Australian Journal of Basic and Applied Sciences, 8(16) October 2014, Pages: 378-387

AENSI Journals

Australian Journal of Basic and Applied Sciences

ISSN:1991-8178

Journal home page: www.ajbasweb.com

Image Segmentation using a Hybrid Genetic Algorithm with Tabu list for Maximum Tsallis Entropy Thresholding

[1]L. Jubair Ahmed and [2]A. Ebenezer Jeyakumar

[1]*Research Scholar, Department of Electronics & Communication Engineering, Anna University Coimbatore, India*
[2]*Director, Sri Ramakrishna Engineering College, Anna University, India*

ARTICLE INFO

Article history:
Received 20 August 2014
Received in revised form
19 September 2014
Accepted 23 October 2014
Available online 16 November 2014

Keywords:
Image Segmentation, Maximum Tsallis entropy thresholding, Comprehensive Learning PSO, Ant Colony Optimization

ABSTRACT

Background: Image thresholding is an important technique for image processing and pattern recognition. Many thresholding techniques have been proposed in the literature. Objective: In this paper to compute optimum thresholds for Maximum Tsallis entropy thresholding (MTET) model, a new hybrid algorithm is proposed based on the fusion of a Genetic Algorithm with the Tabu Search method. In this new theory a system dependent parameter '*q*' measuring the degree of non-extensivity is introduced. The q parameter in the Tsallis entropy is used as an adjustable value, which plays an important role as a tuning parameter in the image segmentation. Thus replacing the traditional maximum entropy thresholding (MET) with a maximum Tsallis entropy. This new multilevel thresholding technique is called hybrid genetic algorithm (HGA) and tabu search algorithm for MTET**. Results:** Experimental results over multiple images with different range of complexities validate the efficiency of the proposed technique with regard to segmentation accuracy, speed, and robustness in comparison with other techniques reported in the literature. **Conclusion:** This article presents an extensive study on the application of a hybrid algorithm integrating two powerful metaheuristic techniques for multilevel thresholding for image segmentation problem. The proposed method was evaluated on various types of images, and the experimental results show the efficiency and the feasibility of the proposed method on the real images. It is demonstrated that the simulation results obtained using the hybrid HGA & tabu search method are superior to that of ABC, PSO and Genetic Algorithm methods in terms of producing quality thresholds. The validity and stability of the method is justified both qualitatively and quantitatively.

To Cite This Article: L. Jubair Ahmed and A. Ebenezer Jeyakumar., Image Segmentation using a Hybrid Genetic Algorithm with Tabu list for Maximum Tsallis Entropy Thresholding. ***Aust. J. Basic & Appl. Sci.,*** *8(16): 378-387, 2014*

INTRODUCTION

Image segmentation has been the subject of intensive research and a wide variety of segmentation techniques have been reported in the last two decades. In general, image segmentation divides an image into related sections or regions which consist of image pixels and their relationship among several data-feature values. Many thresholding techniques have been proposed to solve image segmentation problems and are classified by their differences (Pal, N.R., S.K. Pal, 1993). For instance, some techniques have been classified as either optimal or property-based , while other methods are identified as either global or local thresholdings based on the role of the intensity value . The difference between global and local thresholdings is that a global thresholding technique cuts the entire image with a single threshold value, whereas a local thresholding technique divides the image into sub-images, and for each a threshold is determined.

Segmentation algorithms are based on two significant criteria: the homogeneity of a region (thresholding) and the discontinuity between adjacent disjoint regions (finding edges). Since methods based on the homogeneity offer the advantage of a smaller storage space, fast processing speed and easy manipulation, thresholding techniques are considered the most popular among the computer vision communities (Otsu, N., 1979). Many thresholding techniques have been proposed to solve image segmentation problems and are classified by their differences. For instance, some techniques have been classified as either optimal or property-based, while other methods are identified as either global or local thresholdings based on the role of the intensity value.

Corresponding Author: L. Jubair Ahmed, Research Scholar, Department of Electronics & Communication Engineering, Anna University, India,
Tel: 91 8854124569; E-mail: sahmedjubair@gmail.com

Parametric approaches assume that each group (sub-image or class) has the probability density function (PDF) of a Gaussian distribution and finds an estimate of the parameters of such a distribution which will best fit the given histogram data . Unfortunately, when the desired number of classes is much lower than the number of peaks in the original histogram, the computation time to find the threshold values often becomes expensive.

The nonparametric approach is based on a search of the thresholds optimizing an objective function such as the between-class variance (Otsu's function) and entropy (Kapur's function). Recent developments in statistical mechanics based on Tsallis entropy have intensified the interest of investigating it as an extension of Shannon's entropy (Zhang, Yudong 2011). It appears in order to generalize the Boltzmann/Gibbs' traditional entropy to non-extensive physical systems. In this new theory a system dependent parameter '*q'* measuring the degree of non-extensivity is introduced. The q parameter in the Tsallis entropy is used as an adjustable value, which plays an important role as a tuning parameter in the image segmentation. Thus replacing the traditional maximum entropy thresholding (MET) with a maximum Tsallis entropy. This replacement not only improves the effectiveness of the segmentation process by increasing the number of thresholds, but also makes cumbersome in finding the optimum thresholds. Meanwhile, the computation time increases sharply when the number of thresholds increases, so the traditional exhaustive method finds it difficult to identify the optimal thresholds. Thus researchers tend to use meta-heuristics methods to find the optimum threshold values quickly.

To eliminate such problems the evolutionary techniques have been applied in solving multilevel thresholding problems. The GA, ACO and PSO, (Zhang, R., J. Liu, 2006; Peng-Yeng Yin, 1999) which are forms of probabilistic heuristic algorithm, have been successfully used in multilevel thresholding. The GA method is usually faster because the GA has parallel search techniques, which emulate natural genetic operations. Though the GA methods have been employed successfully to solve complex nonlinear optimization problems, recent research has identified some deficiencies in GA performance. This degradation in efficiency is apparent when the optimized parameters are highly correlated and the premature convergence of the GA reduces its search capability. The particle swarm optimization (PSO) has been applied to the multilevel thresholding for image segmentation. Then the DE algorithm is applied to find the optimal threshold values by minimizing the dissimilarity between the input image, likewise Particle Swarm Optimization (PSO) [5] is another latest evolutionary optimization technique, which is used for the multilevel thresholding, similarly the Ant Colony Optimization (ACO) and Simulated Annealing (SA) algorithms are used for multilevel image thresholding .

Taking into account the advantages of the meta-heuristics to escape from local optima with a reasonable time, some meta- heuristic techniques have been extensively employed to search more fastly the optimal thresholds. As is known, normal global meta- heuristic techniques conduct only one search operation in one iteration, for example (GA) is a search technique used in computing to find exact or approximate solutions to optimization and search problems.In (Hou's method *et al* (1990) ,the author claims restricted genetic operators which is found to render parts of the solution space unreachable and the order in which tasks are specified affects the likelihood that two tasks will be crossed over. A novel GA approach for scheduling tasks for parallel execution on multiprocessor systems can be implemented.

Conversely, this paper proposes a new hybrid technique HGA which integrates Genetic Algorithm with Tabu list for MTET, to perform segmentation with better solutions with less execution time. This hybrid technique is claimed very powerful compared to ABC, because Tsallisparameter'q' can be used as a tuning parameter for improvising image thresholding results. Thus finding the optimal thresholds at a more reasonable time is guaranteed by the proposed hybrid HGA method compared to ABC (Zhang, Yudong, 2011).

The remainder of this paper is organized as follows. In Section 2, the problem of the multilevel thresholding is formulated as an optimization problem and the objective function treated are briefly presented. The Section 3 deals with the proposed hybrid technique and its coding for the multilevel thresholding problem modelled as MTET. Section 4 gives numerical experiments and comparative results of the proposed method with the other techniques reported in the literature. Concluding remarks are given in Section 5.

Problem Formulation:

The optimal thresholding methods search the thresholds such that the segmented classes on the histogram satisfy the desired property. This is performed by fusion of a genetic algorithm with the tabu search method using Tsallisparameter'q' as a tuning parameter. In this paper, Maximum Tsallis entropy thresholding method is used (Zhang, Yudong, 2011).

A. *Initial Population:*

The individuals in the GA population can be represented as a vector of cells. A cell is a pair of task and processor and is represented by (t, p). Each cell indicates that task t is assigned to processor p. Figure 1 shows an example individual for eight tasks and three processors.

In an individual the cell determines which tasks are assigned to which processor. The processor assignments for the individual shown in Figure 1 are as shown in Figure 2. The length of an individual will be equal to the number of tasks in the DAG

(1, 2) (2, 1) (3, 3) (4, 2) (5, 1) (6, 3) (7, 1) (8, 3)

Fig. 1: Example for an individual.

Processor 1	Task 2	Task 5	Task 7
Processor 2	Task 1	Task 4	
Processor 3	Task 3	Task 6	Task 8

Fig. 2: Processor assignments for the individual.

The individuals are generated randomly to form the initial population. Each individual have one copy of each task, so that the length of individuals in the population is the number of tasks in the DAG. Each task is assigned randomly to the processor.

*B. **Fitness Function:***

In Genetic Algorithm, fitness function is an objective function that is need to optimize the problem. It is used to evaluate the individuals and it controls the genetic operators. The factors considered for multiprocessor scheduling problem are throughput, finishing time, processor utilization etc. The fitness function used in this research work is based on the makespan of the schedule. The makespan of a schedule S is defined as the following Equation (3.1)

$$MS (S) = \max ft (Pj) \qquad (3.1)$$

where ft (Pj) is the finishing time for the last task in processor Pj. The makespan should be converted into maximization problem as genetic operators will try to maximize the fitness function. So the fitness value of the schedule S is as shown in Equation (3.2)

$$F_{max} - MS (S) \qquad (3.2)$$

where F_{max} is the maximum finishing time observed. Thus, the optimal schedule has minimum makespan and larger fitness value than other schedules.

C. Genetic Operators:

Genetic Algorithm is a global optimization algorithm. Arriving at a proper balance between the local and the global phases of the optimization is a precondition for successful global optimization. Experience in using GA has shown that, GA does not make optimum use of the available information, because the process of reaching to the optimum solution is slow. This deficiency, resulting from GA's weakness in the fine-tuning can be remedied by using tools to improve the individuals. It is essential in designing these algorithms that a compromise is reached between the speed of convergence to a solution and the reliability of the solution.

Function of genetic operators is to create new population based on the current population of individuals. By combining good structures of two individuals, it may result in a better one. For multiprocessing scheduling problem, the genetic operators used must enforce the intraprocessor precedence relations, as well as completeness and uniqueness of the tasks in the schedule. Certain portions of the schedule may belong to the optimal schedule. By combining several of these optimal parts, it schedules the tasks efficiently. The three basic genetic operations are

(i) Crossover
(ii) Mutation
(iii) Selection

(i) Crossover:

The crossover operation is performed by parent population by swapping a part of one parent with a part of other. In multiprocessor scheduling, it exchanges substrings of cells between two individuals. This allows GA to find out new solutions while retaining parts of previous solutions. A random crossover involves two parent individuals. Crossover point is chosen randomly for parents. The portions to the right of the crossover points are exchanged two form two offspring’s. There are two .different basic considerations for designing crossover operators

(1) To make less change in crossing over so as to inherit parents’ features as much as possible - all variations of multipoint crossover operator which belong to this class. In two points crossover, two random points are selected and two portions of each parent are exchanged.

(2) To make more change in crossing over so as to explore new patterns of permutation and thereby enhance the search ability, variations of uniform crossover which belong to this class. In order to inherit parents’ features as much as possible, the proposed method adapts three point crossover operator as shown in Fig. 3(b) with a high probability. In order to improve the individual at the same time, it uses adapted one point crossover operator as shown Fig. 3(a) with a low probability.

Australian Journal of Basic and Applied Sciences, 8(16) October 2014, Pages: 378-387

The crossover rate gives the probability a pair of parents that will undergo crossover. Parents that do not crossover transform unchanged as shown in Figure 3(a) into offspring. Parents that do not crossover may still undergo mutation

Randomly select a crossover point: 3
Parent1 (1,1) (2,2) (3,1) (4,2) (5,3) (6,3) (7,1) (8,2)
Parent2 (1,2) (2,1) (3,2) (4,3) (5,3) (6,2) (7,3) (8,1)
Offspring1 (1,1) (2,2) (3,1) (4,3) (5,3) (6,2) (7,3) (8,1)
Offspring2 (1,2) (2,1) (3,2) (4,3) (5,3) (6,2) (7,3) (8,1)

Fig. 3: (a) Random one point crossover.

Randomly select first crossover point: 2
Randomly select second crossover point: 4
Randomly select second crossover point: 6
Parent1 (1,1) (2,2) (3,1) (4,2) (5,3) (6,3) (7,1) (8,2)
Parent2 (1,2) (2,1) (3,2) (4,3) (5,3) (6,2) (7,3) (8,1)
Offspring1 (1,1) (2,2) (3,2) (4,3) (5,3) (6,3) (7,3) (8,1)
Offspring2 (1,2) (2,1) (3,1) (4,2) (5,3) (6,2) (7,1) (8,2)

Fig. 3: (b) Random three point crossover.

(ii) Mutation:

After the each crossover Childs are produced that undergoes mutation with a low probability. The mutation rate indicates the probability that a cell will be changed. As a result, the expected number of mutations per Individual is equal to the mutation rate multiplied by the length of an individual. If a cell is selected to be mutated, then either the task number or the processor number of that cell will be randomly changed is given in Figure4(a).The proposed method includes a new randomized mutation operation in which the processor numbers of two cells interchanged are introduced Figure 4(b) shows an example for the proposed mutation operator. The numbers in bold letters show the selected mutation bits at random. The second individual shows the cells after mutation.

(1,1) (2,2) (3,1) (4,2) (5,**3**) (6,3) (7,1) (8,2)

Fig. 4: (a) Before Mutation.

(1,1) (2,3) (3,1) (4,2) (5,**2**) (6,3) (7,1) (8,2)

Fig. 4: (b) After Mutation.

(iii) Selection:

The selection operator allows the algorithm to take biased decisions favoring good individuals when changing generations. For this, some of the good individuals are replicated, while some of the bad individuals are removed. As a consequence, after the selection, the population is likely to be dominated by good individuals. Starting from a population P_1, this transformation is implemented iteratively by generating a new population P_2 of the same size as P_1, as follows: Initially, the best individual of P_1 is replicated, with only one copy kept in P_1 and the other inserted in P_2. Then, at each iteration, select an individual $S_1 \in P_1$ is selected according to its fitness. Then, S_1 is duplicated into a new individual S_1', and s1 is kept in P_1 while S_1' is inserted into P_2. This process is repeated until P_2 reaches the size of P_1. Notice that, using this scheme, each individual can be selected more than once or not at all. Thus, some individuals are eliminated from generation to generation. Selection operation forms a new population of strings by selecting string in the old population based on their fitness values.

The selection criterion is that the strings with higher fitness value should have higher chance of surviving to next generation. Good strings have high fitness value and hence should be preserved into next generation. The selection operation that is used is the tournament selection method. Tournament Selection method **-** Tournament selection is one of many methods of selection in Genetic Algorithms which runs a "tournament" among a few individuals chosen at random from the population and selects the winner (the one with the best fitness) for crossover easily adjusted by changing the tournament size. If the tournament size is larger, weak individuals have a smaller chance to be selected.

Australian Journal of Basic and Applied Sciences, 8(16) October 2014, Pages: 378-387

D. Methodology and Design:

Genetic Algorithms are different from more normal optimization and search procedures in five ways

- working with a coding of the parameter set, not the parameters themselves
- searching from a population of points, not a single point
- using payoff (objective function) information, not derivatives or other auxiliary knowledge
- using probabilistic transition rules, not deterministic rules and coding.

The majority of optimization methods move from a single point in the decision space to the next using some transition rule to determine the next point. This point-to point method is dangerous as it can locate false peaks in multimodal (many-peaked) search spaces. GAs overcome this by working from a database of points simultaneously (a population of strings), climbing many peaks in parallel. The probability of finding a false peak is reduced compared to methods that go point to point. Many search techniques require auxiliary information in order to work properly. For an example, gradient-based techniques need derivatives able to climb the current peak and other local search procedures like the greedy techniques of combinational optimization require access to most if not all tabular parameters .

GAs have no need for all this auxiliary information, they are blind. To perform an effective search for better and better structures they only require payoff values (objective function values) associated with individual strings. This characteristic makes a GA a more canonical method than many search schemes. Different search problems have vastly different forms of auxiliary information. By not using this auxiliary information a broadly based scheme can be developed. The mechanics of a simple GA are surprisingly simple, involving nothing more complex than copying strings and swapping partial strings. Simplicity of operation and power of effect are two main attractions of the GA approach.

Typical GA designs incorporate a large number of arbitrary human decisions which can potentially bias the algorithm’s performance. For example, Hou’s
method *et al* (1990) uses restricted genetic operators which is found to render parts of the solution space unreachable and the order in which tasks are specified affects the likelihood that two tasks will be crossed over. A novel GA approach for scheduling tasks for parallel execution on multiprocessor systems can be implemented. The proposed Our GA extends the traditional GA in two key ways

There are two different basic considerations for designing crossover operators

(1) To make less change in crossing over so as to inherit parents’ features as much as possible - all variations of multipoint crossover operator belong to this class. In two point crossover two random points are selected and two portions of each parent are exchanged.

(2) To make more change in crossing over so as to explore new patterns of permutation and thereby enhance the search ability, variations of uniform crossover which belong to this class. In order to inherit parents’ features as much as possible, this research uses three point crossover operator with a high probability. In order to improve the individual at the same time, it uses adapted one point crossover operator with a low probability. In the proposed method, a new mutation operation in which the processor numbers of two cells interchanged are introduced. For the selection operation, used the tournament selection method is used. Tournament selection is one of many methods of selection in Genetic Algorithms which runs a "tournament" among a few individuals chosen at random from the population and selects the winner for crossover easily adjusted by changing the tournament size.

Second, a tabu list is added in the algorithm. One of the most important disadvantages of GA is that the execution time needed to run the program will be more as suggested by Reza Akbari et. al. (2009). In most of the cases, the complexity of the algorithm is ignored for the quality of solution. So, in order to reduce the execution time of the algorithm the tabu list is added in the basic algorithm. The tabu list is added in the algorithm in order to reduce the repetition of parents in the next generation whose children already have been selected in the previous generations. In the case of scheduling algorithms, the execution time should also be scalable to solution quality. By using the tabu list restrict the solutions which do not result in the considered value can be restricted. This prevention is because performing operation on such individuals will not produce children whose fitness is better than current generation.

E. Tabu search algorithm:

Glover (1977) described the Tabu Search is an effective meta heuristic algorithm. It provides solution to large scale combinatorial optimization problems in different areas. Tabu Search is a mathematical optimization method, belonging to the class of local search techniques. Tabu Search enhances the performance of a local search method by using memory structures: once a potential solution has been determined, it is marked as tabu so that the algorithm does not visit that possibility repeatedly .The basic principle of TS is that whenever it encounters a local optimum by allowing non-improving moves; cycling back to previously visited solutions is prevented by the use of memories, called tabu lists, that record the recent history of the search, a key idea that can be linked to Artificial Intelligence concepts. The Tabu Search algorithm combines a few simple ideas into a remarkably efficient framework for heuristic optimization. Among the main elements of this framework are

Australian Journal of Basic and Applied Sciences, 8(16) October 2014, Pages: 378-387

- A (usually) greedy local search; the next solution is usually the best not-yet visited solution in the current neighborhood.
- A mechanism (the tabu list) discouraging returns to recently visited solutions.
- A mechanism that changes the solution path (perhaps by a random move) when no progress has been made for a long time.

The basic steps in a Tabu Search algorithm is as follows:

Step 1: Choose an initial solution x
Step 2: Find a subset of N(x) the neighbor of x which are not in the tabu list.
Step 3: Find the best one (x') in N(x).
Step 4: If F(x') > F(x) then set x=x'.
Step 5: Modify the tabu list.
Step 6: If a stopping condition is met then stop, else go to the second step.

F.Tabu List:

A key premise of the Tabu Search is that usually it is better to visit new solutions than to return to recently visited ones. Toward this end, lists of recently visited solutions and/or recently used moves are maintained. Moves are disallowed if they, or the solution they lead to, are found on these lists. Items placed on such a tabu list remain there for a fixed number of iterations (equal to the length of the list). Tabu can be explored either as tasks solution.

- Tasks as tabus - The use of a task as a tabu has the advantage that a separate tabu list is not required. Instead, each task is marked by the time of its most recent move. This simplifies implementation. Reentry to a recently visited solution is now avoided by preventing tasks from being moved back to their previously assigned processor. However, the task is also prohibited from moving to other processors. This reduces the size of the neighborhood, and, perhaps, the efficiency of the search. Also, after a while, all the tasks assigned to a given processor may have moved to a different processor, potentially recreating a tabu solution using different processors. To avoid these situations, it is important not to restrict tasks from being moved for too long.
- Solutions as tabus - When solutions are tabus, sufficient information about previous solutions must be saved, such that entry into similar solutions can be recognized. A special challenge for the present problem is the fact that solutions are not unique. Since the processors are assumed to be identical, it does not matter which processor performs a given set of tasks. All such similar solutions should therefore be detected.
- List length -The length of the tabu list L is a critical parameter in most Tabu Search algorithms. The wrong choice of L may lead to a very inefficient algorithm. Glover (1986) suggested that 7 would be a good value for L. Anderson *et al*. (1993) reported that list measures between 7 and 15 worked well for a path assignment problem. Lokketangen *et al* (1995) cites a case where the most efficient algorithm included more than 200 items in the list. Morton and Pentico (1993) suggest that keeping all solutions on the list works best for scheduling problems. Taillard *et al* (1994) reports on the successful use of lists that randomly change in length at certain points of time. It appears that the most efficient length of the list depends on the problem being solved, and the algorithm being used.

F.A New Tabu List Based Genetic Algorithm:

One of the most important disadvantages of GA is that the execution time needed to run the program will be more. In most of the cases, the complexity of the algorithm is ignored for the quality of solution. So, in order to reduce the execution time of the algorithm a tabu list is added in the basic algorithm. The structure of the proposed algorithm is shown in Figure 5. The tabu list is added in the algorithm in order to reduce the repetition of parents in the next generation whose children already have been selected in the previous generations. In the case of scheduling algorithms, the execution time also should be scalable to solution quality. By using the tabu list, restrict the solutions which do not result in the considered value can be restricted.

A key premise of the Tabu Search is that usually it is better to visit new solutions than to return to recently visited ones. Toward this end, lists of recently visited solutions and/or recently used moves are maintained. Moves are disallowed if they, or the solution they lead to, are found on these lists. Items placed on such a tabu list remain there for a fixed number of iterations (equal to the length of the list). Tabus can be explored either as tasks or solution.

Proposed Hybrid Genetic Algorithm And Tabu Search:

The structure of the proposed algorithm is as shown in the Figure 4. In this algorithm, a tabu list for storing the parents which are selected for the next generation, are maintained. For each time when two individuals are selected for performing crossover and mutation to make the next generation, the algorithm will check whether the selected two individuals are already selected in the previous generations. If the parents are already selected in the previous generation those individuals are discarded in the next generations. If the selected individuals are not in the tabu list, the algorithm will perform the genetic operations for that individuals.

While termination criteria not satisfied **do**
  For each individual in current population do
    Calculate the fitness value of each individual
    Create the intermediate generation as follows
    Add the fittest individual to the intermediate population
  Repeat
    For all the individuals that have fitness > 2/3 average fitness
    Apply all crossovers for this individual and calculate fitness
    Replace best individual in the population and put its parents to a tabu list
    Apply tournament selection to select two individuals
    If the selected individuals not in the tabu list
    Apply multipoint crossover
    Calculate fitness value
    Apply mutation and calculate fitness value
  Until
    the intermediate population size is completed
    Copy intermediate population over current population

Fig. 4: Proposed hybrid Genetic Algorithm and Tabu Search algorithm.

The Figure 5 shows the flowchart representation of the proposed algorithm. As shown in the Figure 5, for each generation, the algorithm will check whether the selected individuals are in the tabu list or not for performing genetic operation. If it is present in the list it will select another set of individuals for the next generation. If its not present those individuals will constitute the formation of next generation

RESULTS AND DISCUSSION

In order to evaluate the performance of the proposed GA, the algorithm is coded in MATLAB code. The values of parameters to be used in the GA are as follows:

- Population size = 10,
- Crossover probability = 0.8,
- Mutation probability = 0.05,
- Selection method = Tournament selection,
- The maximum number of generations =1000.

Five images named ‘‘LENA’’ ‘‘HOUSE’’ ‘‘HUNTER’’ ‘‘BUTTERFLY’’ and ‘‘MAP’’ are used for conducting our experiments. These original test images are adopted for simulation using the proposed method for image segmentation.

The Tsalli’s index ‘q’ is a real number to state the nonextensivity of the systems. In image processing, ‘q’ is effective to evaluate the pixels’ long-range correlations so its value is adjustable and can play as a tuning factor in the segmentation process. In this research the value of q=0.7 is estimated based on the empirical value.

The HGA and other three multilevel thresholding methods that are ABC, PSO, and GA are implemented for the purpose of comparisons. Table 1 summarizes the optimum thresholds of the five test images. This result reveals that the segmentation results depend heavily on the objective function that is selected. Similarly, all test images reveal the fact that thresholding results determined by proposed method found qualitatively better compared to other techniques. It is also found that thresholding results are better qualitatively when the number of thresholds increased.As the steps enumerated above, the PCG method will be invoked (Step 6) if and when: T^* of the ABC phase in the present iteration should be less than the previous. Thereby the individual associated with the T^* will be deterministically guided by the PCG method using the gradient information to fine-tune the currently best-explored region. The termination is done when there is no improvement in T^* for a preset number of iterations.

The popular performance indicator, peak signal to noise ratio (PSNR), is used to compare the segmentation results using the proposed and other threshold techniques of this paper. For the sake of completeness we define PSNR, measured in decibel (dB) as

$$PSNR(in\ dB) = 20log_{10}\left(\frac{255}{RMSE}\right) \tag{14}$$

$$RMSE = \sqrt{\frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[I(i,j) - \tilde{I}(i,j)]^2}$$

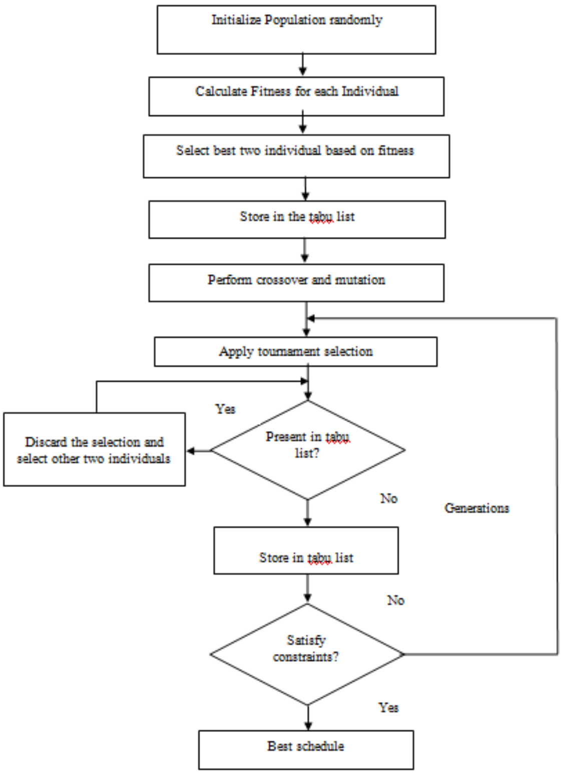

Fig. 5: Flowchart for the proposed hybrid Genetic Algorithm and Tabu Search algorithm.

Table 1: Comparison of Optimum threshold values.

Test image	m	Optimum threshold values			
		GA	ABC	PSO	HGA
Lena	2	98,172	120,164	120,164	120,164
	3	79,118,169	81,124,178	110,149,187	98,159, 181
	4	73,103,129,185	85,124,161,193	85,118,164,200	86,120, 151,205
	5	56,106,153,175,215	76,108,136,164,193	86,117,142,166,196	95,130, 152,173,200
House	2	79,173	87,145	87,145	87,145
	3	56,127,182	88,133,199	90,133,199	82,123,177
	4	61,93,152,212	67,105,146,189	70,112,152,189	73,111,151,189
	5	49,92,131,170,203	66,95,121,155,200	70,104,134,160,212	60,99,114,158,198
Hunter	2	98,170	94,137	94,137	94,137
	3	70,117,170	82,118,171	83,143,174	87,147,173
	4	61,91,150,193	71,110,142,182	78,109,143,187	90,119,150,191
	5	51,91,133,172,203	65,93,123,150,182	70,103,139,174,198	79,114,144,174,198
Butterfly	2	78,150	97,136	97,136	97,136
	3	84,115,167	99,135,197	100,135,185	89,124,169
	4	63,94,150,210	95,120,144,189	89,122,143,178	94,121,141,179
	5	50,90,132,165,196	89,114,141,170,213	70,107,134,162,189	70,119,140,170,214
Map	2	98,172	114,176	114,176	114,176
	3	88,147,194	84,142,198	90,131,183	80,145,172
	4	56,109,152,206	73,113,156,203	78,121,158,189	80,117,157,199
	5	49,92,131,170,202	75,112,147,174,206	79,113,142,170,191	91,118,144,174,206

where and M, N are the size of image, I is the original image and I is the thresholded image at a particular level.

Table 2 summarizes the PSNR values for all the four techniques used in this paper. A higher value of PSNR indicates a better quality of thresholding. For all the test images, the proposed method proves to be better than ABC PSO and GA methods. Similarly, the standard deviation σ is given by,

$$\sigma = \sqrt{\frac{1}{t}\sum_{i=1}^{t}(s_i - \mu)^2} \tag{15}$$

Australian Journal of Basic and Applied Sciences, 8(16) October 2014, Pages: 378-387

where 't' is the number of runs for each algorithm and in this research it is 50. Note that 'S_i' is the best objective value obtained by the i-th run of the algorithm and μ is the mean. Both best objective function value and standard deviation are summarized in Table 3.

Table 2: Comparison of PSNR values for different methods.

Test Image	m	PSNR (dB)			
		HGA	ABC	PSO	GA
Lena	2	18.4215	15.2419	15.2419	15.2419
	3	21.2332	17.4715	17.1425	16.9455
	4	22.1332	19.5070	19.4324	19.0207
	5	23.2841	20.9916	20.5637	19.8703
House	2	18.6233	12.9865	12.9865	12.9865
	3	20.6930	14.0213	13.8104	13.6918
	4	22.4978	16.8884	16.4428	16.1794
	5	23.9747	17.5635	16.7719	16.5772
Hunter	2	16.5926	11.3848	11.3848	11.3848
	3	18.0245	14.5772	14.5135	14.0724
	4	19.2790	16.2874	15.4496	14.1926
	5	19.8811	17.3380	16.6426	15.6197
Butterfly	2	16.8115	13.0516	13.0516	13.0516
	3	19.1106	18.1337	17.8316	17.2964
	4	21.4774	20.0356	18.9792	18.8382
	5	23.7354	21.9096	21.4406	20.2055
Map	2	18.1283	16.6045	16.6045	16.6045
	3	19.6091	18.4286	18.0419	16.2161
	4	21.3499	20.6499	19.7997	19.7340
	5	23.6609	22.1638	21.8968	21.5746

Table 3: Comparison of RABC and ABC algorithms for best Objective function and Standard Deviation.

Test image	m	Best objective function value		Standard Deviation	
		HGA	ABC	HGA	ABC
Lena	2	0.885012	0.888888	0.0000	0.0000
	3	1.223113	1.296294	0.0000	1.68e-006
	4	1.656062	1.654319	0.0000	2.13e−006
	5	1.991421	1.995881	0.0000	3.43e-006
House	2	0.888886	0.888771	0.0000	0.0000
	3	1.296292	1.296192	0.0000	1.70e−006
	4	1.654316	1.653630	0.0000	2.65e−005
	5	1.995879	1.994227	4.0605e−008	4.12e−004
Hunter	2	0.888930	0.888920	0.0000	0.0000
	3	1.296295	1.296275	1.2303e−009	3.69e−006
	4	1.654320	1.654228	1.6831e−008	5.66e−005
	5	1.995883	1.995760	2.727e−007	7.68e−005
Butterfly	2	0.887741	0.888888	0.0000	0.0000
	3	1.295414	1.296286	0.0000	1.68e−006
	4	1.653312	1.654319	0.0000	1.85e−005
	5	1.988451	1.995883	2.6223e−008	6.74e−005
Map	2	0.888885	0.881216	0.0000	0.0000
	3	1.296192	1.283982	0.0000	2.80e−006
	4	1.654020	1.589902	1.0904e−010	3.69e−006
	5	1.993884	1.928422	2.9953e−008	9.63e−006

A lower value of σ indicates a better quality of thresholding. From σ values listed in Table 3, it is evident that the proposed method outperforms other methods, which shows a better stability of the proposed method.

Conclusion:

This article presents an extensive study on the application of a hybrid algorithm integrating two powerful metaheuristic techniques for multilevel thresholding for image segmentation problem. As seen from the experimental results, non-extensive entropy based MTET image thresholding using Hybrid HGA algorithm is effective for image segmentation applications. The proposed hybrid method yielded a near optimum (compared to exhaustive search) threshold value for q=0.7, the segmentation results are promising. It is demonstrated that the simulation results obtained using the hybrid HGA method are superior to that of ABC, PSO and GA methods in terms of producing quality thresholds. The validity and stability of the method is justified both qualitatively and quantitatively The segmentation results are promising and it encourage researches for applying the HGA algorithm calculation to complex and constant picture dissection issue, for example, the immediate target distinguish problems and the unpredictable record examination. It is promising to encourage further research for applying this proposed algorithm to complex problems of image processing and pattern recognition.

REFERENCES

Pal, N.R., S.K. Pal, 1993. A review on image segmentation techniques. Pattern Recognition, 26(1): 1277-1294.

Otsu, N., 1979. A threshold selection method from gray-level histograms, IEEE Transactions on Systems Man Cybernetics, 9(2): 62-66.

Kapur, J.N., P.K. Sahoo, A.K.C. Wong, 1985. A new method for gray-level picture thresholding using the entropy of the histogram, Computer Vision Graphics Image Processing, 29(2): 273-285.

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

Zhang, R., J. Liu, 2006. Underwater image segmentation with maximum entropy based on Particle Swarm Optimization (PSO). In Proceedings of the first international multi-symposiums on computer and computational Sciences (IMSCCS'06): 360-363.

Horng, Ming-Huwi, 2011. Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation, Expert Systems with Applications, 38(11): 13785-13791.

Liang, J.J., A.K. Qin, P.N. Suganthan, S. Baskar, 2006. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEETransactions on Evolutionary Computation, 10(3): 281-295.

Dennis, Jr, E. John and J. Jorge Moré, 1977. "Artificial Bee Colony Optimization methods, motivation and theory." SIAM review, 19(1): 46-89.

Osuna-Enciso, Valentín, Erik Cuevas and HumbertoSossa, 2013. A comparison of nature inspired algorithms for multi-threshold image segmentation." Expert Systems with Applications, 40(2): 1213-1219.

Yin, P.Y., 2007. Multilevel minimum cross entropy threshold selection based on particle swarm optimization, Applied Mathematics and Computation, 184(4): 503-513.

Lin, Q., C. Ou, 2012. Tsallis entropy and the long-range correlation in image thresholding. Signal Processing, 92(12): 2931-2939.

Peng-Yeng Yin, 1999. A fast scheme for multilevel thresholding using genetic algorithms, Signal Processing, 72(3): 85-95.

Cantu-Paz, E., 1998. A survey of parallel genetic algorithms. Calculateurs Parallels, 5(2): 141-171.

Maitra, M. and A. Chatterjee, 2008. A hybrid cooperative-comprehensive learning based pso algorithm for image segmentation using multilevel thresholding. Expert Systems with Applications, 34: 1341-1350.

Qi Ming-yao, Miao Li-xin, Zhang Le, Xu Hua-yu, 2008. A new Tabu Search heuristic algorithm for the Vehicle Routing Problem with Time Windows, International Conference on Management Science and Engineering, ICMSE 15th Annual Conference Proceedings, 3: 1648-1653.