# Comparison of DE strategies for Gray-Level MultiLevel Thresholding

Uroš Mlakar
University of Maribor
Smetanova 17
Maribor, Slovenia
uros.mlakar@um.si

Iztok Fister
University of Maribor
Smetanova 17
Maribor, Slovenia
iztok.fister@um.si

## ABSTRACT
In this paper we investigate the impact of different mutation strategies in differential evolution applied to the problem of gray-level multilevel thresholding. Four different strategies are compared, namely rand/1, best/1, rand to best/1, and current to best/1. These strategies are tested on four standard test images taken from literature. The quality of segmented images was compared on the basis of the PSNR metric, which showed that the best performing strategy was current to best/1 based on the mean PSNR value, but in the case of best result found, the rand/1 strategy performed best. When comparing the mean and best objective values, the best/1 strategy outperformed the others.

## Keywords
Multilevel thresholding, Otsu criterion, evolutionary algorithm, differential evolution

## 1. INTRODUCTION
Image segmentation is a process of dividing an image into disjoint sets, which share similar properties such as intensity or color. Image segmentation is usually the first step for many high-level methods, such as feature extraction, image recognition, and classification of objects. Simply put, image segmentation is a process of dividing an image into regions, which are used as input for further specific applications. Image thresholding is one of the most used and simplest segmentation techniques, which performs image segmentation based on values contained in the image histogram. In the case of separating an image into two classes, the process is called bilevel thresholding, but when separating the image into several regions we deal with multilevel thresholding. The selection of optimal threshold values is crucial, since the results of good segmentation are a good foundation for applications, which further process the segmented images.

Multilevel thresholding can be regarded as an optimization process, usually maximizing certain criteria like between-class variance or various entropy measures. Many heuristic methods have gained a lot of attention recently, since exhaustive methods are usually computationally inefficient.

In this paper we investigate the influence of different mutation strategies on the quality of segmentation by using the between-class variance as the objective function proposed by Otsu [5]. The rest of the paper is organized as follows. In Section 2 some of related work is presented in the area of multilevel thresholding. In Section 3, image segmentation and the Otsu crietrion are described, while in Section 4 the differential evolution algorithm (DE), along with 4 mutation strategies are presented, which have been used for this study. In Section 5 we define the metric, which was used to assess the quality of the segmentation and present the experimental results. In Section 6 we will conclude this paper with some findings.

## 2. RELATED WORK
Many works has been done for multilevel thresholding using evolutionary algorithms. Some of these algorithms can be found in [7] and [4]. Alihodzic et al. [1] introduced a hybrid bat algorithm with elements from the differential evolution and artificial bee colony algorithms. Their result show their algorithm outperforms all other in the study, while significantly improving the convergence speed. Bhandari et al. [2] investigated the suitability of the cuckoo search (CS) and the wind driven optimization algorithm for multilevel thresholding using Kapur's entropy as the objective function. The algorithms were tested on a standard set of satellite images by using various number of thresholds. They concluded that both algorithms can be efficiently used for the multilevel thresholding problem. Duraisamy et al. [3] proposed a novel particle swarm optimization (PSO) algorithm maximizing the Kapur's entropy and the between-class variance. Their algorithm has been tested on 10 images, and the results compared with a genetic algorithm. The PSO proved better in terms of solution quality, convergence, and robustness. Zhang et al. [8] presented an artificial bee colony algorithm (ABC), for the problem of multilevel thresholding. They compared their algorithm to PSO and GA, where their conclusions were that the ABC is more rapid and effective, using the Tsallis entropy.

## 3. IMAGE SEGMENTATION
For the purpose of multilevel threshold selection the Otsu criterion was selected as the objective function. It operates on the histogram of the image, maximizing the between-class

variance. For a bilevel thresholding problem is this formally defined as:

$$\sigma_B^2(t^*) = \max_{1 \le t \le L} \sigma_B^2(t), \tag{1}$$

where

$$\sigma_B^2 = \omega_0 \omega_1 (\mu_1 - \mu_0)^2, \tag{2}$$

with $\omega_0$ and $\omega_1$ being probabilities of class occurrence and $\mu_0$ and $\mu_1$ the mean values of each class.

This problem is easily expandable to a multilevel problem as:

$$\sigma_B^2(t_1^*, t_2^*, ..., t_{n-1}^*) = \max_{1 \le t_1 < t_2 < ... < t_{n-1} < L} \sigma_B^2(t_1, t_2, ..., t_n) \tag{3}$$

# 4. EVOLUTIONARY ALGORITHM

An evolutionary algorithm is a generic population-based optimization algorithm. It uses mechanisms which are inspired by biological evolution, such as reproduction, mutation, recombination, and selection. The solutions for the optimization problems are the individuals in the population, and the fitness function provides a metric for measuring the quality of the solutions. In this paper an EA called differential evolution (DE) has been used for optimal multilevel thresholding. Hereinafter the DE algorithm will be described in detail.

## 4.1 Differential Evolution

Differential evolution (DE) [6] is a population based optimization algorithm used for global optimization. It is simple, yet very effective in solving various real life problems. The idea of DE is a simple mathematical model, which is based on vector differences.

The population of the DE algorithm consists of $Np$ individuals $\mathbf{x}_i$, where $i = 1, 2, ..., Np$, and each vector consists of $D$-dimensional floating-point encoded values $\mathbf{x}_i = \{x_{i,1}, x_{i,2}, x_{i,1}, ..., x_{i,D}\}$. In the evolutionary process, individuals are evolved using crossover, mutation and selection operators, which are controled by a scale factor $F$ and crossover rate $C_r$. The following mutation strategies were considered in this paper:

- "rand/1":

$$\mathbf{v}_i^g = x_{r1}^g + F\,(x_{r2}^g - x_{r3}^g) \tag{4}$$

- "best/1" :

$$\mathbf{v}_i^g = x_{best}^g + F\,(x_{r1}^g - x_{r2}^g) \tag{5}$$

- "current to best/1" :

$$\mathbf{v}_i^g = x_i^g + F\,(x_{best}^g - x_i^g) + F\,(x_{r1}^g - x_{r2}^g) \tag{6}$$

- "rand to best/1" :

$$\mathbf{v}_i^g = x_{r1}^g + F\,(x_{best}^g - x_{r1}^g) + F\,(x_{r2}^g - x_{r3}^g) \tag{7}$$

For the creation of a mutant vector $\mathbf{m}_i$, three random vectors from the population are selected, defined by indexes $r1$, $r2$ and $r3$. indexes are mutually different and different from $i$. They are selected uniformly within the range $\{1, 2, ..., Np\}$. The scale factor $F$ is defined within the range $[0, 2]$. The mutant vector $\mathbf{m}_i$ is obtained by adding a scaled vector difference to a third vector.

The trial vector $\mathbf{t}_i$ is then generated using the crossover rate $C_r$ and a corresponding vector $\mathbf{x}_i$ from the population as:

$$t_{i,j} = \begin{cases} m_{i,j} & \text{if } rand(0,1) \le C_r \text{ or } j = j_{rand}, \\ x_{i,j} & otherwise. \end{cases} \tag{8}$$

As can be seen from Eq. 8, the crossover rate $C_r$ is defined at the interval $[0, 1]$ and it defines the probability of creating the trial vector parameters $t_{i,j}$. The $j_{rand}$ index is responsible for the trial vector to contain at least one value from the mutant vector. After crossover, some values of the vector may fall out of bounds, which means that these values must be mapped back to the defined search space.

The next step in the evolutionary process is the selection of the fittest individuals. During the selection process the trial vector $\mathbf{u}_i$, competes with vector $\mathbf{x}_i$ from the population. The one with the better fitness value survives and is transfered to the next generation.

# 5. RESULTS

In this section, the obtained results are presented, and also the experimental environment is described.

## 5.1 Experimental environment

For the purpose of this study 4 gray scale standard test images were taken form literature. All images are of same size (512 × 512 pixels) and in uncompressed format. All images are presented in Figure 1. For evaluating the quality of the results during the evolution, the Otsu criterion for multilevel thresholding has been used (see Section 3).

All experiments were conducted with the following number of thresholds: 5, 8, 10, 12, and 15. The algorithm was coded in C++.

## 5.2 Image quality assessment

For evaluating the segmentation quality the well established peak-signal-to-noise ratio metric has been used. It gives the similarity of an image against a reference image based on the MSE of each pixel. It is defined as follows:

$$PSNR = 10 \log_{10} \left( \frac{255^2}{MSE} \right) \tag{9}$$

$$MSE = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} [I(i,j)) - J(i,j)]^2 \tag{10}$$

where I and J are the original and segmented images respectively.

(a) Lena (b) PEppers (c) Baboon (d) Woman

Figure 1: Test images Lena, Peppers, Baboon, and Woman.

Table 1: Comparison of best mean objective values, with mean CPU times computed by rand/1, best/1, current to best/1, and random to best/1 mutation strategies using Otsu criterion.

| Test image | M | Mean objective values | | | | | CPU Time(s) | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | rand/1 | best/1 | current to best/1 | random to best/1 | | rand/1 | best/1 | current to best/1 | random to best/1 |
| Baboon | 5 | 0.958493 | **0.958852** | 0.958749 | 0.95865 | | 0.01023 | 0.01213 | 0.01143 | 0.01103 |
| | 8 | 0.979531 | 0.97988 | **0.979964** | 0.979874 | | 0.009967 | 0.01197 | 0.01207 | 0.01323 |
| | 10 | 0.985483 | **0.985823** | 0.985814 | 0.985742 | | 0.01107 | 0.01267 | 0.0124 | 0.013 |
| | 12 | 0.989188 | 0.989286 | **0.989347** | 0.989276 | | 0.01267 | 0.01263 | 0.01283 | 0.01183 |
| | 15 | 0.992333 | 0.992483 | **0.992529** | 0.992401 | | 0.01227 | 0.01283 | 0.0168 | 0.0132 |
| Lena | 5 | 0.968098 | 0.968212 | **0.96839** | 0.968315 | | 0.0115 | 0.001267 | 0.0105 | 0.005033 |
| | 8 | 0.984382 | **0.985277** | 0.985055 | 0.984949 | | 0.01257 | 0.0039 | 0.01413 | 0.01327 |
| | 10 | 0.989208 | **0.990203** | 0.98978 | 0.98956 | | 0.0131 | 0.005233 | 0.01417 | 0.014 |
| | 12 | 0.991884 | **0.992653** | 0.992276 | 0.992222 | | 0.01293 | 0.007367 | 0.01543 | 0.01503 |
| | 15 | 0.994345 | **0.994909** | 0.994507 | 0.994439 | | 0.0145 | 0.008767 | 0.0158 | 0.01633 |
| Peppers | 5 | 0.967926 | **0.96823** | 0.968125 | 0.968077 | | 0.01173 | 0.01117 | 0.0123 | 0.01173 |
| | 8 | 0.984386 | **0.984845** | 0.984844 | 0.984592 | | 0.01073 | 0.01217 | 0.01207 | 0.01263 |
| | 10 | 0.988955 | 0.989191 | **0.989244** | 0.989009 | | 0.0116 | 0.01287 | 0.01317 | 0.01277 |
| | 12 | 0.991711 | **0.991907** | 0.99185 | 0.991807 | | 0.0117 | 0.013 | 0.01307 | 0.01203 |
| | 15 | 0.994071 | 0.994177 | 0.994206 | **0.994216** | | 0.0138 | 0.01337 | 0.0144 | 0.01357 |
| Woman | 5 | 0.963698 | **0.96401** | 0.96393 | 0.963852 | | 0.009533 | 0.0109 | 0.0111 | 0.01127 |
| | 8 | 0.982118 | 0.982415 | **0.982431** | 0.982393 | | 0.01077 | 0.0122 | 0.0132 | 0.01137 |
| | 10 | 0.987687 | 0.987929 | **0.988018** | 0.987783 | | 0.01097 | 0.0124 | 0.01287 | 0.0117 |
| | 12 | 0.990841 | 0.99098 | **0.990999** | 0.990945 | | 0.01157 | 0.01313 | 0.0126 | 0.0126 |
| | 15 | 0.993554 | 0.9937 | **0.993711** | 0.993619 | | 0.01237 | 0.01377 | 0.0137 | 0.01343 |
| Best | | 0 | 10 | 9 | 1 | $\overline{time}$ | 0.0117(2) | 0.0106(1) | 0.0132(4) | 0.0124(3) |

## 5.3 Experimental results

The following settings of the DE were used for the experiments in this paper : $Np$ was set to 50, $G$ to 200, while the $F$ and $Cr$ were set to 0.9 and 0.2 respectively at the start of the evolutionary process. The stopping criteria was set to 10000 function evaluations, while each algorithm was run 30 times.

The results in Table 1 show the mean objective values, computed by each mutation strategy. Additionally the average computation times are reported. Based on mean values of the objective functions, the best results are obtained by using the best/1 mutation strategy, with the current to best/1 being a close second. When comparing average computation times the best/1 strategy again obtained the best result. In Table 2 the results of the PSNR metric is depicted (mean and best values). Based on the PSNR, the best mean value was obtained by the current to best/1 strategy, but when considering the best obtained results the rand/1 was the winner. The overall best results in both tables are marked in bold face.

## 6. CONCLUSION

The influence of different DE strategies has been investigated for the purpose of optimal multilevel thresholding problem. The strategies were tested on a set of four standard test images of size 512 × 512 pixels. The objective function during the evolution was maximizing the between-class variance proposed by Otsu. The obtained thresholds were compared on the basis of mean objective values, while also the well established metric PSNR was used to assess the quality of the segmented images. The experiments showed that the best performing strategy was best/1 when considering the mean objective values, while it also converged the fastest. On the other hand the mean PSNR metric showed that the best segmented images came from the current to best/1 strategy, but the best results were obtained using the classical rand/1 strategy.

Further work includes testing the mutation strategies by using other objective functions (i.e. Kapur's or Tsallis entropy).

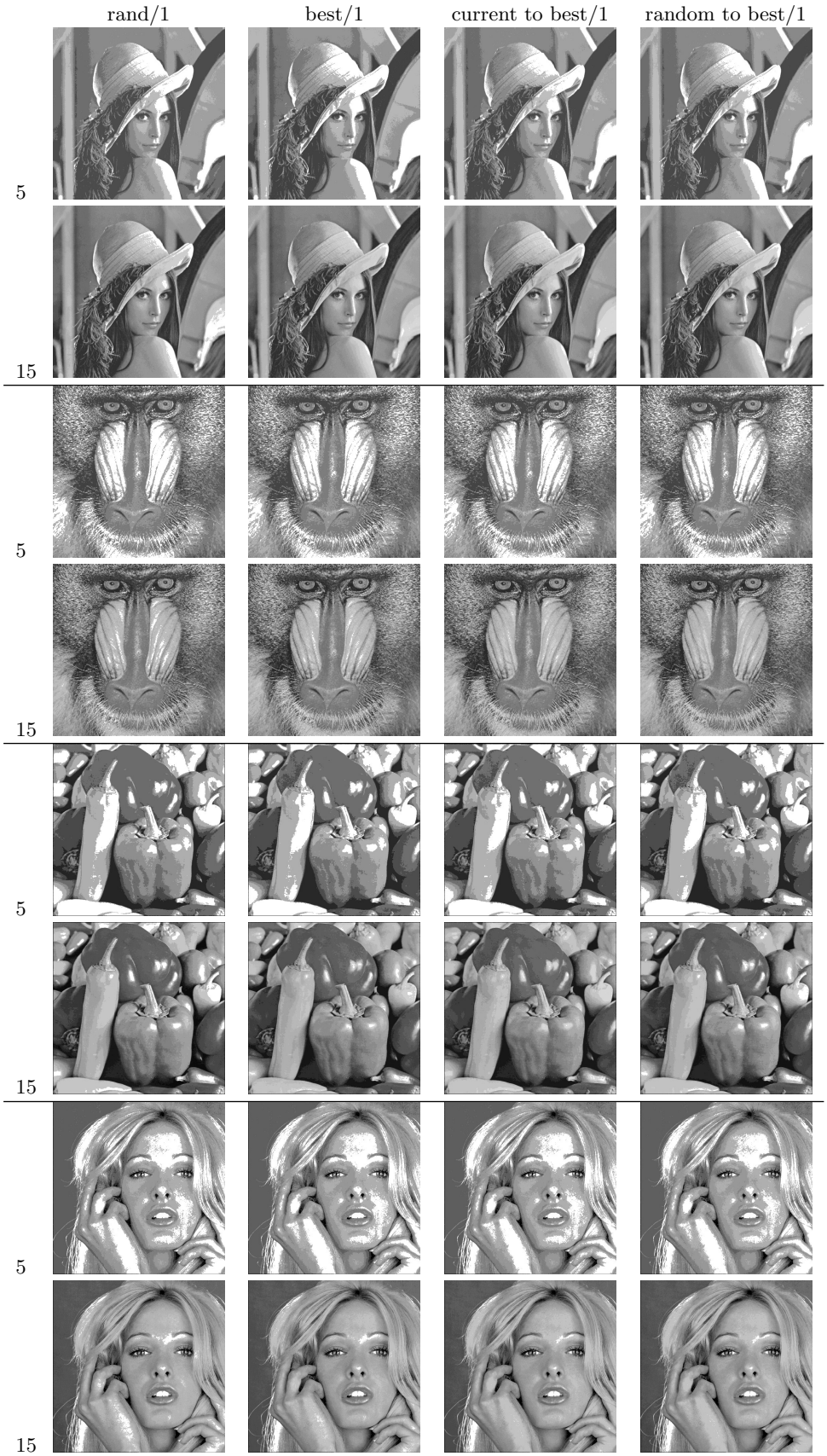


Figure 2: Test images segmented into 5 and 15 levels using different DE mutation strategies.

Table 2: Comparison of best mean objective values, with mean CPU times computed by DE, PSO, ABC, CS, and hjDE using Otsu criterion.

| Test image | M | Mean PSNR values | | | | | Best PSNR values | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| | | rand/1 | best/1 | current to best/1 | random to best/1 | | rand/1 | best/1 | current to best/1 | random to best/1 |
| Baboon | 5 | 18.1344 | **18.3655** | 18.3612 | 18.1425 | | 19.055000 | 19.057400 | 19.060400 | **19.313000** |
| | 8 | 21.0159 | 21.7262 | **22.1329** | 21.3687 | | **24.365000** | 24.010100 | 22.890600 | 23.928900 |
| | 10 | 22.4483 | 22.8946 | 23.3143 | **24.5666** | | **27.102000** | 26.624900 | 26.259100 | 26.598400 |
| | 12 | **28.6212** | 25.8498 | 23.5886 | 24.918 | | **29.052900** | 27.718700 | 28.932900 | 28.730200 |
| | 15 | 24.2914 | 28.7404 | 27.4966 | **28.8786** | | 30.405500 | 31.318300 | 32.102300 | **32.506900** |
| Lena | 5 | 17.3731 | 17.7323 | **17.7754** | 17.4229 | | **18.436100** | 17.807900 | 17.808400 | 18.126300 |
| | 8 | 20.303 | 20.0005 | 19.9081 | **20.8732** | | 22.146300 | 22.228500 | **22.260300** | 21.598900 |
| | 10 | 21.2402 | 21.2492 | **22.5055** | 21.9101 | | 24.120400 | 23.964400 | 23.392400 | **24.931900** |
| | 12 | **25.3207** | 22.7806 | 22.7796 | 21.4321 | | 26.859200 | **28.144400** | 26.572900 | 26.420300 |
| | 15 | 23.8341 | 24.7654 | **27.3944** | 23.1946 | | **30.855000** | 28.871100 | 29.830600 | 29.663600 |
| Peppers | 5 | 17.4975 | 18.1431 | **18.3017** | 17.846 | | **18.612700** | 18.157400 | 18.419600 | 18.447400 |
| | 8 | 23.442 | 23.2891 | 22.9984 | **24.0323** | | **24.656400** | 24.653500 | 24.487500 | 24.623600 |
| | 10 | **25.8584** | 24.9253 | 25.4991 | 25.5769 | | **26.449000** | 26.019600 | 26.139400 | 25.585600 |
| | 12 | 25.9307 | 25.9187 | **26.665** | 26.2836 | | 26.941100 | 27.028100 | 26.770500 | **27.283500** |
| | 15 | **27.4515** | 25.8253 | 26.0986 | 26.9089 | | 28.997200 | 30.301300 | **30.381100** | 28.991100 |
| Woman | 5 | **21.5085** | 21.0176 | 21.0176 | 21.0176 | | **21.508500** | 21.017600 | 21.167400 | 21.089100 |
| | 8 | 23.1922 | 23.2705 | 23.2292 | **23.436** | | 24.397700 | 24.742200 | 24.264600 | **24.787800** |
| | 10 | 25.5869 | 24.1715 | **26.0503** | 25.8962 | | **27.353000** | 26.242400 | 26.260400 | 27.177800 |
| | 12 | 27.7128 | **28.7043** | 26.8407 | 26.298 | | 28.725100 | **28.754200** | 28.746300 | 28.699800 |
| | 15 | 26.8773 | 29.841 | **31.6153** | 27.5219 | | 31.824400 | **31.917100** | 31.646000 | 31.114900 |
| Best | | 5 | 2 | 8 | 5 | | 10 | 3 | 2 | 5 |

## 7. REFERENCES


[1] Adis Alihodzic and Milan Tuba. Improved bat algorithm applied to multilevel image thresholding. *The Scientific World Journal*, 2014, 2014.

[2] Ashish Kumar Bhandari, Vineet Kumar Singh, Anil Kumar, and Girish Kumar Singh. Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using kapur's entropy. *Expert Systems with Applications*, 41(7):3538–3560, 2014.

[3] Sathya P Duraisamy, Ramanujam Kayalvizhi, et al. A new multilevel thresholding method using swarm intelligence algorithm for image segmentation. *Journal of Intelligent Learning Systems and Applications*, 2(03):126, 2010.

[4] Iztok Fister Jr., Xin-She Yang, Iztok Fister, Janez Brest, and Dusan Fister. A brief review of nature-inspired algorithms for optimization. *CoRR*, abs/1307.4186, 2013.

[5] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *Automatica*, 11(285-296):23–27, 1975.

[6] Rainer Storn and Kenneth Price. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. *Journal of global optimization*, 11(4):341–359, 1997.

[7] Xin-She Yang and Xingshi He. Swarm intelligence and evolutionary computation: Overview and analysis. In Xin-She Yang, editor, *Recent Advances in Swarm Intelligence and Evolutionary Computation*, volume 585 of *Studies in Computational Intelligence*, pages 1–23. Springer International Publishing, 2015.

[8] Yudong Zhang and Lenan Wu. Optimal multi-level thresholding based on maximum tsallis entropy via an artificial bee colony approach. *Entropy*, 13(4):841–859, 2011.