Multiobjective Genetic Algorithm for Image Thresholding

Layla Tahri and Mohamed Wakrim

IbnZohr University, Faculty of Sciences, EMMS, Agadir, Morocco
layla.tahri@gmail.com, m.wakrim@uiz.ac.ma

Abstract. In this paper we present a new image thresholding method based on a multiobjective Genetic Algorithm using the Pareto optimality approach. We aim to optimize multiple criteria in order to increase the segmentation quality. Thus, we've adapted the well-known Non Domination Sorting Genetic Algorithm [1] for this purpose so that it takes into consideration the contribution of the objective functions in improving the reproduction step and then improving the optimal Pareto front of solutions. Our method was tested against NSGAI algorithm and has shown effectiveness and convergence speed.

Keywords: Evolutionary approach, Genetic algorithms, Image segmentation, Image thresholding, Multiobjective optimization, Pareto optimization.

1 Introduction

Image segmentation has been approached from different ways of knowledge such as graph theory, statistics (multivariate analysis), artificial neural networks, fuzzy set theory and other areas. One of the most popular methods in this field is to perceive image segmentation as an optimization problem, where the best segmentation of a given image is achieved by optimizing one or more objective functions. In this sense, evolutionary algorithms have captivated since their apparition in 1995, the attention of the practitioners of optimization all over the world, due to their elevated degree of robustness, convergence speed and accuracy.

Moreover, they've shown effectiveness in solving complicated optimization problems and overcame the well-known problem of being trapped in local optima for the classical approaches.

To date, many attempts of applying evolutionary algorithms to find the “best” clustering of an image have been achieved. However, most of them present the disadvantage of being dependent on many parameters making their adaptation to different problems harder. This remains an important issue when it comes to apply the algorithm to different datasets of images.

To overcome this problem we propose a new algorithm based on an adapted version of the Non Sorted Genetic Algorithm II (NSGAI), using the multilevel thresholding extended from Otsu's method and the Shannon Entropy as objective functions.

This work is organized as follows: in the next section we present the extension of the classical Otsu's thresholding method to the multilevel thresholding. In the section 3, we introduce a modified Shannon entropy measure. Then in the 4th section we present the formulation of image segmentation problem as a multiobjective optimization problem. The non-dominated sorting Genetic Algorithm is then introduced in section 5. And finally we present our new method.

2 Otsu's Thresholding algorithm

The Otsu's method aims to automatically segment an image into two classes based on the shape of image histogram[2]. The extension of this method to the multilevel thresholding is given by the following equations:

Let $\{t_1, \dots, t_N\}$ be the set of thresholds to evaluate. Finding the best set of thresholds is performed by maximizing the inter-classes variance:

$$\{t_1^*, \dots, t_N^*\} = \text{Maximize}\{\sigma_B^2(t_1, t_2, \dots, t_N)\}, 1 \leq t_1 < t_2 < \dots < t_N \leq \max L \quad (1)$$

where $\max L$ is the maximum value in the range of the thresholds.

and

$$\sigma_B^2 = \sum_{k=1}^{N+1} \omega_k (\mu_k - \mu_T)^2 \quad (2)$$

is the inter-classes variance,

with ω_k and μ_k are computed as follows:

$$\omega_k = \sum_{t_{k-1}+1}^{t_k} p_i, k \in [1, N] \quad (3)$$

and

$$\mu_k = \sum_{t_{k-1}+1}^{t_k} i * p_i, k \in [1, N] \quad (4)$$

then

$$\mu_T = \sum_{k=1}^N \omega_k \mu_k \quad (5)$$

This algorithm is widely used, easy to implement and gives satisfactory results in terms of clustering quality. Its main disadvantage is the important computational time it needs for the execution. In fact, its exhaustive search involves $(L - N)^N$ possible combinations. This increases considerably with the number of classes in real life images.

Thus, our method has been developed to overcome this problem by combining a modified Genetic algorithm based on NSGA II and the multi-level thresholding method shown above.

3 Shannon Entropy

3.1 Bi-level Thresholding

Shannon entropy [3],[4] and [5] is a statistical criterion used to find the best threshold for segmenting an image into two classes and is defined as follows:

$$H_T(t) = -\sum_{i=1}^t \frac{p_i}{P_t} e^{1-\frac{p_i}{P_t}} - \sum_{i=t+1}^{maxL} \frac{p_i}{P_t} e^{1-\frac{p_i}{P_t}} \quad (6)$$

where t is the gray level, p_i is the frequency of t in the image, P_t is the prior probability of t and $maxL$ is the highest gray level in the image. Maximizing this entropy gives the best threshold for the image segmentation into two classes.

$$t^* = \text{ArgMax}\{H_T(t), t \in [1..maxL]\} \quad (7)$$

3.2 Multilevel Thresholding

The Shannon entropy can be extended to multilevel thresholding as follows:

$$H_T(t_1, \dots, t_N) = -\sum_{k=1}^N \sum_{i=t_{k-1}+1}^{t_k} \frac{p_i}{P_{t_k}} e^{1-\frac{p_i}{P_{t_k}}} \quad (8)$$

where $1 \leq t_1 < t_2 < \dots < t_{k-1} < t_k < \dots < t_N \leq maxL$

and the best set of thresholds is obtained by maximizing this entropy:

$$\{t_1^*, \dots, t_N^*\} = \text{ArgMax}\{H_T(t_1, \dots, t_N)\} \quad (9)$$

4 Formulation of Image Segmentation Problem as a Multiobjective Optimization Problem:

In a problem where there are many (possibly conflicting) objectives to be optimized, there is no accepted notion as optimal solution. Hence, comparing between solutions becomes difficult. Generally, the optimizing methods produce a set of solutions of equivalent quality. The “best” solution is therefore subjective and depends on the decision maker.

The multi-objective optimization problem can be formally stated as:

Find the vector $t^* = (t_1^*, t_2^*, \dots, t_N^*)$ which will satisfy the following properties:

Optimizes the vector function $[f_1(t), f_2(t), \dots, f_{M-1}(t), f_M(t)]$

and satisfies the inequality and inequality constraints:

$$g_i(t) \leq 0, \quad i = 1, 2, \dots, k \quad (10)$$

and

$$h_i(t) = 0, \quad i = 1, 2, \dots, p \quad (11)$$

The constraints of equations (10) and (11) define the region Ω of admissible solutions and the vector t^* represents one of the optimal solutions in the set Ω .

In general, the uniqueness of the optimal solution of a multi-objective optimization problem is not satisfied, and poorly defined. The concept of Pareto optimality is as an appropriate response to this case.

For a minimization problem, the formal definition of the Pareto optimality can be given as follows:

A decision vector t^* is said to be pareto optimal if and only if there is no other vector t that dominates t^* . i.e there is no t such that :

$$\forall i \in \{1, 2, \dots, M\} : f_i(t) \leq f_i(t^*) \quad (12)$$

and

$$\exists j \in \{1, 2, \dots, M\} : f_j(t) < f_j(t^*) \quad (13)$$

this property (*i.e. t^* dominates t*) is written as: $t^* > t$

5 Non Domination Sorting Algorithm II (NSGAI)

There are different approaches to solve multiobjective optimization problems such as aggregating, population based non-pareto and pareto based techniques. The NSGAI belongs to the last category. Its main advantages of this algorithm are convergence speed due to its non-domination sorting applied to the generations and the elitism.

The pseudo code of the NSGAI is given as follows:

Input : the image to be clustered and the number of clusters.

Output: Pareto Front of optimal solutions

Initialization of the population

Evaluation of the objective functions

Non-dominated sort of the population

Selection of the parents

Crossover or mutation

While not (stopping criteria is reached)

 Evaluation of the objective functions

 Merge (population and offspring)

 Fronts $\leftarrow$ Non dominated sort of the merged set

 Parents $\leftarrow$ ensemble vide ; FrontL $\leftarrow$ ensemble vide

 For each Front do the follow

 Compute and assign crowding distance to each individual in the i th Front

 If (size(parents) + size(Front i)) > size(population)

 FrontL $\leftarrow$ Front i ; Break();

 Else

 Parents $\leftarrow$ merge(parents, Front i);

 End

 End

 If (size(parents) < size(population))

 FrontL $\leftarrow$ SortByRankAndDistance(FrontL);

 For (P1 to size(population) - size(FrontL))

 Parents $\leftarrow$ P1

 End

 End

 Selected $\leftarrow$ SelectParentsByRankAndDistance(Parents, size(population))

 Population $\leftarrow$ offspring

 Offspring $\leftarrow$ CrossoverOrMutation(Selected)

End

Return Children

6 The Proposed Multiobjective Method

In order to reach a rapid and non-parametric algorithm, we adapted the NSGAII algorithm at the following levels:

6.1 Initialization

Most of NSGA algorithms initialize the population randomly. They might have a set of individuals in the same range, which might lead to a non-well distributed population and therefore affects the next iterations. i.e a threshold variable t_i is initialized as $t_i \in [1, \dots, \max L]$, where $\max L$ is the highest level in the image.

In our method, we specify a range for each decision variable depending of the number of clusters. i.e knowing that the number of clusters is N , we divide the range $[1, \dots, \max L]$ to $N+1$ equal intervals. Then, each decision variable t_i is initialized within the appropriate interval such as: $t_i \in [t_{k-1} + 1, t_k]$, $k \in [1, N + 1]$

6.2 Selection

In most of NSGA algorithms, the selection process chooses randomly from the sorted population, the parents susceptible of performing the reproduction process. It might lead to redundancy of the parent sets in the reproduction pool.

For this reason, we've added a constraint which implies that the parents should be from different Pareto fronts, but having the best scores in terms of objective functions regarding other individuals from the same front. This can be written as follows:

Choose $Parent_i$ and $Parent_j$ such as:

$$\begin{aligned} &Parent_i \in Front_i \quad \text{and} \quad Parent_j \in Front_j \quad \text{for each } i \neq j \\ &\text{with} \quad f_j(Parent_i) = ArgMax \{f_j(Parent_{i,k})\}, \\ &\text{where} \quad Parent_{i,k} \in Front_i \quad \text{and} \quad j \in [1, \dots, M] \end{aligned} \quad (14)$$

6.3 Crossover

Our third intervention was at the crossover step. In fact, the NSGA algorithms use the SBX (Simulated Binary Crossover) β [6] in the process of generating offspring chromosome as described below:

$$child_1 = 1/2 \left(((1 + \beta) * Parent_1) + ((1 - \beta) * Parent_2) \right) \quad (15)$$

$$child_2 = 1/2 \left(((1 - \beta) * Parent_1) + ((1 + \beta) * Parent_2) \right) \quad (16)$$

Therefore, in order to take into consideration the weight of the parent's objective functions in improving the Pareto solutions, we use the following equations:

$$child_1 = 1/2 * \left(\left(((1 + \omega_{11}) + (1 - \omega_{12})) * Parent_1 \right) + \left(((1 + \omega_{21}) + (1 - \omega_{22})) * Parent_2 \right) \right) \quad (17)$$

$$child_2 = 1/2 * \left(\left(((1 - \omega_{11}) + (1 + \omega_{12})) * Parent_1 \right) + \left(((1 - \omega_{21}) + (1 + \omega_{22})) * Parent_2 \right) \right) \quad (18)$$

where

$$\omega_{11} = \frac{f_{11}(t)}{f_{11}(t) + f_{12}(t)} \text{ and } \omega_{12} = \frac{f_{12}(t)}{f_{11}(t) + f_{12}(t)} \quad (19)$$

represents the contribution weights of objective functions for the first parent $Parent_1$ and

$$\omega_{21} = \frac{f_{21}(t)}{f_{21}(t) + f_{22}(t)} \text{ and } \omega_{22} = \frac{f_{22}(t)}{f_{21}(t) + f_{22}(t)} \quad (20)$$

are the contribution weights of objective functions for the second parent $Parent_2$

Noting also that f_{ij} is the j^{th} objective function for the i^{th} individual.

6.4 Mutation

The NSGA uses a constant η_m called the polynomial mutation spread factor that needs to be tuned. Usually it takes 20 as value. Thus, in order to overcome this parameter-dependence, we mutate the parent as described below:

For each component p_j of the parent do

Convert p_j to the binary format.

Randomly select the i^{th} bit to change.

Mutate if as follows

If $p_j(i) = 0$ then

$p_j(i) = 1$

Else

$p_j(i) = 0$

Re-convert the component p_j to the original number format.

7 Results and Discussion

We've chosen a set of test images from the Berkeley Database [7] in order to evaluate and compare the performance of both NSGAII and the Adapted NSGAII algorithms. This choice is based on the nature of the images: the goat and elephant images present a weak contrast between the foreground objects and the background. In the opposite, the peppers and the water skier have an important contrast regarding the background image.

Table 1. Original images from Berkeley database

Fig. 1. Goat image

Fig. 2. Elephant image

Fig. 3. Water skier image

Fig. 4. Peppers image

Table 2. Comparison of the number of input and output parameters in NSGAII and our proposed method

Algorithm	Input parameters	Output
NSGA II	Image - Number of clusters Crossover spread factor Mutation spread factor Pool size - Tournament size	Pareto front
The adapted NSGA II	Image Number of clusters Pool size Tournament size	Pareto front

Table 3. Comparison between the Otsu’s variance, the modified Shannon entropy and the CPU time in NSGAII and our proposed method

Algorithm	Image	Number of clusters	Otsu’s Variance	Modified Shannon Entropy	CPU time (in seconds)
NSGAII	Goat	3	2445.8986	-10.1730	6.4176
	Elephant	3	1337.7215	-8.9751	6.2460
	Peppers	6	3243.9823	-22.6925	8.8013
	Water skier	4	1224.2536	-16.4316	7.8684
The adapted NSGAII	Goat	3	2338.6183	-9.7920	1.9344
	Elephant	3	1424.8782	-10.3751	1.3884
	Peppers	6	3535.4301	-18.3782	2.9812
	Water skier	4	1312.6678	-13.0296	2.0280

In table 2, we can see that the NSGA II requires more input parameters than the adapted version we proposed. Moreover, those parameters need tuning in order to be adjusted for different sets of images.

In table 3, we present the mean values of the Otsu’s Variance, the modified Shannon entropy and the CPU execution time for each algorithm over 50 runs and 100 generation.

The overall performance is close with better results for our algorithm for the peppers and Water skier images which can be explained by the introduction of the objective function’s weights in the crossover process.

In terms of execution time, our algorithms performs much better than the NSGAII, this might be owed to the elimination of the spread crossover and mutation factors which has led to less calculations and therefore required less computational time.

Table 4. Results of image segmentation performed by NSGAII and Adapted NSGAII

NSGAII results

The Adapted NSGAII results

Fig. 5. Segmented goat image using the NSGAII **Fig. 6.** Segmented goat image using the Adapted NSGAII

Table 4. (Continued)

Fig. 7. Segmented elephant image using the NSGAII

Fig. 8. Segmented elephant image using the Adapted NSGAII

Fig. 9. Segmented peppers image using the NSGAII

Fig. 10. Segmented peppers image using the Adapted NSGAII

Fig. 11. Segmented water skier image using the NSGAII

Fig. 12. Segmented water skier image using the Adapted NSGAII

Although the goat and elephant image present less contrast, the Adapted NSGAII performed well and gave a good segmentation result.

Furthermore, the proposed adapted algorithm shows better performance for the images with important contrast compared to the segmentation results of the NSGAII. This is consistent in general with the Otsu's variance and the modified Shannon entropy results in table 2.

8 Conclusion

In this article, a new image segmentation based on multiobjective genetic algorithm has been proposed. The question has been formulated as a multiobjective optimization problem.

The objective functions used were the Otsu's variance and a modified Shannon entropy measure, both extended to multilevel thresholding.

The algorithm has been applied to several images presenting different characteristics; the results were presented and compared to the NSGAII ones.

The good performance of our method for different types of images at the level of objective functions, convergence speed and independence of parameter tuning; shows that it might be motivating to use this algorithm in the field of image segmentation.

References

1. Deb, K., Pratap, A., Agarwal, S., Meyarivan, T.: A fast and elitist multi-objective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation* 6(2), 182–197 (2002)
2. Otsu, N.: A Threshold Selection Method from Gray-Level Histograms. *IEEE Trans. on Syst., Man and Cyb.* 9(1), 62–66 (1979)
3. Kapur, J.N., Sahoo, P.K., Wong, A.C.K.: A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision, Graphics and Image Processing* 29, 273–285 (1985)
4. Pal, N.K., Pal, S.K.: Entropy: A new definition and its applications. *IEEE Trans. Syst. Man. Cybern.* 21, 1260–1270 (1991)
5. Nakib, A.: Conception de métaheuristiques d'optimisation pour la segmentation d'images. Application à des images biomédicales, pp. 9–10 (2008)
6. Deb, K., Kumar, A.: Real-coded Genetic Algorithms with Simulated Binary Crossover: Studies on Multimodal and Multiobjective Problems. *Complex Systems*, 431–454 (1995)
7. The Berkeley database site,
<http://www.oracle.com/technetwork/database/berkeleydb/overview/index.html>