



# A New Local Search Adaptive Genetic Algorithm for the Pseudo-Coloring Problem

Rodrigo Colnago Contreras<sup>1</sup>, Orides Morandin Junior<sup>2</sup>,  
and Monique Simplicio Viana<sup>2</sup>(✉)

<sup>1</sup> University of São Paulo, São Carlos, Brazil  
contreras@alumni.usp.br

<sup>2</sup> Federal University of São Carlos, São Carlos, Brazil  
{orides,monique.viana}@ufscar.br

**Abstract.** Several applications result in a gray level image partitioned into different regions of interest. However, the human brain has difficulty in recognizing many levels of gray. In some cases, this problem is alleviated with the attribution of artificial colors to these regions, thus configuring an application in the area of visualization and graphic processing responsible for categorizing samples using colors. However, the task of making a set of distinct colors for these regions stand out is a problem of the NP-hard class, known as the pseudo-coloring problem (PsCP). In this work, it is proposed to use the well-known meta-heuristic Genetic Algorithm together with operators specialized in the local search for solutions as well as self-adjusting operators responsible for guiding the parameterization of the technique during the resolution of PsCPs. The proposed methodology was evaluated in two different scenarios of color assignment, having obtained the best results in comparison to the techniques that configure the state of the art.

**Keywords:** Genetic Algorithm · Local search · Adaptive operator · Visualization · Pseudo-coloring problem

## 1 Introduction

Many problems today consist of extracting visual patterns from images that are only available in gray levels. Thus, in this type of situation, a professional should view an image that presents details in the same tones and infer complex classification hypotheses. However, studies show that human vision has difficulty in differentiating monochrome tones [15]. This fact is related to the low capacity of the brain to categorize signals of similar frequencies. Therefore, many researchers have dedicated themselves to the development of specialized methodologies for assigning artificial colors to images originally arranged in gray levels to facilitate the extraction of visual patterns. For example, we can see advances in detecting edges in sonar images [2], detecting tumors in the chest [11], detecting weapons in baggage [10], visual enhancement of medical images [18], etc.

In the aforementioned works, the central strategy used in the visual enhancement of information consists of the generation of a set of colors, which must be as distinct as possible, with a different coloring attributed to each region of the evaluated image. This situation is known as the Pseudo-Colorization Problem (PsCP), or construction of high contrast sets [7]. The main complication presented in PsCP is the similarity between colors used. That is, neighboring regions in an image may have similar colors and, consequently, their visualization may be compromised. To solve this need it is necessary to use some optimization technique in order to distance the colors used as far as possible. However, this optimization becomes unfeasible as the number of regions in the image increases, since this problem is contained in the class of problems NP-hard. In this way, many authors mitigate the problem with the use of meta-heuristics. Radlak and Smolka [17] propose a methodology using a measure of color contrast optimized with a Genetic Algorithm (GA). The method performs searches in the RGB color domain, making neighboring regions colored with the most distinct colors possible. However, simplified versions of GA can present several problems such as inefficient search and premature convergence [22].

To get around these problems, Asadzadeh [3] proposes aLSGA, which consists of a GA that uses specialized operators in local search to solve the combinatory problem Job Shop Scheduling. In detail, the author presents a GA with a local search operator who works in conjunction with the mutation operator and a local search operator who performs massive exploitation in search space. The methodology proved to be superior to traditional GA and similar evolutionary methods in solving this problem, which also belongs to the NP-hard class. In this work, we propose to improve aLSGA so that it is specialized in the PsCP solution. Also, we propose the addition of adaptive rules [23], [13], [19] so that the method developed makes automatic adjustments during its execution to perform the search process without premature convergence or inefficient search.

The paper is organized into 6 sections. Specifically, in Sect. 2 we describe the mathematical formulation of PsCP. In Sect. 3, the details of the proposed algorithm and each of the operators that compose it are presented. In Sect. 4, experiments, results, and comparison with other methods in the literature are presented in two different test scenarios. Finally, in Sect. 5, conclusions and possible directions for future work are presented.

## 2 Formulation of Pseudo-Coloring Problem

In this work, we will adopt a model similar to that of Radlak and Smolka [17] to approach PsCP. To this end, the objective is to allocate to a  $I$  image, already segmented into  $K$  disjoint regions, a set of  $K$  colors so that these colors are as distinct as possible in neighboring image regions. Mathematically, let  $I_1, I_2, \dots, I_K$  be the pre-segmented sub-regions of  $I$ , and the neighborhood matrix  $\Delta$  whose coordinates  $\delta_{i,j}$  are equal to 1 if the region  $I_i$  is neighbor to  $I_j$  or  $\delta_{i,j}$  are equal to 0 if the region  $I_i$  is not neighbor to  $I_j$  or in the case of  $i = j$ . Thus, the situation consists in determining a set of  $K$  colors  $\mathcal{C} = \{c_1, c_2, \dots, c_K\}$  to coloring

the regions of  $I$  such that the value of the function  $F$  of Eq. (1) is the **largest** possible value, configuring this problem in a maximization problem.

$$F(\mathcal{C}) := \min \{ \delta_{i,j} \cdot d(c_i, c_j) \mid \delta_{i,j} \neq 0, \quad i, j \in \{1, 2, \dots, K\} \}, \quad (1)$$

in which  $d(\cdot, \cdot)$  is a distance function in some color space.

Specifically, each color  $c_i \in \mathcal{C}$  is represented by a triple of integer values that define the RGB coloration to be associated to the region  $I_i$ . Thus, each color  $c_i$  belongs to the sRGB set, which in this work is represented by the set  $\{0, 1, \dots, 255\}^3$ , and, consequently,  $\mathcal{C} \subset \{0, 1, \dots, 255\}^{3 \cdot K}$ . Furthermore, 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 [8] with illuminant D50, since this space is favorable for performing visual distinction of colors [14].

### 3 Local Search Adaptive Genetic Algorithm for PsCP

In this section, we describe in detail the proposed algorithm, which was developed specifically to present good solutions for PsCP. In such a way that each operator of the method has in its description the explanation of the idea that composes the technique so that the reproducibility is done without difficulty. For the best of our knowledge, there is no record in the specialized literature of any GA-based technique that has a dedicated operator in carrying out a massive local search around the same individual and that its functioning is adjusted with an applied adaptative operator for the PsCP. Thus, this work has the main purpose of introducing a technique with these characteristics, which is entitled Local Search Adaptive Genetic Algorithm (LSAGA). Therefore, we present, in summary, the contributions of this work:

1. An adaptation of the Asadzadeh method [3] to PsCP. In particular, a new operator of massive local search for the best individual in the population is proposed;
2. Addition of an adaptation operator, inspired by [13], which coordinates the use of the basic operators of the algorithm in order to avoid premature convergence and optimize the method's exploitation and exploration capacity,
3. The advancement of experimental results in benchmarks that define the state-of-the-art.

#### 3.1 Chromosome Decoding

In [17], three chromosome populations evolve in parallel in a GA, one dedicated to the R component, another to the G component and another to the B component of each of the  $K$  segmented regions in  $I$ . In our work, we propose that a single population of chromosomes is evolved during the execution of the method. In detail, each chromosome in the proposed modeling is associated with a set of colors for the regions in  $I$ . That is, each gene on a chromosome is formed by a

color in sRGB which must be associated with a specific region of  $I$ . Mathematically, the genetic representation of a  $\mathcal{C}$  chromosome adopted in this work follows the definition of Eq. (2).

$$\mathcal{C} = [(R_1, G_1, B_1), (R_2, G_2, B_2), \dots, (R_K, G_K, B_K)], \tag{2}$$

in which  $(R_i, G_i, B_i) \in \text{sRGB}$  is the color to be associated with the region  $I_i$ .

It is important to note that, in practice, the proposed encoding models the chromosome as a vector of  $3 \cdot K$  coordinates with values between 0 and 255 and not a subset of  $K$  colors in sRGB. As a consequence of this, the coordinates of each chromosome carry an order that refers to the sub-regions of  $I$ . That is, the first three coordinates of a chromosome represent the RGB color of the first region of  $I$ , the subsequent three coordinates represent the RGB color of the second region of  $I$ , and so on.

### 3.2 Fitness Function

The genetic representation proposed in the Sect. 3.1 makes it natural to define the fitness function as a simple isomorphism with the  $F$  function of the Eq. (1). Mathematically, we define the fitness of a chromosome  $\mathcal{C} = [(R_1, G_1, B_1), (R_2, G_2, B_2), \dots, (R_K, G_K, B_K)] \in \text{sRGB}^K$  the function  $\bar{F}$ , presented in Eq. (3).

$$\begin{aligned} \bar{F} : \text{sRGB}^K &\longrightarrow \mathbb{R}_+ \\ (c_1, c_2, \dots, c_K) &\longmapsto \bar{F}(c_1, c_2, \dots, c_K) := F(\{c_1, c_2, \dots, c_K\}) \end{aligned} \tag{3}$$

### 3.3 Selection Process

The process of selecting individuals in evolutionary methods is important to ensure that individuals who are better adapted, or with better fitness value, have a greater chance of exchanging genetic information, or reproducing, during the crossover operator. Besides, the selection method is responsible for preserving individuals with good fitness values during the generation of new populations of the algorithm. In this paper, we make use of the roulette wheel method ensuring that the best individual in the current population will reproduce [21] and will also be transferred to the new population.

### 3.4 Crossover Operator

To ensure that the exchange of genetic information between two individuals is defined by valid colors, it is important that the individuals generated in the crossover process respect the limits of sRGB. For this, we propose an extension of the well-known convex crossover [9], in which each of the colors represented by the generated individuals is the result of the random convex combination of the colors represented by their parents. Specifically, two intermediate offsprings are generated,  $\hat{\text{Kid}}_1$  and  $\hat{\text{Kid}}_2$ , which are the result of the convex combination

between each of the parents' colors,  $\text{Parent}_1$  and  $\text{Parent}_2$ , and therefore belong to  $\mathbb{R}^{3K}$ . Then, these individuals are projected in the feasible space  $\text{sRGB}^K$  using the function  $\text{proj}_{\text{sRGB}^K}(\cdot)$ , which projects a given vector from  $\mathbb{R}^{3K}$  to the closest vector belonging to  $\text{sRGB}^K$ . This projection determines the  $\text{Kid}_1$  and  $\text{Kid}_2$  offsprings generated in this operator.

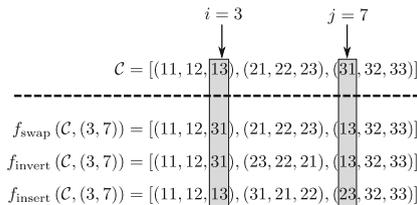
### 3.5 Mutation Operator

The mutation operator proposed in this work is based on the local search procedures presented for the first time in [3] and [16] to solve production scheduling problems. Our technique is an improvement that consists of the combination of these two methodologies. Therefore, the proposed mutation process consists of the eventual application of one of the following mutation subroutines:

- $\text{Mut}_1$ : In this case, successive  $N_{\text{Mut}_1}$  applications of a mutation function, randomly determined in a set of mutation functions, are carried out on a chromosome so that the beneficial perturbation are maintained and the method proceeds from them;
- $\text{Mut}_2$ : Unique and definitive application of a mutation function.

Thus, as soon as an individual is generated in the crossover operator, it has a probability  $p_{\text{mut}}$  of going through the mutation process. If selected, the individual also has a probability  $p_{\text{LS}}$  of going through a local search procedure, represented by the subroutine  $\text{Mut}_1$ , and a probability  $1 - p_{\text{LS}}$  of receiving an only perturbation according to the subroutine  $\text{Mut}_2$ .

In  $\text{Mut}_1$ , a group of mutation functions in the form of  $f_{\text{mut}}(\cdot, \cdot) : \text{sRGB} \times \{1, 2, \dots, K\}^2 \rightarrow \text{sRGB}$  must be defined. For this, we propose the use of the three most used mutation functions in solving combinatorial problems [3]:  $f_{\text{swap}}(\cdot, \cdot)$ ,  $f_{\text{invert}}(\cdot, \cdot)$  and  $f_{\text{insert}}(\cdot, \cdot)$ . In Fig. 1, an example of the perturbation caused by these functions on the same individual is presented. As these functions only perform permutations between the coordinates of a chromosome, it do not compromise the feasibility of the solutions generated.



**Fig. 1.** Schematic diagram of the functioning of three mutation functions on the same chromosome  $C$ .

In  $\text{Mut}_2$ , we propose to use only one mutation function. In this case, a simple Gaussian perturbation [4],  $f_{\text{Gauss}}(\cdot)$ , in which all genes on the chromosome receives a slightly random increase or decrease.

### 3.6 Massive Search Operator

This type of operator intends to carry out a more elaborate and systematic search around an individual from the population that presents a good adaptation, looking for neighbors of the same who are similar, but that has better fitness value. In other words, massive local search operators assume that around a good individual there may be better individuals more probability than in random regions of space. In [3], this operator performs successive swaps between the coordinates of the best individual in the population, maintaining beneficial perturbations. In our work, we propose a new massive local search operator specialized in looking for colors similar to the colors represented by the best individual in a population, through successive Gaussian perturbations. Specifically, considering  $\mathcal{C}_{\text{Best}} = (c_1, c_2, \dots, c_K)$  as the best individual in the population, the procedure consists of performing the following three steps for all colors  $c_i$ :

- **Step 1:** The color  $c_i$  receives a random addition, making it lighter;
- **Step 2:** The perturbation is maintained only if it is beneficial, increasing the fitness value from  $\mathcal{C}_{\text{Best}}$ ,
- **Step 3:** If  $c_i$  was not modified in the previous step, then a random decrease in  $c_i$  is made, making it darker, which should only be maintained if it is beneficial.

### 3.7 Adaptive Rules

Adaptation strategies to control the occurrence of crossover and mutation are well-known methodologies in the specialized literature used to increase the genetic variability of the population and, consequently, prevent the occurrence of premature convergence. These techniques consist of changing the probability of mutation and crossover to control the ability of the meta-heuristic to properly perform exploitation and exploration, respectively. To this end, we propose to adjust the probabilities of mutation and crossover according to the improvement in the fitness value that individuals in a population have in relation to the fitness of the previous population. To model this improvement, we propose to use a measure based on how much the population of one generation of the method has improved compared to the population of the previous generation. Specifically, we propose that the improvement be measured by a weighted average between the differences of the following measures of two consecutive generations of the method: the average of the population's fitness values ( $\mu$ ), the best fitness ( $A$ ), and the worst fitness ( $\lambda$ ). Besides, this average should be multiplied by the standard deviation value of the current generation's fitness values, since this measure is a direct representation of the variability of the current population. Thus, the mathematical representation of the measure that represents the fitness improvement between two consecutive generations of the method is the value  $\text{Improve}_{\text{it}}$ , defined in Eq. (4).

$$\text{Improve}_{\text{it}} := \sigma_{\text{it}} \cdot \frac{\omega_1 |\mu_{\text{it}} - \mu_{\text{it}-1}| + \omega_2 |A_{\text{it}} - A_{\text{it}-1}| + \omega_3 |\lambda_{\text{it}} - \lambda_{\text{it}-1}|}{\omega_1 + \omega_2 + \omega_3}, \quad (4)$$

in which, “it” is the current generation of the method.

In this work, we consider that the most important measure to be considered is the difference between the best fitness values from one generation to the next. For this reason, we have adopted:  $(\omega_1, \omega_2, \omega_3) := (1, 3, 1)$ .

If the improvement is not contained between  $\text{Improve}_{\min}$  and  $\text{Improve}_{\max}$ , then adaptive rules that control the probability of mutation and the probability of crossover must be applied with the intention of controlling the exploitation and exploration of the method, respectively. In detail, the adopted adaptive rules are applied in two cases:

- $\text{Improve}_{\text{it}} < \text{Improve}_{\min}$ : In this case, 10% is added to the mutation probability value and 10% is reduced to the crossover probability value.
- $\text{Improve}_{\text{it}} > \text{Improve}_{\max}$ : In this case, 10% is reduced to the mutation probability value and 10% is added to the crossover probability value.

### 3.8 Proposed Algorithm

The proposed method consists of using all the operators described in this text so that its structure consists of the standard scheme of a GA with additional operators. That is, the method generates an initial population of colors, selects individuals for reproduction, performs the crossover of these individuals, applies the mutation operator to a percentage of the population, performs a massive search in the region of the best individual, generates a new population with the best individuals and assesses the need to change the initial parameters using adaptive rules. Each of these steps is performed a fixed number of times.

## 4 Experiments and Results

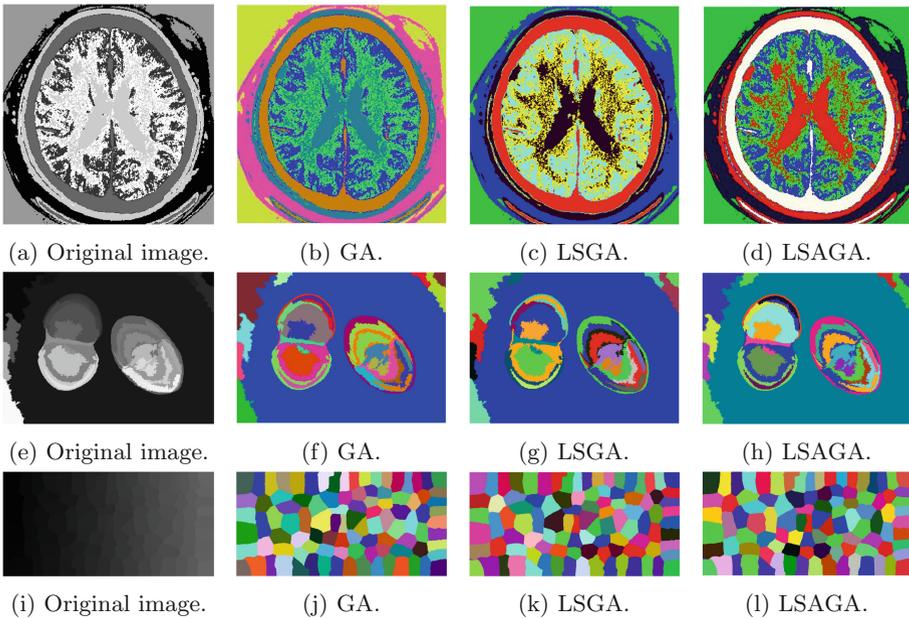
To evaluate the proposed methodology, we defined two distinct test scenarios, as done in [17]. In the first scenario, the proposed method is evaluated on a set of images from the real world. In the second scenario, the proposed method is evaluated at 24 abstract images in which its sub-regions are fully-connected. Thus, the experiments must confirm that the proposed method obtains the best performance applied to real and synthetic problems.

### 4.1 Setup and Implementation

To perform the tests, the authors of [17] provided us their GA code. Thus, in all tests of this work, the technique compared is the technique of [17] and, for this reason, our technique has the most similar configuration possible with the configuration of the GA method. In detail, we use 30 individuals per population, which are taken randomly at  $\text{sRGB}^K$ ; we started the technique with mutation probability  $p_{\text{mut}} = 0.1$  and crossover probability  $p_{\times} = 0.85$ ; 50% of individuals selected for mutation must go through the local search process ( $p_{\text{LS}} = 0.5$ ), with  $N_{\text{Mut}_1} = K$ ; improvement limits are  $\text{Improve}_{\min} = 0.01$  and  $\text{Improve}_{\max} = 122.5$ ; and the method runs for  $10^4$  generations. The computational implementation was done in the MATLAB environment on an i5-4460 PC with 8 GB of ram.

### 4.2 First Scenario

In this first case study, the proposed technique was evaluated on three images, originally arranged in gray levels, which are shown in the left column of Fig. 2. In detail, the images presented that make up the evaluation benchmark are: “brain” (Fig. 2a), with  $K = 6$ ; “two brains” (Fig. 2e), with  $K = 31$ ; and “mosaic” (Fig. 2i), with  $K = 100$ . The technique used for comparison is the basic GA [17] as it is the most recent technique and more similar to the proposed method.<sup>1</sup> Besides, two versions of the material developed in this work are evaluated, one disregarding the adaptive rules, the Local Search Genetic Algorithm (LSGA), and another technique considering these rules, the LSAGA. The best coloring obtained by each of these techniques after 50 executions of each one, is shown in Fig. 2.



**Fig. 2.** Pseudo-Colored Images.

In all the rows of the images in Fig. 2 it is possible to see an evolution in the ease of detecting a greater number of regions if we observe the image in greyscale for the colored version by LSAGA. For example, concerning the images of the two brains (Figs. 2e–2h), we can see that basic GA presents some confusion in the colored regions in shades of pink and red in the midwest region of the image. Something similar occurs with the coloring obtained by LSGA in the central-east

<sup>1</sup> The technique presented in [6] is not used for comparison, since it performs optimization on pre-defined palettes and not on the sRGB space.

region of Fig. 2g, in which we can see that very dark shade of blue is close to a colored region in black. These complications are completely circumvented in the color obtained by the proposed LSAGA method.

Numerical evaluations confirm the superior performance of the proposed methodology. In detail, in Table 1, a set of statistical measures about the fitness value of each technique is presented after 50 executions of each. In all evaluated images, our LSAGA achieve the greatest best result in all cases, and also achieving the greatest worst fitness in two cases. Besides, on all occasions, our LSAGA achieved the best fitness average. This confirms that the adaptation operator tends to guide the LSGA to better solutions. In contrast, in the image “mosaic”, which have a larger number of regions, we can see that the standard deviation (STD) of the fitness values of GA is the smallest. This is precisely because the solutions obtained by GA are restricted to a domain of lower fitness values since the average values presented by this technique are around 25 units less than the average values of the proposed techniques considering the image “mosaic”.

**Table 1.** Statistics about the fitness in the first test scenario. **Bold** numbers are the best values in each situation.

Image	Method	Max	Min	Mean	STD	Average of time (seconds)
Brain	LSAGA	<b>111.5897</b>	<b>103.424</b>	<b>109.5214</b>	<b>2.356342</b>	44.4965625
	LSGA	<b>111.5897</b>	93.81592	107.9427	3.930504	<b>33.9584375</b>
	GA	110.4434	88.66374	100.2385	5.252312	66.953125
Two Brains	LSAGA	<b>105.3691</b>	74.19344	<b>87.03895</b>	<b>6.229127</b>	158.64375
	LSGA	103.2438	<b>74.92773</b>	85.71574	6.917277	144.0959375
	GA	92.47527	62.60709	78.36654	6.915184	<b>97.70625</b>
Mosaic	LSAGA	<b>83.7881</b>	<b>61.2706</b>	<b>73.5591</b>	6.286933	989.3925
	LSGA	81.51334	56.41686	72.11558	5.776788	936.844375
	GA	53.89128	43.38659	48.73923	<b>2.475167</b>	<b>331.4825</b>

With respect to time, in the case of the image “brain”, the low number of regions ( $K = 6$ ) causes the proposed methodology to present low complexity and, therefore, achieve better results in a shorter computational time, since the technique does not build a population for each RGB color component, as basic GA does. However, in more complex images, our methodology still achieves better results, but the computational time taken by these is longer than the time taken by the basic GA. This fact is due to the exaggerated number of generations for our techniques, as can be seen in the convergence analysis presented in Fig. 3, in which the evolution of the best fitness of the techniques in each of the evaluated images is represented. Note that our technique needs less than 20% of the total number of generations to achieve a result that is higher or at least close to the result obtained by GA with the  $10^4$  generations. Therefore, it is clear that the proposed methodology takes less time than GA to obtain a satisfactory solution in all images.

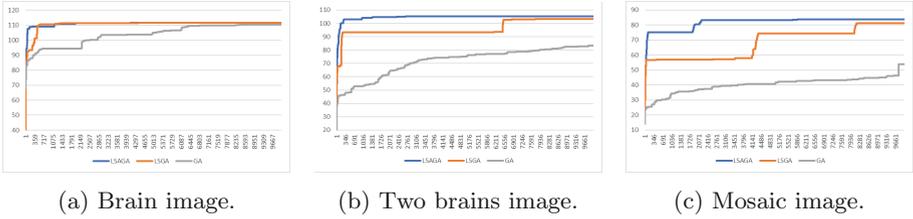


Fig. 3. Convergence analysis.

### 4.3 Second Scenario

The second scenario consists of evaluating the proposed techniques in 24 synthetic and abstract images, which are divided into fully-connected regions. In this evaluation, we compared the proposed techniques (LSGA and LSAGA), with basic GA [17] and a Greedy Algorithm [5]. Table 2 shows the maximum dissimilarity between  $K$  colors (fitness value) presented by each technique after 50 executions.

Table 2. Max distance of colors in a fully-connected images. Green numbers are the best values in raw and red numbers are the worst values in raw.

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

In this scenario, LSAGA presented the best performance in 22 of the 24 images considered, having tied only in the first two ( $K = 2$  and  $K = 3$ ). In addition, LSGA had the worst performance on only one occasion ( $K = 4$ ), and on other occasions, the Greedy Algorithm had the worst performance on images with the least number of regions ( $2 \leq K \leq 14$ ) and GA had the worst performance in the most complex images ( $15 \leq K \leq 24$ ). This demonstrates that the proposed methodology is robust both in images with a smaller number of regions and in images with a larger number of regions, surpassing the performance of GA and Greedy Algorithm in this case study.

## 5 Conclusion

In this work, we propose an adaptation of a GA with operators dedicated to local search and the adaptation of parameters to solve the well-known PsCP. All the necessary steps for the operation of the proposed method were presented with the amount of detail sufficient to make the technique reproducible.

As in [17], we evaluated the proposed material in two test scenarios: one consisting of images from the real world and the other with synthetic images with fully-connected regions. In both cases, the proposed methods, LSGA and LSAGA, exceeded the best techniques available in the specialized literature. In particular, the LSAGA method presents the best results in all the case studies of this work. This is due to the use of the adaptation rules in addition to the local search procedures, which helps the method to avoid premature convergence and to keep the genetic variability of the population high.

In future work, we intend to evaluate the addition of more elaborated diversity control operators at LSGA, such as operators with predator-prey models [12] or operators based on fuzzy rules [1]. Also, we intend to expand this work using the methodology proposed in real problems present in the literature that make direct use of artificial coloring such as weapon detection [20].

**Acknowledgements.** This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001.

## References

1. Amali, S.M.J., Baskar, S.: Fuzzy logic-based diversity-controlled self-adaptive differential evolution. *Eng. Optim.* **45**(8), 899–915 (2013)
2. 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: 2016 International Conference on Emerging Trends in Engineering, Technology and Science (ICETETS), pp. 1–4. IEEE (2016)
3. Asadzadeh, L.: A local search genetic algorithm for the job shop scheduling problem with intelligent agents. *Comput. Ind. Eng.* **85**, 376–383 (2015)
4. Bäck, T., Schwefel, H.P.: An overview of evolutionary algorithms for parameter optimization. *Evol. Comput.* **1**(1), 1–23 (1993)

5. Bianco, S., Citrolo, A.G.: High contrast color sets under multiple illuminants. In: Tominaga, S., Schettini, R., Trémeau, A. (eds.) CCIW 2013. LNCS, vol. 7786, pp. 133–142. Springer, Heidelberg (2013). [https://doi.org/10.1007/978-3-642-36700-7\\_11](https://doi.org/10.1007/978-3-642-36700-7_11)
6. Bianco, S., Schettini, R.: Unsupervised color coding for visualizing image classification results. *Inf. Vis.* **17**(2), 161–177 (2018). <https://doi.org/10.1177/1473871617700682>
7. Carter, R.C., Carter, E.C.: High-contrast sets of colors. *Appl. Opt.* **21**(16), 2936–2939 (1982)
8. Connolly, C., Fleiss, T.: A study of efficiency and accuracy in the transformation from RGB to CIELAB color space. *IEEE Trans. Image Process.* **6**(7), 1046–1048 (1997)
9. Dioşan, L., Oltean, M.: Evolving crossover operators for function optimization. In: Collet, P., Tomassini, M., Ebner, M., Gustafson, S., Ekárt, A. (eds.) EuroGP 2006. LNCS, vol. 3905, pp. 97–108. Springer, Heidelberg (2006). [https://doi.org/10.1007/11729976\\_9](https://doi.org/10.1007/11729976_9)
10. Dmitruk, K., Denkowski, M., Mazur, M., Mikołajczak, P.: Sharpening filter for false color imaging of dual-energy x-ray scans. *SIViP* **11**(4), 613–620 (2017). <https://doi.org/10.1007/s11760-016-1001-7>
11. Etehadtavakol, M., Ng, E.Y.K.: Color segmentation of breast thermograms: a comparative study. In: Ng, E.Y.K., Etehadtavakol, M. (eds.) Application of Infrared to Biomedical Sciences. SB, pp. 69–77. Springer, Singapore (2017). [https://doi.org/10.1007/978-981-10-3147-2\\_6](https://doi.org/10.1007/978-981-10-3147-2_6)
12. Li, X.: A real-coded predator-prey genetic algorithm for multiobjective optimization. In: Fonseca, C.M., Fleming, P.J., Zitzler, E., Thiele, L., Deb, K. (eds.) EMO 2003. LNCS, vol. 2632, pp. 207–221. Springer, Heidelberg (2003). [https://doi.org/10.1007/3-540-36970-8\\_15](https://doi.org/10.1007/3-540-36970-8_15)
13. Lin, C.: An adaptive genetic algorithm based on population diversity strategy. In: 2009 Third International Conference on Genetic and Evolutionary Computing, pp. 93–96. IEEE (2009)
14. Mahyar, F., Cheung, V., Westland, S., Henry, P.: Investigation of complementary colour harmony in CIELAB colour space. In: Proceedings of the AIC Midterm Meeting, China (2007)
15. Moodley, K., Murrell, H.: A colour-map plugin for the open source, Java based, image processing package, imageJ. *Comput. Geosci.* **30**(6), 609–618 (2004)
16. Ombuki, B.M., Ventresca, M.: Local search genetic algorithms for the job shop scheduling problem. *Appl. Intell.* **21**(1), 99–109 (2004). <https://doi.org/10.1023/B:APIN.0000027769.48098.91>
17. Radlak, K., Smolka, B.: Visualization enhancement of segmented images using genetic algorithm. In: 2014 International Conference on Multimedia Computing and Systems (ICMCS), pp. 391–396. IEEE (2014)
18. Smary, N.A.: A proposed HSV-based pseudo-coloring scheme for enhancing medical images. In: Computer Science & Information Technology (2018)
19. Shojaedini, E., Majd, M., Safabakhsh, R.: Novel adaptive genetic algorithm sample consensus. *Appl. Soft Comput.* **77**, 635–642 (2019)
20. Xue, Z., Blum, R.S.: Concealed weapon detection using color image fusion. In: Proceedings of the 6th International Conference on Information Fusion, vol. 1, pp. 622–627. IEEE (2003)
21. Zames, G., et al.: Genetic algorithms in search, optimization and machine learning. *Inf. Technol. J.* **3**(1), 301–302 (1981)

22. Zang, W., Ren, L., Zhang, W., Liu, X.: A cloud model based DNA genetic algorithm for numerical optimization problems. *Future Gener. Comput. Syst.* **81**, 465–477 (2018)
23. 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*, pp. 176–183. IEEE (2003)