Many-level Image Thresholding using a Center-Based Differential Evolution Algorithm

Seyed Jalaleddin Mousavirad*, Shahryar Rahnamayan, SMIEEE†
Nature-Inspired Computational Intelligence Lab
Department of Electrical, Computer, and Software Engineering
Ontario Tech University
Oshawa, Canada
Email: *jalalmoosavirad@gmail.com, †shahryar.rahnamayan@uoit.ca

Gerald Schaefer
Department of Computer Science
Loughborough University
Loughborough, U.K.
gerald.schaefer@ieee.org

***Abstract*—Image thresholding is a crucial image processing task. Most of the time, it plays a pivotal role in an image processing chain, therefore, any error in image thresholding can propagate to other steps such as edge detection, area/volume estimation, or object recognition. Multi-level image thresholding is a popular method for image segmentation, dividing an image into homogeneous regions. Conventional algorithms are time-consuming due to utilising an exhaustive search, especially when the number of threshold levels increases. On the other hand, population-based metaheuristic algorithms have been successfully applied to this problem. In this paper, we propose a center-based differential evolution (DE) algorithm for high-dimensional multi-level image thresholding (many-level image thresholding). While DE has been shown to yield satisfactory performance for various real-world optimisation problems, in our algorithm, DE is further boosted with a center-based sampling strategy. We evaluate our algorithm on a set of benchmark images on high-dimensional search spaces and with regards to an entropy-based objective function and peak signal-to-noise ratio (PSNR). The obtained results demonstrate that the proposed algorithm can improve upon the performance of other metaheuristic image thresholding techniques.**

***Index Terms*—Image thresholding, optimisation, differential evolution, center-based sampling.**

I. Introduction

Image segmentation plays a fundamental role in machine vision applications and divides an image into non-overlapping groups so that pixels located in the same region share similar characteristics, while pixels from distinct regions exhibit more differences. Image thresholding represents a popular approach to image segmentation due to its simplicity, robustness, and accuracy [1]. Bi-level thresholding selects a single threshold, whereas multi-level thresholding selects multiple thresholds and represents a challenging task that has attracted much research attention in recent years.

Conventional image thresholding algorithms work efficiently for bi-level thresholding, but their efficiency decreases dramatically for multi-level thresholding since an increasing number of thresholds leads to a significant increase in term of computational complexity of an exhaustive search. To overcome this problem, population-based metaheuristic algorithms such as genetic algorithm (GA) [2], particle swarm optimisation (PSO) [3], differential evolution (DE) [4] and human mental search (HMS) [5] can be employed and various such approaches have been recently reported for multi-level image thresholding. For example, [6] proposes a PSO-based image thresholding method based on minimum cross-entropy and compares it with other methods for $k = 2, 3, 4$ where k is the number of threshold levels. DE is employed for image thresholding in [7] using an Otsu-based objective function. In [8], teaching-learning-based optimisation (TLBO) is used for image thresholding and evaluated for $k = 2, 3, 4, 5$. Other population-based algorithms such as salp swarm algorithm (SSA) [9], sine cosine algorithm (SCA) [10], and HMS [11] have also been employed for image thresholding. Most studies have concentrated on fewer than 6 thresholds, while higher-dimensional image thresholding, many-level image thresholding, is not often considered [1]. Here, the task becomes much more challenging due to the curse of dimensionality whereby the search space expands exponentially when increasing the number of dimensions while good solutions are sparsely distributed.

Differential evolution is an effective population-based metaheuristic algorithm that has been successfully employed for a variety of applications [12], [13]. DE is based on three main operators, mutation, crossover, and selection. Mutation is the core operator in DE and generates a mutant vector based on a linear combination of distinct candidate solutions. The role of the crossover operator is to combine a mutant vector with its target vector to yield a trial vector. Finally, the selection operator employs a greedy strategy to select the better of trial vector and target vectors.

In the center-based sampling strategy introduced in [14], it was shown that the probability of closeness to an unknown solution for the center point is significantly higher than for other points and that consequently, the center point is a valuable point compared to any point generated randomly. A center-based sampling strategy is used to improve DE in [15] by employing the center of three best candidate solutions as the base vector for mutation, while [16] proposes a center-based SHADE (success history-based parameter adaptation DE) algorithm, with mutation employing a center-based candidate solution using a normal distribution. [17] uses a center-based initialisation algorithm to tackle deceptive optimisation algo-

978-1-7281-6929-3/20/$31.00 ©2020 IEEE

rithms, whereas [14] shows that the probability of the center point being close to the optimal solution increases sharply with an increasing number of dimensions. As a result, we can say that a center-based sampling strategy has a potential ability to boost the performance of high-dimensional multi-level image thresholding.

In this paper, we propose a center-based differential evolution algorithm for high-dimensional multi-level image thresholding (many-level image thresholding). We use an entropy-based objective function and evaluate our algorithm on a set of benchmark images to demonstrate excellent thresholding performance superior to several other population-based metaheuristic algorithms.

The remainder of the paper is organised as follows. Section II provides some background on the underlying techniques, while Section III presents our proposed algorithm. Experimental results are reported in Section IV and Section V concludes the paper.

```
Input  : $D$: dimensionality of problem; $NFE_{\max}$:
         maximum number of function evaluations;
         $N_P$: population size, $F$: scaling factor, $CR$:
         crossover rate
Output: $x^*$: the best solution
generate initial population $Pop$ randomly;
evaluate fitness for each candidate solution;
$NFE = N_P$ ;
while $NFE < NFE_{\max}$ do
    for $i \leftarrow 1$ to $N_P$ do
        select three parents, $x_{r1}$, $x_{r2}$, and $x_{r3}$,
          randomly from current population, with
          $x_{r1} \neq x_{r2} \neq x_{r3}$;
        $v_i = x_{r1} + F * (x_{r2} - x_{r3})$;
        for $j \leftarrow 0$ to $D$ do
            if $rand_j[0,1] < CR \; or \; j == j_{rand}$ then
                $u_{i,j} = v_{i,j}$;
            else
                $u_{i,j} = x_{i,j}$;
            end
        end
        calculate objective function of $u_i$;
        if $f(u_i) < f(x_i)$ then
            $\bar{x} \leftarrow u_i$;
        else
            $\bar{x} \leftarrow x_i$;
        end
        $Pop(i) \leftarrow \bar{x}$;
    end
    $NFE = NFE + N_P$;
end
$x^* \leftarrow$ best candidate solution in $Pop$
```

Algorithm 1: Pseudo-code of DE.

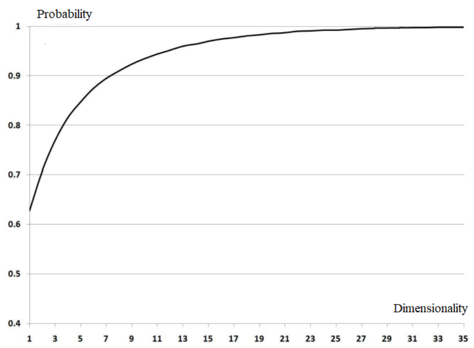


Fig. 1: Probability of center point closeness to solution versus the number of dimensions [14].

II. Background

A. Multi-level Image Thresholding

The aim in multi-level image thresholding is to find threshold values to yield pixel ranges that allow to separate an image into non-overlapping regions. More specifically, multi-level image thresholding can formally be formulated as

$$\begin{aligned} M_0 &= \{g(x,y) \in I | 0 \leq g(x,y) \leq t_1 - 1\} \\ M_1 &= \{g(x,y) \in I | t_1 \leq g(x,y) \leq t_2 - 1\} \\ M_i &= \{g(x,y) \in I | t_i \leq g(x,y) \leq t_{i+1} - 1\} \\ M_m &= \{g(x,y) \in I | t_m \leq g(x,y) \leq 2^L - 1\} \end{aligned} \quad (1)$$

where $t_i, i = 1, ..., m$ is the i-th threshold value, m is the number of threshold values, I is the original image and $g(x,y)$ denotes the pixel value at location (x,y).

B. Canonical Differential Evolution

Differential evolution (DE) [4] is a population-based algorithm which has shown notable performance to tackle different

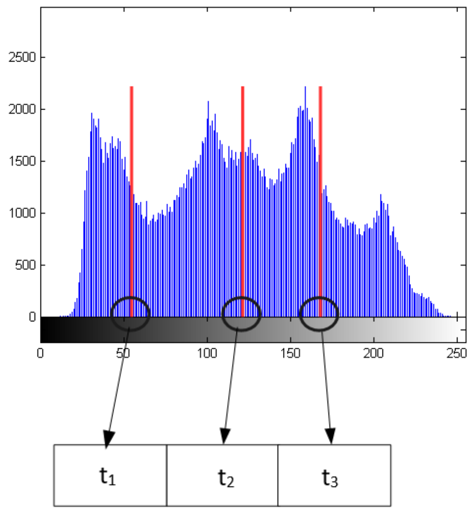
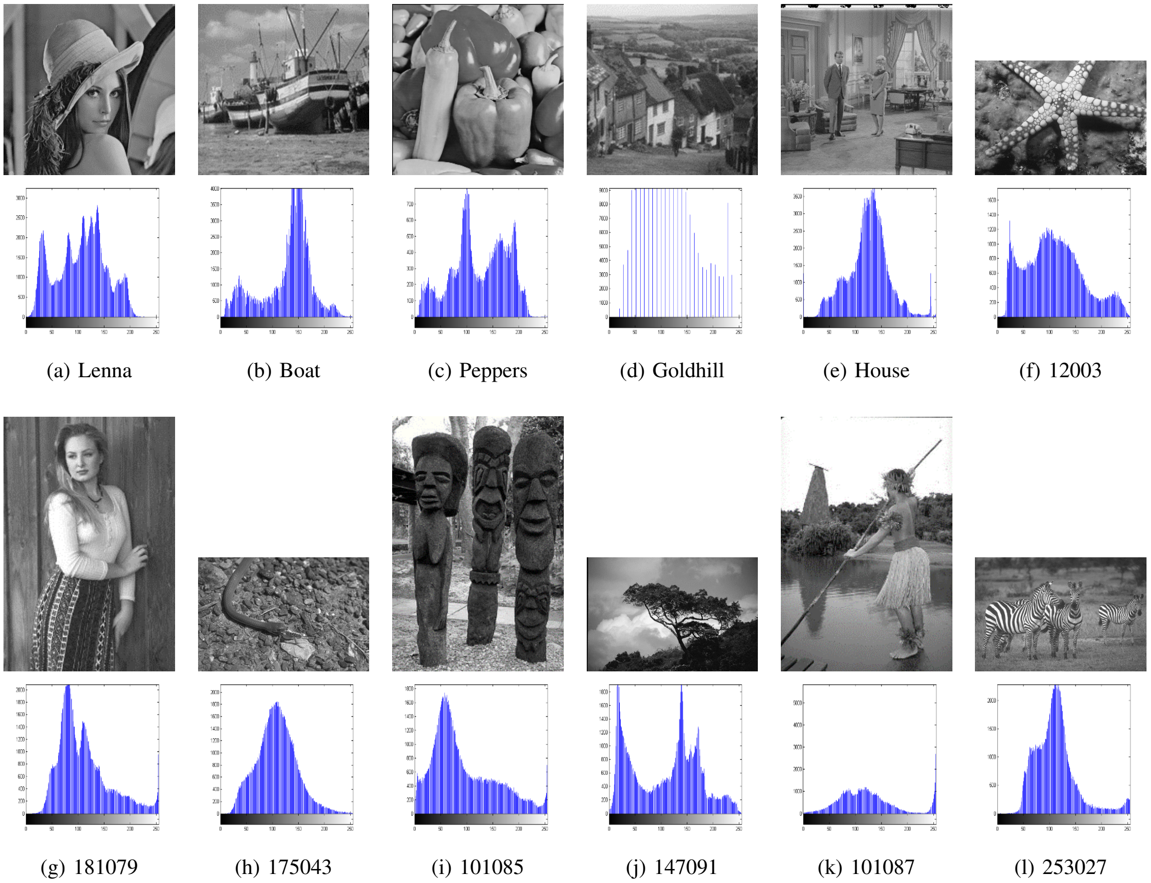


Fig. 2: Data representation example in a three-level image thresholding.



(a) Lenna (b) Boat (c) Peppers (d) Goldhill (e) House (f) 12003

(g) 181079 (h) 175043 (i) 101085 (j) 147091 (k) 101087 (l) 253027

Fig. 3: Test images and their histograms.

optimisation problems. DE starts with N_P randomly generated candidate solutions and proceeds based on three operators, mutation, crossover, and selection.

Mutation generates a mutant vector as

$$v_i = x_{r1} + F * (x_{r2} - x_{r3}), \tag{2}$$

where x_{r1}, x_{r2} and x_{r3} are three candidate solutions and F is a scaling factor.

The aim of crossover is to generate a new candidate solution based on a combination of a mutant vector with a parent vector to yield a trial vector as

$$u_{i,j} = \begin{cases} v_{i,j} & \text{if } rand(0,1) \leq CR \text{ or } j = j_{rand} \\ x_{i,j} & \text{otherwise,} \end{cases} \tag{3}$$

where CR is the crossover rate, and j_{rand} is a random integer number in $[1 : D]$ with D the dimensionality of the search space.

Finally, the selection operator selects the best candidate solution between the new candidate solution (trial vector) and the previous one (its parent).

Algorithm 1 summarises DE in terms of pseudo-code.

C. Center-Based Sampling Strategy

The probability of points being closer to an unknown solution is found to be remarkably higher for the center point than for uniformly generated points, while this probability increases with an increasing number of dimensions. This is investigated using Monte Carlo simulations in [14], while a mathematical proof is provided in [18].

Fig. 1 investigates the probability of closeness of the center point (p_c) to the solution in comparison to a random point. The probability rises steeply with an increasing number of dimensions which verifies that center-based sampling is can significantly aid to solve optimisation problems and in particular higher-dimensional ones.

The center point for a search space is defined as

$$c = \frac{a_i + b_i}{2}, \tag{4}$$

where a and b are the lower and upper bounds, $i = 1, ..., D$, and D is the number of dimensions.

A center-point sampling strategy has been succesfully employed to boost metaheuristic algorithms in [16], [17], [19].

III. Center-Based DE for Many-Level Image Thresholding

In this paper, we propose a center-based differential evolution algorithm (CenDE) for many-level image thresholding. For that, we will first define a data representation and an objective function, and then explain the center-based DE algorithm.

TABLE I: Parameter settings for the experiments.

	parameters	value
DE	scaling factor	0.5
	crossover rate	0.9
PSO	cognitive constant (C_1)	2
	social constant (C_2)	2
	inertia constant (w)	1 to 0
SCA	a	2
SSA	no parameters	

A. Data Representation

The data representation defines the structure of each candidate solution. We employ a real-valued encoding strategy with lower and upper bounds of each component set to 0 and $2^n - 1$ where n is the bitdepth. Fig. 2 gives an example for a problem with 3 threshold levels in an 8-bit image.

B. Objective Function

We evaluate the quality of each candidate solution based on maximising the entropy of the thresholded image. In particular, we employ Kapur's entropy [11], which has been previously used for thresholding [20], and is defined as

$$f([t_1, t_2, ...t_m]) = H_0 + H_1 + ... + H_m, \tag{5}$$

with

$$\begin{aligned} H_0 &= -\sum_{i=0}^{t_1-1} (p_i/\omega_0) ln(p_i/\omega_0), \omega_0 = \sum_{i=0}^{t_1-1} p_i \\ H_1 &= -\sum_{i=t_1}^{t_2-1} (p_i/\omega_1) ln(p_i/\omega_1), \omega_1 = \sum_{i=t_1}^{t_2-1} \\ H_2 &= -\sum_{i=t_2}^{t_3-1} (p_i/\omega_2) ln(p_i/\omega_2), \omega_2 = \sum_{i=t_2}^{t_3-1} p_i \\ H_m &= -\sum_{t_m}^{L-1} (p_i/\omega_m) ln(p_i/\omega_m), \omega_m = \sum_{i=t_m}^{L-1} p_i \end{aligned} \tag{6}$$

where m is the number of threshold values, N is the total number of pixels in the image, L indicates the number of grey levels, and p_i is calculated as $p_i = h_i/N$, where h_i is the probability of occurrence of grey level i.

C. Center-based DE algorithm

As mentioned above, mutation in conventional DE is based on three candidate solutions selected randomly from the current population. In contrast, in center-based DE, a new base vector is introduced at the center of three candidate solutions.

Inspired by [16], [19], the new mutation operator selects five candidate solutions, x_{r1}, x_{r2}, x_{r3}, x_{r4}, and x_{r5}, randomly from the current population. Then, the average of three candidate solutions is computed as

$$x_{center} = \frac{x_{r1} + x_{r2} + x_{r3}}{3}, \tag{7}$$

and mutation performed as

$$v_i = x_{center} + F(x_{r4} - x_{r5}) \tag{8}$$

TABLE II: Objective function results for all algorithms and all images.

image	D	PSO	SSA	SCA	DE	CenDE
Lenna	10	32.09	31.76	30.70	32.18	**32.50**
	15	40.25	40.04	38.51	40.45	**40.99**
	20	47.04	46.46	44.72	46.70	**47.57**
Boat	10	32.68	32.48	31.45	32.77	**32.82**
	15	41.32	40.99	39.28	41.25	**41.65**
	20	48.02	47.67	45.73	47.68	**48.06**
Peppers	10	31.96	31.76	30.42	32.13	**32.49**
	15	40.00	39.75	37.80	39.91	**41.10**
	20	46.34	45.81	43.32	45.87	**47.35**
Goldhill	10	18.41	17.21	16.38	18.87	**21.28**
	15	21.30	19.96	18.84	21.71	**26.32**
	20	23.42	22.05	20.78	23.73	**30.72**
House	10	31.48	31.29	29.99	31.65	**32.23**
	15	39.45	39.24	36.85	39.38	**40.37**
	20	45.58	45.15	43.30	45.24	**46.23**
12003	10	33.02	32.86	31.82	33.10	**33.43**
	15	41.49	41.19	40.01	41.42	**42.02**
	20	48.16	47.85	45.99	47.78	**48.47**
181079	10	32.30	32.09	31.09	32.51	**32.89**
	15	40.95	40.66	39.29	41.09	**41.45**
	20	**47.90**	47.50	45.80	47.65	47.71
175043	10	32.73	32.52	31.44	**32.83**	32.47
	15	**41.32**	40.84	39.54	41.12	41.17
	20	**47.91**	47.35	45.91	47.44	47.72
101085	10	**33.29**	33.17	32.27	33.34	32.02
	15	**42.07**	41.83	40.47	42.02	40.01
	20	**48.85**	48.50	46.76	48.49	46.74
147091	10	33.25	33.09	32.36	**33.40**	33.15
	15	**41.95**	41.68	40.17	41.93	41.78
	20	48.70	48.06	46.74	**48.46**	48.07
101087	10	31.75	31.41	30.49	31.81	**32.25**
	15	**40.77**	40.34	38.93	40.65	40.65
	20	**47.88**	47.02	45.39	47.30	47.16
253027	10	31.93	31.63	30.47	32.03	**32.95**
	15	40.28	39.94	38.12	40.25	**41.29**
	20	46.63	46.26	44.22	46.12	**47.52**

Our algorithm begins with an initial population of N_P candidate solutions. Then, during each iteration, five candidate solutions are randomly selected from the current population. Three of these are employed to calculate x_{center} and a mutant vector is generated based on Eq. (8). Crossover is then applied based on Eq.(3), and finally, selection is performed based on a greedy selection strategy. This process is repeated until a stopping condition is satisfied (a maximum number of objective function evaluations in this paper).

IV. Experimental Results

In our experiments, we evaluate our proposed algorithm on a set of benchmark images. We select five commonly employed images, namely, *Lenna*, *Boat*, *Peppers*, *Goldhill*, and *House*, as well as some images previously used for thresholding evaluation [1], [11], [20] from the Berkley Segmentation Data Set and Benchmark [21], namely, *12003*, *181079*, *175043*, *101085*, *147091*, *101087*, and *253027*. All images as well as their corresponding histograms are shown in Fig. 3. As can be seen from there, some images such as *Lenna* and *Peppers* have multiple peaks and valleys in their histograms, while others such as *175045* have only one peak. Images such as *101087* have a smooth histogram, while *Goldhill* has a histogram with abrupt alterations.

TABLE III: Friedman ranks based on objective function value.

image	D	PSO	SSA	SCA	DE	CenDE
Lenna	10	3	4	5	2	1
	15	3	4	5	2	1
	20	2	4	5	3	1
Boat	10	3	4	5	2	1
	15	2	4	5	3	1
	20	2	4	5	3	1
Peppers	10	3	4	5	2	1
	15	2	4	5	3	1
	20	2	4	5	3	1
Goldhill	10	3	4	5	2	1
	15	3	4	5	2	1
	20	3	4	5	2	1
House	10	3	4	5	2	1
	15	2	4	5	3	1
	20	2	4	5	3	1
12003	10	3	4	5	2	1
	15	2	4	5	3	1
	20	2	3	5	4	1
181079	10	3	4	5	2	1
	15	3	4	5	2	1
	20	1	4	5	3	2
175043	10	2	3	5	1	4
	15	1	4	5	3	2
	20	1	4	5	3	2
101085	10	2	3	4	1	5
	15	1	3	4	2	5
	20	1	2	4	3	5
147091	10	2	4	5	1	3
	15	1	4	5	2	3
	20	1	4	5	2	3
101087	10	3	4	5	2	1
	15	1	4	5	2.5	2.5
	20	1	4	5	2	3
253027	10	3	4	5	2	1
	15	2	4	5	3	1
	20	2	3	5	4	1
average rank		2.11	3.81	4.92	2.40	**1.76**

We compare our proposed CenDE algorithm with conventional DE, as well as some state-of-the-art and recent methods, including PSO [6], SSA [9], and SCA [10]. The same data representation and objective function is used for all algorithms.

The population size and maximum number of function evaluations are set to 50 and 10000, respectively, for all algorithms, while other parameter settings are given in Table I. To evaluate high-dimensional multi-thresholding, we set D to 10, 15, and 20 [1], while due to the stochastic nature of the algorithms, we run each 25 times.

Table II gives the results in terms of mean objective function value for all algorithms and all images. From there we can see that CenDE yields significantly better objective function values in comparison to all other algorithms giving the best result for 24 out of 36 cases.

Table III reports the resulting Friedman ranks (p-value is 6.32E-21) and further demonstrates that CenDE is clearly the best algorithm overall.

To assess the robustness of our algorithm, we list the standard deviations of the algorithms in Table IV. CenDE yields the lowest standard deviation for 23 of the 36 cases indicating superior robustness compared to the other methods.

TABLE IV: Standard deviation of objective function value for all algorithms and images.

image	D	PSO	SSA	SCA	DE	CenDE
Lenna	10	0.22	0.27	0.39	**0.12**	0.16
	15	0.49	0.53	0.68	0.32	**0.31**
	20	0.71	0.54	0.71	0.55	**0.41**
Boat	10	0.19	0.20	0.45	**0.12**	**0.12**
	15	0.51	0.29	0.58	0.34	**0.19**
	20	0.79	0.51	0.69	0.52	**0.51**
Peppers	10	0.27	0.25	0.52	0.14	**0.13**
	15	0.51	0.39	0.76	**0.33**	0.36
	20	0.60	**0.40**	0.94	0.67	0.42
Goldhill	10	**0.39**	0.53	0.62	0.41	0.56
	15	0.76	0.87	0.89	**0.49**	0.68
	20	0.66	0.71	0.83	0.70	**0.64**
House	10	0.20	0.23	0.61	**0.12**	0.16
	15	0.36	0.45	0.63	**0.36**	**0.36**
	20	0.74	0.51	0.73	0.54	**0.35**
12003	10	0.20	0.18	0.36	0.12	**0.11**
	15	0.27	0.31	0.7	0.32	**0.29**
	20	0.50	0.61	1.05	0.49	**0.33**
181079	10	0.23	0.25	0.40	**0.13**	0.24
	15	0.48	0.33	0.92	**0.31**	**0.31**
	20	0.77	0.82	0.93	0.63	**0.52**
175043	10	0.20	0.24	0.49	**0.12**	**0.12**
	15	0.47	0.25	0.77	0.34	**0.24**
	20	0.64	0.56	0.83	0.64	**0.43**
101085	10	0.16	0.18	0.40	**0.14**	**0.14**
	15	**0.32**	0.29	0.60	0.36	0.62
	20	0.54	0.49	0.78	**0.42**	0.57
147091	10	0.19	0.18	0.54	0.12	**0.08**
	15	0.30	0.32	0.77	0.28	**0.27**
	20	0.50	0.58	0.74	0.48	**0.38**
101087	10	0.16	0.24	0.50	0.11	**0.10**
	15	0.33	0.35	0.56	0.31	**0.28**
	20	0.65	0.56	0.83	**0.48**	0.50
253027	10	0.15	0.21	0.51	**0.12**	0.18
	15	0.39	0.44	0.67	**0.26**	0.40
	20	0.65	**0.40**	0.94	0.59	0.52

We further employ a two-sided Wilcoxon signed-rank test [22] with a significance level of 95% to evaluate the proposed algorithms statistically based on objective function results and the manner explained in [22]. The results are given in Table V and show that with p-values below 0.05 CenDE is statistically significantly better compared to the other algorithms.

Peak signal-to-noise ratio (PSNR) is a commonly employed measure to evaluate image quality and calculated as

$$PSNR = 20 \log_{10}(255/RMSE), \tag{9}$$

where $RMSE$ is the root mean squared error defined as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{M} \sum_{j=1}^{N} (I(i,j) - \hat{I}(i,j))^2}{MN}}, \tag{10}$$

where M and N are the image dimensions, and I and $\hat{I}$ are the original and thresholded images, respectively. A higher PSNR signifies better image quality.

TABLE V: Results of Wilcoxon signed rank tests.

CenDE vs.	p-value
DE	0.0218
SCA	3.8429E-04
SSA	3.4045E-04
PSO	0.0048

TABLE VI: PSNR results for all algorithms and all images.

image	*D*	PSO	SSA	SCA	DE	CenDE
Lenna	10	31.23	30.88	29.52	31.32	**31.59**
	15	33.22	33.24	31.79	33.19	**33.36**
	20	34.45	**34.84**	33.05	34.48	34.68
Boat	10	30.75	31.18	29.49	31.14	**31.22**
	15	33.17	33.06	32.17	33.23	**33.48**
	20	34.20	34.68	33.33	34.51	**34.69**
Peppers	10	31.04	30.88	29.64	**31.29**	30.85
	15	33.21	33.18	31.63	**33.39**	32.92
	20	34.77	34.65	33.34	**34.88**	34.27
Goldhill	10	24.96	**26.59**	24.37	25.01	23.49
	15	26.34	**28.29**	26.82	27.08	26.66
	20	28.17	**30.68**	28.54	28.62	30.09
House	10	30.50	30.57	29.66	30.82	**31.15**
	15	32.80	32.60	31.44	32.61	**33.52**
	20	34.46	34.07	33.87	34.67	**34.75**
12003	10	31.10	30.85	29.04	**31.24**	31.07
	15	**33.18**	33.08	31.68	33.00	33.07
	20	34.21	34.43	32.81	33.86	**34.60**
181079	10	30.09	30.75	28.79	30.41	**30.91**
	15	32.41	32.82	30.99	32.70	**33.46**
	20	34.59	34.13	33.34	34.07	**35.24**
175043	10	30.98	30.92	29.03	**31.31**	30.81
	15	32.80	33.14	31.52	32.32	**33.18**
	20	34.46	34.70	32.62	33.84	**34.74**
101085	10	**30.49**	30.05	29.22	30.46	25.55
	15	**32.88**	32.45	31.39	32.62	26.78
	20	34.00	**34.09**	32.70	33.91	28.93
147091	10	31.00	30.84	29.66	31.09	**31.24**
	15	32.92	32.80	31.17	32.89	**33.20**
	20	34.39	34.19	33.14	34.34	**34.48**
101087	10	31.34	31.36	29.88	**31.74**	31.40
	15	**33.82**	34.26	31.98	33.66	32.84
	20	35.11	**35.59**	33.84	35.14	34.58
253027	10	31.19	31.10	30.23	**31.34**	31.08
	15	32.88	**33.87**	32.06	33.31	33.33
	20	34.72	**35.24**	33.44	34.48	34.69

TABLE VII: Friedman ranks based on PSNR.

image	*D*	PSO	SSA	SCA	DE	CenDE
Lenna	10	3	4	5	2	1
	15	3	2	5	4	1
	20	4	1	5	3	2
Boat	10	4	2	5	3	1
	15	3	4	5	2	1
	20	4	2	5	3	1
Peppers	10	2	3	5	1	4
	15	2	3	5	1	4
	20	2	3	5	1	4
Goldhill	10	3	1	4	2	5
	15	5	1	3	2	4
	20	5	1	4	3	2
House	10	4	3	5	2	1
	15	2	4	5	3	1
	20	3	4	5	2	1
12003	10	2	4	5	1	3
	15	1	2	5	4	3
	20	3	2	5	4	1
181079	10	4	2	5	3	1
	15	4	2	5	3	1
	20	2	3	5	4	1
175043	10	2	3	5	1	4
	15	3	2	5	4	1
	20	3	2	5	4	1
101085	10	1	3	4	2	5
	15	1	3	4	2	5
	20	2	1	4	3	5
147091	10	3	4	5	2	1
	15	2	4	5	3	1
	20	2	4	5	3	1
101087	10	4	3	5	1	2
	15	2	1	5	3	4
	20	3	1	5	2	4
253027	10	2	3	5	1	4
	15	4	1	5	3	2
	20	2	1	5	4	3
average rank		2.81	2.47	4.81	2.53	**2.38**

Table VI shows the mean PSNR values for all algorithms and all images[1], whereas Table VII reports the corresponding Friedman ranks (p-value is 2.7787E-12). These results further demonstrate the strong performance of our proposed CenDE thresholding algorithm in comparison with the other techniques. CenDE gives the highest PSNR for 17 of the 36 cases and the second highest for a further 4 cases, and is thus clearly ranked first overall.

Finally, we compare our algorithm visually on image *385028* as a representative example. Fig. 4 shows the manual segmentations provided in the Berkley Segmentation Data Set as well as the 10-level thresholded images by the various methods and further illustrates the good performance of CenDE which for example is able to segment the lake regions with less noise.

V. Conclusions

In this paper, we have proposed a novel algorithm based on differential evolution (DE) to tackle the challenging many-level image thresholding problem. Our proposed CenDE algorithm employs a center-based sampling strategy, using the center of three randomly selected candidate solutions as the base vector for mutation, to boost the performance of DE. Extensive experimental results on a set of benchmark images and in terms of both objective function value and PSNR demonstrate CenDE to delivery very good thresholding performance and to outperform, also statistically, a number of competing algorithms.

References

[1] S. J. Mousavirad, G. Schaefer, and H. Ebrahimpour-Komleh, “A benchmark of population-based metaheuristic algorithms for high-dimensional multi-level image thresholding,” in *IEEE Congress on Evolutionary Computation*, 2019, pp. 2394–2401.

[2] D. Whitley, “A genetic algorithm tutorial,” *Statistics and Computing*, vol. 4, no. 2, pp. 65–85, 1994.

[3] Y. Shi and R. Eberhart, “A modified particle swarm optimizer,” in *IEEE International Conference on Evolutionary Computation*, 1998, pp. 69–73.

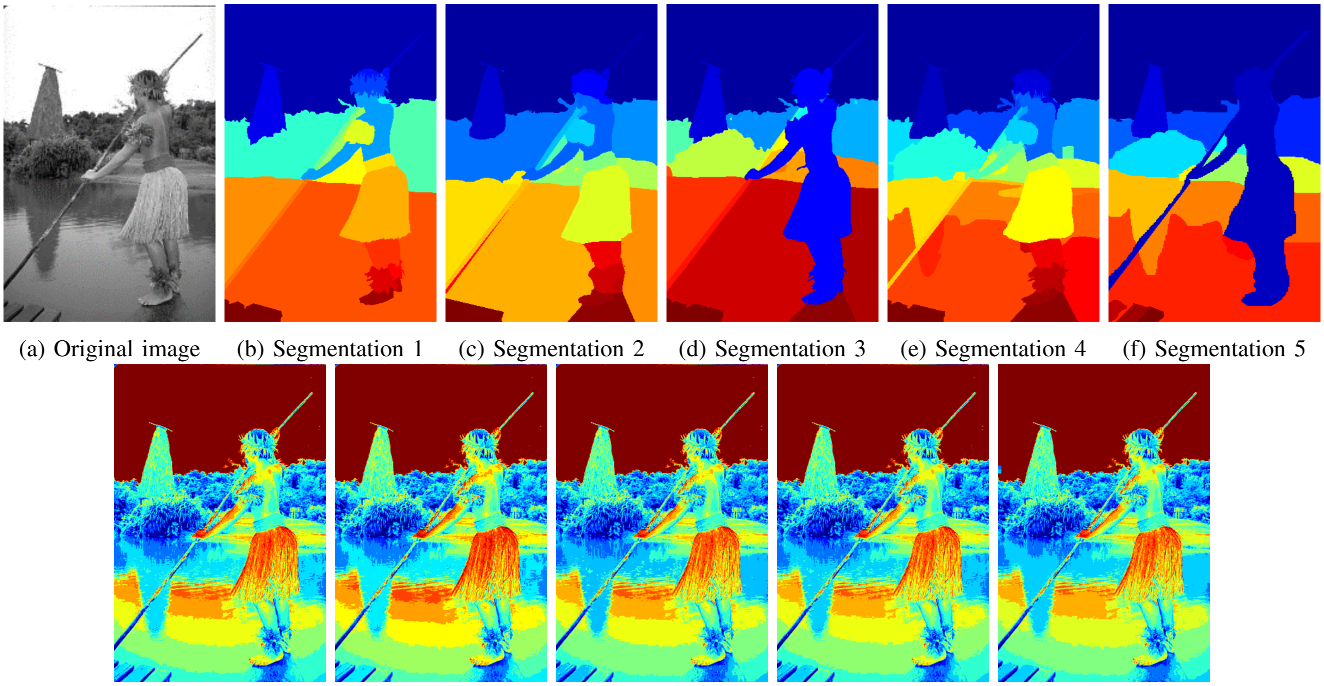
[4] R. Storn and K. Price, “Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces,” *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[5] S. J. Mousavirad and H. Ebrahimpour-Komleh, “Human mental search: a new population-based metaheuristic optimization algorithm,” *Applied Intelligence*, vol. 47, no. 3, pp. 850–887, 2017.

[6] P.-Y. Yin, “Multilevel minimum cross entropy threshold selection based on particle swarm optimization,” *Applied Mathematics and Computation*, vol. 184, no. 2, pp. 503–513, 2007.

[7] K. Charansiriphaisan, S. Chiewchanwattana, and K. Sunat, “A global multilevel thresholding using differential evolution approach,” *Mathematical Problems in Engineering*, vol. 2014, 2014.

[1] Different from e.g. [1] where pixel values were replaced with threshold values, we here replace pixel values by the weighted average within threshold intervals to yield the best possible quality of a thresholded image.


(a) Original image (b) Segmentation 1 (c) Segmentation 2 (d) Segmentation 3 (e) Segmentation 4 (f) Segmentation 5

(g) PSO (h) SSA (i) SCA (j) DE (k) CenDE

Fig. 4: Segmented images for image 1010878.

[8] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Optimal multilevel image thresholding using the teaching-learning-based optimization," *Machine Vision and Image Processing*, vol. 2, no. 2, pp. 51–62, 2016.

[9] S. Wang, H. Jia, and X. Peng, "Modified salp swarm algorithm based multilevel thresholding for color image segmentation," *Mathematical Biosciences and Engineering*, vol. 17, no. 1, pp. 700–724, 2019.

[10] S. Gupta and K. Deep, "Improved sine cosine algorithm with crossover scheme for global optimization," *Knowledge-Based Systems*, vol. 165, pp. 374–406, 2019.

[11] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Human mental search-based multilevel thresholding for image segmentation," *Applied Soft Computing*, 2019.

[12] S. Das and A. Konar, "Automatic image pixel clustering with an improved differential evolution," *Applied Soft Computing*, vol. 9, no. 1, pp. 226–236, 2009.

[13] I. Fister, D. Fister, S. Deb, U. Mlakar, and J. Brest, "Post hoc analysis of sport performance with differential evolution," *Neural Computing and Applications*, pp. 1–10, 2018.

[14] S. Rahnamayan and G. G. Wang, "Center-based sampling for population-based algorithms," in *IEEE Congress on Evolutionary Computation*, 2009, pp. 933–938.

[15] H. Salehinejad, S. Rahnamayan, and H. R. Tizhoosh, "CenDE: Centroid-based differential evolution," in *IEEE Canadian Conference on Electrical & Computer Engineering*, 2018, pp. 1–4.

[16] H. Hiba, M. El-Abd, and S. Rahnamayan, "Improving SHADE with center-based mutation for large-scale optimization," in *IEEE Congress on Evolutionary Computation*, 2019, pp. 1533–1540.

[17] S. J. Mousavirad, A. Asilian Bidgoli, and S. Rahnamayan, "Tackling deceptive optimization problems using opposition-based DE with center-based Latin hypercube initialization," in *14th International Conference on Computer Science and Education*, 2019.

[18] S. Rahnamayan and G. G. Wang, "Toward effective initialization for large-scale search spaces," *Trans Syst*, vol. 8, no. 3, pp. 355–367, 2009.

[19] H. Hiba, S. Mahdavi, and S. Rahnamayan, "Differential evolution with center-based mutation for large-scale optimization," in *IEEE Symposium Series on Computational Intelligence*, 2017, pp. 1–8.

[20] S. J. Mousavirad and H. Ebrahimpour-Komleh, "Multilevel image thresholding using entropy of histogram and recently developed population-based metaheuristic algorithms," *Evolutionary Intelligence*, vol. 10, no. 1-2, pp. 45–75, 2017.

[21] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *8th International Conference on Computer Vision*, vol. 2, 2001, pp. 416–423.

[22] J. Derrac, S. García, D. Molina, and F. Herrera, "A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms," *Swarm and Evolutionary Computation*, vol. 1, no. 1, pp. 3–18, 2011.