LSAGA	Yes	No	Yes	Yes (Equation (16a))
sig-LSAGA	Yes	No	Yes	Yes (Equation (16b))
cub-LSAGA	Yes	No	Yes	Yes (Equation (16c))

The best solution obtained for the PsCP using these techniques after 50 independent executions of each one, is shown in the Figures 9–16, so that in Figures 9–12 we present only the pseudo-colored pre-segmented image, while in Figures 13–16 we present a fusion between the grey-level version of the original image with its pseudo-colored pre-segmented version.

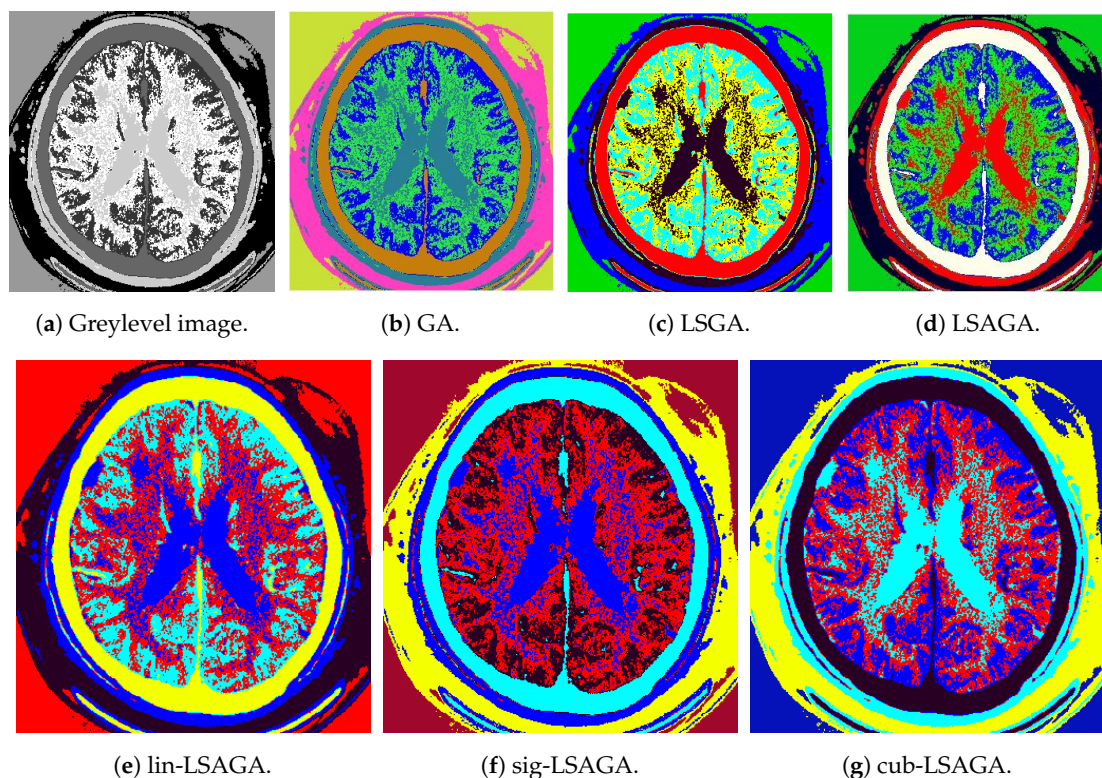


Figure 9. Pseudo-colored results for the grey-level image “Brain”.

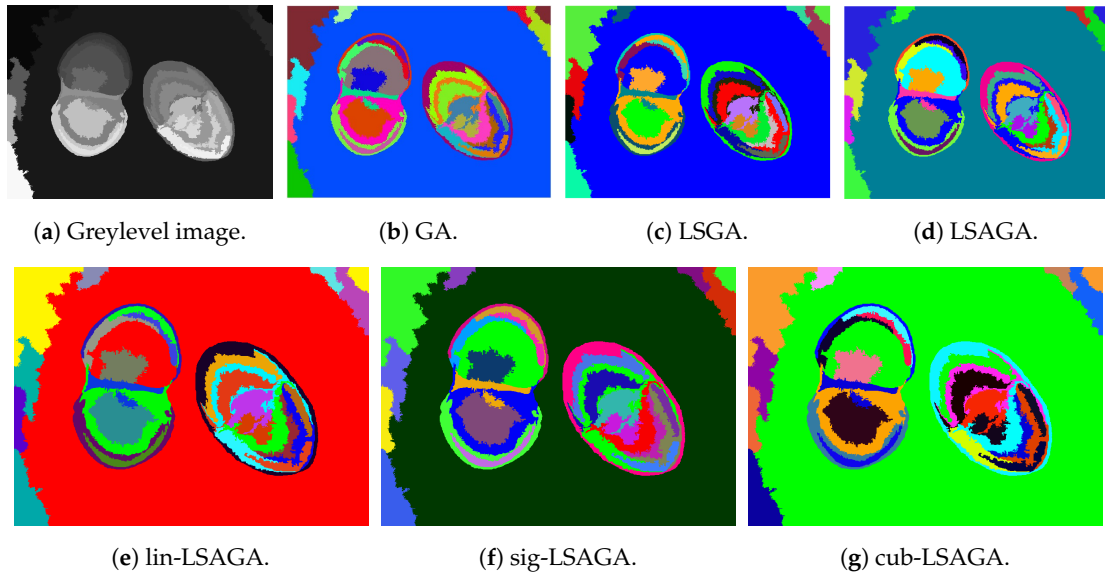


Figure 10. Pseudo-colored results for the grey-level image “Two Brains”.

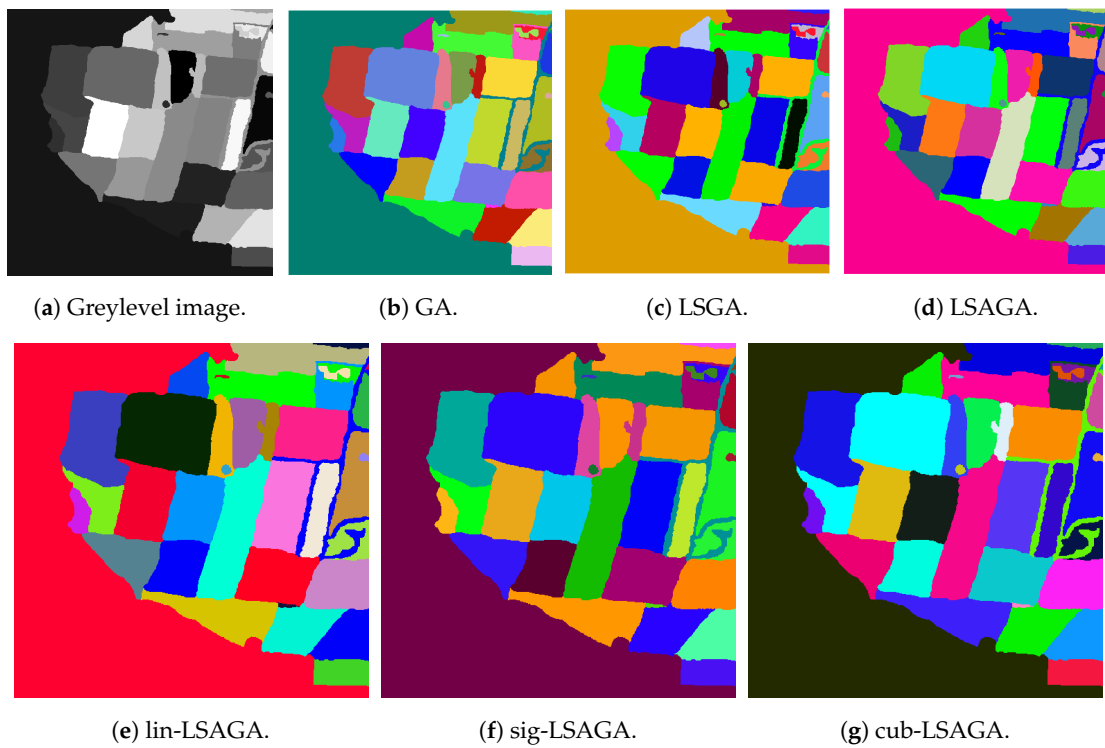


Figure 11. Pseudo-colored results for the grey-level image “Maps”.

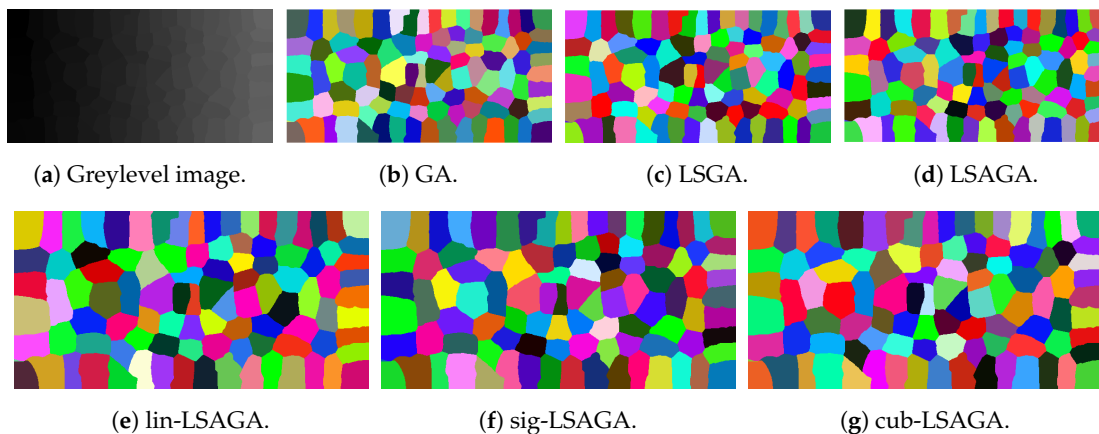


Figure 12. Pseudo-colored results for the grey-level image “Mosaic”.

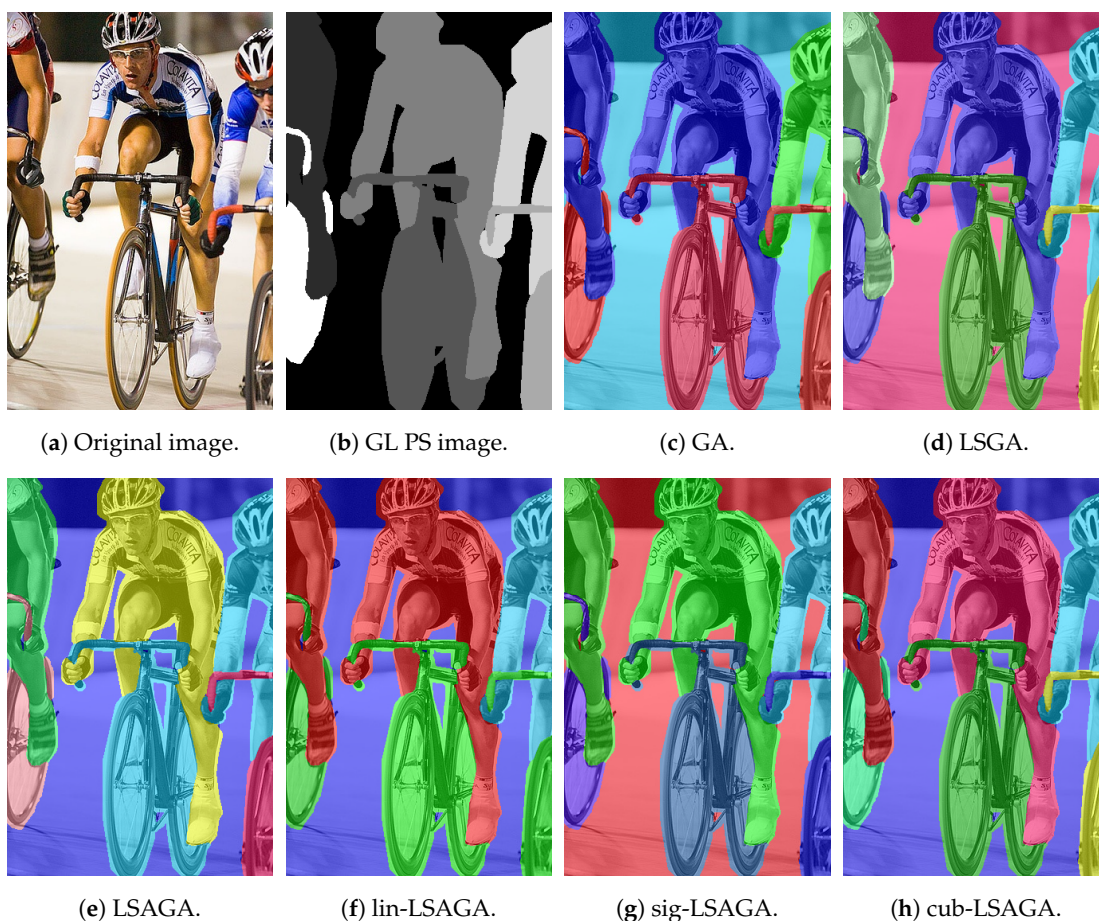


Figure 13. Pseudo-colored results for the grey-level image “Bicycle”.

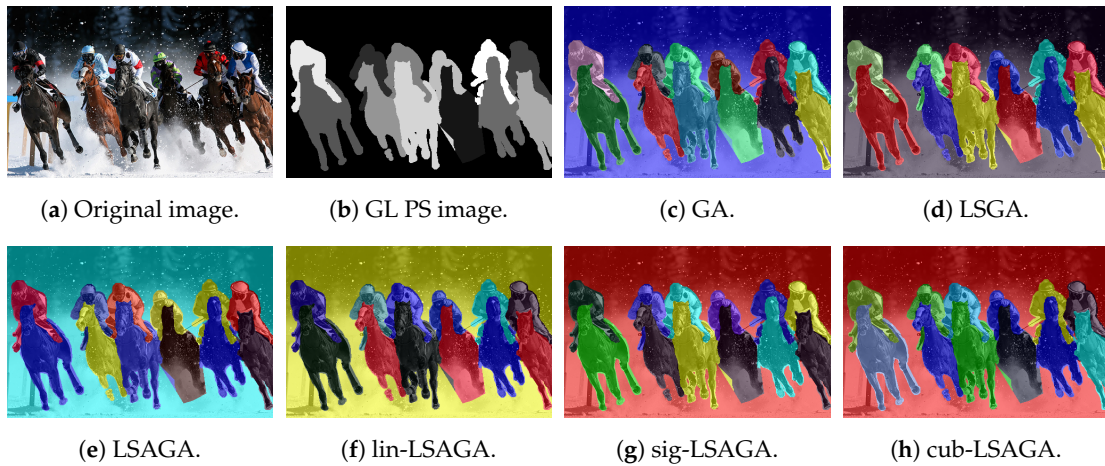


Figure 14. Pseudo-colored results for the grey-level image “Horses”.

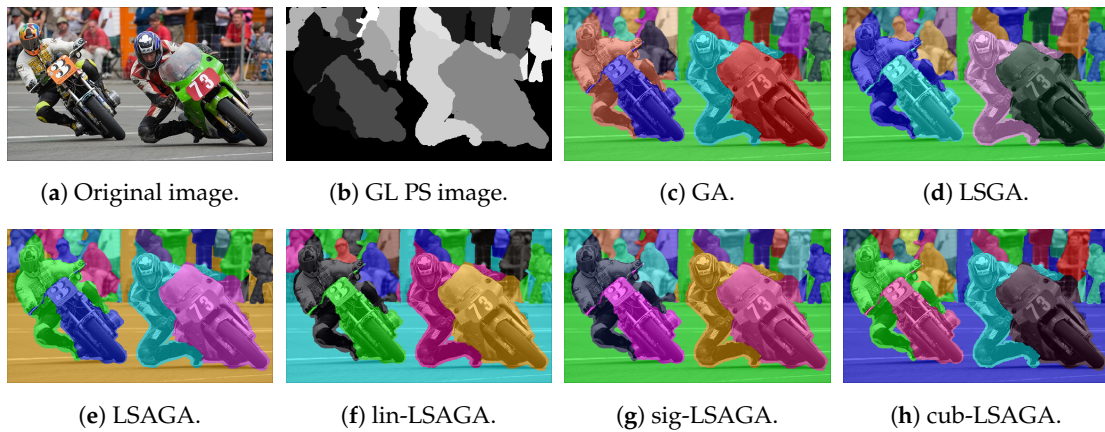


Figure 15. Pseudo-colored results for the grey-level image “Motorcycle”.

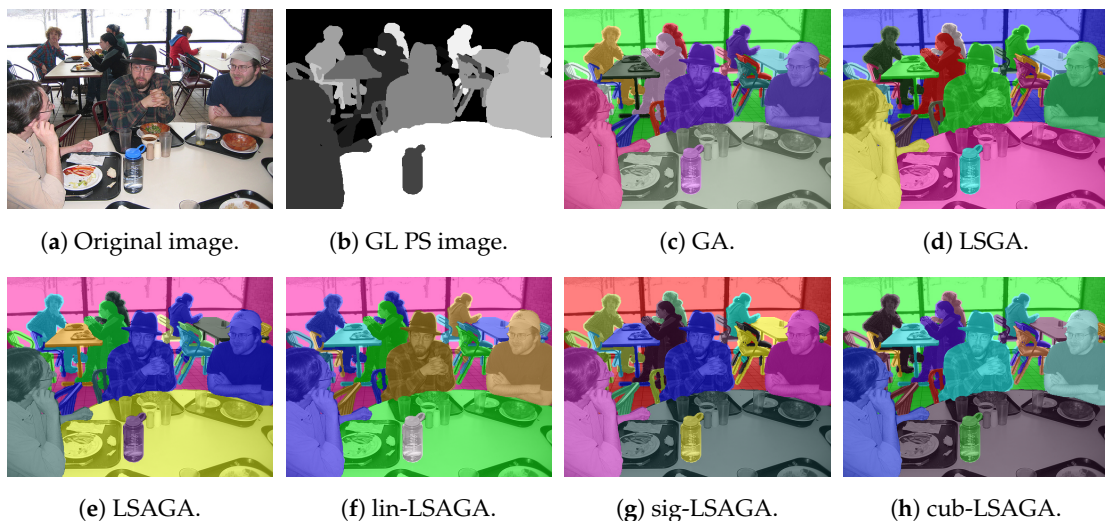


Figure 16. Pseudo-colored results for the grey-level image “Lunch”.

Considering all the images in Figures 9–16, we can see a evolution in the efficiency of visually detect a greater number of regions if we observe the image in greyscale and compare its respective pseudo-colored version. However, in some cases, one technique performs better than another. As highlighted in some examples in the sequence:

- **Brains:** This image is not very complex, as it has only six sub-regions. However, we can see that, in the case of the pseudo-colorization obtained by GA (Figure 9b), the colors used in the central part of the brain are two different shades of green, which can cause some visual confusion. This does not occur with the pseudo-colorizations obtained by the other methods.
- **Two Brains:** Concerning the images of the Two Brains, we can see that basic GA (Figure 10b) obtains a reasonably good visual separation between the sub-regions, presenting some confusion only with the colored regions in shades of red and pink in the central-left region of the image. In the pseudo-colored image calculated by the LSGA method, there is also a confusion problem as well as in the basic GA. When analyzing Figure 10c, the central right region has a dark blue color very close to a region that was colored with black. These problems do not occur in the pseudo-colorization performed by the proposed technique with the addition of adaptive rules.
- **Maps:** All methods used many shades of blue to pseudo-color this image, which can cause some visual confusion. For example, in the top-right section of the pseudo-colored image using the LSAGA technique (Figure 11d), we can see three colored regions with similar shades of blue. In addition, the pseudo-colored image with GA (Figure 11b) presents more serious issues. In detail, note that there is a small circular subregion in the right section of all pseudo-colored images, however, this subregion is almost visually imperceptible in the pseudo-colored image with GA.
- **Mosaic:** That is an image in which the methods have great difficulty for pseudo-coloring due to their high number of sub-regions. However, even though all pseudo-colored images have a high degree of visual separability between their regions, it is possible to see that the pseudo-colored version by GA (Figure 12b) has in its top-left part several shades of yellow in neighboring areas and has in its bottom-right section neighboring regions colored with similar shades of red and pink. Something similar also occurs with the pseudo-colored image with the LSGA (Figure 12c).
- **Bicycle:** The image has $K = 6$ sub-regions and a small number of connections are defined by the neighborhood matrix. Therefore, all techniques have good visual results. However, all images show shades of blue in some pair of neighboring sub-regions, but this does not compromise the visual distinction of each sub-region, since the shades of blue used are very dissimilar.
- **Horses:** Analyzing the image, we can see that all methods present good pseudo-coloring results, with dissimilar color assignments in neighboring areas. However, this fact does not happen, for example, in the GA pseudo-colored image (Figure 14c), in which the blue horse in the central part of the image is pseudo-colored with a color visually similar to the background color.
- **Motorcycle:** This image has a reasonably large number of sub-regions ($K = 18$), yet all the techniques were able to associate a good pseudo-coloring for each sub-region, we highlight some negative points, such as in the case of the pseudo-colored image by LSAGA (Figure 15e), which contains two people in the top-left part of the background that represent adjacent areas and that have been colored with shades visually similar in blue.
- **Lunch:** That image is also highly complex, due to the number of sub-regions ($K = 20$) that define a connection in the neighborhood matrix. Thus, not all techniques perform well. For example, in the pseudo-colored image by our LSGA (Figure 16d), the table in the top-right part of the image receives a color similar to the background color. Something similar occurs with the color of the chair in the top-left section. Another example is presented by the pseudo-coloring obtained by our LSAGA (Figure 16e), which contains shades of blue in very close areas, such as the person in blue, in the central part of the image, sitting in a blue chair, in the case of the pseudo-colored image by mapped methods this does not occur.

The numerical evaluations performed for this set of images confirm that the proposed methodology achieves superior performance. Table 3 shows a set of statistical values that were calculated using the fitness value obtained by each technique after 50 executions, in which the used measures are: the best presented fitness (Max); the worst presented fitness (Min); the mean value of the presented fitness (Mean); the standard deviation (STD); and the average time (AT) for each execution

in seconds. In detail, we can see that the cub-LSAGA, the sig-LSAGA and the LSAGA methods find the best results in most images. Considering the nine images tested, cub-LSAGA found the best result in three images, sig-LSAGA found it in one image and LSAGA found it in five images. The methods that have an adaptation operator obtained the best results in the images, except for the Maps image that the best result was obtained by the LSGA method, however, the second-best value is presented by the sig-LSAGA, which is a method that also contains the mapped adaptive operator. In all the cases, a method that uses a mapped adaptive operator obtains the best Min value. In all images, the best statistical mean value was found by the methods that use the mapped adaptive operator. In addition, these same methods showed the smallest standard deviation in 7 of the 9 images considered, which shows that the methods proposed in this paper obtain more stable results when compared to other methods present in the literature, including our preliminary LSGA and LSAGA [21]. Analyzing the improvement of the results that were obtained by methods that make use of adaptive operator it is possible to verify that this operator has the capacity to guide LSGA and GA to find better solutions. However, GA is the method that obtained the lowest standard deviation in the case of the image "Mosaic", which has a larger number of sub-regions. This fact occurred because the solutions obtained by GA are restricted to a lower set of fitness values, with the average values obtained by GA being around 25 units less than the average values obtained by our techniques in the image "Mosaic".

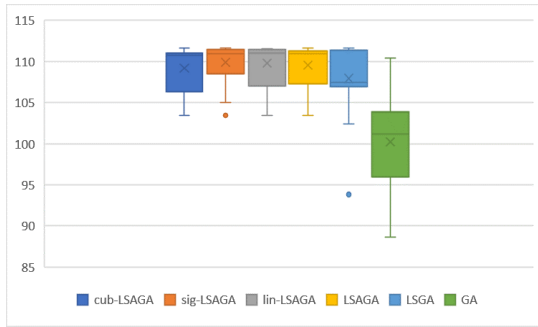
In Figure 17, a visual comparison of the performance of the techniques evaluated in the box plots is presented, so that we highlight information about the fitness values presented by the techniques during the 50 evaluations in each image that makes up the benchmark.

In images with a low number of regions, such as the image "Brains" which has $K = 6$ sub-regions and the image "Bicycle" which has $K = 7$ sub-regions, the proposed methodology presented a lower level of complexity and was able to obtain better results (Figure 17a) with less computational time, as shown in Table 3. This occurs because our methods do not create a parallel population for each color component of the RGB color space as the basic GA of Radlak and Smolka [14] does. If we observe cases where the number of regions increases, such as the cases of the images "Horses" ($K = 13$), "Motorcycle" ($K = 18$) and "Lunch" ($K = 20$), the difference between GA computational time compared to our techniques decreases. It is clear that considering more complex images (Figure 17c,e,g), that is, images with a greater number of sub-regions, our methodology also obtains better numerical performance. However, the technique is more computationally costly compared to basic GA. This fact is due to the number of generations used, which, as we can analyze in the convergence curves presented in Figure 18, is a much larger number than necessary since our technique reaches the value equal to that obtained by GA or, at least, a very close value with less than 20% of the total number of iterations. This is an indication that our methodology could present the same performance with only 20% of the total number of iterations used and, therefore, would consume only 20% of the total time presented.

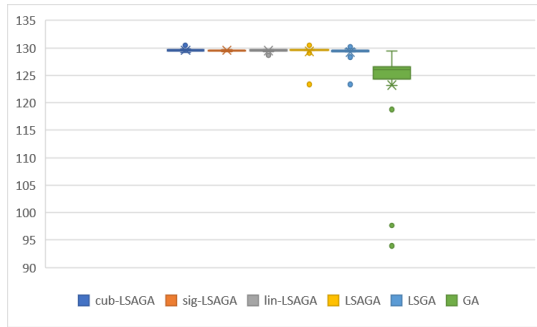
In this way, we conclude this CS noting that all the proposed techniques surpass basic GA in all the compared images. In addition, the mapped adaptive operators presented in this work produce more stable results in the RGB color space, in addition to producing pseudo-colorizations satisfactory to human vision. In contrast, computational time can be an opponent of our methodology in some situations, which must be circumvented by using more flexible configurations with fewer iterations.

Table 3. Statistical measurements of the fitness values presented in 50 runs of each method for the case study (CS) I images. The values highlighted in bold represent the best value of the evaluated measure available in the line.

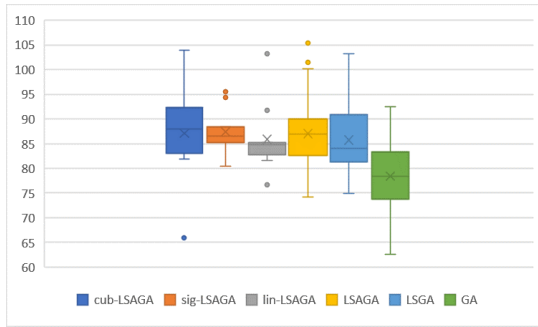
	Cub-LSAGA	Sig-LSAGA	Lin-LSAGA	LSAGA	LSGA	GA
Brain						
Image						
Max	111.5897	111.5897	111.5685	111.5897	111.5897	110.4434
Min	103.4240	103.4240	103.4240	103.4240	93.8159	88.6637
Mean	109.2113	109.9077	109.7841	109.5214	107.9427	100.2385
STD	2.9257	2.3071	2.4198	2.3563	3.9305	5.2523
AT (s)	39.9599	46.4401	45.1142	44.4965	33.9584	66.9531
Two Brains						
Image						
Max	103.8950	95.5579	103.2217	105.3690	103.2437	92.4752
Min	65.8645	80.4354	76.6066	74.1934	74.9277	62.6070
Mean	87.0920	87.4156	85.8846	87.0389	85.7157	78.3665
STD	9.4722	4.3749	6.8571	6.2291	6.9172	6.9151
AT (s)	160.4555	157.1275	163.7412	158.64375	144.0959	97.7062
Maps						
Image						
Max	93.7054	94.5564	91.4757	93.5399	97.0676	85.1661
Min	64.4656	66.1475	71.2507	66.8662	64.9360	55.9863
Mean	79.1163	83.6823	79.2163	81.2258	80.9922	72.4412
STD	8.0998	7.6433	6.1562	6.6415	6.4602	6.2322
AT (s)	222.9510	226.4532	226.4238	224.1458	203.8438	130.4597
Mosaic						
Image						
Max	80.4838	82.2895	80.6392	83.7880	81.5133	53.8912
Min	69.8773	63.7500	64.4427	61.2706	56.4168	43.3865
Mean	75.1833	73.5342	72.8356	73.5591	72.1155	48.7392
STD	3.4294	4.9239	5.1849	6.2869	5.7767	2.4751
AT (s)	990.0295	992.4713	991.2589	989.3925	936.8443	331.4825
Bicycle						
Image						
Max	130.4377	129.6426	129.6426	130.5154	130.2436	129.3809
Min	129.1017	129.3329	128.6836	123.3457	123.3457	93.9292
Mean	129.5630	129.5257	129.4862	129.3585	129.2352	123.2097
STD	0.2439	0.1219	0.2094	1.2629	1.2606	8.5688
AT (s)	40.3541	42.4896	48.4984	45.5781	34.0573	68.9844
Horse						
Image						
Max	111.4548	110.3349	111.0386	110.5609	108.4003	104.8786
Min	87.9770	89.0085	85.7987	85.3845	87.4883	79.3059
Mean	100.5157	100.6137	102.0270	97.4747	98.1981	90.5370
STD	7.6174	5.6810	6.5707	7.2756	5.8545	6.0897
AT (s)	72.5547	73.9412	75.2496	69.1302	53.5729	74.4583
Motorcycle						
Image						
Max	105.8383	101.0411	101.9775	103.5309	97.8846	101.8992
Min	76.7733	77.0004	80.2057	72.0469	74.3101	68.4586
Mean	90.3223	88.7403	89.1553	87.0939	87.0660	82.5259
STD	7.1476	6.4653	6.1515	6.6437	6.0711	8.9604
AT (s)	89.7412	85.9214	86.6647	89.6563	73.5052	85.3021
Lunch						
Image						
Max	98.6577	103.8848	99.3338	105.9357	100.8967	95.3289
Min	75.2950	75.2677	73.4639	70.43786	70.1481	65.2178
Mean	84.7076	88.4268	87.1569	82.81681	85.0893	80.1618
STD	6.0211	7.3464	6.1305	7.3561	7.2866	7.7565
AT (s)	103.2497	99.1463	98.8976	100.8177	82.4479	87.6302



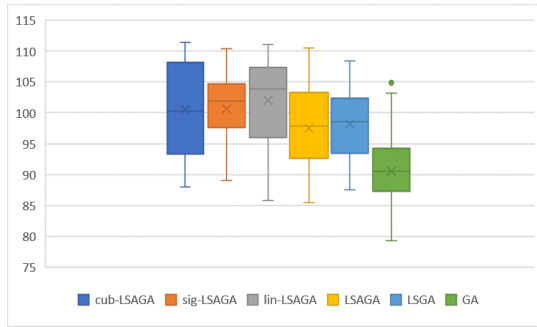
(a) Fitness in Brain.



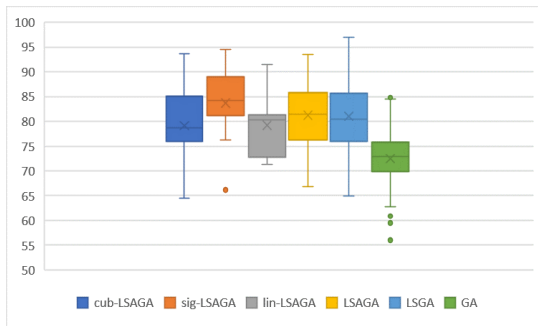
(b) Fitness in Bicycle.



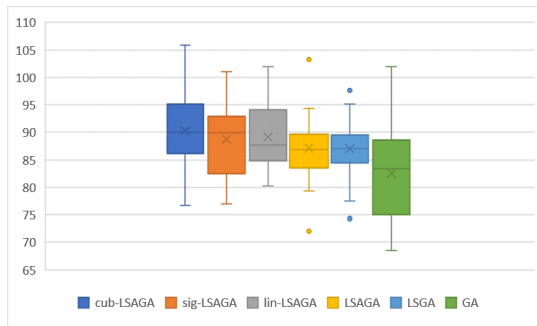
(c) Fitness in Two brains.



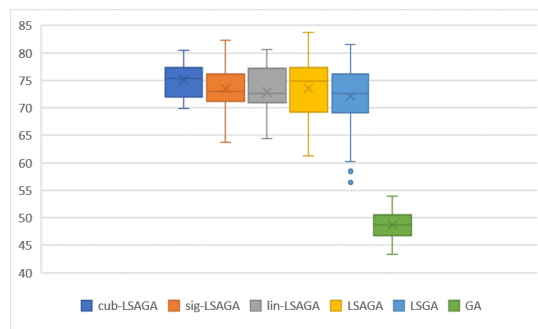
(d) Fitness in Horse.



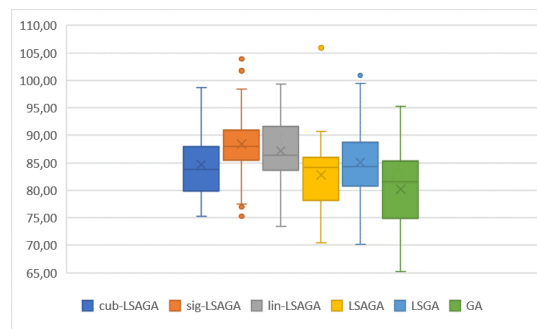
(e) Fitness in Maps.



(f) Fitness in Motorcycle.

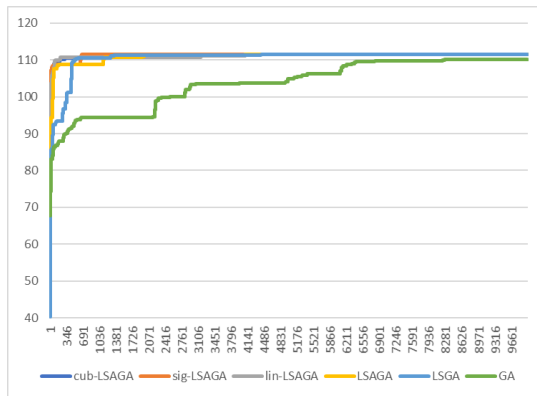


(g) Fitness in Mosaic.

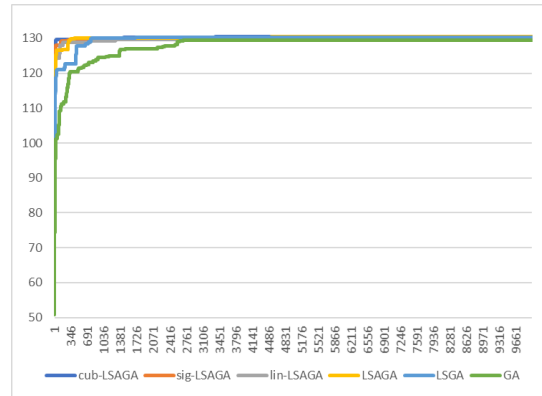


(h) Fitness in Lunch.

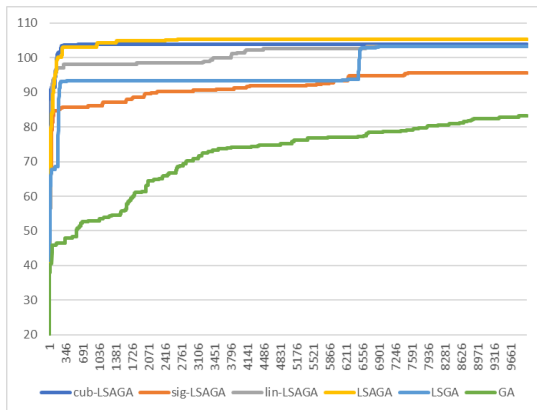
Figure 17. Numerical results (fitness values) displayed in boxplots of 50 independent runs of each technique.



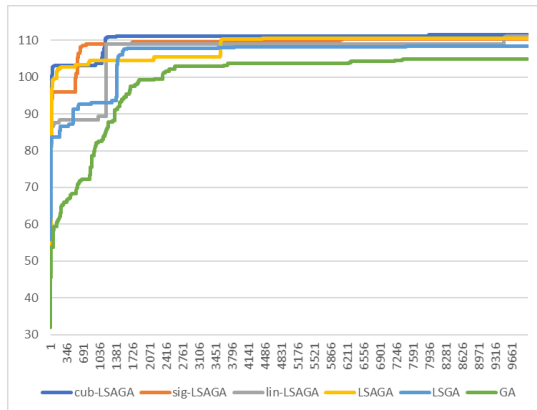
(a) Brain classes.



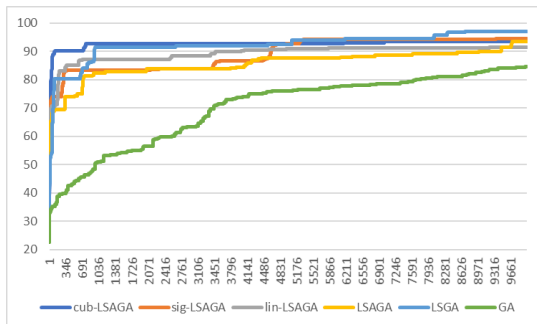
(b) Bicycle.



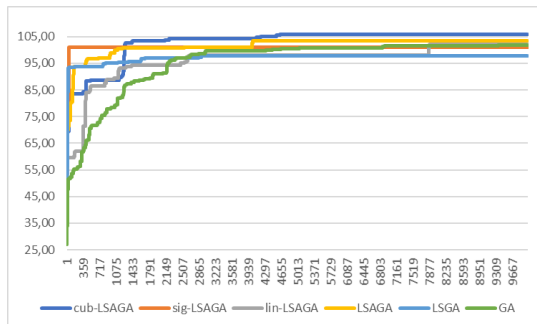
(c) Two brain classes.



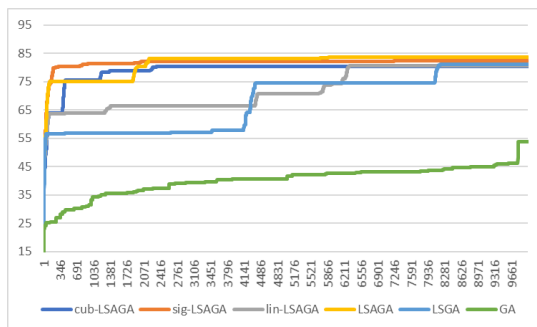
(d) Horses.



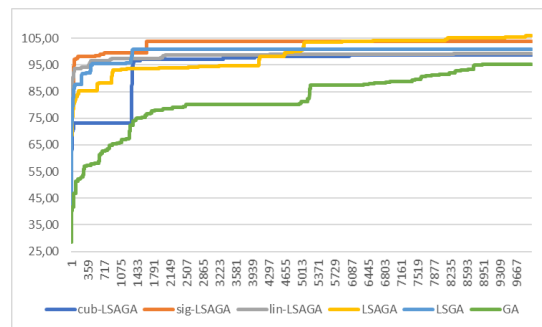
(e) Map classes.



(f) Motorcycle.



(g) Mosaic classes.



(h) Lunch.

Figure 18. Convergence analysis.

6.3. Case Study II: Quantitative Analysis Considering a Synthetic Benchmark of Images and $\mathbb{Y} = sRGB$

This second case study consists of evaluating the proposed techniques in 24 synthetic and abstract images, which are divided into fully-connected regions. In other words, this CS differs from the previous one in that we are going to evaluate the techniques in an extension of the Kelly [54] benchmark containing 24 images that do not necessarily exist in the real world. That is, in this CS, we are going to assume that we have available only the matrix of neighborhoods of a given grey-level image and execute the optimization methods on these matrices considering the \mathbb{Y} as being the sRGB color space. Specifically, we will evaluate the methodology in images I_K divided into K sub-regions $I_{K,1}, I_{K,2}, \dots, I_{K,K}$, such that the neighborhood matrix between these regions is in the form $\Delta_K = (\delta_{i,j}^K)_{i,j}$, with $\delta_{i,j}^K = 1$ in the case of $i \neq j$ and $\delta_{i,j}^K = 0$ otherwise, for $K \in \{2, 3, \dots, 25\}$, as mathematically defined in Equation (17).

$$\Delta_K := \mathbf{1}_{K \times K} - I_{K \times K} = \begin{bmatrix} 0 & 1 & 1 & \cdots & 1 \\ 1 & 0 & 1 & \cdots & 1 \\ 1 & 1 & 0 & \cdots & 1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & 1 & \cdots & 0 \end{bmatrix}_{K \times K}, \quad (17)$$

in which, $\mathbf{1}_{K \times K}$ is the $K \times K$ matrix with all the coordinates equals to 1, $I_{K \times K}$ is the identity matrix with dimension $K \times K$ and K is the number of sub-regions considered for each case for $K \in \{2, 3, \dots, 25\}$.

In this evaluation, we will take in comparison the same techniques considered in the previous CS (Table 2) with the same configuration used in CS I, and a Greedy Algorithm [14,36]. Table 4 shows the maximum fitness found for each technique between K colors, with $K \in \{2, 3, \dots, 25\}$. All techniques were performed 50 times.

Table 4. Max distance of colors in a fully-connected images. Green numbers are the max (best) values in line and red numbers are the min (worst) values in line.

Regions (K)	Cub-LSAGA	Sig-LSAGA	Lin-LSAGA	LSAGA	LSGA	GA [14]	Greedy [14]
2	249.2	249.2	249.2	249.2	249.2	249.2	233.85
3	166.11	166.11	166.11	166.11	166.11	166.11	164.64
4	130.64	130.64	130.64	130.64	129.64	130.21	129.64
5	111.59	111.59	111.59	111.59	111.59	111.43	108.81
6	102.58	102.58	102.58	102.58	102.58	102.48	93.78
7	94.7	94.7	94.7	94.7	93.75	93.04	86.95
8	86.15	86.15	86.15	86.15	86.13	84.78	80.03
9	81.49	81.49	81.49	81.49	80.43	78.68	74.45
10	77.8	77.8	77.8	77.8	74.9	74.65	71.92
11	68.1	69.43	69.43	69.43	68.1	66.71	65.77
12	65.61	65.61	65.61	65.61	64.65	64.84	61.86
13	64.26	64.26	64.26	64.26	62.5	63.13	57.79
14	60.89	60.89	60.89	60.89	59.1	58.8	57.32
15	57.16	57.16	57.16	57.16	56.7	53.52	55.27
16	51.53	55.82	55.82	55.82	51.53	51.01	53.4
17	53.56	53.56	53.56	53.56	52.55	49.67	51.32
18	50.56	50.56	50.56	50.56	50.47	48.17	49.42
19	45.08	50.5	47.9	50.5	48.24	45.08	47.9
20	44.67	49.26	49.26	49.26	45.83	44.67	47.57
21	46.54	45.68	45.68	45.68	44.78	42.66	46.54
22	46.36	46.36	44.23	46.36	44.87	41.63	44.23
23	44.74	44.74	44.74	43.62	43.28	41.3	44.74
24	39.77	43.86	39.77	43.86	42.22	39.77	43.61
25	38.55	43.09	38.55	43.09	41.82	38.55	41.98

In this scenario, sig-LSAGA presented the best performance in 23 of the 24 images considered. The sig-LSAGA was the method that obtained the best results in the comparison. In addition,

this technique did not present any worse results. Subsequently, our preliminary LSAGA method was the second method that presented the best values without presenting any worse value. Our lin-LSAGA method had the best result in 19 cases and the worst result in 2 cases; similar behavior to our cub-LSAGA, which had the best result in 18 cases and the worst result in 4 cases. These techniques present greater difficulties in images with more subregions (K larger). In addition, our LSGA technique presented the worst performance on only one occasion ($K = 4$), unlike the Greedy Algorithm which presents the worst performance in simpler images ($2 \leq K \leq 14$) and the basic GA that presents the worst performance in more complex images ($15 \leq K \leq 24$). This fact demonstrates that the technology proposed in this work is robust in comparison to the other techniques, especially when we use adaptive rules since our methods showed the best results in all the synthetic images considered and numerically surpassing the values presented by the GA and Greedy Algorithm.

6.4. Case Study III: Quantitative Analysis Considering Real World Images and \mathbb{Y} as Munsell Atlas Color Set

In this case study, we will evaluate our methodology on a set of real-world images, with the exception that we will consider \mathbb{Y} to be the well-established Munsell color set [57] with 1269 colors, also known as Munsell atlas color set [58]. This set of colors is a perceptually uniform color space and, like the sRGB space, it is also defined by three components: hue, value and chroma [59].

To the best of our knowledge, the most recent technique in the literature that evaluates this color space to obtain a high contrast set of colors is the local search (LS) technique by Bianco and Schettini [39]. The authors rate their methodology on a set of 5 real-world images. However, not all images used are available in the specialized literature. Therefore, to make the results reproducible, the authors make available the graphs that describe the neighborhood relationships between the subregions of each image. In this way, this CS, like the CS II, has only a quantitative assessment character, since we are not going to deal with the original images, but only with their neighborhood matrices. In this case, to conduct our analysis in this CS, we will consider all images used in [39] in which their sub-regions are connected by different graphs, totaling 4 out of 5. In this case, the images considered here are “Basketball”, “Texture”, “Oscar” and “Webpage”. The graph between the subregions of each of these images is represented in Figure 19. In addition, numerical details about these images can be viewed in Table 5.

Table 5. Details about the used images for the CS III.

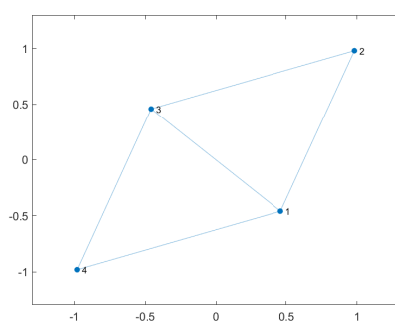
Image	Number of Subregions (K)	Nonzero Values in Δ	Number of Connections in the Graph
Basketball	4	10	5
Texture	8	38	19
Oscar	5	14	7
Webpage	8	24	12

For the evaluations of this case study, we will consider the same techniques considered in the previous case studies with the same configuration for each one. However, the basic GA compared in this situation is a version presented by Bianco and Schettini [39], which, according to the author, is an adaptation of the GA of Radlak and Smolka [14] to solve PsCP in Munsell’s color space. The authors also compare their methodology with a technique of randomly associating colors to images. Thus, we will also use the random association of colors for the sub-regions of the images for comparison, which is denoted by “Random”. This technique consists of randomly producing 1,000,000 possible color settings for each image and selecting the solution with the highest fitness value as a solution. Finally, we will present the values obtained by the Local Search (LS) technique of Bianco and Schettini [39]. In fact, the results of these last three techniques (GA, Random and LS) were collected directly from their original work. Whereas, our LSGA, LSAGA, lin-LSAGA, sig-LSAGA and cub-LSAGA techniques were executed 50 times independently of the configuration presented in the previous case studies. In Table 6, some statistical measures are presented regarding the 50 fitness

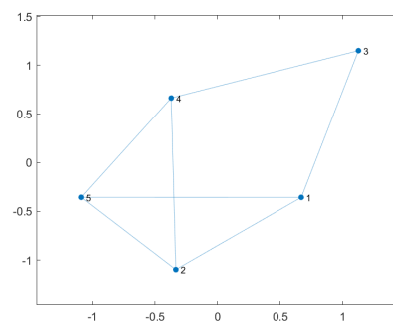
values obtained by each of our techniques and their comparison with the results presented by the techniques discussed by Bianco and Schettini [39].

According to Table 6, the proposed methods show superior performance concerning most of the statistical measures considered. Below we highlight some details regarding each of the images in isolation:

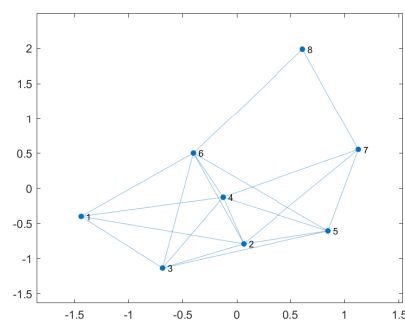
- **Basketball:** All the proposed methods have the same best value (Max) considering the 50 iterations. However, our cub-LSAGA method presents superior results considering the other statistical measures: worst value (Min), mean and standard deviation (STD). The LS method is deterministic and, therefore, presents the same solution in every evaluation. However, all of our methods have the worst values (Min), results higher than the result obtained by LS.
- **Texture:** In this case, LS presents competitive results to our techniques. However, except for our cub-LSAGA, all the techniques proposed in this text have the best fitness value greater than the value obtained by LS. In addition, all techniques that make use of adaptation strategies presented a value of fitness as the worst value (Min), better than the best value presented by the Random technique. Likewise, our lin-LSAGA had the worst value greater than the best value presented by GA. This technique also presented the best average among all the compared techniques.
- **Oscar:** In this case, except for cub-LSAGA, all of our techniques showed the highest best value. In addition, except for computational time, our sig-LSAGA presents the best statistical measures. It is worth mentioning that, all of the proposed techniques obtain a worse value for fitness than a better result than the other techniques with more than 10 units of difference.
- **Webpage:** For this image, our sig-LSAGA presents the best fitness and average value. Meanwhile, our cub-LSAGA stands out considering the worst fitness value and the standard deviation. However, all the proposed techniques have higher average values than the best values presented by the other techniques.



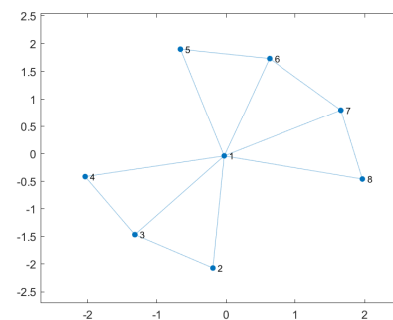
(a) Basketball.



(b) Oscar.



(c) Texture.



(d) Web.

Figure 19. Graph of the neighborhood matrix between the sub-regions of each grey-level image used in the work of Bianco and Schettini [39]

Table 6. Optimization in Munsell Atlas color space with 1269 colors. The symbol “-” means that the respective measurement is not available in the original work. The values highlighted in bold represent the best value of the evaluated measure available in the table line.

	Cub-LSAGA	Sig-LSAGA	Lin-LSAGA	LSAGA	LSGA	GA [14,39]	Random [39]	LS [39]
Image Basketball								
Max	83.8193	83.8193	83.8193	83.8193	83.8193	70.1	60.95	70.1
Min	80.9203	80.2733	80.9203	78.1239	74.6126	-	-	-
Mean	82.5140	82.2057	82.2179	82.0358	81.8048	61.68	17.76	-
STD	0.7331	0.8182	0.7500	1.2027	1.4551	-	-	-
AT (s)	38.3354	40.4001	38.7352	39.6563	32.3594	15.08	0.0013	3.03
Image Texture								
Max	58.7465	63.5075	63.0651	66.3612	63.0651	47.45	45.15	60.63
Min	46.7823	46.0836	47.7114	45.8120	36.5557	-	-	-
Mean	53.0731	54.4095	55.4383	54.6570	52.4303	43.75	17.78	-
STD	3.3279	4.0777	3.4021	4.6342	5.7014	-	-	-
AT (s)	58.1205	59.9801	59.2552	50.5365	35.8229	29.48	0.0021	18.87
Image Oscar								
Max	82.4165	83.8193	83.8193	83.8193	83.8193	59.76	56.44	63.5
Min	78.1239	79.5652	78.1239	77.3148	76.4953	-	-	-
Mean	81.8081	82.0605	81.9420	81.7832	81.2568	54.74	18.74	-
STD	1.3161	0.9360	1.0757	1.4110	1.6553	-	-	-
AT (s)	39.8854	42.1464	40.7552	39.0000	28.5573	17.93	0.0015	5.56
Image Webpage								
Max	80.9203	82.3743	82.2811	82.1702	80.9203	50.13	42.93	62.33
Min	67.2430	62.8774	64.4095	62.8774	62.2342	-	-	-
Mean	75.7342	76.9152	76.1263	76.0651	73.6511	44.15	14.89	-
STD	4.0863	4.7845	4.6217	4.6063	5.5967	-	-	-
AT (s)	50.1254	47.8946	47.7740	48.6667	37.0156	30.36	0.002	18.99

In summary, we can see by observing Table 6 that our techniques surpass the results that make up the current state-of-the-art with special emphasis on the stability of techniques that use mapped adaptive operators. For example, considering the image “Basketball”, our cub-LSAGA, sig-LSAGA and lin-LSAGA methods showed standard deviations less than 1. In this case, our cub-LSAGA presented a standard deviation of 0.7331. Something similar occurs in the image “Oscar”, in which our sig-LSAGA has a standard deviation of 0.9360. Thus, this case study also confirms that the mapped adaptive methods proposed in this text help to maintain the stability of the method.

Regarding the average processing time of each technique, we realized that our methodology is, in fact, more computationally costly in all cases of this CS. Making our methodology used only in offline applications and it is not ideal to use it in applications that require real-time response. However, it presents superior results in comparison to the other techniques that define the state of the art.

7. Conclusions

In this paper, a new GA was proposed with the addition of specialized operators to reinforce the local search and with the addition of rules to adjust the parameters using symmetric mapping functions to treat the PsCP. All the steps contained in the proposed technique were presented in detail to facilitate the reproduction of the methodology.

We evaluated the proposed material in three case studies: in the first, our techniques were used to assign artificial coloring to real-world images and we performed qualitative and quantitative analyzes of the results obtained; in the second, our techniques were evaluated quantitatively in a benchmark of theoretical and synthetic images whose sub-regions are fully-connected; and in the third, we quantitatively evaluate our methodology when performing optimization in the Munsell color space. In the three case studies, the proposed methods obtained superior results when compared with other

techniques presented in the state of the art. The proposed technique with the mapped adaptive rules presented the best results and it is the most stable version of our method. The combination of the inclusion of local search procedures and the use of more elaborate adaptive rules helped the method to obtain better results while maintaining a good genetic variability in the population and avoiding possible premature convergence.

In future works, we intend to expand the use of our mapped adaptation operators to update any GA configuration, such as the number of individuals in the population, the number of generations of the method, etc. In addition, we intend to study the effect that these strategies can have on other types of recent proposed meta-heuristics, such as the squirrel search algorithm [60], when applied to solve the PsCP. We believe that, even though the results obtained by the proposed method surpass the other state-of-the-art techniques, it is still necessary to conduct a study on techniques that can be used to reduce the computational time necessary to obtain the solutions to the problem, since if the image to be colored has a high number of sub-regions then the proposed method will need a very large amount of computational time to obtain good solutions. We also intend to use the proposed methodology to solve another class of problems in the literature, such as visual threat detection in X-ray scans [2,61].

Author Contributions: Conceptualization, M.S.V., O.M.J. and R.C.C.; methodology, M.S.V., O.M.J. and R.C.C.; software, M.S.V. and R.C.C.; validation, M.S.V. and R.C.C.; formal analysis, M.S.V. and R.C.C.; investigation, M.S.V. and R.C.C.; writing—original draft preparation, M.S.V. and R.C.C.; writing—review and editing, O.M.J.; visualization, R.C.C.; supervision, O.M.J.; project administration, O.M.J. All authors have read and agreed to the published version of the manuscript.

Acknowledgments: This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior—Brasil (CAPES)—Finance Code 001, and by the São Paulo Research Foundation (FAPESP), process number 15/14358-0. We also thank the authors Radlak and Smolka [14] for sharing their GA code with us.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

References

1. Anitha, U.; Malarkkan, S.; Premalatha, J.; Manonmani, V. Comparison of standard edge detection techniques along with morphological processing and pseudo coloring in sonar image. In Proceedings of the 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS), Pudukkottai, India, 24–26 February 2016; pp. 1–4.
2. Dmitruk, K.; Denkowski, M.; Mazur, M.; Mikołajczak, P. Sharpening filter for false color imaging of dual-energy X-ray scans. *Signal Image Video Process.* **2017**, *11*, 613–620. [[CrossRef](#)]
3. Etehadtavakol, M.; Ng, E.Y. Color segmentation of breast thermograms: A comparative study. In *Application of Infrared to Biomedical Sciences*; Springer: Singapore 2017; pp. 69–77.
4. Semary, N.A. A proposed Hsv-Based Pseudo-Coloring Scheme For Enhancing Medical Images. *Comput. Sci. Inf. Technol.* **2018**. [[CrossRef](#)]
5. Pipatnoraseth, T.; Phognsuphap, S.; Wiratkapun, C.; Tanawongsuwan, R.; Sajjacholapunt, P.; Shimizu, I. Breast Microcalcification Visualization Using Pseudo-Color Image Processing. In Proceedings of the 2019 12th Biomedical Engineering International Conference (BMEiCON), Ubon Ratchathani, Thailand, 19–22 November 2019; pp. 1–5.
6. Zeng, X.; Tong, S.; Lu, Y.; Xu, L.; Huang, Z. Adaptive Medical Image Deep Color Perception Algorithm. *IEEE Access* **2020**, *8*, 56559–56571. [[CrossRef](#)]
7. Pablo, R.; Verstraelen, P.; Simona, T.; Muthukumar, G.; Marzia, P.; Isabel, P.; Nathalie, C.; De Miguel-Pérez, D.; Riccardo, A.; Bals, S.; et al. Improving extracellular vesicles visualization: From static to motion. *Sci. Rep.* **2020**, *10*, 1–9.
8. Wu, M.; Jin, X.; Jiang, Q.; Lee, S.; Guo, L.; Di, Y.; Huang, S.; Huang, J. Remote Sensing Image Colorization Based on Multiscale SEnet GAN. In Proceedings of the 2019 12th International Congress on Image and Signal Processing, BioMedical Engineering and Informatics (CISP-BMEI), Huaqiao, China, 19–21 October 2019; pp. 1–6.

9. Beauchemin, M. Unsupervised colour coding for visualization of categorical maps. *Remot. Sens. Lett.* **2019**, *10*, 77–85. [[CrossRef](#)]
10. Zheng, P.; Zhang, J. Quantitative nondestructive testing of wire rope based on pseudo-color image enhancement technology. *Nondestruct. Test. Eval.* **2019**, *34*, 221–242. [[CrossRef](#)]
11. Burger, P.; Gillies, D. *Interactive Computer Graphics: Functional, Procedural and Device-Level Methods*; Addison-Wesley: Boston, MA, USA, 1989.
12. Moodley, K.; Murrell, H. A colour-map plugin for the open source, Java based, image processing package, ImageJ. *Comput. Geosci.* **2004**, *30*, 609–618. [[CrossRef](#)]
13. Carter, R.C.; Carter, E.C. High-contrast sets of colors. *Appl. Opt.* **1982**, *21*, 2936–2939. [[CrossRef](#)]
14. Radlak, K.; Smolka, B. Visualization enhancement of segmented images using genetic algorithm. In Proceedings of the 2014 International Conference on Multimedia Computing and Systems (ICMCS), Marrakech, Morocco, 14–16 April 2014; pp. 391–396.
15. Zang, W.; Ren, L.; Zhang, W.; Liu, X. A cloud model based DNA genetic algorithm for numerical optimization problems. *Future Gener. Comput. Syst.* **2018**, *81*, 465–477. [[CrossRef](#)]
16. Asadzadeh, L. A local search genetic algorithm for the job shop scheduling problem with intelligent agents. *Comput. Ind. Eng.* **2015**, *85*, 376–383. [[CrossRef](#)]
17. Lin, C. An adaptive genetic algorithm based on population diversity strategy. In Proceedings of the 2009 Third International Conference on Genetic and Evolutionary Computing, Guilin, China, 14–17 October 2009; pp. 93–96.
18. Shojaedini, E.; Majd, M.; Safabakhsh, R. Novel adaptive genetic algorithm sample consensus. *Appl. Soft Comput.* **2019**, *77*, 635–642. [[CrossRef](#)]
19. Zhu, K.Q. A diversity-controlling adaptive genetic algorithm for the vehicle routing problem with time windows. In Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence, Sacramento, CA, USA, 3–5 November 2003; pp. 176–183.
20. Xing, Y.; Chen, Z.; Sun, J.; Hu, L. An improved adaptive genetic algorithm for job-shop scheduling problem. In Proceedings of the Third International Conference on Natural Computation (ICNC 2007), Haikou, China, 24–27 August 2007; pp. 287–291.
21. Contreras, R.C.; Morandin Junior, O.; Viana, M.S. A New Local Search Adaptive Genetic Algorithm for the Pseudo-Coloring Problem. In *Advances in Swarm Intelligence*; Tan, Y., Shi, Y., Tuba, M., Eds.; Springer International Publishing: Cham, Switzerland, 2020; pp. 349–361.
22. Appel, K.I.; Haken, W. *Every Planar Map is Four Colorable*; American Mathematical Soc.: Providence, RI, USA, 1989; Volume 98.
23. Jensen, T.R.; Toft, B. *Graph Coloring Problems*; John Wiley & Sons: New York, NY, USA, 2011; Volume 39.
24. Christensen, J.; Shieber, S.M.; Marks, J. *Placing Text Labels on Maps and Diagrams*; Academic Press: Cambridge, MA, USA, 1994.
25. Welsh, T.; Ashikhmin, M.; Mueller, K. Transferring Color to Greyscale Images. In Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques; San Antonio, TX, USA, 23–26 July 2002.
26. Zare, M.; Lari, K.B.; Jampour, M.; Shamsinejad, P. Multi-GANs and its application for Pseudo-Coloring. In Proceedings of the 2019 4th International Conference on Pattern Recognition and Image Analysis (IPRIA), Tehran, Iran, 6–7 March 2019; pp. 1–6.
27. Fang, F.; Wang, T.; Zeng, T.; Zhang, G. A Superpixel-based Variational Model for Image Colorization. *IEEE Trans. Vis. Comput. Graph.* **2019**, *26*, 2931–2943. [[CrossRef](#)] [[PubMed](#)]
28. Wan, S.; Xia, Y.; Qi, L.; Yang, Y.; Atiquzzaman, M. Automated Colorization of a Grayscale Image With Seed Points Propagation. *IEEE Trans. Multimed.* **2020**, *22*, 1756–1768. [[CrossRef](#)]
29. Li, B.; Lai, Y.K.; Rosin, P.L. A review of image colourisation. In *Handbook Of Pattern Recognition And Computer Vision*; World Scientific: Singapore, 2020; p. 139.
30. Dai, J.; Zhou, S. Computer-aided pseudocolor coding of gray images: complementary color-coding technique. In *Electronic Imaging and Multimedia Systems*; Li, C.S., Stevenson, R.L., Zhou, L., Eds.; International Society for Optics and Photonics (SPIE): Beijing, China, 1996; Volume 2898, pp. 186–191.
31. Campadelli, P.; Mora, P.; Schettini, R. Color set selection for nominal coding by Hopfield networks. *Vis. Comput.* **1995**, *11*, 150–155. [[CrossRef](#)]
32. Green-Armytage, P. A colour alphabet and the limits of colour coding. *JAIIC J. Int. Colour Assoc.* **2010**, *5*, 1–23.

33. Campadelli, P.; Posenato, R.; Schettini, R. An algorithm for the selection of high-contrast color sets. *Color Res. Appl.* **1999**, *24*, 132–138. [[CrossRef](#)]
34. Brill, M.H.; Robertson, A.R.; Schanda, J. Open problems on the validity of Grassmann's laws. *Color. Underst. Cie Syst.* **2007**, 245–259.
35. Glasbey, C.; van der Heijden, G.; Toh, V.F.; Gray, A. Colour displays for categorical images. *Color Res. Appl.* **2007**, *32*, 304–309. [[CrossRef](#)]
36. Bianco, S.; Citrolo, A.G. High Contrast Color Sets under Multiple Illuminants. In *Computational Color Imaging*; Tominaga, S., Schettini, R., Trémeau, A., Eds.; Springer: Berlin/Heidelberg, Germany, 2013; pp. 133–142.
37. Kirkpatrick, S.; Gelatt, C.D.; Vecchi, M.P. Optimization by simulated annealing. *Science* **1983**, *220*, 671–680. [[CrossRef](#)]
38. Holland, J.H. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*; MIT Press: Cambridge, MA, USA, 1992.
39. Bianco, S.; Schettini, R. Unsupervised color coding for visualizing image classification results. *Inf. Vis.* **2018**, *17*, 161–177. [[CrossRef](#)]
40. Connolly, C.; Fleiss, T. A study of efficiency and accuracy in the transformation from RGB to CIELAB color space. *IEEE Trans. Image Process.* **1997**, *6*, 1046–1048. [[CrossRef](#)] [[PubMed](#)]
41. Mahyar, F.; Cheung, V.; Westland, S.; Henry, P. Investigation of complementary colour harmony in CIELAB colour space. In Proceedings of the AIC Midterm Meeting, Hangzhou, China, 12–14 July 2007.
42. Hoerer, T.; Kuenzer, C. Object Detection and Image Segmentation with Deep Learning on Earth Observation Data: A Review-Part I: Evolution and Recent Trends. *Remot. Sens.* **2020**, *12*, 1667. [[CrossRef](#)]
43. Pare, S.; Kumar, A.; Singh, G.; Bajaj, V. Image segmentation using multilevel thresholding: A research review. *Iran. J. Sci. Technol. Trans. Electr. Eng.* **2020**, *44*, 1–29. [[CrossRef](#)]
44. Taghanaki, S.A.; Abhishek, K.; Cohen, J.P.; Cohen-Adad, J.; Hamarneh, G. Deep semantic segmentation of natural and medical images: A review. *Artif. Intell. Rev.* **2020**, *14*, 1–42. [[CrossRef](#)]
45. Viana, M.S.; Morandin Junior, O.; Contreras, R.C. A Modified Genetic Algorithm With Local Search Strategies And Multi-Crossover Operator For Job Shop Scheduling Problem. *Sensors* **2020**, *20*, 5440. [[CrossRef](#)] [[PubMed](#)]
46. Zames, G.; Ajlouni, N.; Holland, J.; Hills, W.; Goldberg, D. Genetic algorithms in search, optimization and machine learning. *Inf. Technol. J.* **1981**, *3*, 301–302.
47. Goldberg, D.E. Genetic algorithms in search. *Optim. Mach.* **1989**, *1*.
48. Dioşan, L.; Oltean, M. Evolving crossover operators for function optimization. In *European Conference on Genetic Programming*; Springer: Berlin/Heidelberg, Germany, 2006; pp. 97–108.
49. Ombuki, B.M.; Ventresca, M. Local search genetic algorithms for the job shop scheduling problem. *Appl. Intell.* **2004**, *21*, 99–109. [[CrossRef](#)]
50. Amjad, M.K.; Butt, S.I.; Kousar, R.; Ahmad, R.; Agha, M.H.; Faping, Z.; Anjum, N.; Asgher, U. Recent research trends in genetic algorithm based flexible job shop scheduling problems. *Math. Probl. Eng.* **2018**, *2018*. [[CrossRef](#)]
51. Bäck, T.; Schwefel, H.P. An overview of evolutionary algorithms for parameter optimization. *Evol. Comput.* **1993**, *1*, 1–23. [[CrossRef](#)]
52. Hinterding, R. Gaussian mutation and self-adaption for numeric genetic algorithms. In Proceedings of the 1995 IEEE International Conference on Evolutionary Computation, Perth, Australia, 29 November–1 December 1995; p. 384.
53. Bulmer, M.G. *Principles of Statistics*; Courier Corporation: New York, NY, USA, 1979.
54. Kelly, K.L. Twenty-two colors of maximum contrast. *Color Eng.* **1965**, *3*, 26–27.
55. Everingham, M.; Winn, J. The pascal visual object classes challenge 2012 (voc2012) development kit. *Pattern Anal. Stat. Model. Comput. Learn. Tech. Rep.* **2011**, *8*, 1–32.
56. Everingham, M.; Eslami, S.A.; Van Gool, L.; Williams, C.K.; Winn, J.; Zisserman, A. The pascal visual object classes challenge: A retrospective. *Int. J. Comput. Vis.* **2015**, *111*, 98–136. [[CrossRef](#)]
57. Newhall, S.M.; Nickerson, D.; Judd, D.B. Final report of the OSA subcommittee on the spacing of the Munsell colors. *JOSA* **1943**, *33*, 385–418. [[CrossRef](#)]
58. Munsell, A.H. *Munsell Book of Color: Matte Finish Collection*, Munsell Color: Baltimore, MD, USA, 1992.
59. Campadelli, P.; Schettini, R.; Zuffi, S. A system for the automatic selection of conspicuous color sets for qualitative data display. *IEEE Trans. Geosci. Remot. Sens.* **2001**, *39*, 2283–2286. [[CrossRef](#)]

60. Jain, M.; Singh, V.; Rani, A. A novel nature-inspired algorithm for optimization: Squirrel search algorithm. *Swarm Evol. Comput.* **2019**, *44*, 148–175. [[CrossRef](#)]
61. Xue, Z.; Blum, R.S. Concealed weapon detection using color image fusion. In Proceedings of the 6th International Conference on Information Fusion, Cairns, Queensland, Australia, 8–11 July 2003; pp. 622–627.

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).