A Differential Evolution Algorithm for Contrast Optimization

Artur Leandro da Costa Oliveira[1] and André Britto[1,2(✉)]

[1] Post-Graduation Program in Computer Science,
Federal University of Sergipe, Sergipe, Brazil
arturleandrodacosta@gmail.com

[2] Departament of Computing, Federal University of Sergipe, Sergipe, Brazil
andre@dcomp.ufs.br

Abstract. Image Enhancement is one of the most important phases of the image processing system. Contrast Enhancement plays a key role in this step. Histogram Equalization (HE) is one of the main tools used to improve the contrast of an image. However, the use of HE causes an increase in the natural brightness of the image, which is not desirable in many types of applications such as consumer electronics products. To solve these limitations, it is proposed in this paper a variation of the Differential Evolution metaheuristic algorithm for Contrast Optimization called DECO. The results obtained were statistically compared with other techniques and metaheuristic algorithms. The results showed that DECO is competitive compared with other techniques.

Keywords: Differential Evolution · Contrast Optimization

1 Introduction

Despite the evolution of photo camera sensors and processing algorithms in capturing the image, it is still a challenge to obtain images that present a good contrast quality. Image pre-processing is one of the key steps in the image processing area. Its main function is to transform the image obtained to present an improvement for the subsequent processing step, such as detection and identification [7].

Image enhancement is a technique widely used in this step. The main objective of image enhancement is to make changes to the image's attributes so that its use in a given task is more appropriate for a specific observer [11]. For example, image enhancement is essential in medical diagnostics, or iris and fingerprint enhancement systems for biometric recognition systems. It is usually done by increasing the contrast or by suppressing the noise [11].

A wide number of algorithms are used to enhance the image quality. Histogram Equalization (HE) is considered one of the most common techniques used to enhance the contrast of grayscale images [7]. Its goal is to use an evenly distributed histogram with a cumulative density function [3]. The HE can cause

R. Cerri and R. C. Prati (Eds.): BRACIS 2020, LNAI 12319, pp. 179–194, 2020.
https://doi.org/10.1007/978-3-030-61377-8_13

problems related to the image brightness level, mainly in consumer electronics applications. In this case, preserving the brightness is essential for the success of the application [3]. Different methodologies based in HE are found to fix the problem of preservation of brightness. However, these methods fail to produce, with low cost, natural quality images. CLAHE [16] (Contrast Limited Adaptive Histogram Equalization) is a well known adaptive method of HE in the literature. Its improvement allows it to add restrictions to the HE and preserve certain characteristics of the image.

In recent years, with the emergence of several quality metrics for improving images, it has been possible to use metaheuristic algorithms to perform this step. Contrast optimization aims to improve image contrast by defining it as a restricted nonlinear optimization problem. Several works have been developed for this purpose. Evolutionary Genetic Algorithms [9,13,18], Particle Swarm Optimization (PSO) [1] and Ant Colony Optimization (ACO) [10] have good results for the problem. In [9] a new representation of the solution widely used in Contrast Optimization is presented. The genetic algorithm proposed obtains excellent results in improving the contrast of images. However, the defined objective function can be further improved to maximize its results. The algorithm performance is questionable concerning the execution time, because few iterations were used in its tests. They also do not implement statistical testing procedures, which make it difficult to compare the divergent results of experiments between researchers.

This paper presents a metaheuristic algorithm implementation called DECO (Differential Evolution for Contrast Optimization). Its implementation has singularities for the contrast improvement problem, in addition, an objective function with additional parameters is used, based on in [14] that improves the results of the final images. The main contribution of contrast optimization algorithm DECO is related to your high convergence speed for the better solution, besides its simple structure, versatility, and robustness. Our implementation is based on Differential Evolution (DE) developed by Storn and Price [20] which is one of the most superior evolutionary algorithms.

In addition, DECO is evaluated through a comparative analysis between standard methods of contrast improvement and Contrast Optimization techniques. In those set of experiments, a statistical comparison based on [5] is performed between different algorithms. The performance of the algorithms is analyzed using three different criteria. The adapted fitness function proposed by [9], the *Peak Signal-to-Noise Ratio* (PSNR) image quality criterion, and the time that the algorithm took to find the best solution. The results found demonstrate that DECO achieves positive results when solving the problem of improving image quality.

The remainder of the paper is organized in: Sect. 2 details the problem to be addressed. Section 3 presents the algorithm and its specifications, while the experiments and results are provided in Sect. 4. The conclusion is reported in Sect. 5.

2 Background

2.1 Problem Formulation

Exists two categories of approaches for solving the problem of image enhancement: filtering techniques and contrast enhancement methods. Filtering consists of replacing the gray level of a given pixel with another calculated through the neighborhood. The contrast enhancement is the operation that maps the gray levels of the image creating a new set of gray levels distributing more homogeneously. The contrast enhancement can be divided into two types: global or local. Global techniques are based on mapping all gray levels of the image and making changes to that mapping to change the contrast of the whole image. Unlike global techniques, local ones use different functions in different areas of the image to make local changes [1].

Global techniques are suitable for general image enhancement. However, they often fail to adapt to local characteristics of the image, such as brightness. Because gray levels with high frequency often tend to dominate those with low frequency. Local techniques are able to contour with the problems presented by global techniques. However, they have a high cost of performance and tend to bring a high level of improvement in only certain portions of the image and generally increase the noise in the image.

The contrast enhancement problem can be solved using optimization techniques with approaches to global image processing techniques. The representation of the solutions is based on the same used in [9] that manipulates it to represent the chromosome in the Genetic Algorithm. The desired solution can be seen in the upper part of Fig. 1. An ordered vector of size D of integers values that vary in the range of [0,255]. The size D represents the number of gray levels of the input image. Each level of gray found in the image is indicated in an index of the vector. The first index of the solution means the lowest level of gray found in the image. The second index is the second smallest, and so on.

To remap the image the value of the first gray level of the generated solution is used instead of the first gray level value of the original image. In this way, different histograms are generated trough modifications in the solution. Given the example in Fig. 1 the transformation that will be applied to the input image has the following description: $f(40) = 0$, $f(88) = 42$, $f(97) = 79$, $f(121) = 113$, $f(143) = 188$ and $f(201) = 255$.

To convert the solution into an image, it is necessary to use a specific data structure for the problem. Such structure maintains the coordinates of each gray level of the original image. Thus it is possible to change the levels of the original image according to each gray level of the transformed solution. So it is possible to observe the image generated and perform a qualitative analysis of it.

The objective function, also known as fitness, has the role of evaluating and measuring the quality of the generated solution. For the contrast enhancement problem, a grayscale image with good contrast include many intensive edges [18]. The function fitness adopted in [9] is shown in Eq. 1:

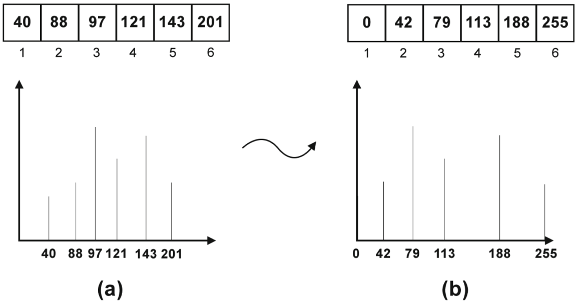

Fig. 1. (a) Histogram of the input image and (b) Histogram of the output image.

$$fitness(s) = log(log(E(I(s)))) * ne(I(s)) \tag{1}$$

Where *fitness(s)* represents the objective function of a solution *s* and *I(s)* represents the image generated after remapping the solution. *E(I(s))* is the sum of the intensities of the generated image edges. This sum is calculated using the following expression [4]:

$$E(I(s)) = \sum_{x} \sum_{y} \sqrt{h(x,y)^2 + v(x,y)^2} \tag{2}$$

On Eq. 2, *h(x,y)* and *v(x,y)* represent, respectively, the horizontal and vertical gradient for the point (x, y) of the image. These values are obtained from the convolution of the Sobel edge detector [7]. The *log(log(E(I(s))))* is used in the sum of the intensities to prevent the production of unnatural images [9]. The number of edges detected in the image is represented by *ne(I(s))* using the convolution result and an automatic threshold value [17].

To evaluate the quality of the solution in our algorithm, a new value will be used that improves the Eq. 1 by adding the entropy of the image. The formula demonstrated in Eq. 3, used by [1,6,8,14,19], has the entropy of the image *H(I(s))* as more one component of the calculation. The higher the entropy value of the image, the more quality it has.

$$fitness(s) = log(log(E(I(s)))) * \frac{ne(I(s))}{PH * PV} * H(I(s)) \tag{3}$$

In addition to entropy, two new values appear in the equation: PH and PV. These values represent respectively the number of horizontal and vertical pixels in the image.

2.2 Related Work

Different approaches for improving contrast using metaheuristics are found in the literature. The first methods that provided the basis for contrast optimization used evolutionary strategies and genetic algorithms as an example in [12,15]. In those works, the standard selection approaches in the genetic algorithm were not performed. Instead, a technique based on user experience was implemented to select the best individuals. The tests in [15] obtained good results on magnetic resonance images and in [12] on satellite images. The work on [18] is one of the pioneers to completely automate the optimization process by metaheuristics. The proposed solution removes the human need by declaring an automatic objective function. The function is performed based on the sum of the edges intensities. Their implementation based on a genetic algorithm uses a Prewitt operator [7] to calculate the objective function. The method proved to be effective in the experiments as it presented results that improved and kept natural the contrast of the tested images.

Modern approaches have come to address the use of different optimization algorithms to solve the problem of image contrast. Different ways of measuring the objective function are adopted, such as entropy, the number of pixels in the image, and the intensity of these pixels. In [19] the Differential Evolution (DE) algorithm with different chaotic sequence approaches is proposed to solve the problem of contrast optimization. Their results prove to be satisfactory when comparing to the different DE approaches proposed. A hybrid algorithm that uses the join of Ant Colony (ACO), Simulated Annealing (SA), and a genetic algorithm is proposed in [10]. The results are compared with the algorithms of [1,13] and it shows to be superior.

The work on [9] proposes a spatial approach to represent the solution of the problem. This methodology is used as a reference by several articles in the literature. Both for its behavior in mapping the solution, and in the comparison of its results. His approach, previously discussed, is the same used in this paper described in the previous section. Their genetic algorithm has specific selection, crossover and mutation operations that lead to excellent results in image contrast and PSNR in the most divergent types of images. The results compared to other optimization algorithms and contrast improvement techniques of the epoch demonstrated superiority. In [6] an algorithm based on Artificial Bee Colony (ABC) is proposed. This proposal uses the concept of [9] to represent the solution, however, it uses a more robust fitness function. Their results surpass the genetic algorithm of [9] in 4 out of 5 cases.

The proposal in [2] diverges from the previous works because it does not represent the solution for optimization. It uses the Artificial Bee Colony to find the best parameters for the Incomplete Beta Function (IBF), which has proven to be effective in enhancing image contrast. Most related works present qualitative and quantitative tests to validate their results. However, as verified in the tests presented in [2,6,9], no statistical methodologies are used to validate the results found. Some works [6,9] has a small number of iterations and is not reported the number of times that the algorithms were executed in order to obtain a diversity of cases for analysis.

3 Differential Evolution for Contrast Optimization - DECO

The algorithm proposed in this paper called DECO (Differential Evolution for Contrast Optimization) is based on the Differential Evolution (DE) algorithm proposed by [21]. The main motivation for choosing this algorithm as a base is that it is one of the evolutionary algorithms that present high consistency and a high degree of performance when compared to other evolutionary algorithms. Its global search capability, effective handling of restrictions, reliable performance, and low need for information mean that the algorithm has a high potential to solve several problems in image processing [19].

3.1 Differential Evolution (DE)

The standard approach of the DE algorithm needs four different parameters for its initialization. A maximum value t_{max} representing the number of iterations performed or the number of generations. The size of the population to be used: *popsize*. The mutation rate factor F used to create the differential vector and CR the probability of performing the crossover.

Mutation Operation (or Differential Operation). DE has variations in its implementation forms. The general convention for the nomenclature of these variations is given by DE/α/β/γ. The α represents how the differential vector will be disturbed. The β is the number of differential vectors used in the perturbation and the γ represents the type of crossover. In this paper, was implemented the standard form of DE, also known as DE/*rand*/1/*bin* [21] represented by Eq. 4:

$$v_{i,G+1} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G}) \tag{4}$$

The *rand* means that the vector will be randomly selected and the *bin* means that the crossover is performed in “Binomial” scheme. A variation of the algorithm was also implemented, which was called DECO-BEST. The proposed algorithm has the form DE/*best*/2/*bin* as base. The *best* means that the best vector will be selected. This form is a high benefit method that deserves special attention [20]. It mutate individuals according to the following Eq. 5:

$$v_{i,G+1} = x_{best,G} + F \cdot (x_{r1,G} + x_{r2,G} - x_{r3,G} - x_{r4,G}) \tag{5}$$

In the above equations, the value x_i represents an individual from the population of size *NP*. The elements r_1, r_2, r_3,and r_4 are random individuals selected from the population. G represents the generation which that individual belongs. The mutation factor is described by $F \in$ [0,2] and $v_{i,G+1}$ represents the differential vector generated to perform the crossover operation with the selected individual. In Eq. 5 the value described by $x_{best,G}$ represents the best fitness solution of that generation.

Crossover Operation. The crossover is applied after the mutation operation to mix individuals with the differential vectors resulting from the operation, increasing the diversity of individuals. The resulting individual is generated by:

$$u_{ji,G+1} = \begin{cases} v_{ji,G+1} & \text{if (randb(j) } \leq \text{ CR) or j = rnbr(i),} \\ x_{ji,G} & \text{if (randb(j) } > \text{ CR) and j } \neq \text{ rnbr(i).} \end{cases} \tag{6}$$

On Eq. 6, j represents each position of the individual's vector. The value of *randb(j)* is the jth random value generated in the range of [0,1]. *rnbr(i)* is also a random value of $i \in [1,D]$ it guarantees that at least one position of the vector will be changed.

Selection Operation. The selection step has as main objective to select which individual will be chosen for the next generation, *G + 1*. Among the choices are the individual from that iteration $x_{i,G}$ and the individual generated by the mutation and crossover $u_{i,G+1}$. The choice is based on the fitness value of the individuals. The one with the highest fitness value will be chosen for the next generation.

3.2 DECO Implementation

The pseudocode in Algorithm 1 describes the main steps that have been implemented for the proposed contrast enhancement algorithm. First, the algorithm receives an image and the same parameters of the DE. Through the input image it is possible to extract basic structures that will serve as support for the rest of the algorithm. After that the steps follow the same structure as the standard implementation of the DE algorithm [21].

An initial population of size *NP* is created with random values. From t_{max} iterations, changes are made to the populations to improve the objective function. The change is based on, for each p of the population a new individual q is generated by adding and subtracting vector from p with 3 other individuals in the population: a, b, and c. The new son q will be compared with the father p and it will be added to the population if it has a greater fitness value, otherwise it will be discarded. Because we are dealing with a maximization problem, higher values of the objective function indicate optimization in the final result. The *best* individual found is maintained and updated in each generation. Thus, at the end of the optimization, the data for the experiments will be extracted from the *best* solution and the image generated through it.

ExtractMapper **Function.** This function, referenced on line 2, is responsible for generating the M structure that will be used to maintain the data from the original input image and perform the solution conversion. A scan of the original image is performed to extract the standard solution and the size of this solution D which represents the number of gray levels of the individuals that will be generated. It is important to note that this value must be ordered so that there is

Algorithm 1: DECO

```
   input  : An image $I$ and the DE parameters: $t_{max}$, $popsize$, $F$, $CR$
   output: The optimized image $O$
1  begin
2      $M \longleftarrow ExtractMapper(I)$;
3      $P \longleftarrow InitializePopulation(M, popsize)$;
4      repeat
5          $Q \longleftarrow \emptyset$;
6          for $p \in P$ do
7              $a$ ⟵a copy of an individual other than $p$, randomly chosen;
8              $b$ ⟵a copy of an individual other than $p$ and $a$, randomly chosen;
9              $c$ ⟵a copy of an individual other than $p$, $a$ and $b$, randomly chosen;
10             $d \longleftarrow GenerateDifferential(a, b, c, F)$;
11             $q \longleftarrow CrossOver(d, p, CR)$;
12             if $Fitness(q, M) < Fitness(p, M)$ then
13                 $q \longleftarrow Copy(p)$
14             if $Fitness(best, M) < Fitness(q, M)$ then
15                 $best \longleftarrow Copy(q)$
16             $Q.add(q)$;
17         $P \longleftarrow Copy(Q)$;
18     until reach the maximum number of iterations: $t_{max}$;
19     $O \longleftarrow GenerateImage(M, best)$;
```

a reference between the original solution and the new individuals. Also, a coordinate map of the original solution is kept in memory. Such map is responsible for transform individuals into images. Each gray level of the original image is changed in its different coordinates, for each gray level position changed in the new individual.

***InitializePopulation* Function.** With the solution size D found in the previous step, it is possible to initialize, line 3, a population of size NP of individuals through a random process. An individual in the population is represented by a vector of size D with values in the range of [0,255]. To maximize the range of gray levels, a value of 0 is assigned to the first element of the vector and a value of 255 to the last one [9]. The vector created is increasingly ordered in order to comply with the requirements of the proposed solution.

***GenerateDifferential* Function.** Also known as the Mutation Operation (or Differential Operation), this function is demonstrated in line 10. Its task is to generate the individual d through the operation defined in the Eq. 4, in the case of DECO-BEST the Eq. 5. It is important to note that the values generated in this operation can escape the domain of the integers in the range of [0,255]. Therefore, it is important to perform rounding and apply the range limits. If a value generated is not within the bounds, a new random value that does not belong to the individual is generated.

***Crossover* Function.** In this function, referenced on line 11, the individual q is generated by mixing the individual from the differential operation d and the individual p from the current iteration. The function implementation follows the steps of the Eq. 6 where there will be at least one change in values between individuals. At the end of the process, it is necessary to reorder the vector of the individual q because the exchange of values in the crossover can generate anomalies in the resulting image for that individual.

***Fitness* Function.** It is necessary to transform the solution into an image to calculate the objective function based on Eq. 3. The structure in M must be used to perform the conversion. With the resulting image it is possible to apply the Sobel edge detector to extract the gradient according to Eq. 2 and extract the remaining values that result in fitness. The fitness calculation of each individual is stored in memory internally in the individual's object to avoid excessive-performance expenses. Because the steps taken to reach fitness proved to be costly in large size images.

***GenerateImage* Function.** After performing the optimization step, the best fitness individual named *best* is chosen. It will be possible to obtain a final image of this individual, on line 19, by remapping *best* with the data kept in M for the entry image I and the initial solution that is kept in memory.

4 Experimental Results

In this section the two DECO and DECO-BEST implementations are applied to a validation process. Different types of grayscale images were used that cover different levels of contrast indicating different scenarios of use of the algorithms. Several algorithms and metrics are used, through a statistical comparison, to obtain validity in the tests performed.

4.1 Algorithms and Parameters

The experiments were performed with algorithms widely used in the literature to improve the contrast, such as Histogram Equalization [7] and CLAHE [16]. Also were included methods of contrast optimization based on metaheuristics, such as the genetic algorithm proposed in [9] and the ABC demonstrated in [6], which use the same representation of the solution and obtained good results in their experiments. First, it is necessary to carry out initial isolated tests on DECO and DECO-BEST to find initial parameters that would bring relevant results to the work.

For DECO, DECO-BEST algorithms, and the genetic proposed by [9] the t_{max} value for the number of iterations it was set to $1,000$, and the value NP for the population size is set to 20. In DECO and DECO-BEST algorithms, the following parameter values were used: $F = 1$, and $CR = 0.9$. Such values obtained good results in initial isolated tests with different types of images. In the genetic algorithm, the crossover rate and mutation rate values were respectively 0.8 and 0.1. As indicated in the tests carried out in [9]. In the ABC algorithm [6] were used the same parameters proposed in the paper: number of solutions 25, both for the number of employed bees and onlookers. And the threshold of activation of the scouts limit is set to 5. Only the number of iterations t_{max} has been changed to the value of 400 to balance the number of executions of the objective function with the other algorithms.

4.2 Images and Metrics

A set of 5 images in grayscale with different sizes, taken from the literature and related works, were used for the experiments. The images are named by: *Airplane* (size: 512×512), *Cameraman* (size: 256×256), *Crowd* (size: 512×512), *Messier83* (size: 640×640), and *Chest X-Ray* (size: 482×551). *Airplane* and *Cameraman* presents a good variation of pixel intensities and frequencies. The *Messier83* and *Chest X-Ray* images have many pixels with low intensity. *Messier83* also has a high region with low frequency of pixels intensity. The *Crowd* presents the highest variation of pixels intensity between the images, which results in a more equalized image. These images will make it possible to evaluate differences in the execution of the algorithms in different image sizes and characteristics.

The performance of the algorithms is analyzed using three different criteria. The fitness value of the best individual in each algorithm is used to analyze which optimization achieved the best performance. With this comparison it is possible to evaluate which algorithm can reach a higher number of intensive edges. The *Peak Signal-to-Noise Ratio* (PSNR) image quality criterion, which measures the noise ratio between the original image and the enhanced images. And the execution time that the algorithm took to find the best solution.

Metrics evaluation will be performed using non-parametric tests with multiple comparisons [5]. Each algorithm was executed 20 times and its values of μ (mean) and σ (standard deviation) for each metric are counted for each algorithm. It is necessary to apply a multiple comparison between the algorithms to know if they are equivalent or not. In this paper the Kruskal–Wallis test was used to verify if one algorithm is significantly different from the other. It is necessary to reach a *p-value* < 0.05 for a $\alpha = 0.05$. This value indicates the lowest level of significance that results in the rejection of the null hypothesis [5].

4.3 Results

In the first step, the algorithms were statistically evaluated for their performance over the fitness value of the best solution. Only metaheuristic algorithms participate in this phase because they are directly influenced by the value of fitness. The values of μ (mean) and σ (standard deviation) for each algorithm are shown in Table 1. It is shown in Table 2 the values of *p-value* resulting from the Kruskal–Wallis test to compare the results of fitness in the algorithms. The values in bold in this table represent that the algorithms are comparable to each other. It is possible to notice that the DECO and DECO-BEST algorithms had comparable results in all images. The Kruskal-Wallis test of these two algorithms results in a *p-value* > 0.05.

Table 1. *Fitness* mean and standard deviation.

Test image	[9]	[6]	DECO	DECO-BEST
Airplane	$\mu = 19.79$	$\mu = 20.46$	$\mu = 19.98$	$\mu = 19.99$
	$\sigma = 0.16$	$\sigma = 0.10$	$\sigma = 0.06$	$\sigma = 0.05$
Cameraman	$\mu = 17.40$	$\mu = 17.79$	$\mu = 17.45$	$\mu = 17.44$
	$\sigma = 0.20$	$\sigma = 0.09$	$\sigma = 0.03$	$\sigma = 0.04$
Crowd	$\mu = 21.22$	$\mu = 21.75$	$\mu = 21.37$	$\mu = 21.40$
	$\sigma = 0.10$	$\sigma = 0.09$	$\sigma = 0.02$	$\sigma = 0.03$
Messier83	$\mu = 5.51$	$\mu = 5.59$	$\mu = 5.56$	$\mu = 5.56$
	$\sigma = 0.03$	$\sigma = 0.09$	$\sigma = 0.01$	$\sigma = 0.01$
Chest X-Ray	$\mu = 16.89$	$\mu = 16.19$	$\mu = 16.29$	$\mu = 16.28$
	$\sigma = 0.28$	$\sigma = 0.15$	$\sigma = 0.03$	$\sigma = 0.03$

Table 2. Kruskal-Wallis test on *fitness*.

Comparison	***Airplane***	***Cameraman***	***Crowd***	***Messier83***	***Chest X-Ray***
[9] versus [6]	5e−14	5.3e−07	1.0e−13	0.0014	4.6e−09
[9] versus DECO	0.00049	**0.99**	**0.02987**	0.0019	0.00044
[9] versus DECO-BEST	0.00014	**1.00**	0.00034	0.0027	3.0e−05
[6] versus DECO	0.00014	3.0e−06	5e−06	**0.9998**	**0.12570**
[6] versus DECO-BEST	0.00049	3.3e−07	0.00133	**0.9980**	**0.38305**
DECO versus DECO-BEST	**0.99068**	**0.98**	**0.58495**	**0.9996**	**0.93247**

$\alpha = 0.05$

In the images *Airplane* and *Crowd* it is possible to observe a better result of algorithm [6] despite its high standard deviation value. At *Cameraman* the results are very similar, but for Kruskal-Wallis, algorithm [6] differs from the others. In *Messier83* only the algorithm [9] obtained a lower result than the rest of the algorithms, differently from *Chest X-Ray* where it obtained the best result. It is notable in all images that the standard deviation of DECO and DECO-BEST are lower than their competitors, demonstrating a standard of reliability in the execution of the algorithm. And despite some draws, the algorithm [6] was more successful in maximizing the objective function.

In the second step, the algorithms are compared using the PSNR. The purpose of this test is to analyze the noise ratio of the generated image compared to the original image. The HE and CLAHE techniques participate in these experiments, but the mean value and standard deviation for these algorithms are not shown because several executions of these techniques arrive at the same result. The test data can be found in Table 3. The results of *PSNR* shows that the HE and CLAHE algorithms add less noise to the images, only in *Messier83* that their results are worse than the metaheuristic algorithms, mainly the HE result. Among the metaheuristic algorithms it is possible to see in Table 4 that there were several equivalences between them. The algorithm in [9] added less noise to the image only in the *Airplane* and in *Cameraman*, where there was equivalence with all the other algorithms. [6] was superior in isolation in the image *Crowd* and the *Cameraman*, in *Messier83* it was equivalent to the others. DECO and DECO-BEST were superior in the image *Chest X-Ray* and equivalent in *Cameraman* and *Messier83*. When matching the data of these two tests, it is interesting to note, for example, that in the image*Chest X-Ray* the algorithms DECO and DECO-BEST despite being comparable with [6] in fitness, maintain a better quality in the final image. In the image *Crowd* the algorithm [6] is superior in isolation in both metrics.

Table 3. *PSNR* mean and standard deviation.

Test image	HE	CLAHE	[9]	[6]	DECO	DECO-BEST
Airplane	21.87	23.64	$\mu = 20.70$	$\mu = 19.43$	$\mu = 19.70$	$\mu = 19.49$
			$\sigma = 0.51$	$\sigma = 0.89$	$\sigma = 0.43$	$\sigma = 0.47$
Cameraman	21.58	22.86	$\mu = 18.33$	$\mu = 18.19$	$\mu = 18.06$	$\mu = 18.46$
			$\sigma = 0.56$	$\sigma = 0.72$	$\sigma = 0.69$	$\sigma = 0.35$
Crowd	15.39	21.38	$\mu = 11.26$	$\mu = 12.60$	$\mu = 11.74$	$\mu = 11.92$
			$\sigma = 0.28$	$\sigma = 0.48$	$\sigma = 0.27$	$\sigma = 0.34$
Messier83	3.68	19.78	$\mu = 20.60$	$\mu = 21.69$	$\mu = 21.67$	$\mu = 21.54$
			$\sigma = 0.45$	$\sigma = 1.32$	$\sigma = 0.36$	$\sigma = 0.38$
Chest X-Ray	14.36	18.54	$\mu = 16.82$	$\mu = 16.98$	$\mu = 17.54$	$\mu = 17.50$
			$\sigma = 0.41$	$\sigma = 0.80$	$\sigma = 0.59$	$\sigma = 0.47$

Table 4. Kruskal-Wallis test on *PSNR*.

Comparison	***Airplane***	***Cameraman***	***Crowd***	***Messier83***	***Chest X-Ray***
[9] versus [6]	5.5e−06	**0.875**	5.3e−11	0.00129	**0.8994**
[9] versus DECO	0.00047	**0.66**	0.01403	6.4e−06	0.0051
[9] versus DECO-BEST	3.8e−06	**0.75**	0.00030	0.00014	0.0046
[6] versus DECO	**0.76870**	**0.98**	0.00078	**0.63272**	0.0437
[6] versus DECO-BEST	**0.99985**	**0.28**	0.02817	**0.94988**	0.0405
DECO versus DECO-BEST	**0.72528**	**0.14**	**0.72528**	**0.91446**	**1.0000**

$\alpha = 0.05$

The third step is characterized by accounting for the total execution time of the algorithms, demonstrated in Table 5. As expected, the HE and CLAHE techniques obtain execution time values in milliseconds, due to the algorithm technique, so only the results of the metaheuristic algorithms will be demonstrated. DECO and DECO-BEST had better results compared to other algorithms, reaching execution times well below expectations, demonstrating the performance power that DE carries.

Table 5. *Execution time* mean and standard deviation.

Test Image	[9]	[6]	DECO	DECO-BEST
Airplane	$\mu = 46.67s$	$\mu = 177.59s$	$\mu = 28.66s$	$\mu = 29.27s$
	$\sigma = 1.16s$	$\sigma = 8.59s$	$\sigma = 2.49s$	$\sigma = 0.03s$
Cameraman	$\mu = 11.34s$	$\mu = 30.21s$	$\mu = 6.92s$	$\mu = 7.07s$
	$\sigma = 0.25s$	$\sigma = 0.82s$	$\sigma = 0.60s$	$\sigma = 0.04s$
Crowd	$\mu = 51.54s$	$\mu = 208.24s$	$\mu = 31.63s$	$\mu = 32.39s$
	$\sigma = 1.65s$	$\sigma = 13.11s$	$\sigma = 2.65s$	$\sigma = 0.84s$
Messier83	$\mu = 69.87s$	$\mu = 213.72s$	$\mu = 42.49s$	$\mu = 44.49s$
	$\sigma = 1.79s$	$\sigma = 11.22s$	$\sigma = 3.73s$	$\sigma = 0.58s$
Chest X-Ray	$\mu = 44.28s$	$\mu = 190.63s$	$\mu = 27.55s$	$\mu = 27.87s$
	$\sigma = 1.04s$	$\sigma = 136.64s$	$\sigma = 2.57s$	$\sigma = 0.03s$

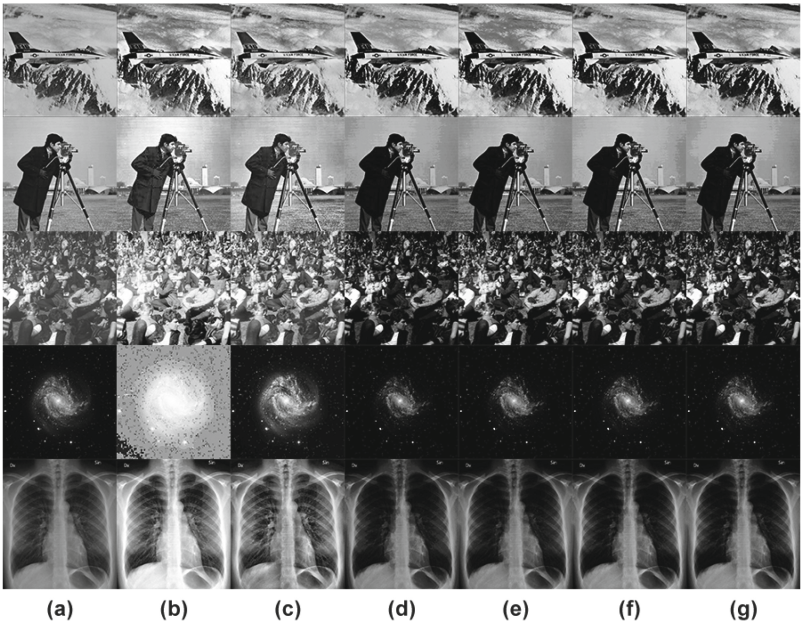

Fig. 2. Visual quality comparison: (a) Original Image, (b) HE, (c) CLAHE, (d) [9], (e) [6], (f) DECO and (g) DECO-BEST. From top to bottom are the respective images: *Airplane*, *Cameraman*, *Crowd*, *Messier83* and *Chest X-Ray*

The last step of the experimental process is characterized by performing a visual analysis of the images generated by the algorithms. Figure 2 shows the comparison of all final solutions taken from the tests of each algorithm for each image. It is noticed that the HE and CLAHE algorithms, despite having obtained excellent results in the PSNR and runtime tests, change the characteristics of the images. Brightness is the most affected feature in these algorithms, the biggest effect of this loss is seen in *Messier83*. That image has a high concentration of pixel frequency in low-level intensities, which caused HE to have difficulties in improving the contrast. Therefore, the use of optimization algorithms is validated to obtain images as close as possible to the original, with enhancement only in contrast levels. It is visually noticeable that there is a lower loss in the natural brightness of the images in the results of the metaheuristic algorithms. The visual differences between the metaheuristic algorithms themselves are difficult to be noticed, but in certain cases there are greater amplitudes of contrasts in certain specific locations of the images. The noise is also little perceived visually, despite the obtained values of *PSNR* which was lower than HE and CLAHE in some cases.

5 Conclusions

In this work, an image contrast improvement based on metaheuristics was implemented using two well-known strategies of the Differential Evolution algorithm which resulted in the DECO and DECO-BEST algorithms. The representation of the optimization problem was based on [9]. The algorithms were compared qualitatively and statistically with other contrast improvement algorithms and techniques: Histogram Equalization (HE), CLAHE, the genetic algorithm proposed in [9], and the ABC proposed in [6].

The results found were promising, the resulting images showed to have a good quality standard, with low noise level, and a visual contrast improvement. Regarding the objective function, the DECO and DECO-BEST algorithms achieved similar results. They demonstrated to be competitive in comparison to the other algorithms and their standard deviation of the obtained fitness values was much smaller than the others which demonstrates a high convergence in the search for the solution. The results obtained in PSNR were equivalent between the metaheuristic algorithms which indicates a small deformation of the image caused by the use of this solution representation. In terms of execution time, DECO and DECO-BEST resulted in much shorter times than the metaheuristic algorithms.

For future works, it is proposed the use of more recent variations of the differential evolution algorithm that obtain excellent results in solving optimization problems. A greater variety of contrast improvement techniques can also be compared in the experiments to improve the results found. Another path to be explored is to change the representation of the problem by mapping it as a multi-objective problem. The motivation is to improve not only edge strength but also PSNR and other combined quality criteria, such as SSIM and IFC. In this way, the final result of the image quality will be improved.

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

References

1. Braik, M., Sheta, A.F., Ayesh, A.: Image enhancement using particle swarm optimization. In: World Congress on Engineering, vol. 1, pp. 978–988 (2007)
2. Chen, J., Yu, W., Tian, J., Chen, L., Zhou, Z.: Image contrast enhancement using an artificial bee colony algorithm. Swarm Evol. Comput. **38**, 287–294 (2018)
3. Chen, S.D., Ramli, A.R.: Contrast enhancement using recursive mean-separate histogram equalization for scalable brightness preservation. IEEE Trans. Consum. Electron. **49**(4), 1301–1309 (2003)
4. DaPonte, J.S., Fox, M.D.: Enhancement of chest radiographs with gradient operators. IEEE Trans. Med. Imaging **7**(2), 109–117 (1988)
5. Derrac, J., García, S., Molina, D., Herrera, F.: A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. Swarm Evol. Comput. **1**(1), 3–18 (2011)
6. Draa, A., Bouaziz, A.: An artificial bee colony algorithm for image contrast enhancement. Swarm Evol. Comput. **16**, 69–84 (2014)
7. Gonzalez, R.C., Woods, R.E., et al.: Digital Image Processing. Prentice Hall, Upper Saddle River (2002)
8. Gorai, A., Ghosh, A.: Gray-level image enhancement by particle swarm optimization. In: 2009 World Congress on Nature & Biologically Inspired Computing (NaBIC), pp. 72–77. IEEE (2009)
9. Hashemi, S., Kiani, S., Noroozi, N., Moghaddam, M.E.: An image contrast enhancement method based on genetic algorithm. Pattern Recogn. Lett. **31**(13), 1816–1824 (2010)
10. Hoseini, P., Shayesteh, M.G.: Hybrid ant colony optimization, genetic algorithm, and simulated annealing for image contrast enhancement. In: IEEE Congress on Evolutionary Computation, pp. 1–6. IEEE (2010)
11. Maini, R., Aggarwal, H.: A comprehensive review of image enhancement techniques. arXiv preprint arXiv:1003.4053 (2010)
12. Munteanu, C., Lazarescu, V.: Evolutionary contrast stretching and detail enhancement of satellite images. In: Proceedings of MENDEL 1999, pp. 94–99 (1999)
13. Munteanu, C., Rosa, A.: Evolutionary image enhancement with user behavior modeling. ACM SIGAPP Appl. Comput. Rev. **9**(1), 8–14 (2001)
14. Munteanu, C., Rosa, A.: Gray-scale image enhancement as an automatic process driven by evolution. IEEE Trans. Syst. Man Cybern. Part B Cybern. **34**(2), 1292–1298 (2004)
15. Poli, R., Cagnoni, S.: Evolution of pseudo-colouring algorithms for image enhancement with interactive genetic programming. Cognitive Science Research Papers-University OF Birmingham CSRP (1997)
16. Reza, A.M.: Realization of the contrast limited adaptive histogram equalization (CLAHE) for real-time image enhancement. J. VLSI Signal Process. Syst. Signal Image Video Technol. **38**(1), 35–44 (2004)
17. Rosin, P.L.: Edges: saliency measures and automatic thresholding. Mach. Vis. Appl. **9**(4), 139–159 (1997)
18. Saitoh, F.: Image contrast enhancement using genetic algorithm. In: IEEE SMC 1999 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028), vol. 4, pp. 899–904. IEEE (1999)
19. dos Santos Coelho, L., Sauer, J.G., Rudek, M.: Differential evolution optimization combined with chaotic sequences for image contrast enhancement. Chaos, Solitons Fractals **42**(1), 522–529 (2009)

20. Storn, R., Price, K.: Minimizing the real functions of the ICEC'96 contest by differential evolution. In: Proceedings of IEEE International Conference on Evolutionary Computation, pp. 842–844. IEEE (1996)
21. Storn, R., Price, K.: Differential evolution-a simple and efficient heuristic for global optimization over continuous spaces. J. Global Optim. **11**(4), 341–359 (1997)