Modified seeker optimization algorithm for image segmentation by multilevel thresholding

Milan Tuba, and Ivona Brajevic

Abstract— Image segmentation is among most often used techniques for image analysis and one standard way to do it is multilevel tresholding. The selection of optimum thresholds has remained a challenge in image segmentation. High computational cost and inefficiency of the conventional multilevel thresholding methods lead to the use of global search heuristics to find the optimal thresholds. The optimal thresholds are often determined by either minimizing or maximizing some criterion functions defined on images. In this paper, a new swarm intelligence algorithm, modified seeker optimization (MSO) algorithm, is presented for image segmentation by multilevel thresholding. This algorithm is used to maximize the Kapur's and Tsallis' objective functions. Experiments have been performed on four test images using various numbers of thresholds. The experimental results demonstrate that the proposed MSO algorithm can find multiple thresholds that are very close to the optimal ones determined by the exhaustive search method. Compared to the particle swarm optimization (PSO) algorithm, the MSO algorithm performs satisfactory in terms of solution quality, robustness and convergence.

Keywords—Maximum entropy thresholding, Image thresholding, Multilevel tresholding, Seeker optimization algorithm, Swarm intelligence

I. Introduction

Image segmentation is one of the most important operations in image analysis and computer vision [1], [2], [3], [4], [5]. Thresholding is one of the simplest techniques for performing image segmentation and it is very useful in separating objects from background image, or discriminating objects from objects that have distinct gray-levels. Thresholding involves bi-level thresholding and multilevel thresholding. For bi-level thresholding the main objective is to determine one threshold which separates the pixels into two groups, one including those pixels with gray levels above certain threshold, the other including the rest. For multilevel thresholding the main objective is to determine multiple thresholds which divide pixels into several groups. The pixels which belong to the same class have gray levels within a specific range defined by several thresholds. The global thresholding methods [6], belonging to parametric and nonparametric approaches, select thresholds by optimizing (maximizing or minimizing) some criterion functions defined from images.

The maximum entropy thresholding has been widely used in determining the optimal thresholding in image segmentation [7]. Of particular interest is an information theoretic approach that is based on the concept of entropy introduced by Shannon. The principle of entropy is to use uncertainty as a measure to describe the information contained in a source. The maximum entropy criterion for image thresholding was first proposed by Pun, and later it was corrected and improved by Kapur [6]. Basically, the entropy thresholding considers an image histogram as a probability distribution, and then selects as an optimal threshold value that yields the maximum entropy. More precisely, a best entropy thresholded image is the one that preserves as much information as possible that is contained in the original unthresholded image in terms of Shannon's entropy. Recent developments of statistical mechanics based on a concept of nonextensive entropy, also called Tsallis entropy, have intensified the interest of investigating a possible extension of Shannon's entropy to information theory [8], [9]. This interest appears mainly due to similarities between Shannon's and Boltzmann/Gibbs entropy functions. The Tsallis entropy is a new proposal in order to generalize the Boltzmann/Gibbs traditional entropy to nonextensive physical systems.

Optimization techniques inspired by swarm intelligence have become very popular during the last decade. Three of the most successful examples of optimization techniques inspired by swarm intelligence are ant colony optimization [10], particle swarm optimization [11] and artificial bee colony optimization [12], [13]. Basic versions of these algorithms are later enhanced to improve performance in general or for some class of the problems [13], [15], [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26]. Also, swarm intelligence algorithms are successfully used for wide range of practical problems. Many of these metaheuristics were adopted to search for the multilevel thresholds [27], [28], [29]. Yin [30] presented a new method that adopts the particle swarm optimization to select the thresholds based on the minimum cross-entropy. Horng applied the honey bee mating optimization (HBMO), the artificial bee colony (ABC) algorithm [31] and the firefly algorithm [32] to search for the thresholds using the maximum entropy criterion. In [33] the adaptation and comparison of six meta-heuristic algorithms: genetic algorithm, particle swarm optimization, differential evolution, ant colony, simulated annealing and tabu search were presented. The experimental results have shown that the

Manuscript received October 12, 2012. Revised April 29, 2013.

The work was supported by the Ministry of Science, Republic of Serbia, Grant No. III-44006

M. Tuba is with the Faculty of Computer Science, Megatrend University, Belgrade, Serbia (e-mail: tuba@ieee.org)

I. Brajevic is with the Faculty of Mathematics, University of Belgrade, Serbia (e-mail: ivona.brajevic@googlemail.com)

genetic algorithm, the particle swarm optimization and the differential evolution outperformed the other algorithms.

Seeker optimization algorithm (SOA) is a novel swarm intelligence algorithm based on simulating the act of human searching, which has been shown to be a promising candidate of search algorithms for unconstrained function optimization [34]. The SOA results for multimodal test functions were not very satisfactory and in order to enhance its performance, the modified seeker optimization algorithm named MSO was proposed [35]. This paper proposes the Kapur and Tsallis based MSO algorithm for solving multilevel thresholding problem. The PSO algorithm is implemented for purposes of comparison. Also, the exhaustive search method is conducted for deriving the optimal solutions for comparison with the results generated from PSO and MSO algorithms.

The rest of the paper is organized as follows. In the Section 2 the problem of the multilevel thresholding is formulated and the objective functions treated are presented. In the Section 3 the SOA algorithm is briefly described. Section 4 presents our proposed MSO algorithm for multilevel thresholding problem. Section 5 gives comparative results of the implemented MSO and PSO algorithms.

II. Multilevel thresholding problem formulation

Let there be L gray levels in a given image I having M pixels and these gray levels are in the range $\{0,1,..L-1\}$. The multilevel thresholding problem can be configured as a k-dimensional optimization problem, for determination of k optimal thresholds $[t_1, t_2, \ldots, t_k]$ which optimizes an objective function.

Several objective functions devoted to the thresholding have been proposed in the literature [6]. Generally, these functions are determined from the histogram of the image, denoted by the SOA algorithm $h(i)$, $i = 0,1..L-1$, where $h(i)$ represents the number of pixels having the gray level i. The normalized probability at level i is defined by the ratio $P_i = h(i)/M$. One of the most popular objective function is defined by Kapur. The aim is to maximize the objective function:

$$f([t_1, t_2, \ldots, t_k]) = H_0 + H_1 + \ldots + H_k \quad (1)$$

where

$$H_0 = -\sum_{i=0}^{t_1-1} \frac{P_i}{w_0} \ln \frac{P_i}{w_0}, \; w_0 = \sum_{i=0}^{t_1-1} P_i,$$

$$H_1 = -\sum_{i=t_1}^{t_2-1} \frac{P_i}{w_1} \ln \frac{P_i}{w_1}, \; w_1 = \sum_{i=t_1}^{t_2-1} P_i,$$

$$H_2 = -\sum_{i=0}^{t_3-1} \frac{P_i}{w_2} \ln \frac{P_i}{w_2}, \; w_2 = \sum_{i=t_2}^{t_3-1} P_i, \ldots$$

$$H_k = -\sum_{i=t_k}^{L-1} \frac{P_i}{w_k} \ln \frac{P_i}{w_k}, \; w_K = \sum_{i=t_k}^{L-1} P_i$$

The method similar to the maximum entropy sum method of Kapur is Tsallis non-extensive entropy concept. The aim is to maximize the objective function:

$$f([t_1, t_2, \ldots, t_k]) = S_q^{A_0} + S_q^{A_1} + \ldots + S_q^{A_k} + (1-q\} \cdot S_q^{A_0} \cdot S_q^{A_1} \cdot \ldots \cdot S_q^{A_k} \quad (2)$$

where

$$S_q^{A_0} = \frac{1 - \sum_{i=0}^{t_1-1} (\frac{p_i}{p^{A_0}})^q}{q-1}, \; p^{A_0} = \sum_{i=0}^{t_1-1} p_i$$

$$S_q^{A_1} = \frac{1 - \sum_{i=t_1}^{t_2-1} (\frac{p_i}{p^{A_1}})^q}{q-1}, \; p^{A_1} = \sum_{i=t_1}^{t_2-1} p_i$$

$$S_q^{A_2} = \frac{1 - \sum_{i=t_2}^{t_3-1} (\frac{p_i}{p^{A_2}})^q}{q-1}, \; p^{A_2} = \sum_{i=t_2}^{t_3-1} p_i, \ldots$$

$$S_q^{A_k} = \frac{1 - \sum_{i=t_k}^{L-1} (\frac{p_i}{p^{A_k}})^q}{q-1}, \; p^{A_k} = \sum_{i=t_k}^{L-1} p_i$$

III. Seeker optimization algorithm

Seeker optimization algorithm (SOA) mimics the behaviour of human search population based on their memory, experience, uncertainty reasoning and communication with each other. SOA is a population-based heuristic algorithm. The algorithm operates on a set of solutions called search population. Each individual of the population is called a seeker or agent. The total population is equally categorized into K subpopulations according to the indexes of the seekers (all the subpopulations have the same size, shown as Fig. 1). All the seekers in the same subpopulation constitute a neighborhood which represents the social component for the social sharing of information.

The main characteristic features of this algorithm are the following:

1. The algorithm uses search direction and step length to update the positions of seekers

2. The calculation of the search direction is based on a compromise among egotistic behaviour, altruistic behaviour and pro-activeness behaviour

3. Fuzzy reasoning is used to generate the step length because the uncertain reasoning of human searching could be the best described by natural linguistic variables and a simple if-else control rule: If {objective function value is small} (i.e., condition part), then {step length is short} (i.e., action part)

A search direction $d_{ij}(t)$ and a step length $\alpha_{ij}(t)$ are separately computed for each individual i on each dimension j at each iteration *iter*, where $\alpha_{ij}(\text{iter}) \geq 0$ and $d_{ij}(\text{iter}) \in \{-1,0,1\}$. At each iteration the position of each seeker is updated by:

$$x_{ij}(\text{iter}+1) = x_{ij}(\text{iter}) + \alpha_{ij}(\text{iter}) \cdot d_{ij}(\text{iter}) \quad (3)$$

where $i = 1,2,..SN$; $j = 1,2,..D$ (*SN* is the number of seekers). Also, at each iteration, the current positions of the worst two individuals of each subpopulation are exchanged for both of the best one in each of the other two subpopulations, which is called inter-subpopulation learning.

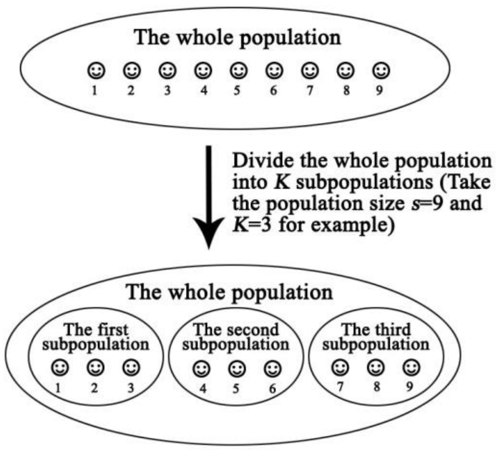

Fig.1: Relationship chart of population and subpopulation

Short pseudo–code of the SOA algorithm is given below:

1. Generating *s* positions uniformly and randomly in search space;
2. cycle = 0;
3. **Repeat**
4. For i = 1 to s do
   Computing $d_i(\text{iter})$;
   Computing $\alpha_i(\text{iter})$;
   Updating each seeker's position using Eq. (3);
5. End of For
6. Evaluating all the seekers and saving the historical best position;
7. Implementing the inter-subpopulation learning operation;
8. cycle = cycle+1;

Until the termination criterion is satisfied

Since its invention, SOA has been applied to solve the other kinds of problems beside numerical function optimization. In [36], the application of the SOA to tuning the structures and parameters of artificial neural networks is presented, while in [37] SOA-based evolutionary method is proposed for digital IIR filter design. Also, a new optimized model of proton exchange membrane fuel cell (PEMFC) was proposed by using SOA [38].

IV. Modified Seeker Optimization Algorithm for Multilevel Thresholding Problem

SOA was analyzed with a challenging set of benchmark problems for function optimization. The simulation results showed that the proposed algorithm is a promising candidate of swarm algorithms for numerical function optimization. For multimodal test functions the results were not very satisfactory because it was noticed that for this type of problems SOA may be stuck at a local optimum. In order to enhance the performance of SOA, the modified seeker optimization (MSO) algorithm is developed [35]. MSO algorithm uses two search equations for producing new population: search equation of artificial bee colony (ABC) algorithm [18] and the search equation of seeker optimization algorithm. Also, MSO algorithm implements the modified inter-subpopulation learning using the binomial crossover operator.

The proposed MSO algorithm based on maximum entropy criterion tries to obtain this optimum *K*-dimensional vector $[t_1, t_2, ..., t_k]$ which can maximize Eq.(1) in the case of Kapur's method Eq.(2) in the case of Tsallis' method. The details of the developed approach are introduced as follows.

Step 1. Initialize population

MSO algorithm generates a randomly distributed initial population of *SN* solutions or seekers t_i *(i = 1, 2, ..., SN)* with *K* dimensions denoted by matrix *T*.

$$T = [t_1, t_2, ..t_{SN}] \text{ and } t_i = (t_{i,1}, t_{i,2}, ..t_{i,K}) \quad (4)$$

where t_{ij} is the j^{th} component value that is restricted into [0,…,L-1] and the $t_{ij} < t_{ij+1}$ for all *j*. Each seeker t_i *(i = 1, 2, ..., SN)* is generated by:

$$t_{i,j} = t_{\min} + rand(0,1) \cdot (t_{\max} - t_{\min}) \quad (5)$$

where $t_{\min}$ and $t_{\max}$ are the minimum and the maximum gray values in the image, the rand(0, 1) is a random number between 0 and 1. In MSO algorithm, as in SOA, the total population is categorized into *N* subpopulations according to the indexes of the seekers. Each seeker t_i, beside of its current position and its objective function value, has the following attributes: the personal best position p_{ibest} so far and the neighborhood best position g_{best} so far.

Step 2. Evaluate population

For each seeker t_i *(i = 1, 2 ..., SN)* evaluate the objective function values by Eq.(1) or Eq.(2).

Step 3. Record the best solution

In this step, the best solution vector, i.e. the solution vector with the highest objective function value is recorded.

Step 4. is repeated a fixed number of iterations. It consists of three parts. The details of each part are described as follows.

Part 1. Calculate new population

Perform an update process for each solution in the search population using a randomly selected search equation. The MSO included a new control parameter which is called behaviour rate (*BR*) in order to select the search equation in the following way: If a uniformly distributed real random number between [0,1) is less then *BR*, the SOA search equation is used, otherwise the search equation of ABC algorithm is performed.

The variant of ABC search equation for producing a new solution v_i, $i \in \{1, 2, ..., SN\}$ which is used in MSO algorithm is:

$$v_{i,j} = \begin{cases} t_{i,j} + \varphi_i (t_{i,j} - t_{k,j}), if \ R_j \le 0.5 \\ t_{i,j} \quad , \quad otherwise \end{cases} \quad (6)$$

where k is a randomly chosen index of a solution from the subpopulation to which the i^{th} seeker belongs, *k* has to be different from *i*, $j = 1,2,...,K$, φ_i is a uniformly distributed real random number between [-1, 1) and R_j is a uniformly random real number within [0, 1).

The SOA search solution equation uses search direction $d_{i,j}$ and step length $\alpha_{i,j}$ for producing a new solution v_i, $i \in \{1, 2, ..., SN\}$. It can be described by:

$$v_{i,j} = t_{i,j} + \alpha_{i,j} \cdot d_{i,j} \ , \quad j = 1,2,..K \quad (7)$$

A search direction $d_{i,j}$ and a step length $\alpha_{i,j}$ are separately computed for each individual *i* on each dimension *j* at each iteration. The calculation of the search direction is based on a compromise among egotistic behavior, altruistic behavior and pro-activeness behavior. The egotistic behavior of each seeker t_i may be modeled by vector called egotistic direction d_{iego} by:

$$d_{iego,j} = p_{ibest,j} - t_{i,j} \ , \ j = 1,2,..K \quad (8)$$

The altruistic behavior of each seeker t_i may be modeled by vector called altruistic direction d_{ialt} by:

$$d_{ialt,j} = g_{best,j} - t_{i,j} \ , \ j = 1,2,..K \quad (9)$$

where g_{best} represents the neighbourhood best position so far. The pro-active behavior of each seeker t_i may be modeled by vector called pro-activeness direction d_{ipro} by:

$$d_{ipro,j} = t_{i,j}(iter_1) - t_{i,j}(iter_2) \ , \ j = 1,2,..K \quad (10)$$

where $iter_1, iter_2 \in \{iter, iter-1, iter-2\}$, $t_i(iter_1)$ and $t_i(iter_2)$ are the best and the worst positions in the set $\{t_i(iter-2), t_i(iter-1), t_i(iter)\}$ respectively. Here, *iter* denotes the current iteration, while $iter-1$ and $iter-2$ denote the previous two iterations.

The expression of search direction for the i^{th} seeker is set to the stochastic combination of egotistic direction, altruistic direction and pro-activeness direction by:

$$d_{ij} = sign(w \cdot d_{ipro,j} + \varphi_1 \cdot d_{iego,j} + \varphi_2 \cdot d_{ialt,j}) \quad (11)$$

where $j = 1,2,..K$, the function sign $(\cdot)$ is a signum function on each dimension of the input vector, w is the inertia weight and φ_1 and φ_2 are real numbers chosen uniformly and randomly in the range [0,1]. Inertia weight is used to gradually reduce the local search effect of pro-activeness direction d_{ipro} and provide a balance between global and local exploration and exploitation. Inertia weight is linearly decreased from 0.9 to 0.1 during a run.

Fuzzy reasoning is used to generate the step length because the uncertain reasoning of human searching. From the view point of human searching behavior, it may be assumed that better points are likely to be found in the neighborhood of families of good points. For calculating the step length of i^{th} seeker we need to calculate vector μ_i by:

$$\mu_i = \mu_{\max} - \frac{S - I_i}{S - 1} \cdot (\mu_{\max} - \mu_{\min}) \quad (12)$$

where *S* denotes the size of the subpopulation to which the seekers belong, I_i is the sequence number of t_i after sorting the objective function values in ascending order, $\mu_{\max}$ is the maximum membership degree value which is equal to or a little less than 1.0, $\mu_{\min}$ is set to 0.0111. Beside of vector μ_i, we need to calculate vector δ_i by :

$$\delta_i = w \cdot abs(t_{\max} - t_{\min}) \quad (13)$$

In Eq.(11), the absolute value of the input vector as the corresponding output vector is represented by the symbol abs $(\cdot)$, $t_{\max}$ and $t_{\min}$ are the positions of the best and the worst seeker in the subpopulation to which the i^{th} seeker belongs, respectively. In order to introduce the randomness in each variable and to improve the local search capability, the following equation is introduced to convert μ_i into a vector with elements as given by:

$$\mu_{ij} = rand(\mu_i, 1) \ , \ j = 1,2,..K \quad (14)$$

The equation used for generating the step length $\alpha_{i,j}$ for i^{th} seeker is :

$$\alpha_{i,j} = \delta_{i,j} \cdot \sqrt{-\ln(\mu_{i,j})} \ , \ j = 1,2,..K \quad (15)$$

For each seeker t_i *(i = 1, 2, ..., SN)* evaluate the objective function values.

Part 2. Evaluating all the seekers and saving the historical best position.

Part 3. Apply the modified inter-subpopulation learning operation

The modified inter-subpopulation learning is implemented as follows: The positions of seekers with the lowest objective function values of each subpopulation *l* are combined with the positions of seekers with the highest objective function values of *(l+z) mod N* subpopulations respectively, where *z=1,2,.. NSC*. *NSC* denotes the number of the worst seekers of each population which are combined with the best seekers. The appropriate seekers are combined using the following binomial crossover operator as expressed in:

$$t_{l_n j,worst} = \begin{cases} t_{i_j,best} & ,if \quad R_j \le 0.5 \\ t_{l_n j,worst} & , \quad otherwise \end{cases} \qquad (16)$$

where R_j is a uniformly random real number within [0, 1), $t_{l_n j,worst}$ is denoted as the j^{th} variable of the n^{th} worst position in the l^{th} subpopulation, $t_{i_j best}$ is the j^{th} variable of the best position in the *i*th subpopulation. Additionally, we included a new parameter which we named inter-subpopulation learning increase period (*ILIP*). After *ILIP* iterations the number of the worst seekers of each subpopulation which are combined with the best seekers is increased to $2 \cdot NSC$.

Step 5. Output best recorded solution

After a predefined number of iterations the positions of the best recorded solution are the optimal threshold values.

V. EXPERIMENTAL RESULTS AND DISCUSSION

The MSO and PSO algorithms have been implemented in Java programming language. Four well-known images, namely Lena, Peppers, Cameraman and Boats with 256 gray levels are taken as the test images. All the images are of size (512 x 512). These original images with their histograms are shown in Fig 2. Tests were done on a PC with Intel® Core™ i3-2310M processor @2,10 GHz with 2GB of RAM and Windows 7 x64 Professional operating system.

In all experiments for both algorithms the same size of population (*SP*) of 40 is used and the same size of maximum number of iterations (*MCN*) of 100 is taken. In proposed MSO algorithm the number of subpopulations (*N*) is 5, the behavior rate (*BR)* is 0.4, the number of seekers of each subpopulation for combination (*NSC*) is 1 and the inter-subpopulation learning increase period (*ILIP*) is $0.4 \cdot MCN$. Parameters of PSO algorithm are: inertia weight (*w*) is 0.5, minimum velocity ($v_{\min}$) is -5, maximum velocity ($v_{\max}$) is 5 , $\phi_{\min}$ is 0 and $\phi_{\max}$ is 2. Since MSO and PSO algorithms are of stochastic type and therefore the results of experiments are not absolutely the same in each run of algorithm, each experiment was repeated 50 times.

Table 1 shows the optimal thresholds, the optimal objective function values and the processing time provided by the exhaustive search method for Kapur’s and Tsallis' method. Table 2 and Table 3 present the mean values, standard deviations and average processing time over 50 runs provided by both algorithms for each image with a threshold numbers from 1 to 5 for Kapur's and Tsallis' method respectively.

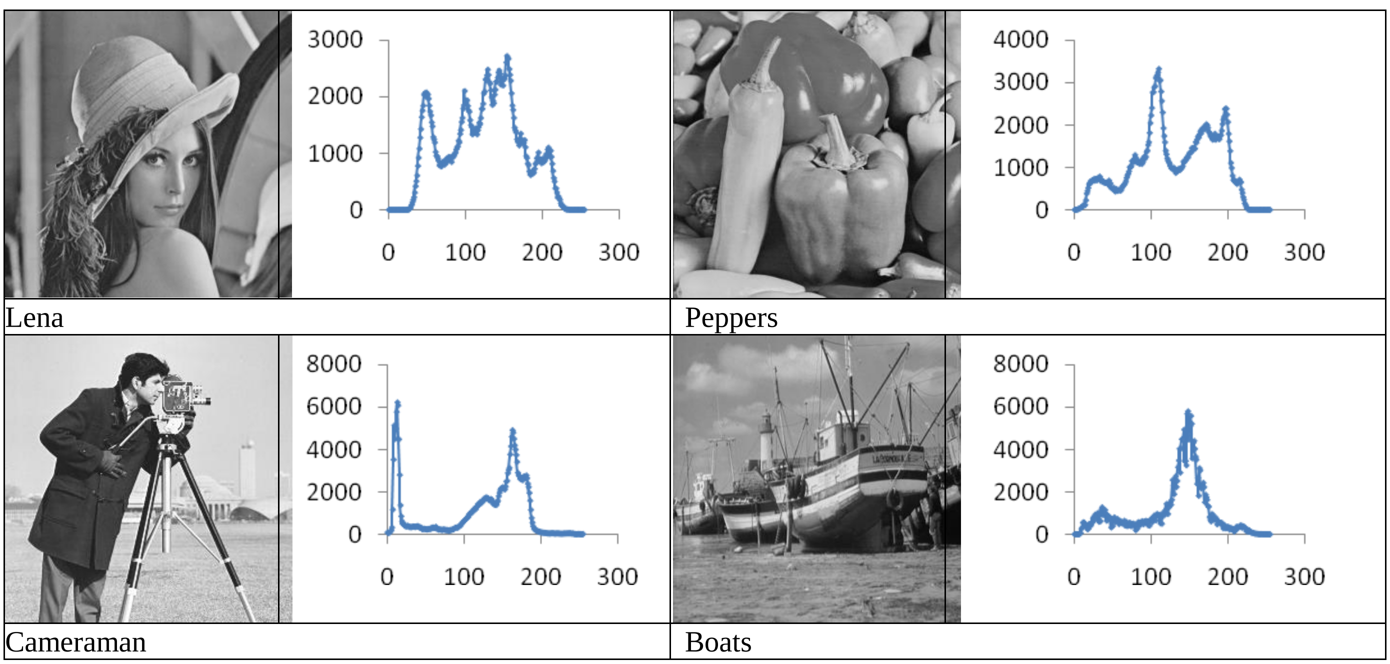

Fig 2: Test images and their histograms

TABLE I

THRESHOLDS, OBJECTIVE FUNCTION VALUES AND TIME PROCESSING PROVIDED BY THE EXHAUSTIVE SEARCH FOR KAPUR'S AND TSALLIS' METHOD

Image	k	Kapur			Tsallis		
		Threshold values	Objective function	Time (s)	Threshold values	Objective function	Time (s)
Lena	1	123	8.941944	0.015	164	0.333333	0.031
	2	97, 164	12.347015	0.827	104, 164	0. 888885	2.291
	3	82, 126, 175	15.318053	30.576	84, 126, 173	1.296279	203.582
	4	64, 97, 138, 179	18.012432	1873.643	NA	NA	NA
Pepper	1	97	9.118984	0.015	94	0.333333	0.035
	2	74, 149	12.557434	0.773	72, 153	0.888885	5.266
	3	69, 119, 167	15.621959	28.461	66, 120, 166	1.296281	223.095
	4	55, 94, 134, 177	18.400522	1881.104	NA	NA	NA
Cameraman	1	196	8.786829	0.022	201	0.333333	0.042
	2	125, 196	12.286490	0.847	145, 201	0.888877	6.523
	3	44, 102, 196	15.394271	27.311	124, 155, 203	1.296252	250.41
	4	42, 96, 145, 198	18.556655	1900.083	NA	NA	NA
Boats	1	115	8.964219	0.024	119	0.333333	0.045
	2	107, 176	12.574798	0.846	64, 119	0.888882	6.446
	3	64, 119, 176	15.820903	30.687	64, 119, 186	1.296281	246.327
	4	48, 88, 128, 181	18.655734	1856.990	NA	NA	NA

TABLE II

MEAN VALUES, STANDARD DEVIATIONS AND AVERAGE PROCESSING TIME OVER 50 RUNS FOR KAPUR'S METHOD

Image	k	PSO			MSO		
		Mean value	St. Dev.	Time(s)	Mean value	St. Dev.	Time(s)
Lena	1	8.941944	1.24E-14	0.0631	8.941944	1.24E-14	0.1781
	2	12.347015	5.33E-15	0.0635	12.347015	5.33E-15	0.1769
	3	15.318053	1.24E-14	0.0656	15.318053	1.24E-14	0.1829
	4	18.000658	1.02E-02	0.0618	18.008059	6.98E-03	0.1826
	5	20.610531	1.40E-04	0.0682	20.610500	1.98E-04	0.1856
Pepper	1	9.118984	7.11E-15	0.0551	9.118984	7.11E-15	0.1923
	2	12.557434	1.07E-14	0.0568	12.557434	1.07E-14	0.1934
	3	15.621959	1.42E-14	0.0668	15.621959	1.42E-14	0.1926
	4	18.392850	2.42E-02	0.0719	18.400501	1.24E-04	0.1920
	5	21.037588	2.17E-02	0.0686	21.052916	1.92E-03	0.2013
Cameraman	1	8.786829	8.88E-15	0.0606	8.786829	8.88E-15	0.1954
	2	12.286490	5.33E-15	0.0657	12.286490	5.33E-15	0.2078
	3	15.392280	1.39E-02	0.0661	15.394271	1.07E-14	0.2102
	4	18.556636	1.35E-04	0.0691	18.556655	2.13E-14	0.2132
	5	21.294869	3.68E-02	0.0656	21.310750	2.67E-02	0.2149
Boats	1	8.964219	5.33E-15	0.0642	8.964219	5.33E-15	0.2010
	2	12.574798	1.42E-14	0.0648	12.574798	1.42E-14	0.2058
	3	15.820754	7.29E-04	0.0679	15.820828	5.21E-04	0.2091
	4	18.624686	3.77E-02	0.0706	18.652739	1.38E-02	0.2110
	5	21.384345	5.88E-02	0.0752	21.400989	1.74E-03	0.2104

TABLE III

MEAN VALUES, STANDARD DEVIATIONS AND AVERAGE PROCESSING TIME OVER 50 RUNS FOR TSALLIS' METHOD

Image	k	PSO			MSO		
		Mean value	St. Dev.	Time(s)	Mean value	St. Dev.	Time(s)
Lena	1	0.333333	4.44E-16	0.3035	0.333333	3.02E-15	1.2168
	2	0.888885	3.33E-16	0.3101	0.888885	3.33E-16	1.2287
	3	1.296279	1.65E-08	0.3111	1.296279	1.37E-09	1.2253
	4	1.654273	1.28E-06	0.3163	1.654273	3.16E-08	1.2494
	5	1.995795	1.67E-06	0.3229	1.995796	8.32E-07	1.2944
Pepper	1	0.333333	5.55E-17	0.3405	0.333333	5.55E-17	1.3112
	2	0.888885	6.66E-16	0.3413	0.888885	6.66E-16	1.3120
	3	1.296281	1.33E-15	0.3432	1.296281	1.33E-15	1.3750
	4	1.654282	1.03E-06	0.3491	1.654283	7.97E-10	1.3802
	5	1.995809	7.31E-06	0.3531	1.995811	6.89E-08	1.3942
Cameraman	1	0.333333	5.55E-17	0.3593	0.333333	5.55E-17	1.3897
	2	0.888877	6.66E-16	0.3616	0.888877	6.66E-16	1.3095
	3	1.296252	1.55E-15	0.3646	1.296252	1.55E-15	1.3241
	4	1.654207	1.16E-05	0.3846	1.654210	1.11E-05	1.3386
	5	1.995700	1.02E-05	0.3890	1.995701	1.00E-05	1.3674
Boats	1	0.333333	2.22E-16	0.3563	0.333333	2.22E-16	1.3470
	2	0.888882	7.97E-08	0.3617	0.888882	6.35E-08	1.3610
	3	1.296280	1.76E-08	0.3647	1.296281	6.66E-16	1.3717
	4	1.654283	2.66E-06	0.3708	1.654283	5.94E-08	1.3738
	5	1.995800	2.72E-06	0.3709	1.995802	9.69E-07	1.3909

The mean values and standard deviations obtained by MSO and PSO algorithms can be compared to the optimal objective function values derived by the exhaustive search method. From Table 1 we found that the computation times of exhaustive search method is exponential. For Kapur function in the case $k=5$ and for Tsallis function in the case $k=4$ and $k=5$, the optimal thresholds and objective function values aren't counted because the time needed to find these values was unacceptable.

From Table 2 it can be seen that both algorithms give good results both in terms of accuracy (mean fitness) and robustness (similar results of repeated runs or small standard deviation), for the threshold numbers from 1 to 2. For each image, for the threshold numbers from 1 to 2, MSO and PSO algorithms converged consistently to the same solution which is equal to the optimal solution. In this case, the standard deviations provided by both algorithms are very low. In the case when the number of thresholds is higher or equal to 3, the MSO algorithm performs better than PSO algorithm for each image, except for the image Lena (k=5). We can see that for the threshold numbers from 3 to 5, the mean values of MSO are closer to the optimal ones than the same of PSO. Also, in that case, the standard deviations obtained by MSO are lower than the standard deviations obtained by PSO, which is specially noticeable for the image Cameraman. It can be concluded that MSO algorithm is superior to PSO in terms of precision and robustness of the results for the Kapur's method.

From Table 3 it is observed that for the threshold numbers from 1 to 5, the MSO algorithm perform well as compared with the PSO algorithm. The mean results show that MSO algorithm performs slightly better than PSO algorithm for each image. Also, MSO algorithm gives smaller standard deviations than the same of PSO. It can be concluded that MSO algorithm is more stable than PSO algorithm for the Tsallis' method.

The reported results from these tables show that as for the exhaustive search, for both algorithms, the number of iterations and the run time increase with the threshold number, but not in the same manner. The convergence times of the MSO and PSO are faster than those of the exhaustive search, except for $k=1$ for both methods. From Table 2 and Table 3, for the threshold numbers from 1 to 5, for each image, we can see that PSO is more efficient in terms of time execution than MSO. It is also observed that the computation time of Tsallis-based PSO is higher than the Kapur-based MSO.

VI. CONCLUSION

In this paper, the modified seeker optimization (MSO) algorithm based on simulating the act of human searching is proposed for multilevel thresholds selection. In order to verify the effectiveness of the proposed MSO approach, four standard test images are investigated. Particle swarm optimization (PSO) algorithm is also implemented for comparison. The experimental results show that MSO algorithm performs better than PSO algorithm with respect to precision and robustness, while in term of execution time the PSO is more efficient than MSO. Even though the Tsallis-based MSO gives lower standard deviation values, compared with all the cases, the Kapur-based MSO converges faster than the Tsallis-based MSO and the Tsallis-based PSO. Therefore, the proposed Kapur-based MSO method is a promising approach for image segmentation due to quality of its segmentation results and computational efficiency.

REFERENCES

[1] Batlle J.,Casals A.,Freixenet J., Martı J., A review on strategies for recognizing natural objects in colour images of outdoor scenes, Image and Vision Computing, Volume 18, Issues 6–7, 2000, pp. 515–530

[2] Chen Y.L., Nighttime Vehicle Light Detection on a Moving Vehicle using Image Segmentation and Analysis Techniques, WSEAS transactions on computers, Volume 8, Issue 3, 2009, pp. 506-515

[3] X. D. Zhuang, N. E. Mastorakis, Image Analysis Based on the Discrete Magnetic Field Generated by the Virtual Edge Current in Digital Images, WSEAS transactions on computers, Volume 10, Issue 8, August 2011, pp. 259-273

[4] I.A. Yusoff, N.A.M. Isa, Two-Dimensional Clustering Algorithms for Image Segmentation, WSEAS transactions on computers, Volume 10, Issue 10, 2011, pp. 332-342

[5] Jiang M., Mastorakis N., Yuan D., Lagunas M. A., Image Segmentation with Improved Artificial Fish Swarm Algorithm, LNEE Vol 28, Springer, 2009, pp. 133-138.

[6] M. Sezgin, B. Sankur, Survey over image thresholding techniques and quantitative performance evaluation, Journal of Electronic Imaging, Vol. 13, No. 1, 2004, pp. 146–165

[7] C.-I Chang, Y. Du, J. Wang, S.-M. Guo, P.D. Thouin, Survey and comparative analysis of entropy and relative entropy thresholding techniques, IEE Proceedings in Vision, Image and Signal Processing, Volume 153, Issue 6, 2006, pp. 837–850

[8] C. Tsallis. E. Brigatti , Nonextensive statistical mechanics: A brief introduction, Continuum Mechanics and Thermodynamics, Volume 16, Number 3, 2004, pp.223-235

[9] M. Portes de Albuquerque, I.A. Esquef, A.R. Gesualdi Mello, M. Portes de Albuquerque, Image thresholding using Tsallis entropy, Pattern Recognition Letters, Volume 25, Issue 9, 2004, pp. 1059-1065

[10] M. Dorigo, T. Stützle, Ant Colony Optimization, MIT Press, Cambridge, MA, 2004

[11] Kennedy, J., Eberhart, R.C., Particle swarm optimization, Proceedings of the 1995 IEEE International Conference on Neural Networks, Piscataway, NJ: IEEE Service Center, 1995, pp. 1942–1948

[12] D. Karaboga, An Idea Based on Honey Bee Swarm for Numerical Optimization, Technical Report-TR06, Erciyes University, Engineering Faculty, Computer Engineering Department, 2005

[13] D. Karaboga, B. Akay, A Modified Artificial Bee Colony (ABC) Algorithm for Constrained Optimization Problems, Applied Soft Computing, Volume 11, Issue 3, 2011, pp. 3021-3031

[14] Jovanovic R. and Tuba M.: An ant colony optimization algorithm with improved pheromone correction strategy for the minimum weight vertex cover problem, Applied Soft Computing, Vol. 11, Issue 8, Dec. 2011, pp. 5360–5366

[15] Tuba M., Bacanin N., Stanarevic N.: Adjusted artificial bee colony (ABC) algorithm for engineering problems, WSEAS Transaction on Computers, Volume 11, Issue 4, April 2012, pp. 111-120

[16] Tuba M., Artificial Bee Colony (ABC) Algorithm Exploitation and Exploration Balance, Proceedings of the 1st Internatonal Conference on Computing, Information Systems and Communications (CISCO '12), 2012, Singapore, pp.252-257

[17] Bacanin N., Tuba M.: Artificial Bee Colony (ABC) Algorithm for Constrained Optimization Improved with Genetic Operators, Studies in Informatics and Control, Vol 21, Issue 2, June 2012, pp. 137-146

[18] Brajevic I. and Tuba M.: An upgraded artificial bee colony algorithm (ABC) for constrained optimization problems, Journal of Intelligent Manufacturing, published Online First, Jan 2012, DOI: 10.1007/s10845-011-0621-6

[19] Tuba M., Jovanovic R.: An Analysis of Different Variations of Ant Colony Optimization to the Minimum Weight Vertex Cover Problem, WSEAS Transactions on Information Science and Applications, Volume 6, Issue 6, June 2009, pp. 936-945

[20] Tuba M., Subotic M., Stanarevic N.: Modified cuckoo search algorithm for unconstrained optimization problems, Proceedings of the European Computing Conference (ECC '11), pp. 263-268, Paris, France, April 2011

[21] Subotic M., Tuba M. and Stanarevic N.: Different approaches in parallelization of the artificial bee colony algorithm, International Journal of Mathematical Models and Methods in Applied Sciences, Vol. 5, Issue 4, 2011, pp. 755-762

[22] Tuba M., Subotic M., Stanarevic N. : Performance of a modified cuckoo search algorithm for unconstrained optimization problems, WSEAS Transactions on Systems, Volume 11, Issue 2, February 2012, pp. 62-74

[23] Jovanovic R. and Tuba M.: Ant Colony Optimization Algorithm with Pheromone Correction Strategy for Minimum Connected Dominating Set Problem, Computer Science and Information Systems (ComSIS), Vol 10, No. 1, 2013, pp.133-149

[24] Tuba M., Jovanovic, R.: Improved Ant Colony Optimization Algorithm with Pheromone Correction Strategy for the Traveling Salesman Problem, International Journal of Computers, Communications & Control, Vol. 8, Issue 3, 2013, pp. 409-417

[25] Tuba M., Brajevic I., Jovanovic R.: Hybrid Seeker Optimization Algorithm for Global Optimization, Applied Mathematics and Information Sciences, Vol. 7, No. 3, 2013, pp. 867-875

[26] Tuba M., Jovanovic R., Brajevic I.: Parallelization of the Cuckoo Search Using CUDA Architecture, Proceedings of the 19th American Conference on Applied Mathematics, Cambridge, MA, USA, 2013, pp. 137-142

[27] S. P. Duraisamy, R. Kayalvizhi, A new multilevel thresholding method using swarm intelligence algorithm for image segmentation, Journal of Intelligent Learning Systems and Applications, Volume 2, Number 3, 2010, pp.126-138

[28] Z. Ye, Z. Hu, X. Lai,H. Chen, Image Segmentation Using Thresholding and Swarm Intelligence, Journal of Software, Volume 7, Issue 5, 2012, pp. 1074-1082

[29] M.H. Horng, Multilevel Minimum Cross Entropy Threshold Selection based on Honey Bee Mating Optimization, Proc. of the 3rd WSEAS Int. Conf. on Circuits, Systems, Signal and Telecommunications (CISST'09), 2009, pp. 25-30.

[30] P. Y. Yin, Multilevel minimum cross entropy threshold selection based on particle swarm optimization, Applied Mathematics and Computation, Volume 184, Issue 2, 2007, pp. 503-513

[31] M.H. Horng, Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation, Expert Systems with Applications, Volume 38, Issue 11, 2011, pp. 13785–13791

[32] M.H. Horng, T.W. Jiang, Multilevel Image Thresholding Selection Based on the Firefly Algorithm, 2010 Symposia and Workshops on Ubiquitous, Autonomic and Trusted Computing, 2010, pp.58-63.

[33] K. Hammouche, M. Diaf, P. Siarry, A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem, Engineering Applications of Artificial Intelligence, Volume 23, Issue 5, 2010, pp. 676–688

[34] Dai C., Chen W., Song Y., Zhu Y., Seeker optimization algorithm: a novel stochastic search algorithm for global numerical optimization, Journal of Systems Engineering and Electronics, Vol. 21, No. 2, 2010, pp. 300-311

[35] Brajevic I., Tuba M., Modified Seeker Optimization Algorithm for Unconstrained Optimization Problems, Proceedings of the

American Conference on Applied Mathematics (AMERICAN-MATH '12), pp. 395-400, Cambridge, USA, January, 2012

[36] Dai C., Chen W., Zhu Y., Jiang Z. , You Z. , Seeker optimization algorithm for tuning the structure and parameters of neural networks, Neurocomputing, Volume 74, Issue 6 , 2011, pp. 876-883

[37] Dai C., Chen W., Zhu Y., Seeker optimization algorithm for digital IIR filter design, IEEE Transactions on Industrial Electronics, Volume 57, Issue 5, 2010, pp.1710-1718

[38] Dai C., Chen W., Cheng Z., Li Q., Jiang Z., Jia J.. Seeker optimization algorithm for global optimization: A case study on optimal modelling of proton exchange membrane fuel cell (PEMFC), International Journal of Electrical Power and Energy Systems, 2011, Volume 33, Issue 3, pp. 369-376

Milan Tuba received B.S. in mathematics, M.S. in mathematics, M.S. in computer Science, M.Ph. in computer science, Ph.D. in computer science from University of Belgrade and New York University.

From 1983 to 1994 he was in the U.S.A. first as a graduate student and teaching and research assistant at Vanderbilt University in Nashville and Courant Institute of Mathematical Sciences, New York University and later as an assistant professor of electrical engineering at Cooper Union Graduate School of Engineering, New York. During that time he was the founder and director of Microprocessor Lab and VLSI Lab, leader of scientific projects and supervisor of many theses. From 1994 he was associate professor of computer science and Director of Computer Center at University of Belgrade, Faculty of Mathematics, and from 2004 also a Professor of Computer Science and Dean of the College of Computer Science, Megatrend University Belgrade. He was teaching more than 20 graduate and undergraduate courses, from VLSI design and Computer architecture to Computer networks, Operating systems, Image processing, Calculus and Queuing theory. His research interest includes mathematical, queuing theory and heuristic optimizations applied to computer networks, image processing and combinatorial problems. He is the author of more than 100 scientific papers and a monograph. He is coeditor or member of the editorial board or scientific committee of number of scientific journals and conferences.

Prof. Tuba is member of the ACM since 1983, IEEE 1984, New York Academy of Sciences 1987, AMS 1995, SIAM 2009, IFNA. He participated in many WSEAS Conferences with plenary lectures and articles in Proceedings and Transactions.

Ivona Brajevic received B.S. in mathematics in 2006 and M.S. in mathematics in 2008 from University of Belgrade, Faculty of Mathematics.

She is currently Ph.D. student at Faculty of Mathematics, Computer science department, University of Belgrade and works as teaching assistant at College of Business, Economy and Entrepreneurship in Belgrade. She is the coauthor of six research papers. Her current research interest includes nature inspired metaheuristics, particularly swarm intelligence.

Ms. Brajevic participated in WSEAS conferences.