Multilevel Thresholding Segmentation of T2 weighted Brain MRI images using Convergent Heterogeneous Particle Swarm Optimization

Mohammad Hamed Mozaffari¹, Won-Sook Lee
School of Electrical Engineering and Computer Science, University of Ottawa, 800 King-Edward Avenue, Ottawa, Ontario, Canada, K1N 7S6. Tel: +1-613-562-5800 ext. 2191

Abstract

This paper proposes a new image thresholding segmentation approach using the heuristic method, Convergent Heterogeneous Particle Swarm Optimization algorithm. The proposed algorithm incorporates a new strategy of searching the problem space by dividing the swarm into subswarms. Each subswarm particles search for better solution separately lead to better exploitation while they cooperate with each other to find the best global position. The consequence of the aforementioned cooperation is better exploration, convergence and it able the algorithm to jump from local optimal solution to the better spots. A practical application of this method is demonstrated for the problem of medical image thresholding segmentation. We considered two classical thresholding techniques of Otsu and Kapur separately as the objective function for the optimization method and applied on a set of brain MR images. Comparative experimental results reveal that the proposed method outperforms another state of the art method from the literature in terms of accuracy, computation time and stable results.

Keywords: Convergent Heterogeneous Particle Swarm Optimization; Multilevel Image Thresholding Segmentation; Otsu’s and Kapur’s criteria; Magnetic Resonance image; Multiswarm search.

1 Introduction

Image processing techniques are used in any fields of science and technology such as astronomy, industry, and electronics. For medicine¹–⁶, image processing is a vital tool that helps the physicians to interpret images of the patients to diagnose diseases easier and assist the surgeons to perform the operation with more accuracy and effectiveness. Generally, the goal of these techniques is to extract the valuable information in the images that is usually a critical and challenging task. Image segmentation is an absolutely essential preparatory process in almost all image processing approaches in which the image is divided into classes, the background, foreground and region of interest (ROI)⁷–⁹. As an optimum segmentation, pixels that contribute in each class should be similar in terms of gray- or RGB- level intensities. Thresholding methods¹⁰,¹¹ are the easiest image segmentation approaches that in their simplest case (bi-level) endeavor to identify one optimum threshold value $t_b$ for the pixels intensity. Thus, the pixels whose intensity values are bigger than $t_b$ are labeled as the first class and the rest labeled as the second class, usually one class is for the background and another for the foreground region.

¹ Corresponding author
Email address: mmoza102@uottawa.ca, wslee@uottawa.ca
of the image\textsuperscript{12,13}. Multilevel thresholding is the extended version of the bi-level in which the image pixels are separated into more than two classes corresponding to threshold values.

According to a prevalent taxonomy, thresholding methods are divided into parametric and nonparametric methods. In the parametric approaches, the gray-level distribution of each class leads to find the thresholds. In order to identify the model of each class, the parameters of the probability density function must be estimated. In the non-parametric techniques, optimizing one criteria such as the entropy, the error rate, and between-class variance results optimal threshold values\textsuperscript{12–16}. Time-consuming and computationally expensive of the parametric methods cause that the non-parametric techniques become a desired and common alternative for the thresholding. For better results, criteria in these methods can also be considered as a fitness function for one optimization algorithm which is the main part of this study\textsuperscript{17}.

Amongst thresholding methods, Otsu’s\textsuperscript{18–22} and Kapur’s\textsuperscript{17,23,24} approach are two popular one because of their powerfulness, accuracy, and efficiency especially in the case of bi-level thresholding problem\textsuperscript{25}. Although these classical methods perform well in bi-level thresholding problem but for multilevel thresholding problem, because of their exhaustive searching strategy, and greater dimension of the problem space, the computation time grows exponentially with increasing the number of thresholds\textsuperscript{26}. Thus, in real-time multilevel thresholding applications, the classical methods are weak and necessity of new techniques is clear. Many studies try to find an alternative way to solve this issue and a common way is to consider multi-level thresholding as an optimization problem.

Heuristic and Meta-heuristic optimization algorithms are known and distinguished for their fast and efficient performance. A combination of heuristic methods and classical thresholding methods not only can solve the problem of computation time, it also can increase the efficiency, accuracy, and robustness of the results\textsuperscript{27}. Numerous heuristic techniques successfully have been used for the problem of multilevel thresholding segmentation. Heuristic optimization algorithms are inspired by the laws of physics, mechanics and mimic the developmental process in natural and biological phenomena such as Darwinian evolution in creations (Genetic Algorithm)\textsuperscript{15,28–30}, crystal and molecular structure shaping in metallurgy (Simulated Annealing)\textsuperscript{31}, foraging of ants (Ant Colony Optimization)\textsuperscript{32}, social behaviours of flock of birds or school of fishes in migration (Particle Swarm Optimization)\textsuperscript{33–45}, accumulation of water in catchment basins (Inclined Planes system Optimization)\textsuperscript{46,47} and numerous more examples.

In order to use aforementioned techniques for thresholding segmentation problem, the common technique is to use the extended version of one classical thresholding method and consider it as a fitness (objective or criteria) function, then heuristic optimization method can find the best threshold values set iteratively by maximizing or minimizing that fitness function\textsuperscript{8,28,48–50}. In spite of the fact that heuristic methods are powerful enough in solving high dimensional problems but developing of these methods (better convergence, less sensitive to the parameters, more robust, more accurate and etc.) is still a hot topic for researchers.

In this paper, application of a modified version of the Particle Swarm Optimization (PSO) algorithm with a new strategy of searching, named Convergence Heterogeneous Particle Swarm Optimization (CHPSO)\textsuperscript{51} on the problem of multi-level thresholding segmentation of Brain T-2 Magnetic Resonance (MR) images is investigated. Otsu’s and Kapur’s method is considered as the objective functions and 10 test images are used for the experiments. The experimental results reveal that the proposed method can reach to the best solution within fewer iteration numbers in comparison to the similar approaches without sticking into local solutions.

The rest of the paper is organized as follows: In Section 2, the CHPSO algorithms are introduced with details. In Section 3, the Otsu's and Kapur's methods explained in details and Section 4 shows
experimental results and discussion about similar methods as a comparison study. Section 5 concludes the paper.

2 Methodology

Heuristic optimization algorithms are designed for solving sophisticated optimization problems (non-linear in most cases) without having knowledge about the problem. Using a reasonable number of constant parameters, computation time and cost lead to an accurate and convergent optimal solution. In almost all heuristic optimization algorithms, there is a trade-off between two methodologies of exploration and exploitation which help the algorithm to search for the best solution efficiently and liberate from capturing in local optima as much as possible. Exploration reinforces algorithm to search the whole range of the problem space by expanding agents (particles) as far as possible in the search space while the exploitation enables the algorithm to find more accurate solutions with better searching locally.

One of the most popular heuristic population-based optimization method is the Particle Swarm Optimization (PSO) which is inspired by the behavior of a flock of birds or a school of fishes during the migration and foraging which their social network helps them to find the best place of food and accommodation. The first version of this algorithm was very powerful and easy to implement, but since 1995 that the PSO algorithm was introduced, many improvements have been suggested to update and enhance this method. Convergence Heterogeneous Particle Swarm Optimization (CHPSO) Algorithm is an enhanced version of the PSO which guarantees better optimal solution due to its new strategy of searching and better exploration and exploitation. In the following sections, the original version of the PSO and modified one CHPSO is explained clearly.

2.1 Particle Swarm Optimization

In the PSO algorithm, cooperation between particles causes to find the best solution for the search space and each of them can be a candidate for that position. Particles are random selected in the initialization step, then each particle compares its current position with two relative information to update its position and move to a new position, iteratively. The criterion for comparison is the fitness function and the information is the personal best position from the previous iterations and the best position between particles acquired so far.

The position and velocity of particles are calculated respect to this two information in each iteration. Thus, if the position and velocity of \( i \)-th particle in the time \( t \) and \( D \) dimensions are represented by \( X_i = [x_i^1, x_i^2, \ldots, x_i^D] \), and \( V_i = [v_i^1, v_i^2, \ldots, v_i^D] \), respectively, then the position of the particle in the next iteration \((t+1)\) is updated according to the following equations.

\[
\begin{align*}
X_i(t+1) &= X_i(t) + V_i(t+1) \\
V_i(t+1) &= \omega V_i(t) + c_1 R (P_{best_i}(t) - X_i(t)) + c_2 R (G_{best}(t) - X_i(t))
\end{align*}
\]

where \( c_1 \) and \( c_2 \) are two constants for controlling the exploration and exploitation affects, respectively, \( R \) is uniformly distributed random vector from zero to one, \( \omega \) is an inertia weight constant which scales the effect of the velocity in the previous iteration, \( P_{best_i}(t) \) and \( G_{best}(t) \) are the best personal position and the best position which have been acquired so far (from the time zero to \( t \)). Note that the equations (1) and (2) are for the global version of the PSO and there is also the local version which is outside the scope of this report.
In a nutshell, for each particle, the corresponding score using the fitness function is calculated then the best personal particle position respect to the previous iteration is stored for using in the next iteration and also the best global particle position respect to the neighbor’s particles is selected. This is one of the advantages of the PSO that it needs just one stack of memory and there is no need to calculate complicated formula such as gradient which caused the PSO become a fast and easy to implement the algorithm. The pseudocode of the PSO is tabulated in Algorithm 1.

**Algorithm 1 The PSO algorithm pseudocode**

1. **Initialization.**
   a. Set the constant parameters
   b. Randomize the vector of positions $X$ and the velocities $V$ of the particles in the search space
   c. Set the best personal position to current positions $P_{best} = X$
   d. Set $G_{best} = \arg\{\min f(X)\}$
2. **Termination Check.**
   a. If the termination criterion is satisfied stop.
      The $G_{best}$ will be the output solution.
   b. Else go to Step 3.
3. For $i = \ldots, \text{number of iteration}$ Do
   a. Update the velocity and position according to Equation (2) and (1) and check to be in the range space
   b. Evaluate fitness function of each particle.
   c. Save the best personal and global fitness value and position if better than previous
End for
   Update the $G_{best} = \arg\{\min f(P_{best})\}$
4. Goto step 2

### 2.2 Convergent Heterogeneous Particle Swarm Optimization

The ideal optimization algorithm with a perfect convergence and accurate solution (not local solutions) in a reasonable computational time and cost is always desirable and attracts many researchers from all fields of science to have a contribution in this area, at least, make an update to enhance previous optimization methods for a particular application. Depends on the search strategy of the optimization algorithm, trapping in local optimum solutions is an inevitable event and less probability of local trapping better results. For the particles of a fast convergent algorithm such as the original PSO, due to its homogeneous searching behavior (diversity of particles is lost) and their weak exploration ability, it is more likely for the algorithm to capture in local optimal spots within the first iterations.

One way to solve this problem is to use a big coefficient for the exploration part of the algorithm at the expense of losing of exploitation effect and losing of accuracy. A new approach to solving this issue is to use multiswarm techniques which are designed to improve the exploration ability of the original heuristic optimization algorithms as well as the exploitation. Cheung et al. in 2014 proposed the CHPSO which uses the multiswarm technique to enhance the weakness of the PSO in jumping from local solutions to the global one.

The main idea of the CHPSO is to divide particles into the four subswarms that each subswarm can search separately the problem space. Each subswarm moves heterogeneously in its local region but also share their information to the other two subswarms to explore new regions and specify the trajectory of the whole swarm in the solution domain. Totally, two subswarms are named basic subswarm which is used for exploitation search and the other two subswarms are called the adaptive subswarm and the exploration subswarm. The adaptive subswarm updates flight path adaptively with using the knowledge that was sent from the two basic subswarms in advanced. Unknown regions in the problem space are discovered by the exploration subswarm which uses...
the information of the other three subswarms. Because, in CHPSO cooperation between particles is different from the original PSO, new update rules based on positions, velocities and the value of fitness function are defined for CHPSO and are added to the original PSO.

As mentioned before, although each subswarm searches the problem space individually and shares its information with other subswarms (except exploration subswarm), it also contributes in the global exploitation which is the goal of the optimization algorithm. Consequently, all four subswarms search the problem space for the optimum solution heterogeneously while two of them (basic subswarms) supply their velocity and fitness values information for the adaptive subswarm for determining a better trajectory for all particles and finally the velocity information of these three subswarms are shared with the exploration subswarm to enable the algorithm to jump from local optima in order to find the areas of the problem space which never checked.

For better understanding the particles situations, Figure 1 represents a schematic of four subswarms contain five particles that try to find the global best position of the two-dimensional function \( f(x_1, x_2) = x_1^2 + x_2^2 \). This cooperation between the particles is used for updating their positions in the next generation. As illustrated in Figure 1, for minimization problem like this, the global position \( G_{\text{best}} \) is calculated according to the global position in the four subswarms and defined as equation (3).

\[
G_{\text{best}} = \arg \min \left\{ f(P_{\text{best}_{\text{sub(s)}}}) \mid s = 1,2,3,4 \right\}
\] (3)

where, \( P_{\text{best}_{\text{sub(s)}}} \) is the best particle in subswarms. For updating the position of particles in the adaptive subswarm (\( \text{sub(3)} \)), the velocity and fitness function of the basic subswarms are needed and for updating the exploration subswarm (\( \text{sub(4)} \)) velocity of particles in the basic and adaptive subswarms are needed. Position updating rules for basic subswarms (\( \text{sub(1)} \) and \( \text{sub(2)} \)) and adaptive subswarm (\( \text{sub(4)} \)) are identical to the original PSO i.e. equations (1). The velocity of the basic subswarms is updated using equation (4).
\[ V_{(i, (\text{sub}(1|&2)))(t+1)} = \omega V_{(i, (\text{sub}(1|&2)))(t)} + c_1 R(P_{i, (\text{sub}(1|&2))}(t) - X_{i, (\text{sub}(1|&2))}(t)) + c_2 R(\text{Gbest}_i - X_{i, (\text{sub}(1|&2))}(t)) \]

where \(c_1\) and \(c_2\) are two constants for controlling the exploration and exploitation effects, respectively, \(R\) is uniformly distributed random vector from zero to one, \(\omega\) is an inertia weight constant which scales the effect of the velocity in the previous iteration, \(P_{i, (\text{sub}(1|&2))}(t)\) and \(\text{Gbest}_i\) are the best personal position in the subswarm and the best position which have been gained so far (from the time zero to \(t\)).

Using the information are acquired from basic subswarms \(\text{sub}(1)\) and \(\text{sub}(2)\), particles of the adaptive subswarm \(\text{sub}(3)\) can adjust the direction of the whole flight. Velocity of the \(i-th\) particle in the adaptive subswarm is updated using the following equation:

\[
\begin{align*}
V_{i, (\text{sub}(3))}(t+1) &= \omega \left( \frac{\gamma_1}{\gamma_2} V_{i, (\text{sub}(1))}(t+1) + \frac{\gamma_1}{\gamma_2} V_{i, (\text{sub}(2))}(t+1) + V_{i, (\text{sub}(3))}(t) \right) + \\
&\quad + c_1 R(P_{i, (\text{sub}(3))}(t) - X_{i, (\text{sub}(3))}(t)) + c_2 R(\text{Gbest}_i - X_{i, (\text{sub}(3))}(t)) \quad \gamma = \gamma_1 + \gamma_2
\end{align*}
\]

where \(\gamma_1\) and \(\gamma_2\) are the fitness values of the particles in the basic subswarms \(\text{sub}(1)\) and \(\text{sub}(2)\), respectively and other parameters are similar to equation (2). The strategy of using \(\gamma_1\) and \(\gamma_2\) can be explained in this way that the subswarm with better fitness values has a larger effect on the particles in the adaptive subswarm. As discussed, particles of the subswarm \(\text{sub}(4)\) should attempt to explore new areas as far as possible from other subswarms, thus, the velocity of the \(i-th\) particle in the subswarm \(\text{sub}(4)\) is defined by the difference between the basic subswarms and adaptive one as the equation (6). The \(i-th\) particle position in subswarm \(\text{sub}(4)\) is updated according to its previous information and velocity of the \(i-th\) particle in the other subswarm as bellow:

\[
\begin{align*}
V_{i, (\text{sub}(4))}(t+1) &= V_{i, (\text{sub}(1))}(t+1) + V_{i, (\text{sub}(2))}(t+1) - V_{i, (\text{sub}(3))}(t+1) \\
X_{i, (\text{sub}(4))}(t+1) &= \alpha_1 X_{i, (\text{sub}(4))}(t) + \alpha_2 P_{i, (\text{sub}(4))} + \alpha_3 \text{Gbest}_i + V_{i, (\text{sub}(4))}(t+1)
\end{align*}
\]

where \(\alpha_1\), \(\alpha_2\) and \(\alpha_3\) are named impact factors which indicate how much the previous information of the particle can contribute in updating process and they must satisfy this equation: \(\alpha_1 + \alpha_2 + \alpha_3 = 1\). The bigger impact factor \(\alpha_1\) causes the previous information about the \(i-th\) particle effect more on updating process than information shared from other subswarms. Generally, diversity and fitness value of the particles in exploration subswarm is considerable because the particles use all the information in the whole swarm. In this paper, the impact factors \(\alpha_1\), \(\alpha_2\) and \(\alpha_3\) are set to \(\sqrt[6]{1}\), \(\sqrt[3]{1}\) and \(\sqrt{1}\), respectively. For better Understanding of the CHPSO algorithm, its pseudocode is illustrated in Algorithm 1.

**Algorithm 2. The CHPSO algorithm pseudocode**

1. **Initialization.**
   a. **Set** the constant parameters
   b. **For** \(i = 1, ..., 4\) do
      - **Randomize** the positions of all particles in subswarm \(i\) in the search space
      - **Randomize** the velocities of all particles in subswarm \(i\) in the search space
• Set the $G_{best_{sub(i)}} = \arg\{\min f(x_{sub(i)})\}$

End For

c. Set the swarm global $G_{best} = \arg\{\min G_{best_{sub(1,2,3,4)}}\}$

2. Termination Check.

a. If the termination criterion is satisfied stop.
   The $G_{best}$ will be the output solution.

b. Else go to Step 3.

3. For $i=1,...,number\ of\ iteration$ Do

   For $j=1,...,number\ of\ particles$ Do

   a. Update the velocity and position of subswarms 1 and 2 according to Equation (4) and (1) and check to be in the range space

   b. Calculate $\gamma_1$ and $\gamma_2$

   c. Update the velocity and position of subswarms 3 according to Equation (5) and (1) and check to be in the range space

   d. Update the velocity and position of subswarms 4 according to Equation (6) and (7) and check to be in the range space

   e. Evaluate fitness function of each subswarm and particles.

   f. Save the best subswarm personal and global fitness value and position if better than previous

End for

Update the $G_{best} = \arg\{\min G_{best_{sub(1,2,3,4)}}\}$

End for

4. Goto step 2

In the next chapter, using of the CHPSO for optimizing the Otsu and Kapur criteria is explained.

3 Otsu and Kapur Criteria

The aim of image thresholding techniques is to find the optimal threshold values and can be defined and formulated as equations (8) and (9).

Bi-level: $$\begin{cases} p_{(i,j)} \in S_1 & \text{if } 0 \leq p_{(i,j)} < t_b \\ p_{(i,j)} \in S_2 & \text{if } t_b \leq p_{(i,j)} < L-1 \end{cases}$$

Multi-level: $$\begin{cases} p_{(i,j)} \in S_1 & \text{if } 0 \leq p_{(i,j)} < t_1 \\ p_{(i,j)} \in S_2 & \text{if } t_1 \leq p_{(i,j)} < t_2 \\ \vdots \\ p_{(i,j)} \in S_m & \text{if } t_m \leq p_{(i,j)} < L-1 \end{cases}$$

where $p_{(i,j)}$, $i=\{1,2,...,r\}$, $j=\{1,2,...,c\}$ is the pixel intensity of the image (with size $r \times c$ and with $n=L$ gray levels from zero to $L-1$), $t_k$ is one of the selected different thresholds (in bi-level case $k=b$ with just one threshold value and in the case of multi-level $k=\{1,2,...,L-1\}$ with $m$ number of desired thresholds), $S_k$ is one of the sets in which pixels with intensities between thresholds $t_k$ and $t_{k+1}$ are located.

Otsu$^{52}$ proposed a thresholding technique based on probability principles such as zero-, first- and second-order statistics to find the thresholds which can maximize between-class variance. To calculate the probability of each gray-level occurrence in the image, the number of the pixels with
that intensities is needed. Histogram diagram is a way to find this numbers as a gray-level distribution. The probability of each gray-level is determined by equation (10).

\[
\begin{align*}
\Pr_i &= \frac{h_i}{N_p}, \\
& \quad 0 \leq i \leq L-1,
\end{align*}
\]  

(10)

where \( h_i \) is the value of histogram (the number of pixels with the specific intensity level \( i \)), \( N_p \) is the number of pixels in the image \( (r \times c) \). In fact, equation (10) is the normalized version of the histogram by \( N_p \) and so \( 0 \leq \Pr_i < 1 \), \( \sum_{i=1}^{N_p} \Pr_i = 1 \).

Similar to the Otsu’s method, Kapur attempts to find the maximum entropy of the image after segmentation. Normalized Image histogram data again is used as the probability distribution of the pixels intensities. Equations for bi-level and multi-level Otsu’s and Kapur’s method are tabulated in Table 1 and Table 2, respectively. From the tables, the objective function that is used for the CHPSO is defined by equations (11) and (12) for Otsu’s and Kapur’s methods respectively.

\[
\text{fitness}(T) = \max(\sigma^2(T)),
\]

(11)

\[
\text{fitness}(T) = \max(H(T)),
\]

(12)

where vector \( T = \{t_1,t_2,\ldots,t_m\} \), \( 0 \leq t_i \leq L-1 \), \( i = 1,2,\ldots,m \) (for bi-level value \( t_b \)) is a series of thresholds values.

**Table 1: Otsu’s method equations**

| Bi-level | Multilevel |
|----------|------------|
| Image after thresholding | Image after thresholding (\( m = 1 \)) |
| Image is divided into two sets \( S_1 \) and \( S_2 \) (background and foreground pixels) using a threshold at the level \( t_b \) | Image after thresholding is divided into \( m \) sets \( \{S_1,S_2,\ldots,S_m\} \) using thresholds at the levels \( \{t_1,t_2,\ldots,t_m\} \) |
| Classes Gray-levels | Classes Gray-levels |
| \( S_1 = [0,\ldots,t_b-1] \) | \( S_1 = [0,\ldots,t_1-1] \) |
| \( S_2 = [t_b,\ldots,L-1] \) | \( S_2 = [t_1,\ldots,t_2-1] \) |
| \( \vdots \) | \( \vdots \) |
| \( S_m = [t_m,\ldots,L-1] \) | \( S_m = [t_m,\ldots,L-1] \) |
| Probabilities distributions for each class \( \omega(k) = \sum_{i=0}^{k-1} \Pr_i \) | |

(13)
\[ \omega_0 = \sum_{i=0}^{t_0-1} \Pr_i = \omega(t_0) \]
\[ \omega_h = \sum_{i=t_0}^{t_2-1} \Pr_i = \omega(t_2) - \omega(t_1) \]
\[ \omega_1 = \sum_{i=t_b}^{t_2-1} \Pr_i = \omega(L) - \omega(t_b) \]
\[ \omega_2 = 1 \]
\[ \omega_i = \sum_{i=t_b}^{t_2-1} \Pr_i = \omega(L) - \omega(t_1) \]
\[ \omega_m = \sum_{i=t_1}^{t_m-1} \Pr_i = \omega(L) - \omega(t_m) \]
\[ \sum_{i=0}^{m} \omega_1 = \omega_1 + \omega_2 = 1 \]

Mean levels for each class
\[ \mu(k) = \sum_{i=0}^{k-1} i \Pr_i \]
\[ \mu_0 = \frac{\sum_{i=0}^{t_0-1} i \Pr_i}{\omega_0} = \mu(t_0) \]
\[ \mu_1 = \frac{\sum_{i=t_0}^{t_2-1} i \Pr_i}{\omega_1} = \mu(t_2) - \mu(t_1) \]
\[ \mu_m = \frac{\sum_{i=t_0}^{t_m-1} i \Pr_i}{\omega_m} = \mu(L) - \mu(t_m) \]

Mean levels for the whole image
\[ \mu_I = \sum_{i=0}^{t_0-1} i \Pr_i = \mu(L) = \sum_{i=0}^{m} \omega_i \mu_i \]

The Otsu variance between classes
\[ \sigma^2 = \sigma_0^2 + \sigma_1^2 = \omega_0 \mu_0^2 \mu_L^2 + \omega_1 \mu_L^2 \mu_T^2 \]
\[ \sigma^2 = \sigma^2 \sum_{i=0}^{m} \sigma_i = \sum_{i=0}^{m} \omega_1 \mu_1^2 \mu_T^2 \]

Table 2: Kapur’s method equations

| Bi-level | Multilevel |
|----------|------------|
| Image after thresholding | Image after thresholding |
| Image is divided into two sets \( S_1 \) and \( S_2 \) (background and foreground pixels) using a threshold at the level \( t_b \), \( m = 1 \) | Image is divided into \( m \) sets \( \{S_1, S_2, \ldots, S_m\} \) using thresholds at the levels \( \{t_1, t_2, \ldots, t_m\} \) |
| Classes Gray-levels | Classes Gray-levels |
| \( S_1 = [0, \ldots, t_1 - 1] \) | \( S_1 = [0, \ldots, t_1 - 1] \) |
| \( S_2 = [t_b, \ldots, L - 1] \) | \( S_2 = [t_1, \ldots, t_2 - 1] \) |
| The Kapur’s classes entropies | }
The Kapur’s overall entropy

\[ H = H_1 + H_2 \]  

(24)  

\[ H = \sum_{i=0}^{m} H_i \]  

(25)

In this paper, we proposed a combination of the CHPSO and these two criteria to find the best threshold values of the test image. In the next chapter, the application of the proposed method for medical image threshold segmentation is investigated and compared to the similar method.

4 Experimental results and Discussion

In order to evaluate the performance of the CHPSO, it has been applied on 10 benchmark images and the results compared with a novel heuristic algorithm named Amended Bacterial Foraging (ABF) algorithm\(^5\). Images are T2-weighted Brain-hemispheric MR and downloaded from the whole brain atlas database\(^5\). Transaxial slices from 22 to 112 with 10 slice gap interval are selected from the database. All the images have the same size (256 × 256 pixels) and they are in PNG format. A representation of the test images is depicted in Figure 2.
Since random numbers are used in the CHPSO algorithm like other heuristic methods in each run of the algorithm the outcome most likely to different from another run. Thus, it is valuable to employ an appropriate statistical metrics to measure the efficiency of the method. The results have been reported after executing the algorithm for 50 times for each image. It is common in the literature that the number of threshold points is selected from numbers 2, 3, 4, and 5. The number of iterations is set as stopping criteria for the CHPSO and there are not any limit for the best fitness values remains with no change. Therefore, algorithm run for the whole of the time interval and the parameters of the method are tabulated in Table 3. We try to make a similar situation to the ABF method for better comparison.
Calculating the Standard Deviation (STD) from equation (26) is a desirable way to show the dispersion of the data after 50 times repetition and minimum STD for algorithms the better stability.

\[
STD = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (\text{Bestfit}_i - \text{Meanfit})^2}
\]  
(26)

where \( N \) is the number of algorithm run \((N = 50)\), \( \text{bestfit}_i \) is the best fitness acquired from the \( i-th \) algorithm execution and \( \text{Meanfit} \) is the average result of all best finesses. Peak Signal to Noise Ratio (PSNR) is another way of assessing the proposed method in respect to quality and noise effects on the image. It compares the threshold delineated image with original one as a reference to find how much of original data are conveyed to the segmented image. The bigger PSNR the better signal quality and for defining the PSNR a common way is first to determine the mean squared error (MSE) by using both image data from the equation (27), then PSNR in decibels unit (dB) can be calculated by equation (28)\(^7\).

\[
MSE = \frac{1}{r \times c} \sum_{j=0}^{r-1} \sum_{j=0}^{c-1} [I(i, j) - I_s(i, j)]^2
\]  
(27)

\[
PSNR = 20\log_{10}\left(\frac{255}{\sqrt{MSE}}\right)
\]  
(28)

where \( r \) and \( c \) are the number of row and columns of gray-level original and segmented images \((I \text{ and } I_s)\), respectively. In order to qualitatively assessment of the method, the results can be evaluated by the popular uniformity measure as equation (29)\(^53\).

\[
u = 1 - 2 \times m \times \sum_{j=0}^{m} \sum_{i \in R_j} (f_i - \mu_j)^2 \quad \sum_{i \in R_j} (f_{\text{max}} - f_{\text{min}})^2
\]  
(29)

where, \( m \) is the number of thresholds, \( R_j \) is the \( j-th \) region of the image that is segmented, \( f_i \) is the gray level of the pixel \( i \), \( \mu_j \) is the mean of the gray levels inside the region \( R_j \), \( N \) is the number of total pixels in the image, \( f_{\text{min}} \) and \( f_{\text{max}} \) are the minimum and maximum gray level of pixels within the image respectively. The value of the uniformity measure is between 0 and 1 \(^9\). Uniformity with higher value means the better thresholding. Also for having a comparison with ABF results, misclassification error (in present) which is the difference between the best thresholding with \( \nu = 1 \) and the value calculated from equation (29) is considered as a criterion.

The CHPSO is applied over the complete set of benchmark images first considering the Otsu’s method (equation (12) as the fitness function) and then using the Kapur’s method as the fitness function (equation (11)) to find the optimum thresholds. Note that, because of the CHPSO optimization algorithm has the type of minimization, fitness functions values are multiplied by minus to change the problem space from maximization to minimization.
A visualization of the results after applying the proposed method on the images are illustrated in Figure 3 and Figure 4 using Otsu’s and Kapur’s criteria, respectively. As we can see in the figures, the white matter, gray matter, and cerebrospinal fluid are delineated perfectly with more details. In the case of the bigger number of thresholds, the quality of the image is significantly considerable and reveals that the proposed method can save similar patterns in the original image and convey them to the segmented one. In medical applications, the details in the images mean better diagnosis and better treatment and the images proof that our method can be used for this purposes.
Figure 3. CHPSO results using Otsu’s method – (a to d) are images after thresholding with m = 2, 3, 4 and 5, respectively.
The performance of the CHPSO for image thresholding was astonishing and clearly shows the powerfulness of this method, but this results should be compared with other similar techniques to figure out the advantages and disadvantages of the CHPSO. For this reason, the results of the ABF algorithm are tabulated beside the CHPSO in the following tables.

From the Table 4, we can assert that two methods have a similar amount of fitness function, especially for Kapur’s method. In the case of Otsu, the results of the two proposed method are different but they could converge well. The difference between these results is that no pre- or post-processing have been done on the images for the case of CHPSO. For instance, we can see threshold value of 0 in the slice 112 that is because in the original image there is a line with zero intensity.

Table 4. Comparison study of threshold and fitness function values obtained by CHPSO and ABF using Otsu and Kapur

| Slice | m | CHPSO | | | | | ABF | | | |
|-------|---|-------|---|-------|---|-------|---|-------|---|-------|
|       |   | Optimum Thresholds values | Fitness function | Optimum Thresholds values | Fitness function | Optimum Thresholds values | Fitness function | Optimum Thresholds values | Fitness function |
| Slice22 | 2 | 35.99 | 2273.7 | 94.182 | 9.2155 | 93.176 | 1808.847 | 95.184 | 9.2155 |
|       | 3 | 22.66, 117 | 2370.3 | 56.114.184 | 11.733 | 68.118.188 | 2165.1669 | 68.115.186 | 11.7123 |
|       | 4 | 16.71, 81, 128 | 2409.3 | 47.96.146.192 | 13.946 | 45.109.100.184 | 2288.3476 | 43.89.136.186 | 13.9481 |
|       | 5 | 14.42, 73, 105, 140 | 2429.2 | 41.83.124.170.205 | 16.105 | 47.90.128.174.191 | 2321.4813 | 44.85.127.173.207 | 16.2228 |
| Slice32 | 2 | 43.112 | 2607.1 | 108.184 | 9.2645 | 107.178 | 1809.339 | 110.185 | 9.2644 |
|       | 3 | 23.69, 122 | 2712.4 | 52.114.184 | 11.688 | 68.125.168 | 2557.334 | 82.130.190 | 11.6166 |
|       | 4 | 18.58, 96, 135 | 2751.8 | 38.84.132.188 | 13.936 | 51.94.129.188 | 2672.369 | 44.92.136.188 | 13.9222 |
|       | 5 | 17.51, 84, 116, 150 | 2773.8 | 33.77.121.168.203 | 16.043 | 45.79.115.152.194 | 2714.2533 | 42.92.134.184.216 | 16.0040 |
| Slice42 | 2 | 45.121 | 3044.4 | 112.182 | 9.2585 | 112.182 | 2118.4914 | 114.184 | 9.2585 |
|       | 3 | 30.81, 135 | 3143.9 | 82.130.186 | 11.578 | 78.127.178 | 2691.1869 | 83.136.187 | 11.5743 |
|       | 4 | 23.67, 113, 161 | 3199.7 | 28.75.126.186 | 13.865 | 57.112.138.177 | 3120.0412 | 48.102.144.190 | 13.8155 |
|       | 5 | 18.55, 91, 128, 174 | 3234.6 | 26.71.116.162.201 | 16.026 | 53.90.126.162.202 | 3166.4774 | 40.93.137.183.217 | 15.9644 |
| Slice52 | 2 | 45.115 | 2858.8 | 116.184 | 9.2447 | 118.188 | 1569.4257 | 117.186 | 9.2447 |
|       | 3 | 36.88, 132 | 2946.9 | 108.164.202 | 11.58 | 98.130.176 | 2099.4181 | 109.163.203 | 11.5552 |
|       | 4 | 19.58, 97, 137 | 2997.4 | 83.124.167.206 | 13.734 | 84.109.140.180 | 2488.3686 | 92.131.173.210 | 13.7902 |
|       | 5 | 18.57, 92, 125, 164 | 3023.6 | 25.69.117.165.203 | 15.874 | 59.99.129.162.206 | 2934.7700 | 48.89.131.175.209 | 15.7152 |
| Slice62 | 2 | 46.123 | 3369.4 | 120.186 | 9.3367 | 118.183 | 2158.5697 | 119.186 | 9.3314 |
|       | 3 | 39.96, 147 | 3484.6 | 100.146.194 | 11.674 | 94.132.186 | 2781.2290 | 99.150.195 | 11.6688 |
|       | 4 | 35.84, 122, 169 | 3537.5 | 86.126.169.207 | 13.77 | 78.118.151.198 | 3217.2777 | 93.134.176.212 | 13.7812 |
|       | 5 | 19.59, 96, 132, 179 | 3577.9 | 28.72.117.161.201 | 15.866 | 75.102.130.162.201 | 3312.4446 | 82.114.147.181.216 | 15.6953 |
| Slice72 | 2 | 45.126 | 3206.2 | 116.178 | 9.4205 | 111.173 | 2082.9210 | 117.179 | 9.4205 |
Although, the results for the Otsu’s method is different but we can compare them in terms of statistical analysis.

In terms of the Standard deviation, the Peak Signal to Noise Ratio, we can understand from Table 5 that the CHPSO is highly robustness and in almost all cases gained a better STD rather than ABF and consequently more trustable respect to changes of output in each execution of the algorithm. The difference of PSNR for all images about 2 times bigger, reveal another property of the proposed method. The performance of the CHPSO in skipping from the local minima to better solutions lead to outstanding results in compare to the ABF and proof the power of multiswarm optimization algorithms for better exploration of the search space.

Table 5. Comparison study of the CHPSO and ABF in terms of STD and PSNR

| Image | CHPSO | ABF |
|-------|-------|-----|
|       | Otsu  | Kapur | Otsu  | Kapur |
|       | STD   | STD   | STD   | STD   |
|       | PSNR  | PSNR  | PSNR  | PSNR  |
| Slice2 | 2     | 0      | 24.5006 | 4.10766e-14 | 24.269 | 0.0021 | 10.0804 | 1.7121e-4 | 10.3797 |
|        | 3     | 2.2968e-12 | 24.6904 | 5.3832e-15 | 24.416 | 0.7238 | 13.4236 | 0.0009 | 13.4504 |
|        | 5     | 1.3781e-12 | 24.8633 | 0.011992 | 24.573 | 0.8222 | 15.5969 | 0.0021 | 16.2311 |
| Slice3 | 1     | 24.6657 | 24.6535 | 0.094308 | 24.654 | 1.0210 | 16.8073 | 0.0165 | 17.7098 |
|        | 2     | 2.7562e-12 | 24.4469 | 7.1776e-15 | 24.315 | 0.3119 | 9.1680 | 1.0022e-4 | 9.2958 |
|        | 4     | 1.8375e-12 | 24.5709 | 1.0766e-14 | 24.444 | 0.4844 | 12.5778 | 0.0018 | 13.2597 |
|        | 5     | 3.6749e-12 | 24.7943 | 0.023538 | 24.567 | 0.5001 | 15.5448 | 0.0111 | 16.9498 |
| Slice4 | 0     | 24.5606 | 8.972e-15 | 24.339 | 0.2813 | 9.0416 | 1.5006e-4 | 9.0892 |
|        | 1     | 0.1378e-12 | 24.3693 | 8.972e-15 | 24.443 | 0.4063 | 12.4376 | 0.0175 | 12.9165 |
|        | 2     | 0.016243 | 24.7001 | 0.00032519 | 24.585 | 0.7813 | 15.0468 | 0.0527 | 16.3489 |
|        | 3     | 0.0802017 | 24.8189 | 0.0073274 | 24.689 | 1.2969 | 15.9160 | 0.0729 | 17.7189 |
|        | 4     | 2.9187e-13 | 24.4490 | 5.3832e-15 | 24.359 | 0.1278 | 9.1512 | 1.3018e-4 | 9.2423 |
|        | 5     | 9.1873e-13 | 24.6998 | 8.972e-15 | 24.479 | 0.5000 | 9.9752 | 0.0015 | 10.7372 |
| Slice5 | 2     | 0.0042822 | 24.7770 | 0.048056 | 24.661 | 0.9375 | 11.4083 | 0.068 | 11.9692 |
|        | 3     | 0.1792 | 24.9605 | 0.046858 | 24.698 | 1.7656 | 15.4337 | 0.0996 | 16.8916 |
|        | 4     | 2.5936e-12 | 24.3949 | 3.5888e-15 | 24.354 | 0.1474 | 9.1676 | 1.1012e-4 | 9.2923 |
|        | 5     | 2.2496e-12 | 24.6499 | 5.3832e-15 | 24.574 | 1.0001 | 10.7943 | 0.0047 | 11.0396 |
| Slice6 | 2     | 3.6749e-12 | 24.8458 | 0.036804 | 24.718 | 1.2031 | 11.1523 | 0.0169 | 11.9759 |
|        | 3     | 0.66916 | 24.8808 | 0.080143 | 24.777 | 1.8906 | 13.8409 | 0.0867 | 14.8511 |
|        | 4     | 2.7562e-12 | 24.3542 | 8.972e-15 | 24.358 | 0.2594 | 8.9376 | 5.5610e-5 | 9.4718 |
|        | 5     | 2.5936e-12 | 24.7595 | 8.972e-15 | 24.357 | 0.5156 | 10.7279 | 0.0097 | 11.0624 |
| Slice7 | 2     | 1.3781e-12 | 24.8080 | 0.00020507 | 24.656 | 0.6875 | 11.1622 | 0.0378 | 11.7246 |
|        | 3     | 0.017521 | 24.8603 | 0.053892 | 24.794 | 1.0625 | 11.8284 | 0.0642 | 12.6370 |
|        | 4     | 2.7562e-12 | 24.3593 | 5.3832e-15 | 24.427 | 0.2731 | 9.4512 | 2.73 C7e-4 | 9.7528 |
|        | 5     | 4.5936e-13 | 24.5585 | 3.5888e-15 | 24.59 | 1.1406 | 10.0912 | 0.0049 | 10.5192 |
From the Error! Not a valid bookmark self-reference., we can see that the CHPSO is also significantly faster than ABF algorithm. In this report, the algorithm is implemented using MATLAB software on the PC with a dual core of 3GHz with 2 GB memory. Faster results mean the ability of the proposed method for using in the real-time application that is common in medicine.

In term of misclassification error in percentage, CHPSO could achieve better results in most cases especially when the number of thresholds is five. For the sake of representation the results of only three and five thresholds values are illustrated in

Table 7.

In general, the experimental results reveals that the CHPSO approach using both Otsu and Kapur’s criteria is a robust, fast and accurate for the problem of medical image thresholding segmentation in comparison to the similar methods. The CHPSO for getting perfect results owe to the multi-swarm searching strategy and the behavior of the particles that able them to search more new regions in the problem space and jump from local minima to a global one and discover a bigger area. For the speed of the algorithm, it was predictable, because of the simple equations of the original PSO that have not any complex calculations.

| image  | m  | CHPSO        | ABF          |
|--------|----|--------------|--------------|
|        |    | Ct Otsu      | Ct Kapur     | Ct Otsu      | Ct Kapur     |
| Slice22| 2  | 0.1358       | 0.28989      | 3.0993       | 6.5245       |
|        | 3  | 0.1353       | 0.29003      | 3.2681       | 6.9201       |
|        | 4  | 0.1334       | 0.27506      | 3.7904       | 7.4347       |
|        | 5  | 0.1313       | 0.26038      | 3.9115       | 8.1779       |
| Slice32| 2  | 0.1375       | 0.28806      | 2.7974       | 6.7466       |
|        | 3  | 0.1347       | 0.28466      | 3.2821       | 7.2019       |
|        | 4  | 0.1352       | 0.27242      | 3.9708       | 7.5040       |
|        | 5  | 0.1313       | 0.2554       | 4.6637       | 8.1982       |
| Slice42| 2  | 0.1347       | 0.28964      | 2.7200       | 6.7982       |
|        | 3  | 0.1347       | 0.27923      | 3.2588       | 7.1599       |
|        | 4  | 0.1347       | 0.27337      | 3.8201       | 7.8631       |
|        | 5  | 0.1333       | 0.255      | 4.5712       | 8.1029       |
| Slice52| 2  | 0.1362       | 0.28743      | 2.7831       | 6.7001       |
|        | 3  | 0.1352       | 0.28174      | 3.3111       | 7.0938       |
|        | 4  | 0.1346       | 0.27972      | 3.6868       | 7.2184       |
|        | 5  | 0.1313       | 0.25377      | 4.4951       | 8.2670       |
| Slice62| 2  | 0.1353       | 0.30375      | 2.9800       | 6.5281       |
|        | 3  | 0.1351       | 0.27981      | 3.2529       | 6.8914       |
|        | 4  | 0.1341       | 0.28061      | 3.2801       | 7.1803       |
|        | 5  | 0.1301       | 0.25346      | 4.8144       | 8.1060       |
| Slice72| 2  | 0.1344       | 0.29204      | 2.7206       | 6.6929       |
|        | 3  | 0.1361       | 0.27807      | 3.0875       | 7.3030       |
Table 7. Comparison study of CHPSO and ABF in term of misclassification error in percentage (ME)

| Image   | m | CHPSO ME (in %) Otsu | CHPSO ME (in %) Kapur | ABF ME (in %) Otsu | ABF ME (in %) Kapur |
|---------|---|----------------------|-----------------------|--------------------|--------------------|
| Slice 022 | 3 | 0.1578               | 1.4977                | 1.98               | 2.04               |
|          | 5 | 0.0744               | 1.2437                | 1.41               | 1.44               |
| Slice 032 | 3 | 0.1312               | 1.1307                | 1.82               | 2.18               |
|          | 5 | 0.0990               | 0.5876                | 0.85               | 1.01               |
| Slice 042 | 3 | 0.2319               | 3.6760                | 3.24               | 3.05               |
|          | 5 | 0.0999               | 0.2194                | 1.1                | 1.27               |
| Slice 052 | 3 | 0.3038               | 11.6213               | 8.03               | 9.16               |
|          | 5 | 0.0772               | 0.1719                | 1.33               | 1.40               |
| Slice 062 | 3 | 0.3995               | 9.2941                | 7.47               | 7.73               |
|          | 5 | 0.0675               | 0.2305                | 4.33               | 5.57               |
| Slice 072 | 3 | 0.2954               | 9.7454                | 10.98              | 8.01               |
|          | 5 | 0.0735               | 0.7059                | 4.97               | 4.31               |
| Slice 082 | 3 | 0.3304               | 11.2382               | 12.77              | 12.33              |
|          | 5 | 0.0542               | 0.2791                | 9.02               | 7.24               |
| Slice 092 | 3 | 0.2265               | 11.8704               | 9.49               | 10.29              |
|          | 5 | 0.0476               | 0.1634                | 5.53               | 5.97               |
| Slice 102 | 3 | 0.2475               | 7.9017                | 7.25               | 7.63               |
|          | 5 | 0.0465               | 0.0231                | 5.49               | 6.98               |
| Slice 112 | 3 | 0.0849               | 0                    | 1.77               | 2.38               |
|          | 5 | 0.0479               | 0                    | 1.11               | 1.59               |

5 Conclusion

This paper has presented a new approach for thresholding segmentation of MR brain images using a combination of Convergent Heterogeneous Particle Swarm Optimization (CHPSO) and two classical thresholding techniques. We have shown that utilizing a new strategy of searching named multiswarm can improve the performance of a heuristic algorithm for the problem of image thresholding. In this strategy, the particles are divided into subswarms and each subswarm separately searches the problem space to find the best solution and improve the exploitation while the subswarms have a cooperation with each other and share their information for better exploration to find the global best position and jump from local optimal answers. Empirical testing on a set of 10 medical images, shows that the proposed approach is significantly robustness with better convergence, in comparison to similar techniques. Instead of reporting classical techniques, we compare our method with the state of the art and powerful heuristic
method Amended Bacterial Foraging (ABF). In terms of speed and accuracy also, the CHPSO outperforms the ABF. For visualization of the results, all segmented images have been illustrated. Images show that the details form the original image properly conveyed into the segmented image with more details. In future work, we hope to apply other kinds of multiswarm heuristic algorithms and provide more obvious demonstrations of this technique by applying it to more complex kinds of segmentation and images.

References

1. Hamamci A, Kucuk N, Karaman K, Engin K, Unal G. Tumor-cut: segmentation of brain tumors on contrast enhanced MR images for radiosurgery applications. Med Imaging, IEEE Trans. 2012;31(3):790-804.
2. Banerjee S, Mitra S, Shankar BU. Single seed delineation of brain tumor using multi-thresholding. Inf Sci (Ny). 2016;330:88-103.
3. Balan NS, Kumar AS, Raja NSM, Rajinikanth V. Optimal Multilevel Image Thresholding to Improve the Visibility of Plasmodium sp. in Blood Smear Images. In: Proceedings of the International Conference on Soft Computing Systems. ; 2016:563-571.
4. Dawngliana M, Deb D, Handique M, Roy S. Automatic brain tumor segmentation in MRI: Hybridized multilevel thresholding and level set. In: Advanced Computing and Communication (ISACC), 2015 International Symposium on. ; 2015:219-223.
5. Remamany KP, Chelliah T, Chandrasekaran K, Subramanian K. Brain Tumor Segmentation in MRI Images Using Integrated Modified PSO-Fuzzy Approach. Int Arab J Inf Technol. 2015;12.
6. Feng Y, Shen X, Chen H, Zhang X. Internal Generative Mechanism Based Otsu Multilevel Thresholding Segmentation for Medical Brain Images. In: Advances in Multimedia Information Processing--PCM 2015. Springer; 2015:3-12.
7. Akay B. A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. Appl Soft Comput. 2013;13(6):3066-3091.
8. Zhou C, Tian L, Zhao H, Zhao K. A method of two-dimensional Otsu image threshold segmentation based on improved firefly algorithm. In: Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2015 IEEE International Conference on. ; 2015:1420-1424.
9. Bhandari AK, Kumar A, Singh GK. Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur’s, Otsu and Tsallis functions. Expert Syst Appl. 2015;42(3):1573-1601.
10. Sezgin M, others. Survey over image thresholding techniques and quantitative performance evaluation. J Electron Imaging. 2004;13(1):146-168.
11. Liu Y, Xue J, Li H. The Study on the Image Thresholding Segmentation Algorithm. 2015.
12. Horng M-H. Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation. Expert Syst Appl. 2011;38(11):13785-13791.
13. Ayala HVH, dos Santos FM, Mariani VC, dos Santos Coelho L. Image thresholding segmentation based on a novel beta differential evolution approach. Expert Syst Appl. 2015;42(4):2136-2142.
14. Oliva D, Cuevas E, Pajares G, Zaldivar D, Osuna V. A multilevel thresholding algorithm using electromagnetism optimization. Neurocomputing. 2014;139:357-381.
15. Hammouche K, Diaf M, Siarry P. A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. Comput Vis Image Underst.
2008;109(2):163-175.
16. Horng MH. A multilevel image thresholding using the honey bee mating optimization. *Appl Math Comput*. 2010;215(9):3302-3310. doi:10.1016/j.amc.2009.10.018.
17. Cuevas E, Zaldivar D, Perez-Cisneros M. Otsu and Kapur Segmentation Based on Harmony Search Optimization. In: *Applications of Evolutionary Computation in Image Processing and Pattern Recognition*. Springer; 2016:169-202.
18. Otsu N. A threshold selection method from gray-level histograms. *Automatica*. 1975;11(285-296):23-27.
19. Vala HJ, Baxi A. A review on Otsu image segmentation algorithm. *Int J Adv Res Comput Eng Technol*. 2013;2(2):pp - 387.
20. Bindu CH, Prasad KS. An efficient medical image segmentation using conventional OTSU method. *Int J Adv Sci Technol*. 2012;38:67-74.
21. Huang M, Yu W, Zhu D. An improved image segmentation algorithm based on the Otsu method. In: *Software Engineering, Artificial Intelligence, Networking and Parallel & Distributed Computing (SNPD), 2012 13th ACIS International Conference on*. ; 2012:135-139.
22. Huang D-Y, Wang C-H. Optimal multi-level thresholding using a two-stage Otsu optimization approach. *Pattern Recognit Lett*. 2009;30(3):275-284.
23. Kapur JN, Sahoo PK, Wong AKC. A new method for gray-level picture thresholding using the entropy of the histogram. *Comput Vision, Graph Image Process*. 1985;29(1):140. doi:10.1016/0734-189X(85)90125-2.
24. Banerjee S, Jana ND. Bi level kapurs entropy based image segmentation using particle swarm optimization. In: *Computer, Communication, Control and Information Technology (C3IT), 2015 Third International Conference on*. ; 2015:1-4.
25. Oliva D, Cuevas E, Pajares G, Zaldivar D, Perez-Cisneros M. Multilevel thresholding segmentation based on harmony search optimization. *J Appl Math*. 2013:2013.
26. Gao H, Xu W, Sun J, Tang Y. Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm. *Instrum Meas IEEE Trans*. 2010;59(4):934-946.
27. Hammouche K, Dif M, Siarry P. A comparative study of various meta-heuristic techniques applied to the multilevel thresholding problem. *Eng Appl Artif Intell*. 2010;23(5):676-688.
28. Fu Z, He JF, Cui R, et al. Image Segmentation with Multilevel Threshold of Gray-Level & Gradient-Magnitude Entropy Based on Genetic Algorithm. *psq*. 2015;255:9.
29. Yin P-Y. A fast scheme for optimal thresholding using genetic algorithms. *Signal Processing*. 1999;72(2):85-95.
30. Sun G, Zhang A, Yao Y, Wang Z. A novel hybrid algorithm of gravitational search algorithm with genetic algorithm for multi-level thresholding. *Appl Soft Comput*. 2016.
31. Zhang Y, Yan H, Zou X, Tao F, Zhang L. Image Threshold Processing Based on Simulated Annealing and OTSU Method. In: *Proceedings of the 2015 Chinese Intelligent Systems Conference*. ; 2016:223-231.
32. Kaur U, Sharma R, Dosanjh M. Cancers Tumor Detection using Magnetic Resonance Imaging with Ant Colony Algorithm.
33. Eberhart R, Kennedy J. A new optimizer using particle swarm theory. *MHS’95 Proc Sixth Int Symp Micro Mach Hum Sci*. 1995:39-43. doi:10.1109/MHS.1995.494215.
34. Hamdaoui F, Ladgham A, Sakly A, Mtibaa A. A new images segmentation method based on modified particle swarm optimization algorithm. *Int J Imaging Syst Technol*. 2013;23(3):265-271.
35. Ait-Aoudia S, Guerrout E-H, Mahiou R. Medical Image Segmentation Using Particle Swarm Optimization. In: Information Visualisation (IV), 2014 18th International Conference on. ; 2014:287-291.
36. Wei K, Zhang T, Shen X, Liu J. An improved threshold selection algorithm based on particle swarm optimization for image segmentation. In: Natural Computation, 2007. ICNC 2007. Third International Conference on. Vol 5. ; 2007:591-594.
37. Mishra D, Bose I, De UC, Das M. Medical Image Thresholding Using Particle Swarm Optimization. In: Intelligent Computing, Communication and Devices. Springer; 2015:379-383.
38. Esmin AAA, Coelho RA, Matwin S. A review on particle swarm optimization algorithm and its variants to clustering high-dimensional data. Artif Intell Rev. 2015;44(1):23-45.
39. Liu Y, Mu C, Kou W, Liu J. Modified particle swarm optimization-based multilevel thresholding for image segmentation. Soft Comput. 2015;19(5):1311-1327.
40. Sayed GI, Hassanien AE. Abdominal CT Liver Parenchyma Segmentation Based on Particle Swarm Optimization. In: The 1st International Conference on Advanced Intelligent System and Informatics (AISI2015), November 28-30, 2015, Beni Suef, Egypt. ; 2016:219-228.
41. Cheung NJ, Ding X-M, Shen H-B. OptiFel: A convergent heterogeneous particle swarm optimization algorithm for Takagi--Sugeno fuzzy modeling. Fuzzy Syst IEEE Trans. 2014;22(4):919-933.
42. Eberhart RC, Shi Y. Particle swarm optimization: developments, applications and resources. In: Evolutionary Computation, 2001. Proceedings of the 2001 Congress on. Vol 1. ; 2001:81-86.
43. Song M-P, Gu G-C. Research on particle swarm optimization: a review. In: Machine Learning and Cybernetics, 2004. Proceedings of 2004 International Conference on. Vol 4. ; 2004:2236-2241.
44. Li Y, Jiao L, Shang R, Stolkin R. Dynamic-context cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation. Inf Sci (Ny). 2015;294:408-422.
45. Muthiah-Nakarajan V, Noel MM. Galactic Swarm Optimization: A new global optimization metaheuristic inspired by galactic motion. Appl Soft Comput J. 2016;38:771-787. doi:10.1016/j.asoc.2015.10.034.
46. Mozaffari MH, Zahir SH. UNSUPERVISED DATA AND HISTOGRAM CLUSTERING USING INCLINED PLANES SYSTEM OPTIMIZATION ALGORITHM. Image Anal Stereol. 2014;33(1).
47. Mozaffari MH, Abd H, Zahir SH. Application of inclined planes system optimization on data clustering. In: Pattern Recognition and Image Analysis (PRIA), 2013 First Iranian Conference on. ; 2013:1-3.
48. Pal SS, Kumar S, Kashyap M, Choudhary Y, Bhattacharya M. Multi-level Thresholding Segmentation Approach Based on Spider Monkey Optimization Algorithm. In: Proceedings of the Second International Conference on Computer and Communication Technologies. ; 2016:273-287.
49. Sathya PD, Kayalvizhi R. Optimal multilevel thresholding using bacterial foraging algorithm. Expert Syst Appl. 2011;38(12):15549-15564. doi:10.1016/j.eswa.2011.06.004.
50. Rajinikanth V, Raja NSM, Satapathy SC. Robust Color Image Multi-thresholding Using Between-Class Variance and Cuckoo Search Algorithm. In: Information Systems Design and Intelligent Applications. Springer; 2016:379-386.
51. Cheung A, Ding X-M, Shen H-B. OptiFel: A Convergent Heterogeneous Particle Swarm Optimization Algorithm for Takagi-Sugeno Fuzzy Modeling. *IEEE Trans Fuzzy Syst.* 2013;PP(99):1-1. doi:10.1109/TFUZZ.2013.2278972.

52. Otsu N, Smith PL, Reid DB, et al. A Threshold Selection Method from Gray-Level Histograms. 1979;20(1):62-66.

53. Sathya PD, Kayalvizhi R. Amended bacterial foraging algorithm for multilevel thresholding of magnetic resonance brain images. *Measurement.* 2011;44(10):1828-1848.

54. Johnson KA, Becker JA, Williams L. The whole brain atlas. 1999.