



**HAL**  
open science

# Social spiders optimization and flower pollination algorithm for multilevel image thresholding: A performance study

Salima Ouadfel, Abdelmalik Taleb-Ahmed

## ► To cite this version:

Salima Ouadfel, Abdelmalik Taleb-Ahmed. Social spiders optimization and flower pollination algorithm for multilevel image thresholding: A performance study. *Expert Systems with Applications*, 2016, 55, pp.566-584. 10.1016/j.eswa.2016.02.024 . hal-03426984

**HAL Id: hal-03426984**

**<https://uphf.hal.science/hal-03426984v1>**

Submitted on 14 Jan 2025

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Social spiders optimization and flower pollination algorithm for multilevel image thresholding: A performance study

Salima Ouadfel <sup>a, \*</sup>, Abdelmalik Taleb-Ahmed <sup>b</sup>

<sup>a</sup> NTIC Faculty, University of Constantine 2-Abdelhamid Mehri, Constantine, Algeria

<sup>b</sup> LAMIH UMR CNRS 8201 UVHC, Laboratory of Industrial and Human Automation, Mechanics and Computer Science Université de Valenciennes et du Hainaut Cambrésis, Le mont Houy, 59313 Valenciennes Cedex 9, France

## A B S T R A C T

In this paper, we investigate the ability of two new nature-inspired metaheuristics namely the flower pollination (FP) and the social spiders optimization (SSO) algorithms to solve the image segmentation problem via multilevel thresholding. The FP algorithm is inspired from the biological process of flower pollination. It relies on two basic mechanisms to generate new solutions. The first one is the global pollination modeled in terms of a Levy distribution while the second one is the local pollination that is based on random selection of local solutions. For its part, the SSO algorithm mimics different natural cooperative behaviors of a spider colony. It considers male and female search agents subject to different evolutionary operators. In the two proposed algorithms, candidate solutions are firstly generated using the image histogram. Then, they are evolved according to the dynamics of their corresponding operators. During the optimization process, solutions are evaluated using the between-class variance or Kapur's method. The performance of each of the two proposed approaches has been assessed using a variety of benchmark images and compared against two other nature inspired algorithms from the literature namely PSO and BAT algorithms. Results have been analyzed both qualitatively and quantitatively based on the fitness values of obtained best solutions and two popular performance measures namely PSNR and SSIM indices as well. Experimental results have shown that both SSO and FP algorithms outperform PSO and BAT algorithms while exhibiting equal performance for small numbers of thresholds. For large numbers of thresholds, it was observed that the performance of FP algorithm decreases as it is often trapped in local minima. In contrary, the SSO algorithm provides a good balance between exploration and exploitation and has shown to be the most efficient and the most stable for all images even with the increase of the threshold number. These promising results suggest that the SSO algorithm can be effectively considered as an attractive alternative for the multilevel image thresholding problem.

### Keywords:

Multilevel thresholding  
Optimization  
Social spider optimization  
Flower pollination algorithm  
Particle swarm optimization  
Bat algorithm

## 1. Introduction

Image segmentation is an important task for meaningful analysis and interpretation of acquired image in many fields. Many segmentation techniques have been proposed in the literature. Among all the existing techniques, thresholding method is one of the most popular due to its simplicity, robustness and accuracy (Pal & Pal, 1993; Sahoo, Soltani, & Wong, 1988).

Image thresholding is a process by which the whole image is segmented into one or several regions on the basis of one or more threshold values (Sahoo et al., 1988). Bi-level thresholding refers to the process of dividing an input image into two classes, such as the

background and the object of interest. An extension to more than two classes leads naturally to multilevel thresholding where more than two distinct objects or regions need to be depicted. In bi-level thresholding, image pixels with gray values greater than a certain value or threshold are classified as object pixels while the others are classified as background pixels. In multilevel thresholding, image pixels belonging to the same class have gray levels within a specific range defined by several thresholds (Maitra & Chatterjee, 2008). A great interest is devoted to multilevel thresholding within the community of researchers because of the intrinsic nature of real world images where it is required to identify several objects and for which bi-level thresholding fails to produce satisfactory results.

The thresholds can be obtained with a local or a global process (Sezgin & Sankur, 2004). In local thresholding, a different threshold is assigned to each part of the image. In global thresholding, a

\* Corresponding author. Tel.: +213 541552999.

E-mail addresses: [souadfel@yahoo.fr](mailto:souadfel@yahoo.fr), [ouadfel@gmail.com](mailto:ouadfel@gmail.com) (S. Ouadfel), [taleb@univ-valenciennes.fr](mailto:taleb@univ-valenciennes.fr) (A. Taleb-Ahmed).

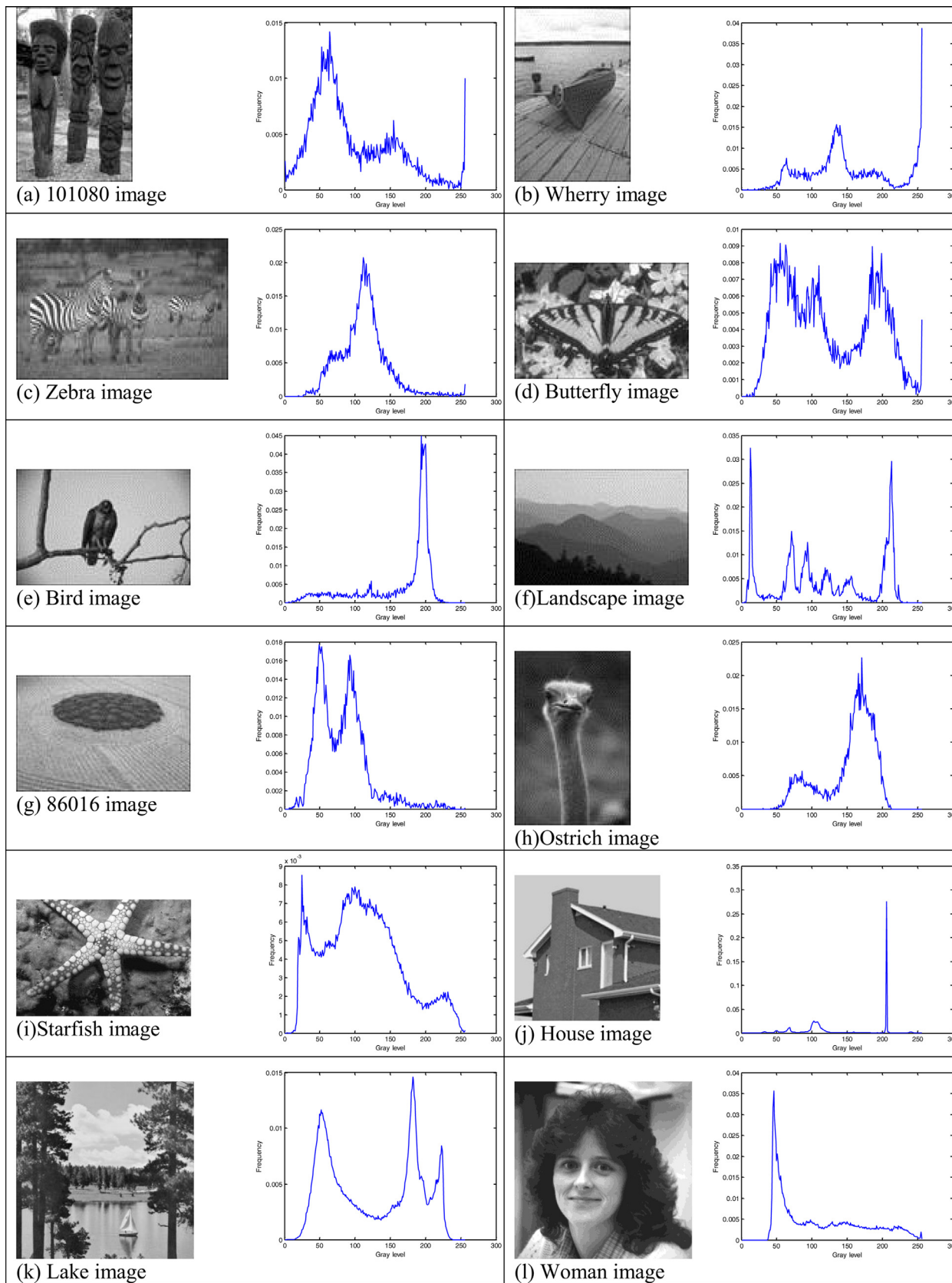


Fig. 1. Real images used in the paper with their histograms.

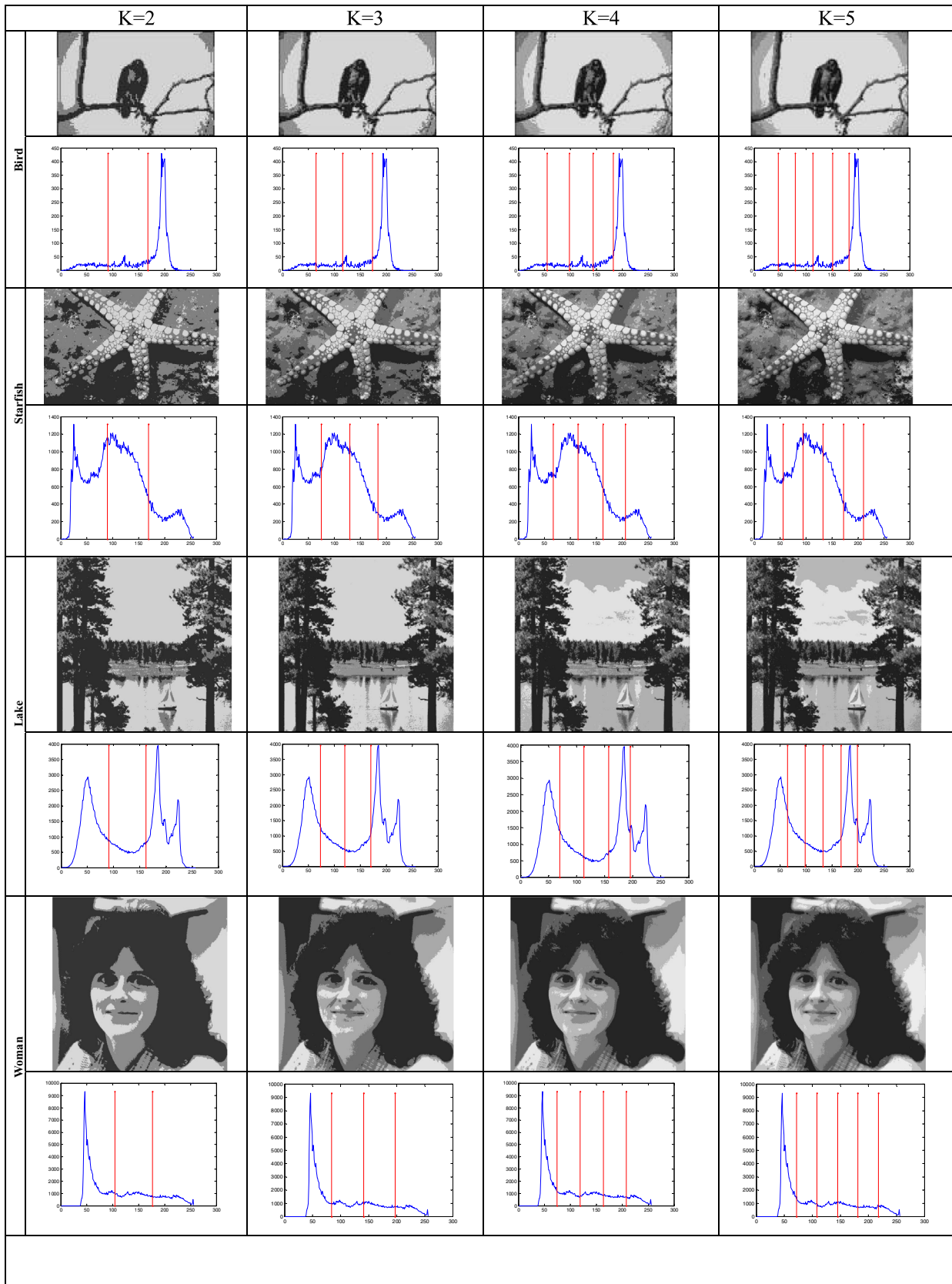


Fig. 2. Results after applying the SSO algorithm using Kapur's entropy over test images.

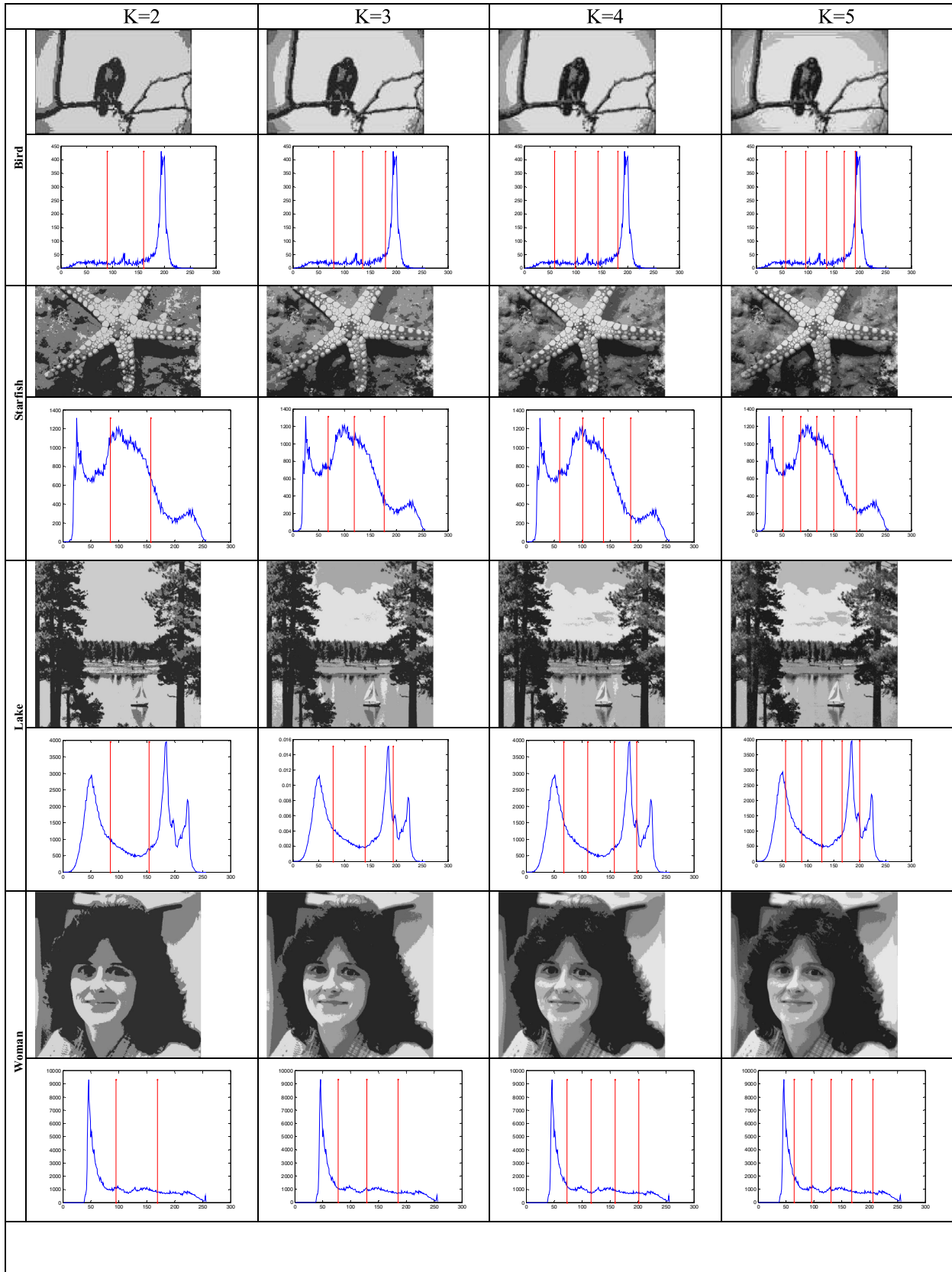


Fig. 3. Results after applying the SSO algorithm using between-class variance over test image.

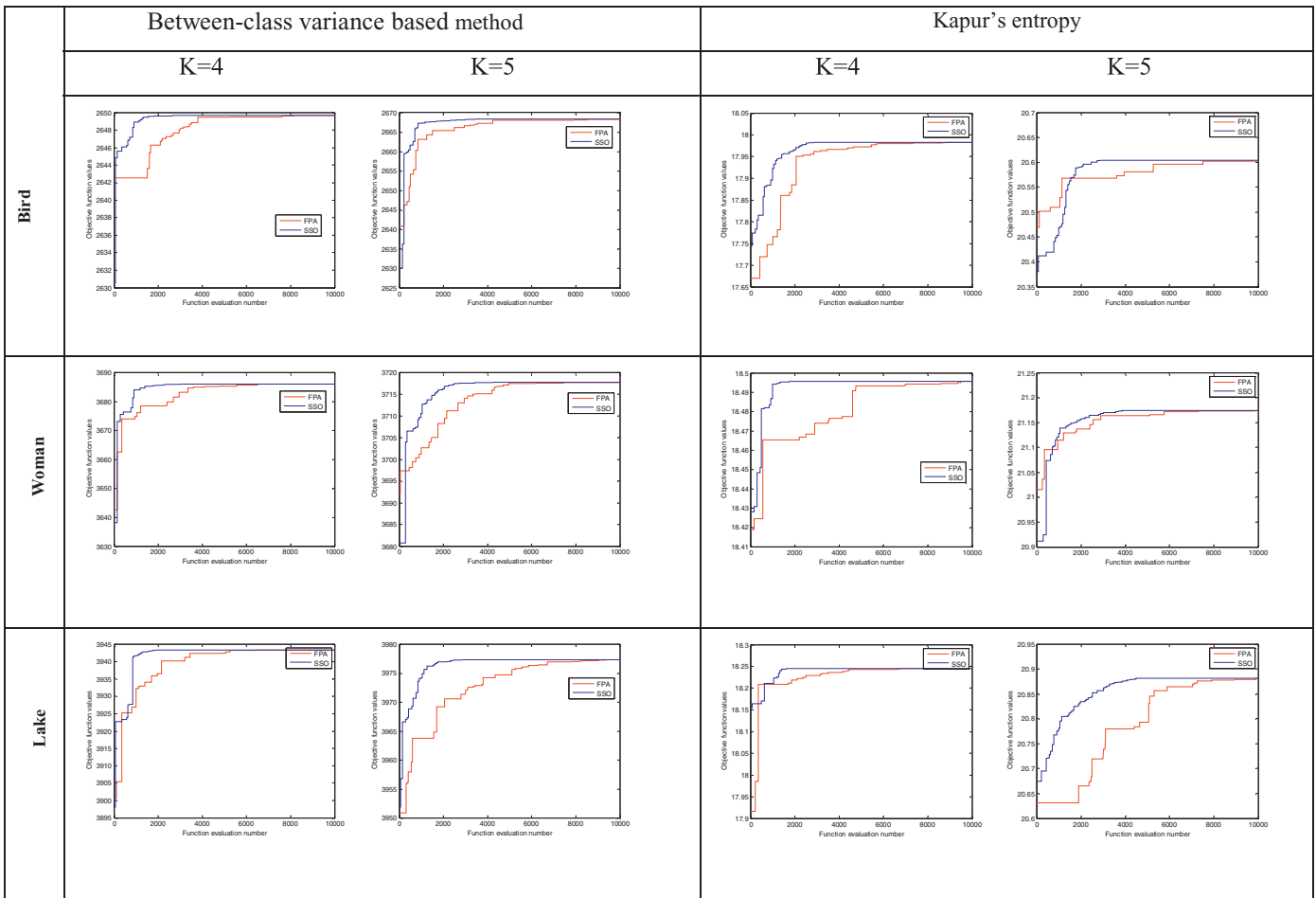


Fig. 4. The convergence curves for SSO and FP algorithms using between-class variance based method and Kapur's entropy.

single global threshold is derived from the whole image. Basically, approaches to find the suitable thresholds can be either parametric or nonparametric (Hammouche et al., 2010). In the parametric approaches, the statistical parameters of the classes in the image are estimated. These methods are computationally expensive, and their performance is very sensitive to the initial conditions. In the nonparametric approaches, the number of thresholds is given in advance and the optimal thresholds are selected by maximizing some criteria among which the two more popular and successful are the Otsu's criterion (Otsu, 1979) and the Kapur's entropy (Kapur, Sahoo, & Wong, 1985). In Otsu based method, the best threshold is the one that maximizes the between-class variance of the regions. Entropy is a useful criterion in communication; it was first introduced to image thresholding by (Pun, 1981) and improved by Kapur (Kapur et al., 1985). Kapur's entropy uses the maximization of the entropy to measure the homogeneity of the classes.

Most of the existing bi-level thresholding methods are easily extendable to multilevel thresholding as well. However, the computational time for finding multiple thresholds grows exponentially with the number of desired thresholds making unfeasible and questionable any attempt to use an exhaustive search. Indeed, for  $K$  thresholds, and  $L$  gray levels, the time complexity grows up to  $(L + K)!/L!K!$  (Cheng et al., 1999; Yin, 2007). Therefore, it might be very difficult to derive systematically and analytically a solution when the number of thresholds increases. Due to its time-consuming computation, multilevel thresholding is often an obstacle in the real time application system. Hence, performing an

image segmentation task through multilevel thresholding should be driven by two main requirements related to solutions quality and algorithm complexity and that can be summarized in the following statement: finding the best values of thresholds in a reasonable amount of time. These issues have been subject of several studies in the literature among which (Dirami, Hammouche, Diaf, & Siarry, 2013; Ghamisi et al., 2014; Manikandan, Ramar, Willjuice, & Srinivasagan, 2014; Peng, Zhang, & Zhang, 2013; Sathya & Kayalvizhi, 2011a, 2011b; Shapiro & Stockman, 2002; Xue & Titterton, 2011).

Casting image multilevel thresholding as a hard optimization problem is a logical consequence given its complexity. In fact, several biologically inspired global optimization algorithms have been recently proposed to deal with image multilevel thresholding. Within this context, we can refer to examples of genetic algorithms (GA) (Fan & Lin, 2007; Hammouche, Diaf, & Siarry, 2008; Manikandan et al., 2014; Tang, Yuan, Sun, Yang, & Gao, 2011; Yin, 1999; Zhang et al., 2014), particle swarm optimization algorithm (PSO) (Gao, Xu, Sun, & Tang, 2010; Gao, Xu, Sun, & Tang, 2013; Ghamisi et al., 2014; Maitra & Chatterjee, 2008; Yin, 2007), ant colony optimization algorithms (ACO) (Tao, Jin, & Liu, 2007), bacterial foraging algorithm (BFO) (Bakhshali & Shamsi, 2014; Sanyal, Chatterjee, & Munshi, 2011; Sathya & Kayalvizhi, 2011a, 2011b), honey bee mating optimization (HBMO) (Horng, 2010a, 2010b), firefly algorithm (FFA) (Horng & Liou, 2011; Raja, Rajinikanth, & Latha, 2014; Yang, 2009), wind driven optimization (WDO) (Bayraktar, Komurcu, Bossard, & Werner, 2013; Bayraktar, Turpin, & Werner, 2011; Bhandari, Singh, Kumar, & Singh, 2014), cuckoo search (CS) (Bhandari

et al., 2014; Panda, Agrawal, & Bhuyan, 2013; Sanjay, Rutuparna, Sudipta, & Panigrahib, 2013; Brajevic, Tuba & Bacanin, 2012), artificial bee colony (ABC) (Akay, 2013; Bhandari, Kumar, & Singh, 2015; Cuevas, Senci3n, Zaldivar, P3rez-Cisneros, & Sossa, 2012; Horng, 2011; Kumar, Kumar, Sharma, & Pant, 2013; Ma, Liang, Guo, Fan, & Yin, 2011; Zhang & Wu 2011), harmony search (HS) algorithm (Oliva, Cuevas, Pajares, Zaldivar, & Perez-Cisneros, 2013), differential evolution (DE) (Cuevas, Zaldivar, & P3rez-Cisneros, 2010; Ouadfel & Meshoul, 2014; Sarkar, RanjanPatra, & Das, 2011), electromagnetism optimization (EMO) (Oliva et al., 2014).

More recently, new metaheuristics have been developed based on the simulation of natural systems such as the flower pollination (FP) algorithm (Yang, 2012) and the social spider optimization (SSO) (Cuevas, Cienfuegos, Zaldivar, & Perez, 2013). The flower pollination (FP) algorithm has been introduced by Yang in 2012 (Yang, 2012) to solve optimization problems through mimicking the natural flower pollination behavior. In FP algorithm, four different rules related to the characteristic of the pollination process, the pollination behavior and the flower constancy are considered. Pollination is ensured by abiotic forces, such as water and wind, and biotic agents, such as insects, birds, bats and other animals (Yang, 2012). Pollination can occur either over short distances like self-pollination or over long distances like cross-pollination, in which the biotic pollinators may travel according to a L3vy flight.

Since its inception, the FP algorithm has gained a growing interest and it has been proposed for solving different problems. In Oliva et al. (2014) authors used the flower pollination algorithm for optimal control in a multimachine system with generalized unified power flow controller. In (Chakraborty, Asha, & Dutta, 2014), the authors propose the hybridizing of FP algorithm, with DE. In Abdel-Raouf, Abdel-Baset, and El-Henawy (2014a) authors extend the basic FP algorithm by including chaotic search. The same authors hybrid in (Abdel-Raouf, Abdel-Baset, & El-henawy, 2014b) the FP algorithm with the HS algorithm to solve the Sudoku Puzzles. In Prathiba, Balasingh, and Moses (2014), different economic load dispatch problems have been handled using the FP algorithm. In Platt (2014) the FP algorithm was applied to nonlinear algebraic systems with multiple solutions. In Rodrigues, Yang, de Souza, and Papa (2015) a binary version of the FP algorithm has been developed for feature selection purpose. In (Yang et al., 2014) the FP algorithm was used in the context of multi-objective optimization. In Lukasik and Kowalski (2015) authors study the application of the FP algorithm for continuous optimization. In Wang and Zhou (2014), a new version of the FP algorithm with dimension improvement is introduced. In Ibrahim and Mahmoud (2014), the FP algorithm has been used to solve Large Integer Programming. In (Mulya et al., 2014) authors applied the flower pollination algorithm for optimal control in a multi-machine system with generalized unified power flow controller. Recently, in Bekdař, Nigdeli, and Yang (2015) the FP algorithm is used to minimize the weight of truss structures, including sizing design variables. According to these studies, it has been shown that the FP algorithm exhibits a better performance when compared to other state-of-the-art metaheuristic algorithms such PSO and GA.

On the other hand and more recently, Cuevas (Cuevas et al., 2013) has presented a nature-inspired algorithm based on the social dynamics of spiders termed as Social-Spider Optimization (SSO). The SSO algorithm is based on the simulation of the mating behavior of social-spiders. The SSO algorithm is different from most of existing swarm algorithms, which model individuals as unisex entities that perform virtually the same behavior. This basic model is far away for the natural model and waste the possibility of adding new and selective operators as a result of considering individuals with different characteristics such as sex, task-responsibility, etc. (Cuevas et al., 2013). In the SSO algorithm, a more realistic model is introduced in which the entire population

**Table 1**  
Parameter settings for the BAT and PSO algorithms.

	Parameters	Value
BAT algorithm	Loudness ( $A$ )	0.25
	Pulse rate ( $R$ )	0.5
	Frequency min ( $Q_{\min}$ )	0
	Frequency max ( $Q_{\max}$ )	0.2
PSO algorithm	Maximum Inertia weight ( $w_{\max}$ )	0.9
	Minimum Inertia weight ( $w_{\min}$ )	0.4
	Maximum velocity ( $V_{\max}$ )	+1.0
	Minimum velocity ( $V_{\min}$ )	-1.0
	Cognitive coefficient ( $C_1$ )	1.429
	Cognitive coefficient ( $C_2$ )	1.429

is divided into different search-agent categories and a set of specialized operators are applied selectively to each of them (Cuevas et al., 2013). Two different search agents (spiders) are considered: males and females. Depending on gender, each individual undergoes a set of different evolutionary operators that mimic different cooperative behaviors typically found in the colony (Cuevas et al., 2013).

In Cuevas et al. (2013), the SSO algorithm has been experimentally tested considering 19 benchmark functions. The performance of SSO algorithm has been also compared to PSO, and ABC algorithms. Results have confirmed an acceptable performance of the proposed method in terms of the solution quality of the solution for all tested benchmark functions. In recent works, SSO has been used to solve different sorts of engineering problems. In Priyadharshini, Divya, Preethi, Pazhaniraja, and Paul (2015) a simple and efficient solution for web service publishing operation is proposed based on SSO algorithm. In Pereira, Delpiano, and Papa (2014) the SSO is used in the context of optical flow parameter estimation and its results are evaluated with those of PSO and HS algorithms.

Encouraged by the successful applications of the FP and SSO algorithms, this paper further examines their feasibility for solving image segmentation problem via a multilevel thresholding approach. As an objective function, Otsu's criterion and Kapur's entropy are used to compare the best performance of segmented images using these two optimization algorithms. In addition, FP and SSO algorithms have only one pre-defined control parameter that must be set. This is important because tuning the control parameters of an algorithm might be more difficult than the problem itself (Akay, 2013).

Therefore, the main contributions of this paper are two-fold: (i) to introduce SSO and FP algorithms for optimal threshold value selection, (ii) to compare them in such context. To the best of our knowledge, this issue has not been investigated before at least during the period this work has been developed.

The two proposed algorithms encode random samples from a feasible search space inside the image histogram as candidate solutions. During the optimization process, the quality of each solution is evaluated using the between-class variance and Kapur's method. Guided by these objective values, the set of candidate solutions is evolved according to the dynamics of the FP and SSO operators until a termination criterion is met. Obtained results from SSO and FP algorithms have been compared against the PSO and the BAT algorithms and against ground truth data that have been derived by implementing an exhaustive search algorithm. The performance of the different algorithms has been evaluated on several images using the objective function and two criteria: the peak-to-signal-noise ratio (PSNR) and the structural similarity index (SSIM).

The rest of the paper is organized as follows: in Section 2, the multilevel thresholding problem is formulated using Otsu's method and Kapur's entropy. In Section 3, we describe the proposed SSO and FP algorithms for multilevel image thresholding.

**Table 2**

The best fitness values and the corresponding best thresholds obtained from the methods based Otsu's criterion and Kapur's entropy for the test images.

Image	K	OTSU_Exhaustive search			Kapur_Exhaustive search		
		Threshold values	Fitness values	CPU time	Threshold values	Fitness values	CPU time
101085	2	95 169	2755.807806	0.364	90 172	12.847082	0.593
	3	60 113 182	2960.771941	20.663	74 127 180	15.963371	32.666
	4	52 92 139 197	3063.499639	1580.821	40 88 138 189	19.013666	2444.41
Wherry	2	110 188	3083.794184	1.051	105 163	12.177692	1.546
	3	103 158 216	3326.678303	70.934	100 148 191	15.141712	85.505
Zebra	4	85 121 161 217	3390.479285	4623.436	54 105 148 191	18.004651	4856.378
	2	94 141	817.642915	1.345	93 169	12.258460	1.506
Butterfly	3	90 125 173	915.103043	20.451	89 134 179	15.338314	30.046
	4	82 110 136 183	964.935915	4166.145	49 93 136 179	18.179248	5670.755
	2	85 154	3519.369256	0.564	110 174	12.683959	0.583
Bird	3	80 137 189	3646.019968	41.857	74 123 176	15.801767	55.506
	4	67 103 150 196	3712.057146	4310.123	74 121 171 217	18.804220	5449.518
	2	90 160	2535.450499	0.434	91 168	12.088437	0.496
Landscape	3	79 135 179	2611.877298	42.714	65 116 173	15.198964	46.451
	4	59 99 143 181	2649.686448	5512.452	55 98 144 183	17.982624	5826.815
	2	56 146	4419.790621	0.534	96 148	11.551304	0.63
Ostrish	3	49 109 173	4721.188536	50.714	96 161 196	14.515739	141.181
	4	44 89 129 180	4783.783312	5734.342	75 123 162 196	17.329207	5876.358
	2	74 132	1035.428099	0.634	119 171	12.372877	0.741
86016	3	67 98 144	1099.382854	51.234	73 122 171	15.466152	63.435
	4	62 89 124 176	1139.970653	5786.312	72 119 155 197	18.139022	5856.343
	2	121 168	1236.707999	0.554	100 145	11.516534	0.657
Starfish	3	105 145 174	1289.941891	52.234	98 139 173	14.216395	0.123
	4	92 127 157 178	1321.742604	5734.342	84 112 142 176	16.704122	5894.302
	2	84 156	2560.900708	0.534	90 169	12.992038	0.675
House	3	68 119 177	2795.302084	50.714	75 130 184	16.156850	70.876
	4	59 100 138 187	2881.589917	5714.652	67 115 163 207	19.105518	5824.222
	2	83 156	3163.410896	0.724	95 208	10.722930	0.987
Lake	3	82 131 182	3235.720474	56.765	47 97 208	13.572520	65.45
	4	56 88 132 182	3263.470178	5443.354	22 61 98 208	16.294258	5567.334
	2	88 155	3741.345417	0.624	94 163	12.450088	0.765
Woman	3	79 139 193	3876.946247	70.453	75 121 169	15.463663	76.503
	4	69 112 158 197	3943.268841	5735.516	72 113 156 194	18.244970	5845.256
	2	95 169	3472.162112	0.675	104 177	12.464464	0.745
	3	78 129 18	3623.515977	60.345	84 141 197	15.58394	78.605
	4	74 116 159 201	3686.974065	5987.312	74 119 164 208	18.495784	6012.982

Experimental results are presented in Section 4. Finally, we summarize our findings and discuss future work in Section 5.

## 2. Problem formulation

Given a gray level image  $I$  to be segmented into meaningful regions. Let there be  $L$  gray level values lying in the range  $\{0, 1, 2, \dots, (L-1)\}$ . Each gray level  $l$  is associated with  $h(l)$  which represents the number of pixels having the  $l$ th gray level as a value. Therefore, the probability  $P_l$  of the  $l$ th gray level is defined as  $P_l = h(l)/N$  where  $N$  denotes total number of pixels in the image  $I$ .

Suppose image  $I$  is composed of  $(K+1)$  classes. Hence,  $K$  thresholds,  $\{t_1, t_2, \dots, t_K\}$  are required to achieve the subdivision of the image into classes:  $C_0$  for  $[0, \dots, t_1 - 1]$ ,  $C_1$  for  $[t_1, \dots, t_2 - 1]$ , ...,  $C_K$  for  $[t_K, \dots, L - 1]$ , such that  $t_1 < t_2 < \dots < t_{K-1} < t_K$ . The multi-level thresholding problem solving aims to find the optimal thresholds  $(t_1^*, t_2^*, \dots, t_K^*)$  that maximizes a given objective function  $fit(t)$ , therefore following equation:

$$(t_1^*, t_2^*, \dots, t_K^*) = \arg \max (fit(t_1, t_2, \dots, t_K)) \quad (1)$$

Optimal thresholds can be obtained by maximizing some desired criteria such as the between class variance (Otsu based method) and the Kapur's entropy.

### 2.1. The Between\_class variance (Otsu's method)

The Otsu's method is one of the most popular methods proposed for bi-level and multilevel thresholding. Using Otsu's

method (Otsu, 1979), optimal thresholds are selected by maximizing the variance between the different classes.

$$(t_1^*, t_2^*, \dots, t_K^*) = \arg \max \left( \sum_{i=0}^K \sigma_i \right) \quad (2)$$

where

$$\sigma_i = \omega_i (\mu_i - \mu_T)^2$$

$$\mu_i = \sum_{j=t_i}^{t_{i+1}-1} \frac{i P_j}{\omega_j}, \omega_i = \sum_{j=t_i}^{t_{i+1}-1} P_j$$

$\mu_T$  is the mean intensity of the original image with  $t_0=0$  and  $t_{K+1}=L$ .

### 2.2. Kapur's entropy

This measure is based on the entropy and the probability distribution of the image histogram. The method aims at finding the optimal threshold that maximizes the overall entropy. With Kapur's entropy (Kapur et al., 1985), optimal thresholds can be found using the following objective function:

$$(t_1^*, t_2^*, \dots, t_K^*) = \arg \max \left( \sum_{i=0}^K H_i \right) \quad (3)$$

where



**Table 3**

The best fitness values and the corresponding best thresholds obtained from the methods based on the between-class variance for test images. The bold numbers in the tables represent optimal values obtained by each method.

Image	K	Fitness values				Threshold values				
		SSO	FP	BAT	PSO	SSO	FP	BAT	PSO	
101085	2	<b>2755.807806</b>	<b>2755.807806</b>	<b>2755.807806</b>	<b>2755.807806</b>	95 169	95 169	95 169	95 169	
	3	<b>2960.771941</b>	<b>2960.771941</b>	<b>2960.771941</b>	2960.757483	60 113 182	60 113 182	60 113 182	59 114 185	
	4	<b>3063.499639</b>	3063.474206	3063.354489	3063.420250	52 92 139 197	51 92 140 198	52 91 139 198	55 96 157 202	
	5	<b>3115.753552</b>	3115.627397	3115.481655	3115.598551	44 73 108 150 202	44 72 106 148 201	45 75 112 153 205	51 85 134 158 190	
	Wherry	2	<b>3083.794184</b>	<b>3083.794184</b>	<b>3083.794184</b>	<b>3083.794184</b>	110 188	110 188	110 188	110 188
Zebra	3	<b>3326.678303</b>	<b>3326.678303</b>	<b>3326.678303</b>	<b>3326.678303</b>	103 158 216	103 158 216	103 158 216	103 158 216	
	4	<b>3390.479285</b>	3390.478293	3390.425138	3390.308413	85 121 161 217	85 121 162 217	85 122 162 218	91 122 159 219	
	5	<b>3422.878517</b>	3422.751096	3422.724955	3422.089093	83 118 149 179 222	83 119 151 182 221	83 117 149 179 220	81 122 163 193 229	
	2	<b>817.642915</b>	<b>817.642915</b>	<b>817.642915</b>	<b>817.642915</b>	94 141	94 141	94 141	94 141	
	3	<b>915.103043</b>	<b>915.103043</b>	915.072336	<b>915.103043</b>	90 125 173	90 125 173	90 125 174	90 125 173	
Butterfly	4	<b>964.935915</b>	964.881275	964.703830	964.927553	82 110 136 183	81 109 135 182	83 111 137 185	78 106 134 189	
	5	<b>990.048155</b>	989.834309	989.708108	989.669316	75 99 119 142 187	74 99 118 143 190	73 99 120 143 189	65 94 115 140 176	
	2	<b>3519.369256</b>	<b>3519.369256</b>	<b>3519.369256</b>	<b>3519.369256</b>	85 154	85 154	85 154	85 154	
	3	<b>3646.019968</b>	<b>3646.019968</b>	<b>3646.019968</b>	3645.995573	80 137 189	80 137 189	80 137 189	81 138 189	
	4	<b>3712.057146</b>	3712.050265	3711.666080	3711.936444	67 103 150 196	67 103 149 196	66 100 149 195	74 110 147 201	
Bird	5	<b>3760.339045</b>	3759.937206	3759.720298	3759.309047	59 90 128 169 205	60 90 127 168 205	59 90 126 166 202	74 113 159 187 215	
	2	<b>2535.450499</b>	<b>2535.450499</b>	<b>2535.450499</b>	<b>2535.450499</b>	90 160	90 160	90 160	90 160	
	3	<b>2611.877298</b>	<b>2611.877298</b>	<b>2611.877298</b>	2611.870728	79 135 179	79 135 179	79 135 179	80 139 177	
	4	<b>2649.686448</b>	<b>2649.686448</b>	<b>2649.686448</b>	2649.544555	59 99 143 181	59 100 144 181	58 100 145 182	62 106 144 181	
	5	<b>2668.341043</b>	2668.245452	2668.139831	2668.259866	57 96 136 170 192	57 95 134 170 192	59 96 135 169 191	57 99 133 176 194	
Landscape	2	<b>4419.790621</b>	<b>4419.790621</b>	<b>4419.790621</b>	<b>4419.790621</b>	56 146	56 146	56 146	56 146	
	3	<b>4721.188536</b>	<b>4721.188536</b>	<b>4721.188536</b>	<b>4721.188536</b>	49 109 173	49 109 173	49 109 173	49 109 173	
	4	<b>4783.783312</b>	4783.763752	4783.656837	4783.587370	44 89 129 180	44 89 129 181	45 90 130 182	47 100 139 191	
	5	<b>4823.804266</b>	4823.732280	4823.458808	4823.269270	42 80 108 139 182	44 81 108 140 182	42 78 107 138 184	36 85 106 143 187	
	Ostrich	2	<b>1035.428099</b>	<b>1035.428099</b>	<b>1035.428099</b>	<b>1035.428099</b>	74 132	74 132	74 132	74 132
3		<b>1099.382854</b>	<b>1099.382854</b>	<b>1099.382854</b>	<b>1099.382854</b>	67 98 144	67 98 144	67 98 144	67 98 144	
4		<b>1139.970653</b>	1139.859419	1139.878713	1139.386748	62 89 124 176	62 89 125 178	61 88 124 176	51 75 109 157	
5		<b>1168.719767</b>	1168.427904	1168.434159	1168.461658	51 74 98 129 179	50 73 97 127 174	50 73 97 131 178	60 84 114 141 170	
2		<b>1236.707999</b>	<b>1236.707999</b>	<b>1236.707999</b>	<b>1236.707999</b>	121 168	121 168	121 168	121 168	
86016	3	<b>1289.941891</b>	<b>1289.941891</b>	<b>1289.941891</b>	<b>1289.941891</b>	105 145 174	105 145 174	105 145 174	105 145 174	
	4	<b>1321.742604</b>	<b>1321.742604</b>	1321.663320	1321.439373	92 127 157 178	92 127 157 178	93 127 158 178	94 134 161 178	
	5	<b>1337.349287</b>	1336.967926	1336.981420	1336.168946	90 121 149 167 183	89 119 148 167 184	91 122 151 169 185	92 125 155 173 187	
	Starfish	2	<b>2546.884578</b>	<b>2546.884578</b>	<b>2541.902196</b>	<b>2546.884578</b>	85 157	85 157	85 157	85 157
		3	<b>2779.925224</b>	<b>2779.925224</b>	2774.459905	2774.488014	69 119 175	68 119 175	69 119 177	68 119 175
4		<b>2865.620446</b>	2865.619444	2860.065110	2859.779080	63 101 135 184	59 100 138 186	60 101 139 187	58 99 145 180	
5		<b>2912.249371</b>	2912.176312	2906.850485	2904.828042	52 86 117 150 194	50 83 115 149 192	53 86 118 150 192	46 78 105 139 179	
House		2	<b>3163.410896</b>	<b>3163.410896</b>	<b>3147.094108</b>	<b>3163.410896</b>	83 156	83 156	83 156	83 156
	3	<b>3235.720474</b>	<b>3235.720474</b>	3219.148283	3222.821715	82 131 182	82 131 182	82 131 181	82 134 180	
	4	<b>3263.470178</b>	3263.453851	3245.839006	3249.877231	56 88 132 182	55 88 133 184	54 87 133 182	54 78 135 185	
	5	<b>3273.415346</b>	3287.148819	3269.144916	3272.947774	56 88 131 180 221	56 87 130 180 218	56 88 132 180 226	63 93 131 177 218	
	Lake	2	<b>3741.345417</b>	<b>3741.345417</b>	<b>3741.345417</b>	<b>3741.345417</b>	88 155	88 155	88 155	88 155
3		<b>3876.946247</b>	<b>3876.946247</b>	3876.945225	3875.477915	79 139 193	79 139 193	80 140 193	80 140 193	
4		<b>3943.268841</b>	<b>3943.268841</b>	3943.079841	3936.963569	69 112 158 197	69 112 157 197	70 113 159 196	71 115 160 198	
5		<b>3977.344042</b>	<b>3977.344042</b>	3976.849952	3960.859529	60 90 128 166 199	60 90 128 166 199	59 86 123 164 199	56 87 129 166 198	
Woman		2	<b>3472.162112</b>	<b>3472.162112</b>	<b>3472.099744</b>	<b>3472.162112</b>	95 69	95 69	95 69	95 69
	3	<b>3623.515977</b>	<b>3623.515977</b>	3623.451238	3623.451238	78 129 185	78 129 185	78 129 185	76 23 180	
	4	<b>3686.974065</b>	3686.030425	3685.970438	3685.480031	74 116 159 201	73 116 159 201	73 116 159 202	72 09 146 195	
	5	<b>3717.736687</b>	3717.642678	3717.432958	3717.504527	65 96 131 168 206	64 96 132 169 208	62 92 127 167 206	72 11 149 184 213	

$$H_i = - \sum_{j=t_i}^{t_{i+1}-1} \frac{P_j}{\omega_j} \ln \frac{P_j}{\omega_j}, \quad \omega_i = \sum_{j=t_i}^{t_{i+1}-1} P_j$$

with  $t_0 = 0$  and  $t_{K+1} = L$ .

The optimal thresholds are chosen in a way the corresponding objective function is maximized. Performing an exhaustive search requires evaluation of all possible values of the thresholds and chooses the ones that give the best objective function score. Despite the simplicity and the straight forwardness of such a process, the computational time grows sharply when the number of threshold increases. In fact, for  $K$  thresholds, and  $L$  gray levels, the time complexity grows up to  $O((L+K)!/L!K!)$  (Cheng et al., 1999; Yin, 2007).

### 3. Proposed approaches for multilevel thresholding

The purpose of this paper is to investigate the search abilities of SSO and FP algorithms for multilevel thresholding. The two metaheuristics are used to find the optimal vector of intensity levels

$t^* = (t_1^*, t_2^*, \dots, t_K^*)$  that maximizes a given objective function. In our study, two functions have been considered: the between-class variance (Eq. (2)) and the Kapur's entropy (Eq. (3)) described in the above section.

#### 3.1. The SSO based multilevel thresholding algorithm

The SSO algorithm was first introduced by Cuevas et al. (2013) in 2012 for modeling the cooperative behavior of social spiders in a colony. In the SSO algorithm, the search space and the search agents stand for the communal web and the spiders of the colony in the natural system respectively. The spider's position represents a potential solution. The search agents are divided into two types: males and females. A particularity of social-spiders is the female-biased population. In SSO algorithm, the number of male spiders hardly reaches 30% of the total colony members while the number of females  $N_f$  is randomly selected within a range of 65-90% of the entire population  $N_p$  (Cuevas et al., 2013). Every spider (male and female) has a weight according to the fitness value of its solution. During optimization, spiders communicate through

**Table 4**  
The best thresholds and the corresponding best fitness obtained from the methods based on the Kapur's entropy for test images. The bold numbers in the tables represent optimal values obtained by each method.

Image	K	Fitness values				Threshold values			
		SSO	FP	BAT	PSO	SSO	FP	BAT	PSO
101085	2	<b>12.847082</b>	<b>12.847082</b>	<b>12.847082</b>	<b>12.847082</b>	90 172	90 172	90 172	90 172
	3	<b>15.963371</b>	<b>15.963371</b>	15.962560	15.963119	74 127 180	74 127 180	78 132 184	78 129 181
	4	<b>19.013666</b>	19.010370	19.006546	19.010788	40 88 138 189	40 88 135 185	40 87 138 186	45 90 136 189
	5	<b>21.744755</b>	21.736601	21.741902	21.735468	37 83 122 161 196	37 82 120 161 196	40 83 121 160 196	32 60 93 153 193
		<b>12.177692</b>	<b>12.177692</b>	<b>12.177692</b>	<b>12.177692</b>	105 163	105 163	105 163	105 163
Wherry	2	<b>15.141712</b>	<b>15.141712</b>	15.141113	15.141689	100 148 191	100 148 191	100 148 188	95 147 187
	3	<b>18.004651</b>	18.001443	17.994920	18.003399	54 105 148 191	54 104 148 191	56 104 150 193	54 113 152 198
	4	<b>20.696338</b>	20.687249	20.686953	20.671377	48 85 120 152 193	48 81 117 151 194	56 107 150 201 237	79 120 161 202 238
	5	<b>12.258460</b>	<b>12.258460</b>	<b>12.258460</b>	<b>12.258460</b>	93 169	93 169	93 169	93 169
		<b>15.338314</b>	<b>15.338314</b>	<b>15.338314</b>	<b>15.338314</b>	89 134 179	89 134 179	89 134 179	89 134 179
Zebra	2	<b>18.179248</b>	18.176303	18.171796	18.158891	49 93 136 179	49 93 136 182	49 94 136 179	80 132 168 204
	3	<b>20.996743</b>	20.981965	20.950633	20.950453	49 93 132 169 204	49 93 131 167 200	50 95 131 169 201	49 81 106 147 203
	4	<b>12.683959</b>	<b>12.683959</b>	<b>12.683959</b>	<b>12.683959</b>	110 174	110 174	110 174	110 174
	5	<b>15.801767</b>	<b>15.801767</b>	15.801176	15.801159	74 123 176	74 123 176	76 124 176	76 127 184
		<b>18.804220</b>	<b>18.804220</b>	18.798495	18.790101	74 121 171 217	74 121 171 217	72 122 169 217	78 116 154 204
Butterfly	2	<b>21.482543</b>	21.474797	21.470672	21.470206	60 96 133 174 217	58 92 128 171 217	70 110 146 180 217	70 94 130 174 216
	3	<b>12.088437</b>	<b>12.088437</b>	<b>12.088437</b>	<b>12.088437</b>	91 168	91 168	91 168	91 168
	4	<b>15.198964</b>	<b>15.198964</b>	<b>15.198964</b>	<b>15.198964</b>	65 116 173	65 116 173	65 116 173	65 116 173
	5	<b>17.982624</b>	17.982427	17.981291	17.979551	55 98 144 183	57 100 144 183	55 98 143 181	58 92 138 182
		<b>20.604057</b>	20.602413	20.598996	20.572345	46 79 113 151 183	50 83 116 153 182	48 79 113 151 183	51 104 137 159 182
Bird	2	<b>11.551304</b>	<b>11.551304</b>	<b>11.551304</b>	<b>11.551304</b>	96 148	96 148	96 148	96 148
	3	<b>14.515739</b>	<b>14.515739</b>	<b>14.508739</b>	<b>14.515739</b>	96 161 196	96 161 196	95 161 196	96 161 196
	4	<b>17.329207</b>	17.322814	17.321985	17.298804	75 123 162 196	74 122 160 196	78 123 162 196	63 101 136 190
	5	<b>19.911790</b>	19.885866	19.877387	19.861553	25 59 106 161 196	25 59 109 161 196	26 59 110 163 196	36 63 107 160 195
		<b>12.372877</b>	<b>12.372877</b>	<b>12.372877</b>	<b>12.372877</b>	119 171	119 171	119 171	119 171
Landscape	2	<b>15.466152</b>	<b>15.466152</b>	<b>15.466152</b>	<b>15.466152</b>	73 122 171	73 122 171	73 122 171	72 119 185
	3	<b>18.139022</b>	18.137106	18.132819	18.127389	72 119 155 197	73 119 159 197	71 119 150 187	27 78 120 174
	4	<b>20.803834</b>	20.791327	20.775555	20.789842	28 74 119 155 197	28 74 121 155 197	28 69 119 151 185	58 84 121 166 196
	5	<b>11.516534</b>	<b>11.516534</b>	<b>11.516534</b>	<b>11.516534</b>	100 145	100 145	100 145	100 145
		<b>14.216395</b>	<b>14.216395</b>	<b>14.216395</b>	<b>14.216395</b>	98 139 173	98 139 173	98 139 173	98 139 173
Ostrish	2	<b>16.704122</b>	<b>16.704122</b>	16.701520	16.703562	84 112 142 176	84 112 142 176	84 111 142 174	81 104 138 173
	3	<b>19.049748</b>	19.036004	19.036047	19.027661	63 90 118 146 177	63 92 120 146 178	63 93 126 153 180	85 109 135 154 180
	4	<b>12.992038</b>	<b>12.992038</b>	<b>12.992038</b>	<b>12.992038</b>	90 169	90 169	90 169	90 169
	5	<b>16.156850</b>	<b>16.156850</b>	16.156490	<b>16.156850</b>	75 130 184	75 130 184	75 131 184	75 130 184
		<b>19.105518</b>	<b>19.105518</b>	19.105054	19.102281	67 115 163 207	67 115 163 207	67 116 163 206	75 113 156 209
86016	2	<b>21.861960</b>	21.861377	21.854004	21.842028	56 94 133 172 211	56 95 134 171 209	57 94 134 174 210	59 106 154 189 217
	3	<b>10.722930</b>	<b>10.722930</b>	<b>10.722930</b>	<b>10.722930</b>	95 208	95 208	95 208	95 208
	4	<b>13.572520</b>	<b>13.572520</b>	13.528305	<b>13.572520</b>	47 97 208	47 97 208	59 97 208	47 97 208
	5	<b>16.034367</b>	16.027469	15.932783	16.030183	46 71 96 208	47 74 97 208	44 69 95 207	55 127 177 211
		<b>18.590445</b>	18.581950	18.564674	18.537709	64 122 163 202 209	65 121 158 201 209	64 122 167 202 208	46 67 97 209 230
Starfish	2	<b>12.450088</b>	<b>12.450088</b>	<b>12.450088</b>	<b>12.450088</b>	94 163	94 163	94 163	94 163
	3	<b>15.463663</b>	<b>15.463663</b>	<b>15.463663</b>	<b>15.463597</b>	75 121 169	75 121 169	75 121 169	75 120 169
	4	<b>18.244970</b>	<b>18.244970</b>	18.244085	18.241909	72 113 156 194	72 113 156 194	72 113 156 195	71 111 155 192
	5	<b>20.881141</b>	20.880414	20.867196	20.850953	66 99 133 166 197	66 100 133 166 197	66 100 130 167 198	66 93 128 165 197
		<b>12.464464</b>	<b>12.464464</b>	<b>12.464464</b>	<b>12.464464</b>	104 177	104 177	104 177	104 177
House	2	<b>15.583941</b>	<b>15.583941</b>	15.583832	<b>15.583941</b>	84 141 197	84 141 197	84 140 197	84 141 197
	3	<b>18.495784</b>	<b>18.495784</b>	18.494771	18.491966	74 119 164 208	74 119 164 208	75 120 165 208	73 116 161 204
	4	<b>21.174324</b>	21.173518	21.172040	21.168137	72 108 145 181 218	71 107 143 180 218	70 107 144 182 218	79 116 148 179 213
	5								

vibrating the strings in the web (search space). The vibration that a spider receives depends on the weight and distance of the sender spider. The spider's positions are updated according to their gender. A new population of search agents (spiders) is generated using the mating operator. Male spiders that tend to be attracted toward females are mating with them according to certain radius called mating radius. After producing new spiders, their fitness values are calculated and then compared to the worst spider in the population. The best spiders are added to the population, while the worst ones are discarded.

For the multilevel thresholding problem, a population of  $N_p$  solutions is created, where  $N_p$  is the number of female and male spiders. The number of females  $N_f$  and the number of males  $N_m$  are calculated as follows (Cuevas et al., 2013):

$$N_f = \text{floor}[(0.9 - \text{rand} * 0.25) * N_p]$$

$$N_m = N_p - N_f \quad (4)$$

where  $\text{rand}$  stands for a random number within the range of [0,1] and  $\text{floor}$  is a function that maps a real number to the largest previous integer.

The population of spiders  $X$  is divided into two sub\_groups according to the gender criterion  $F = \{f_1, f_2, \dots, f_{N_f}\}$  and  $M = \{m_1, m_2, \dots, m_{N_m}\}$ .

Such that  $X = F \cup M = \{x_1 = f_1, x_2 = f_2, \dots, x_{N_f} = f_{N_f}, x_{N_f+1} = m_1, x_{N_f+2} = m_2, \dots, x_{N_p} = m_{N_m}\}$ .

First, we need to formulate each spider as a potential solution to the multilevel thresholding problem. Therefore, a suitable solution encoding should be adopted. Each individual spider is represented as a vector of possible real values corresponding to thresholds. The goal of SSO algorithm is to determine the position in the search space (the optimal threshold values) that optimizes the considered objective function given by Eq. (2) or Eq. (3).

Each individual spider  $x_i$  ( $i=1 \dots N_p$ ) in the generation  $G$  is formulated as follows:

$$x_i(G) = (t_{i,1}, t_{i,2}, \dots, t_{i,K})^T, \text{ subject to } 1 \leq i \leq N_p$$

and

$$0 < t_{i,1} < t_{i,2} < \dots < t_{i,K} < L \quad (5)$$

**Table 5**

Comparison of the PSNR, SSIM and CPU time of the SSO, FP, BAT and PSO applied over the test images using between-class variance based method.

Image	K	PSNR				SSIM				CPU time			
		SSO	FP	BAT	PSO	SSO	FP	BAT	PSO	SSO	FP	BAT	PSO
101085	2	21.3264	21.3264	21.3264	21.3264	0.6035	0.6035	0.6035	0.6035	2.459028	1.254739	0.946760	0.273415
	3	23.6907	23.6907	23.6907	23.6796	0.7874	0.7874	0.7874	0.7867	2.537371	1.256710	0.982512	0.293686
	4	25.6673	25.6647	25.2503	25.6630	0.8388	0.8387	0.8290	0.8385	2.685562	1.289742	0.992497	0.305026
	5	27.1589	27.1544	26.3393	27.1503	0.8852	0.8834	0.8612	0.8810	2.772673	1.303155	1.012615	0.320994
	2	21.8803	21.8803	21.8803	21.8803	0.6784	0.6784	0.6784	0.6784	2.453083	1.225623	0.962395	0.285584
Wherry	3	25.6023	25.6023	25.6023	25.6023	0.7245	0.7245	0.7245	0.7245	2.630997	1.282276	0.953957	0.287585
	4	27.5130	27.5130	27.3975	27.5100	0.8102	0.8078	0.8075	0.8078	2.636265	1.346396	0.979375	0.289598
	5	28.9462	28.9356	28.3475	28.9264	0.8300	0.8296	0.8210	0.8285	2.724980	1.366749	1.013808	0.325162
	2	24.3495	24.3495	24.3495	24.3495	0.7434	0.7434	0.7434	0.7434	2.342889	1.230021	0.942246	0.324007
	3	26.6244	26.6244	26.6231	26.6244	0.8090	0.8090	0.8085	0.8090	2.556389	1.279778	0.961021	0.341796
Zebra	4	28.5061	28.5035	28.4115	28.4953	0.8646	0.8643	0.8565	0.8631	2.624582	1.303657	0.972996	0.356705
	5	29.8900	29.8716	29.5381	29.8683	0.9020	0.9013	0.8929	0.8991	2.674423	1.337754	0.995900	0.369267
	2	22.7294	22.7294	22.7294	22.7294	0.8507	0.8507	0.8507	0.8507	2.459777	1.221231	0.926971	0.335281
	3	24.7027	24.7027	24.7027	24.7018	0.8971	0.8971	0.8971	0.8965	2.627975	1.260529	0.983980	0.339839
	4	26.2500	26.2496	26.0010	26.2375	0.9226	0.9199	0.9118	0.9196	2.633520	1.291418	1.003228	0.343794
Bird	5	27.8768	27.8755	27.0193	27.8551	0.9451	0.9449	0.9305	0.9446	2.811071	1.326125	1.018796	0.356456
	2	25.3233	25.3233	25.3233	25.3233	0.9002	0.9002	0.9002	0.9002	2.397951	1.221849	0.925212	0.313565
	3	27.5441	27.5441	27.5441	27.4852	0.9155	0.9134	0.9134	0.9134	2.423095	1.257078	0.939507	0.332616
	4	29.2758	29.2758	29.2367	29.2641	0.9275	0.9275	0.9272	0.9274	2.677935	1.289072	0.971672	0.349337
	5	30.4788	30.4676	30.2577	30.4675	0.9306	0.9267	0.9233	0.9262	2.661684	1.320999	0.996501	0.351411
Landscape	2	21.5350	21.5350	21.5350	21.5350	0.7353	0.7353	0.7353	0.7353	2.342621	1.226173	0.948603	0.332414
	3	26.2213	26.2213	26.2213	26.2213	0.8147	0.8147	0.8147	0.8147	2.504546	1.245867	0.952583	0.351115
	4	28.4631	28.4620	28.2820	28.4554	0.8530	0.8379	0.8344	0.8377	2.607799	1.287891	0.968467	0.355252
	5	30.9138	30.8902	30.4759	30.8870	0.8851	0.8851	0.8696	0.8851	2.661609	1.340079	1.119093	0.362564
	2	25.2369	25.2369	25.2369	25.2369	0.7716	0.7716	0.7716	0.7716	2.442588	1.215649	0.933036	0.332122
Ostrish	3	26.9599	26.9599	26.9599	26.9599	0.7922	0.7922	0.7922	0.7922	2.559097	1.253733	0.949071	0.333817
	4	28.5800	28.5769	28.2548	28.5680	0.8167	0.8166	0.8032	0.8166	2.692355	1.273697	0.997101	0.341639
	5	30.2418	30.2185	29.3111	30.2171	0.8393	0.8392	0.8349	0.8387	2.750087	1.342492	1.254936	0.352101
	2	26.6320	26.6320	26.6320	26.6320	0.6877	0.6877	0.6877	0.6877	2.361919	1.228202	0.929288	0.329950
	3	28.6811	28.6811	28.6811	28.6811	0.7662	0.7662	0.7662	0.7662	2.549376	1.285986	0.982550	0.335779
86016	4	30.6293	30.6293	30.6195	30.4672	0.8493	0.8493	0.8440	0.8440	2.654843	1.292234	1.014851	0.341702
	5	32.0377	32.0002	31.7955	31.9981	0.8854	0.8840	0.8798	0.8815	2.690042	1.327665	1.065408	0.349288
	2	21.3714	21.3714	21.3714	21.3714	0.6152	0.6152	0.6152	0.6152	2.303670	1.209609	0.954879	0.332101
	3	24.3066	24.3066	24.2990	24.2990	0.7077	0.7077	0.7064	0.7073	2.546539	1.267890	0.965137	0.346260
	4	26.2146	26.2146	25.8579	26.2115	0.7659	0.7658	0.7583	0.7657	2.624687	1.281790	0.988659	0.350483
House	5	27.7868	27.7749	27.3764	27.7718	0.8149	0.8127	0.8122	0.8124	2.705957	1.301520	0.999515	0.364032
	2	25.6575	25.6575	25.6575	25.6575	0.8428	0.8428	0.8428	0.8428	2.282339	1.207549	0.937697	0.401029
	3	27.9418	27.9418	27.9398	27.9149	0.8701	0.8701	0.8699	0.8663	2.556601	1.241663	0.943291	0.413305
	4	29.2840	29.2761	28.9586	29.2724	0.8896	0.8881	0.8794	0.8879	2.647658	1.272546	1.026674	0.418903
	5	30.9179	30.8978	30.5992	30.8870	0.8979	0.8969	0.8909	0.8958	2.674876	1.316153	1.042771	0.424032
Lake	2	22.9475	22.9475	22.9475	22.9475	0.7583	0.7583	0.7583	0.7583	2.373555	1.225698	0.926867	0.331071
	3	25.6152	25.6152	25.6150	25.5649	0.8197	0.8197	0.8140	0.8139	2.507886	1.265674	0.935174	0.338103
	4	27.4884	27.4884	27.4884	27.2985	0.8442	0.8442	0.8436	0.8436	2.642477	1.295845	0.970481	0.345550
	5	28.8756	28.8756	28.6628	28.8600	0.8737	0.8737	0.8625	0.8709	2.751755	1.330833	1.027854	0.358301
	2	23.1353	23.1353	23.1353	23.1353	0.8457	0.8457	0.8457	0.8457	2.350430	1.206614	0.907696	0.339457
Woman	3	25.5847	25.5847	25.4343	25.5828	0.8817	0.8817	0.8764	0.8764	2.483750	1.236018	0.966295	0.346003
	4	27.5863	27.5815	27.5102	27.5815	0.9089	0.9054	0.9044	0.9044	2.623252	1.281098	1.001836	0.351110
	5	29.1484	29.1211	28.9547	29.1211	0.9280	0.9274	0.9270	0.9274	2.709261	1.356787	1.013647	0.364433

Here, the  $K$  parameters in the vector representation correspond to the  $K$  multiple thresholds,  $L$  is the number of gray levels in the original image and  $Np$  is the size of the population.

Initially, the position of each individual spider (female and male) of the population is generated by randomly chosen threshold values within the range  $[g_{min}, g_{max}]$ , where  $g_{min}$  and  $g_{max}$  are the minimum and the maximum gray levels in the image, respectively. This can be expressed by the following equation:

$$t_{i,j} = g_{min} + rand(0, 1) * (g_{max} - g_{min}) \quad (6)$$

$(i = 1, \dots, Np \text{ and } j = 1, \dots, K)$

Each spider moves in the search space according to some vibrations that it receive from another spider in the colony. Cuevas proposed the following mathematical model for the spider's vibration (Cuevas et al., 2013):

$$Vibc_i = w_j * e^{-d_{i,j}^2} \quad (7)$$

where  $d_{i,j}$  is the Euclidean distance between spiders  $i$  and  $j$ , such that  $d_{i,j} = \|x_i - x_j\|$ .  $w_j$  is the weight of the  $j$ th spider (male and

female) and it is defined according to the fitness value of its solution as follows:

$$w_j = \frac{fitness_j - worst}{best - worst} \quad (8)$$

where  $fitness_j$  is the fitness value obtained by the evaluation of the  $j$ th spider's position  $j=1, 2, \dots, Np$ .  $worst$  and  $best$  refer respectively to the worst fitness value and the best fitness value of the entire population.

As social-spiders perform cooperative interaction over other colony members depending on the gender, specialized operators are proposed for each spider.

In SSO algorithm, it is assumed that females can make a decision randomly to move toward or outward the source. When a female is attracted toward the source of vibrations, its position is updated as follows (Cuevas et al., 2013):

$$f_i(G+1) = f_i(G) + \alpha \cdot Vibc_i \cdot (x_c - f_i(G)) + \beta \cdot Vibb_i \cdot (x_b - f_i(G)) + \delta \cdot (rand - 0.5) \quad (9)$$

**Table 6**

Comparison of the PSNR, SSIM and CPU time of the SSO, FP, BAT and PSO applied over the test images using Kapur's entropy.

Image	K	PSNR				SSIM				CPU time			
		SSO	FP	BAT	PSO	SSO	FP	BAT	PSO	SSO	FP	BAT	PSO
101085	2	21.2575	21.2575	21.2575	21.2575	0.6103	0.6103	0.6103	0.6103	2.961597	1.720125	1.401807	0.444230
	3	23.4152	23.4152	23.3156	23.3108	0.7245	0.7245	0.7044	0.7023	2.966144	1.747918	1.403577	0.458996
	4	25.4756	25.2380	25.1839	25.1915	0.8364	0.8268	0.8236	0.8260	3.069850	1.795526	1.460270	0.476639
	5	26.6220	26.4628	26.1837	26.4093	0.8796	0.8544	0.8441	0.8464	3.118941	1.911327	1.470989	0.504385
	2	20.9999	20.9999	20.9999	20.9999	0.6996	0.6996	0.6996	0.6996	2.764352	1.677162	1.373324	0.295391
Wherry	3	24.0994	24.0994	23.7709	23.5813	0.7143	0.7143	0.7138	0.7068	2.977019	1.691961	1.412826	0.300112
	4	24.9928	24.7093	24.4327	24.4364	0.7437	0.7360	0.7307	0.7331	3.020446	1.741585	1.428064	0.316073
	5	28.0984	26.6210	25.7230	25.8287	0.8127	0.7984	0.7619	0.7922	3.041107	1.747067	1.483519	0.321127
	2	23.6696	23.6696	23.6696	23.6696	0.6769	0.6769	0.6769	0.6769	2.761618	1.655143	1.344605	0.315527
	3	26.3709	26.3709	26.3709	26.3709	0.7757	0.7757	0.7757	0.7757	2.964809	1.713047	1.383778	0.455130
Zebra	4	26.8206	26.8118	26.3984	26.8015	0.7909	0.7870	0.7534	0.7860	3.054631	1.715195	1.407235	0.464841
	5	28.1055	27.6801	27.6016	27.6646	0.8376	0.8179	0.8080	0.8134	3.065015	1.744258	1.410154	0.472657
	2	21.6641	21.6641	21.6641	21.6641	0.7918	0.7918	0.7918	0.7918	2.723934	1.686450	1.376477	0.451263
	3	24.5991	24.5991	24.5443	24.5443	0.8941	0.8941	0.8939	0.8933	2.971536	1.714955	1.404676	0.460000
	4	26.0162	26.0162	25.8675	25.8252	0.9151	0.9151	0.9149	0.9149	3.068704	1.726741	1.422739	0.472775
Bird	5	27.4963	27.4921	27.1804	27.3404	0.9395	0.9386	0.9285	0.9366	3.086111	1.750145	1.444432	0.483442
	2	25.1523	25.1523	25.1523	25.1523	0.8959	0.8959	0.8959	0.8959	2.812351	1.641063	1.388612	0.442849
	3	27.0861	27.0861	27.0861	27.0861	0.9159	0.9159	0.9159	0.9159	2.863975	1.673715	1.356706	0.452109
	4	29.2490	29.2467	29.1315	29.2298	0.9276	0.9264	0.9241	0.9263	3.028040	1.724956	1.370563	0.464614
	5	30.2722	30.2558	29.6592	30.2227	0.9363	0.9355	0.9343	0.9344	3.041001	1.736210	1.432050	0.479062
Landscape	2	20.0585	20.0585	20.0585	20.0585	0.6384	0.6384	0.6384	0.6384	2.738977	1.645850	1.362234	0.341924
	3	20.9237	20.9237	20.9237	20.9237	0.6887	0.6887	0.6887	0.6887	2.862937	1.674559	1.372831	0.351900
	4	27.3356	23.5923	23.3688	23.4812	0.8185	0.7178	0.7020	0.7111	3.007773	1.717894	1.391252	0.367694
	5	27.6839	27.4814	27.4025	27.4726	0.8422	0.8417	0.8376	0.8409	3.076331	1.756947	1.423722	0.379703
	2	20.5335	20.5335	20.5335	20.5335	0.7209	0.7209	0.7209	0.7209	2.706791	1.691292	1.359691	0.355561
Ostrich	3	26.5331	26.5331	26.5331	26.5065	0.7933	0.7933	0.7933	0.7902	3.023298	1.710694	1.385007	0.361285
	4	27.0187	26.9966	26.7414	26.9682	0.8037	0.8023	0.7995	0.8016	3.036903	1.743599	1.409078	0.374199
	5	28.9203	27.3581	27.3107	27.3167	0.8223	0.8062	0.8031	0.8061	3.103176	1.755994	1.434556	0.386538
	2	25.2653	25.2653	25.2653	25.2653	0.6357	0.6357	0.6357	0.6357	2.700087	1.562200	1.281102	0.358317
	3	28.6330	28.6330	28.6330	28.6330	0.7660	0.7660	0.7660	0.7660	2.857073	1.642736	1.317125	0.365084
86016	4	29.7898	29.7898	29.7091	29.4635	0.8039	0.8039	0.7959	0.7945	2.920554	1.650255	1.334014	0.384680
	5	31.3104	30.6439	30.1738	30.2465	0.8464	0.8326	0.8104	0.8151	2.991276	1.655886	1.347105	0.407015
	2	21.2617	21.2617	21.2617	21.2617	0.6084	0.6084	0.6084	0.6084	2.821286	1.693793	1.367917	0.406467
	3	24.0868	24.0868	24.0515	24.0868	0.6955	0.6955	0.6950	0.6940	3.026398	1.737140	1.369265	0.437353
	4	25.5818	25.5728	25.4149	25.5728	0.7429	0.7393	0.7386	0.7389	3.060717	1.723074	1.394794	0.454178
Starfish	5	27.3063	27.2897	26.2656	27.2291	0.7870	0.7866	0.7564	0.7846	3.082178	1.759838	1.481318	0.463923
	2	15.6351	15.6351	15.6351	15.6351	0.7264	0.7264	0.7264	0.7264	2.727263	1.665078	1.359054	0.351376
	3	15.8137	15.8137	15.7855	15.8137	0.7475	0.7475	0.7434	0.7434	2.990548	1.681797	1.402546	0.366535
	4	25.6659	15.8652	15.8519	15.7802	0.8483	0.7565	0.7539	0.7501	3.005655	1.729754	1.420579	0.383350
	5	27.0430	26.8268	15.8349	26.7580	0.8589	0.8585	0.7485	0.8562	3.064057	1.748185	1.452156	0.406165
Lake	2	22.8895	22.8895	22.8895	22.8895	0.7621	0.7621	0.7621	0.7621	2.802157	1.641250	1.351624	0.437132
	3	25.3543	25.3543	25.3543	25.3542	0.8201	0.8201	0.8201	0.8013	2.963534	1.671405	1.368192	0.440634
	4	27.4155	27.4155	27.3945	27.4043	0.8436	0.8436	0.8417	0.8417	2.990511	1.707990	1.371039	0.462556
	5	28.6813	28.6686	28.2919	28.6669	0.8668	0.8644	0.8627	0.8629	3.015879	1.745189	1.421993	0.472585
	2	23.0710	23.0710	23.0710	22.6121	0.8366	0.8366	0.7953	0.7953	2.634894	1.636786	1.392799	0.439318
Woman	3	25.5057	25.5057	24.8928	25.5057	0.8791	0.8791	0.8746	0.8743	2.858834	1.657808	1.404298	0.445408
	4	27.5331	27.5331	27.2864	27.4419	0.9067	0.9067	0.9010	0.9021	2.991491	1.677934	1.420751	0.450746
	5	28.9783	28.9443	27.5908	27.6430	0.9207	0.9201	0.8991	0.9010	3.052162	2.014114	1.430122	0.471498

where  $\alpha$ ,  $\beta$ ,  $\delta$  and  $rand$  are random numbers which fall within the range of [0,1], whereas  $G$  represents the current generation number. The individual spiders  $x_c$  and  $x_b$  represent the nearest member to the female spider  $x_i$  that hold a higher weight and the best individual of the entire population, respectively.

If female spiders decide to move away from the source, the following formula is used to update their position (Cuevas et al., 2013):

$$f_i(G + 1) = f_i(G) - \alpha \cdot Vibc_i \cdot (x_c - f_i(G)) - \beta \cdot Vibb_i \cdot (x_b - f_i(G)) + \delta \cdot (rand - 0.5) \quad (10)$$

Therefore, during the optimization process the motion of a female spider is subject to either an attraction force or a repulsion force. The choice of one of them is done using a simple probabilistic rule. A uniform random number  $r_m$  is generated within the range [0,1]. If  $r_m$  is smaller than a given threshold  $PF$ , an attraction movement is generated using Eq. (9); otherwise, a repulsion movement is produced using Eq. (10) (Cuevas et al., 2013).

Male spiders have different position updating operators. First, the male spider population is divided into dominated ( $D$ ) and non-dominated ( $ND$ ) according to their fitness values. The  $D$  spiders have better fitness values compared to  $ND$  spiders. To define the class of each male spider ( $D$  or  $ND$ ), the median of the fitness (or weight) of all male spiders is calculated at every generation. Then, the spiders with fitness value below the calculated median value are considered as  $D$  spiders and the rest as  $ND$  spiders. The  $D$  spiders are attracted to the closest female whereas the  $ND$  spiders tend to move toward the center of the male population. This is governed by the following position updating equation (Cuevas et al., 2013),

$$m_i(G + 1) = \begin{cases} m_i(G) + \alpha \cdot Vibf_i \cdot (x_f - m_i(G)) + \delta \cdot (rand - 0.5) & \text{if } w_{N_{f+i}} > w_{N_{f+m}} \\ m_i(G) + \alpha \cdot \left( \frac{\sum_{h=1}^{N_{pm}} m_{h_i}(G) \cdot w_{N_{f+h}}}{\sum_{h=1}^{N_{pm}} w_{N_{f+h}}} - m_i(G) \right) & \text{if } w_{N_{f+i}} \leq w_{N_{f+m}} \end{cases} \quad (11)$$

**Table 7**

The mean of fitness function and standard deviation values obtained by between-class variance-based optimization algorithms.

Image	K	SSO		FP		BAT		PSO	
		Mean values	STD values	Mean values	STD values	Mean values	STD values	Mean values	STD values
101085	2	2755.807806	0.000000	2755.807806	0.000000	2755.709380	1.2439e-01	2755.514658	9.4594e-01
	3	2960.771941	4.6252e-13	2960.749455	6.3406e-02	959.861346	9.0470e-01	957.396103	3.5315e+00
	4	3063.479956	1.0781e-01	3062.916110	3.8369e-01	3061.550907	1.3762e+00	3055.395609	4.8360e+00
Wherry	5	3115.684102	1.3362e-01	3114.134016	9.1868e-01	3113.374993	1.3758e+00	3102.561189	9.4533e+00
	2	3083.794184	0.000000	3083.794184	0.000000	3083.631775	1.8101e-01	3083.177982	1.0331e+00
	3	3326.678303	0.000000	3326.671884	1.3365e-02	3325.611918	9.4991e-01	3323.968090	4.9648e+00
Zebra	4	3390.260120	1.2004e-01	3389.934790	3.8786e-01	3388.306526	2.4101e+00	3381.968514	1.0207e+01
	5	3422.878517	1.3876e-12	3420.781415	1.3441e+00	3418.157973	5.7730e+00	3415.935186	6.0745e+00
	2	817.642915	0.000000	817.642915	0.000000	817.434638	3.9118e-01	817.176908	1.2271e+00
Butterfly	3	915.103043	5.7815e-13	914.071840	4.4477e-02	914.181937	1.4492e+00	913.304935	1.9136e+00
	4	964.935915	5.7815e-13	964.442422	4.6398e-01	963.216303	1.1328e+00	960.558617	3.6351e+00
	5	990.048155	3.4689e-13	988.551978	7.9168e-01	987.137965	2.2357e+00	984.000586	3.6525e+00
Bird	2	2535.369256	0.000000	2535.369256	0.000000	2535.386521	9.4801e-02	2535.148632	7.1213e-01
	3	3646.019968	0.000000	3645.977107	6.6443e-02	3644.852933	9.1125e-01	3643.228657	2.4584e+00
	4	3712.025517	1.7324e-01	3711.485492	4.0190e-01	3709.907325	1.4669e+00	3704.655071	4.6232e+00
Landscape	5	3760.337745	7.1252e-03	3758.141228	1.0948e+00	3756.853872	2.9815e+00	3749.859524	6.0015e+00
	2	2535.369256	0.000000	2535.450499	0.000000	2535.386521	9.4801e-02	2535.148632	7.1213e-01
	3	2611.868216	4.9742e-02	2611.869820	5.0121e-02	2611.494569	3.3590e-01	2610.941891	1.1415e+00
Ostrish	4	2649.686448	0.000000	2649.479977	1.3132e-01	2648.711394	8.9887e-01	2646.235553	3.5071e+00
	5	2667.652853	1.0935 e-01	2657.261579	9.1886e-01	2666.463349	1.3928e+00	2663.287780	3.3216e+00
	2	4419.698879	0.000000	4419.790621	0.000000	4419.656468	2.0865e-01	4419.132930	1.1754e+00
86016	3	4721.188536	0.000000	4721.186371	7.7080e-03	4720.672689	6.5879e-01	4718.012327	4.9092e+00
	4	4783.767168	8.8425e-02	4783.497446	1.9785e-01	4782.958775	7.0401e-01	4779.764388	3.9817e+00
	5	4823.642659	7.8516e-01	4822.612106	8.1439e-01	4820.468547	5.9719e+00	4815.048972	7.3354e+00
Starfish	2	1035.428099	0.000000	1035.428099	0.000000	1035.170949	3.5470e-01	1035.214138	3.3275e-01
	3	1099.342110	2.1812e-01	1099.334561	8.5927e-02	1098.335241	1.3024e+00	1097.106325	2.6940e+00
	4	1139.970298	1.9417e-03	1139.477774	3.6071e-01	1138.701162	8.9237e-01	1135.243262	3.8515e+00
House	5	1168.711705	3.1840e-02	1166.733125	7.8721e-01	1164.751001	4.2556e+00	1161.120424	5.3029e+00
	2	1236.707999	0.000000	1236.707999	0.000000	1236.576422	2.9094e-01	1236.249554	7.3786e-01
	3	1289.941891	2.3126e-13	1288.918155	3.7457e-02	1289.691850	2.3173e-01	1289.277615	6.9473e-01
Woman	4	1321.742604	0.000000	1321.410057	2.5886e-01	1320.867619	6.6656e-01	1318.412984	2.9343e+00
	5	1337.349287	9.2504e-13	1336.227317	4.9664e-01	1334.570132	3.2463e+00	1332.960010	2.7690e+00
	2	2546.884578	0.000000	2546.884578	0.000000	2541.696914	2.0367e-01	2541.564276	6.9538e-01
Lake	3	2779.488014	4.6252e-13	2774.917215	2.3302e-02	2773.175135	1.6711e+00	2770.278087	6.9775e+00
	4	2865.126347	2.3126e-12	2860.099720	4.2017e-01	2858.568678	1.3098e+00	2854.082159	5.1571e+00
	5	2911.108870	4.4304e-01	2910.038012	9.8584e-01	2904.294574	2.0440e+00	2897.004696	6.8328e+00
Starfish	2	3163.410896	0.000000	3163.410896	0.000000	3146.975278	2.1562e-01	3150.553708	2.6547e-01
	3	3235.843484	1.3876e-12	3235.705426	1.8646e-02	3218.388845	8.2865e-01	3221.217793	1.6325e+00
	4	3263.057955	3.1126e-02	3263.054904	2.7060e-01	3243.329175	2.1116e+00	3244.698498	5.2817e+00
Wherry	5	3268.331684	3.5951e-02	3264.231674	6.9700e-01	3263.799040	3.5602e+00	3263.344190	5.5993e+00
	2	3741.345417	1.3876e-12	3741.345417	1.3876e-12	3741.230175	1.3800e-01	3741.238245	1.6949e-01
	3	3876.946247	1.3876e-12	3876.946247	1.3876e-12	3876.084605	1.2676e+00	3875.477913	4.1545e+00
Zebra	4	3943.249678	2.4847e-02	3943.233990	4.0957e-02	3942.043374	1.1259e+00	3936.963570	1.2028e+01
	5	3977.214694	1.4233e-01	3977.143001	2.5661e-01	3974.927120	2.3478e+00	3960.859530	1.7562e+01
	2	3472.199744	1.8501e-12	3472.162112	0.000000	3472.003337	1.3413e-01	3471.596965	8.3963e-01
Butterfly	3	3623.491238	1.3876e-12	3622.444523	3.5895e-02	3622.956625	5.1605e-01	3621.043654	3.2475e+00
	4	3685.973763	1.6552e-03	3684.623013	3.5565e-01	3684.924941	9.3334e-01	3681.119284	4.3159e+00
	5	3717.736687	9.2504e-13	3715.922921	5.3218e-01	3715.556258	1.5291e+00	3711.587688	3.4906e+00

where  $x_f$  indicates the nearest female spider to the male individual, with the vibration  $Vibf_i$  whereas  $\sum_{h=1}^{N_{pm}} ma_h(G) \cdot w_{N_f+h} / \sum_{h=1}^{N_{pm}} w_{N_f+h}$  correspond to the weighted mean of the male population  $Ma$ .

Once the position of each spider in the population (female and male) is updated, the mating operation is performed to generate new population. Mating in a social-spider colony is performed by dominant males and the female members of the population (Cuevas et al., 2013). When a dominant spider male locates a set of female members who are within a certain radius called mating radius  $r$ , it mates, generating a new brood (Cuevas et al., 2013).

The mating radius  $r$  is defined by the following equation:

$$r = \frac{\sum_{j=1}^K (g_{max} - g_{min})}{2K} \quad (12)$$

where  $K$  is the given threshold number,  $g_{min}$  and  $g_{max}$  are the minimum and the maximum gray levels in the image, respectively.

In the mating process, the weight of each involved spider defines the probability of influence of each individual into the new

brood. The influence probability of each member is set using the roulette method (Cuevas et al., 2013).

If the new spider is better than the worst spider, the worst spider is replaced by the new one. Otherwise, the new spider is discarded and the population is not changed.

This process is repeated until the maximum number of cycles (fixed by the users) is reached. The optimal solution is represented by the spider (solution) with the higher fitness value.

The outline of the SSO algorithm for multilevel image thresholding is given in Algorithm 1.

### 3.2. The FP based multilevel thresholding algorithm

The flower pollination (FP) Algorithm is a new nature inspired metaheuristic proposed by Yang in the year 2012 (Yang, 2012). Flower pollination can be achieved either through self-pollination or cross-pollination. Self-pollination occurs without the aid of pollen vectors such as wind or insects and when pollen from one flower pollinates the same flower or other flowers of the same

### Algorithm 1

```

1: Input:  $Np$ =population size,  $K$ =the number of thresholds,
2: Define the number of females  $N_f$  and male  $N_m$  using Eq. (4)
3: Initialize a population of  $Np$  spiders in random positions using Eq. (6)
4: Evaluate each solution  $x_i$  ( $i=1\dots Np$ ) using Eq. (2) or Eq. (3)
5: Find the best solution  $best$ , and the worst solution  $worst$ 
6: Define the threshold  $PF \in [0, 1]$ 
7: Calculate the radius of mating using Eq. (12)
8: Max_gen=maximum number of generation
9:  $G=0$ 
10: while  $G < \text{Max\_gen}$  Do
11:   for  $i=1 : Np$  do
12:     Calculate the weight of each spider  $x_i$  using Eq. (8)
13:   end if
14:   for  $i=1 : N_f$  do
15:     Move female spiders according to Eqs. (9) and (10)
16:   end if
17:   for  $i=1 : N_m$  do
18:     Move the male spiders according to Eq. (11)
19:   end if
20:   Mate the dominant males and females located in the mating radius to create new spiders.
21:   Evaluate each new spider using Eq. (2) or Eq. (3)
22:   Calculate the weight of each new spider using Eq. (8)
23:   Produced new spiders are substituted with the worst spiders if they have higher weight values
24:   Find the best solution  $best$ , and the worst solution  $worst$ 
25:    $G=G+1$ 
26: end while
27: Output best recorded solution  $best$  with the highest fitness value of the population

```

individual. It is also known as abiotic pollination. Conversely, cross-pollination is biotic as it occurs only when the pollen is delivered to a flower from a different plant through pollinators (Yang, 2012). Biotic cross-pollination may occur at long distance and lead to global pollination as the pollinators such as bees, bats, birds and flies can fly a long distance. In addition, bees and birds may behave according to a Levy flight behavior (Pavlyukevich, 2007) where jump or fly distance steps obey a Levy distribution. Furthermore, flower constancy can be used as an increment step using the similarity or difference of two flowers (Yang, 2012).

The FP algorithm is based on the four following simple rules (Yang, 2012):

- Rule 1*) Biotic cross-pollination can be considered as a global pollination process, and pollen-carrying pollinators move according to Lévy flight.
- Rule 2*) For local pollination, abiotic and self-pollination are used.
- Rule 3*) Pollinators such as insects can develop flower constancy, which is equivalent to a reproduction probability that is proportional to the similarity of two flowers involved.
- Rule 4*) The interaction or switching of local pollination and global pollination can be controlled by a switch probability  $p \in [0, 1]$ .

Yang's model is an attempt to implement the aforementioned four rules with the assumption that each plant has only one flower and each flower produces only one pollen gamete. As a consequence, in the FP algorithm proposed by Yang, a problem's solution refers equivalently to a flower or to a pollen gamete.

Similar to existing metaheuristic algorithms, the FP algorithm uses local and global search techniques in quest to find the optimal solution. The global search simulates the cross-pollination process and uses the best solution to generate new ones according to a Lévy distribution. The local search simulates the self-pollination process, and generates a new candidate solution from a previous one and two others solutions chosen randomly from the population.

In our proposal, the FP algorithm has been adapted to solve the multilevel thresholding problem as follows. A population  $X = (x_1, x_2, \dots, x_{Np})$  of  $Np$  individual pollen is created. Each individ-

ual pollen or solution  $x_i$  in generation  $G$  is encoded as shown in Eq. (5). Each pollen  $x_i$  ( $i=1\dots Np$ ) uses  $K$  decision variables such that each one represents a different threshold point  $t_{ij}$  ( $j=1\dots K$ ) that is used for the multilevel image thresholding.

The position of each individual pollen in the population is initialized using the Eq. (6) and evaluated using Eq. (2) or Eq. (3) depending on the used objective function.

Yang's mathematical model simulates the global pollination using the following equation:

$$x_i(G+1) = x_i(G) + \gamma \cdot L \cdot (x_i(G) - best) \quad (13)$$

where  $x_i(G)$  is the pollen  $x_i$  at generation  $G$ , and  $best$  is the current best solution found among all solutions in the current generation.  $\gamma$  is a scaling factor to control the step size. In addition,  $L$  is the parameter that corresponds to the strength of the pollination, which is also the step size randomly drawn from the Lévy distribution (Yang, 2012).

In FP algorithm, the global pollination occurs with probability  $p$  defined by the so-called switch probability. If this phase is omitted local pollination takes place instead.

In the local pollination process, the second rule of flower pollination is used together with the flower constancy (the third rule). The solution in the next generation is updated according to the solutions of two flowers in the previous generation as follows (Yang, 2012):

$$x_i^1(G+1) = .x_i(G) + \varepsilon(x_j(G) - x_k(G)) \quad (14)$$

where  $x_j^{G+1}$  and  $x_k(G)$  are pollen from different flowers of the same plant species in generation  $G$ , which is equivalent to the random choice of two solutions of the population. This essentially mimics the flower constancy in a limited neighborhood.  $\varepsilon$  is drawn from a uniform distribution in  $[0,1]$  used to implement a random walk (Yang, 2012).

Computationally speaking, after the initialization step, pollen individuals or candidate solutions undergo an iterative process where their positions are updated according to Eq. (13) or Eq. (14) depending on the probabilistically triggered pollination mechanism global or local. The best pollen/solution obtained at the last generation provides the optimal solution to the multilevel thresholding problem i.e., the optimal set of thresholds  $(t_1^*, t_2^*, \dots, t_K^*)$ .

**Table 8**

The mean of fitness function and Standard Deviation values obtained by Kapur-based optimization algorithms.

Image	K	SSO		FP		BAT		PSO	
		Mean values	STD values	Mean values	STD values	Mean values	STD values	Mean values	STD values
101085	2	12.847082	9.0336e-15	12.847082	9.0336e-15	12.846467	1.0063e-03	12.845013	2.6568e-03
	3	15.959464	7.1675e-05	15.953126	2.1800e-04	15.953650	8.5334e-03	15.951629	1.1977e-02
	4	19.013666	3.6134e-15	19.004215	3.7700e-03	18.991473	1.1628e-02	18.952526	6.4997e-02
Wherry	5	21.743031	2.7665e-04	21.724600	7.1860e-03	21.722931	1.0391e-02	21.652565	6.5708e-02
	2	12.177692	7.2269e-15	12.177692	7.2269e-15	12.177356	4.4009e-04	12.176716	1.5701e-03
	3	15.130379	2.5784e-04	15.121212	9.0460e-03	15.121468	2.2908e-02	15.115549	2.6030e-02
Zebra	4	17.971397	2.7893e-03	17.968485	9.1520e-02	17.939004	3.3044e-02	17.903408	5.1382e-02
	5	20.667979	3.3372e-03	20.649552	8.5003e-02	20.603944	4.4341e-02	20.560340	6.9285e-02
	2	12.258460	1.8067e-15	12.258460	1.8067e-15	12.254829	3.0102e-03	12.252177	7.7476e-03
Butterfly	3	15.338314	5.4202e-15	15.337369	9.5600e-04	15.328512	6.5807e-03	15.319011	2.7735e-02
	4	18.161613	2.4011e-04	18.160416	8.7320e-03	18.123871	2.7703e-02	18.100533	3.2702e-02
	5	20.953416	7.5330e-03	20.920384	3.2676e-02	20.862875	6.9188e-02	20.768997	9.5542e-02
Bird	2	12.683959	5.4202e-15	12.683959	5.4202e-15	12.682811	9.1336e-04	12.680177	4.9820e-03
	3	15.798697	1.6816e-05	15.791334	6.0700e-04	15.795200	6.0511e-03	15.780660	2.5981e-02
	4	18.804220	7.2269e-15	18.794949	6.2310e-03	18.778072	1.3287e-02	18.710935	8.3932e-02
Landscape	5	21.478276	4.3865e-03	21.459091	1.1698e-02	21.440207	2.4808e-02	21.347822	8.5569e-02
	2	12.088437	1.8067e-15	12.088437	1.8067e-15	12.087852	5.7062e-04	12.086797	2.6193e-03
	3	15.198959	2.4005e-05	15.198916	2.0900e-04	15.192090	7.0867e-03	15.187560	1.2662e-02
Ostrish	4	17.982371	1.2412e-03	17.977678	3.9190e-03	17.968736	1.0673e-02	17.916202	6.6405e-02
	5	20.577719	7.1433e-03	20.569326	1.6997e-02	20.549839	4.5753e-02	20.446394	8.5110e-02
	2	11.551304	5.4202e-15	11.551304	5.4202e-15	11.548819	3.9589e-03	11.547993	8.1371e-03
86016	3	14.474184	1.6247e-03	14.466690	8.6250e-03	14.461234	2.3029e-02	14.448404	2.4348e-02
	4	17.281434	8.7432e-03	17.296670	1.7801e-02	17.228584	6.6439e-02	17.147524	1.0129e-01
	5	19.848489	1.0449e-03	19.835974	2.7728e-02	19.769436	7.6097e-02	19.650448	1.0780e-01
Starfish	2	12.372877	7.2269e-15	12.372877	7.2269e-15	12.365772	6.8435e-03	12.364024	9.0320e-03
	3	15.466152	1.0840e-14	15.465845	3.1800e-04	15.443076	6.8122e-02	15.436767	2.7665e-02
	4	18.117494	3.0689e-04	18.109530	4.4580e-02	18.092351	3.6142e-02	18.053506	4.3628e-02
House	5	20.725508	4.1389e-03	20.755583	2.2649e-02	20.709123	3.3722e-02	20.634236	8.3558e-02
	2	11.516534	0.0000000	11.516534	0.0000000	11.515755	1.1329e-03	11.514123	3.8304e-03
	3	14.215585	4.4360e-05	14.215238	3.6000e-04	14.210531	6.0335e-03	14.190446	3.4878e-02
Woman	4	16.703804	8.6865e-04	16.698729	4.3500e-03	16.687526	1.5114e-02	16.646412	3.9767e-02
	5	19.034767	1.0753e-04	19.011485	8.8180e-02	19.006144	1.5295e-02	18.957200	4.9132e-02
	2	12.992038	3.6134e-15	12.992038	3.6134e-15	12.991546	6.6527e-04	12.991467	1.0921e-03
Lake	3	16.156850	7.2269e-15	16.156850	7.2269e-15	16.153591	2.7725e-03	16.147936	1.0496e-02
	4	19.105459	2.3236e-04	19.104837	3.9122e-04	19.095579	8.3154e-03	19.065675	3.4642e-02
	5	21.861960	7.2269e-15	21.857416	3.0217e-03	21.837400	1.3432e-02	21.741322	7.3193e-02
86016	2	10.688713	1.6872e-15	10.688693	1.6872e-15	10.688663	8.7706e-02	10.676531	5.2601e-02
	3	13.520511	1.5061e-04	13.511932	1.9450e-03	13.348679	2.4402e-01	13.394050	2.1309e-01
	4	15.725498	1.6436e-04	15.715339	4.4195e-02	15.623268	1.1541e-01	15.596690	1.2815e-01
Lake	5	18.446604	1.0414e-03	18.425864	2.8896e-2	18.377499	9.4385e-02	18.235316	1.1597e-01
	2	12.464464	5.4202e-15	12.464464	5.4202e-15	12.464173	4.0922e-04	12.463629	1.8478e-03
	3	15.583941	3.6134e-15	15.583941	3.6134e-15	15.582078	1.5886e-03	15.577445	9.2046e-03
Woman	4	18.495784	1.0840e-14	18.495410	3.9483e-04	18.487788	5.7690e-03	18.442217	4.6505e-02
	5	21.174135	2.5348e-04	21.170475	2.7359e-03	21.151935	1.5822e-02	21.070425	7.9133e-02
	2	12.450088	5.4202e-15	12.450088	5.4202e-15	12.449493	6.0649e-04	12.449622	4.7934e-04
Woman	3	15.463663	7.2269e-15	15.463660	1.1965e-05	15.455102	8.8160e-03	15.455396	1.2938e-02
	4	18.244637	2.3402e-04	18.244584	5.2695e-04	18.231574	1.3184e-02	18.184194	9.6761e-02
	5	20.860843	6.1621e-02	20.875967	3.7054e-03	20.832177	3.1360e-02	20.663857	2.3620e-01

The outline of the FP algorithm for multilevel image thresholding is given [Algorithm 2](#).

#### 4. Results and discussions

In order to evaluate the performance of the proposed SSO and FP based multilevel thresholding algorithms, twelve benchmark images with different complexities have been used as illustrated in [Fig. 1](#). As can be seen from [Fig. 1](#), these images have irregular distributions. Under such circumstances, classical methods face great difficulties to find the best threshold values.

The results of the proposed algorithms have been compared with those obtained by the PSO ([Kennedy & Eberhart, 1995](#)) and BAT ([Alihodzic and Tuba, 2013](#); [Yang, 2011](#)) algorithms. All the used metaheuristics in this study have two common control parameters: the size of the population  $N_p$  and the maximum number of iteration or generation  $Max\_gen$ . The number of generations cannot be accepted as a time measure since the algorithms perform a different amount of works in their inner loops. For this reason, we use

in this paper, the function evaluation number ( $Max\_FEN$ ) instead of the maximum number of generations.

As the values of these two control parameters ( $N_p$  and  $Max\_FEN$ ) have a great impact on the convergence and the computing time, they were chosen to be the same for all experiments in this paper.  $N_p$  was fixed to 50 and  $Max\_FEN$  was fixed to 10,000.

Using the guidelines provided in the respective literatures, the parameter PF has been set to 0.7 in the SSO algorithm and the parameter P was fixed to 0.8 in the FP algorithm. The parameter setting of the parameters in the PSO and the BAT algorithms is tabulated in [Table 1](#). All the methods are validated through numerical simulations in Matlab on a computer having an Intel Core Duo processor (1.66 GHz) and 2 GB memory.

An exhaustive search method has also been conducted to derive the optimal solution for each number  $K$  of thresholds. When the number of thresholds is over 4, the consumption of CPU time of the exhaustive search exceeds reasonable limits. These ground truth data are summarized in [Table 2](#) where the values of the thresholds along with the corresponding fitness function values

**Table 9**

Statistical analysis of 30 runs for each of 48 cases for the experiments on between-class variance method and Kapur's entropy criterion.  $p$ : probability of the statistic,  $h=0$  indicates null hypothesis cannot be rejected.

Image	K	Between-class variance						Kapur's entropy					
		SSO vs. FP		SSO vs. BAT		SSO vs. PSO		SSO vs. FP		SSO vs. BAT		SSO vs. PSO	
		P	h	P	h	P	h	P	h	P	h	P	h
101085	2	NaN	0	0.000000	1	0.000000	1	NaN	0	0.000001	1	0.000000	1
	3	0.023012	1	0.000000	1	0.000000	1	0.063101	0	0.000563	1	0.000003	1
	4	0.000000	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1
	5	0.009896	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1
	Wherry	2	NaN	0	0.000000	1	0.000005	1	NaN	0	0.000000	1	0.000000
	3	NaN	0	0.000000	1	0.000000	1	0.021538	1	0.000001	1	0.000001	1
	4	0.002689	1	0.000000	1	0.000000	1	0.041890	1	0.000346	1	0.000000	1
	5	0.000000	1	0.000000	1	0.000000	1	0.030174	1	0.000006	1	0.000000	1
Zebra	2	NaN	0	0.000000	1	0.000005	1	NaN	0	0.000425	1	0.000000	1
	3	0.033711	1	0.000000	1	0.000000	1	NaN	0	0.000000	1	0.000000	1
	4	0.000000	1	0.000000	1	0.000000	1	0.040569	1	0.000111	1	0.000000	1
	5	0.000001	1	0.000000	1	0.000000	1	0.007657	1	0.007279	1	0.000000	1
Butterfly	2	NaN	0	0.000000	1	0.000001	1	NaN	0	0.000000	1	0.000000	1
	3	0.151529	0	0.000000	1	0.000000	1	0.333711	0	0.000000	1	0.000000	1
	4	0.002181	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1
	5	0.000004	1	0.000000	1	0.000000	1	0.000154	1	0.000000	1	0.000000	1
Bird	2	NaN	0	0.000066	1	0.000000	1	NaN	0	0.003596	1	0.000000	1
	3	0.333711	0	0.000000	1	0.000000	1	0.333711	0	0.000000	1	0.000000	1
	4	0.000026	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1
	5	0.040528	1	0.000032	1	0.000000	1	0.000081	1	0.000000	1	0.000000	1
Landscape	2	NaN	0	0.000002	1	0.000001	1	NaN	0	0.000000	1	0.000001	1
	3	0.003241	1	0.000000	1	0.000000	1	0.000000	1	0.000006	1	0.000000	1
	4	0.000000	1	0.000000	1	0.000000	1	0.001943	1	0.000446	1	0.000001	1
	5	0.000000	1	0.000000	1	0.000000	1	0.023444	1	0.000000	1	0.000000	1
Ostrich	2	NaN	0	0.000000	1	0.000001	1	NaN	0	0.000000	1	0.000000	1
	3	0.289207	0	0.000000	1	0.000000	1	NaN	0	0.000000	1	0.000000	1
	4	0.042891	1	0.000000	1	0.000000	1	0.031433	1	0.000000	1	0.000000	1
	5	0.001594	1	0.000000	1	0.000000	1	0.000001	1	0.000000	1	0.000001	1
86016	2	NaN	0	0.000013	1	0.000313	1	NaN	0	0.000002	1	0.000000	1
	3	0.013920	1	0.000000	1	0.000000	1	0.333711	0	0.000000	1	0.000000	1
	4	0.000000	1	0.000000	1	0.000000	1	0.000006	1	0.000000	1	0.000000	1
	5	0.000000	1	0.000000	1	0.000000	1	0.039004	1	0.000000	1	0.000000	1
Starfish	2	NaN	0	0.000000	1	0.000000	1	NaN	0	0.000000	1	0.000013	1
	3	0.023410	1	0.000000	1	0.000000	1	NaN	0	0.000000	1	0.000000	1
	4	0.000000	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1
	5	0.013231	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1
House	2	NaN	0	0.000000	1	0.000001	1	0.000313	1	0.007739	1	0.004433	1
	3	0.333711	0	0.000000	1	0.000000	1	0.000000	1	0.000698	1	0.000000	1
	4	0.934079	0	0.007667	1	0.003841	1	0.000063	1	0.005056	1	0.000038	1
	5	0.000397	1	0.005616	1	0.002163	1	0.004804	1	0.006192	1	0.000000	1
Lake	2	NaN	0	0.000002	1	0.000000	1	NaN	0	0.000000	1	0.000000	1
	3	NaN	0	0.000000	1	0.000000	1	NaN	0	0.000000	1	0.000000	1
	4	0.000002	1	0.000001	1	0.000000	1	0.093869	0	0.077296	0	0.000001	1
	5	0.000000	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1
Woman	2	NaN	0	0.000000	1	0.000013	1	NaN	0	0.000000	1	0.000000	1
	3	0.004321	1	0.000000	1	0.000000	1	0.005643	1	0.000000	1	0.000000	1
	4	0.000000	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1
	5	0.034232	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1	0.000000	1

and CPU time are given. It is apparent from Table 2 that the computation time of the exhaustive search method grows in many orders of magnitude with the number of required thresholds. It becomes unacceptable for  $K$  greater than 3. For this reason, when the number of thresholds is greater than 4, there are no values for the exhaustive search listed in our experiments. The other results attained by the optimization methods will be compared with them.

Tables 3 and 4 show the optimal threshold and fitness values obtained by the four algorithms over 30 runs for all test images when using Otsu's method and Kapur's entropy as objective functions respectively. The results in Tables 3 and 4 indicate that all the methods performed equally when the number of thresholds  $K$  is equal to two. When  $K$  is equal to three, the SSO and PF algorithms have the same performance and are better than PSO and BAT algorithms in most of the images. For a higher value of  $K$ , the SSO algorithm is better than the FP algorithm in most of the cases, which indicate that the FP algorithm is more sensitive to

the increase of the threshold level than the SSO algorithm. From the above tables, we find also that the threshold values along with the corresponding fitness values found by SSO algorithm are equal to or very close to optimal thresholds derived by the exhaustive search method based Otsu's method or Kapur's entropy for all values of  $K$ .

Quantitatively speaking, the quality of the segmented images using the obtained thresholds has been evaluated using the two popular performance measures: the indicator peak signal to noise ratio (PSNR) measure (Yin, 2007) and the structural similarity index (SSIM) (Wang, Bovik, Sheikh, & Simoncelli, 2004).

The PSNR quality metric measures the extent to which the segmented image is similar to the original one. Higher is the value of this metric, better is the quality of the achieved segmentation. This measure is defined as:

$$\text{PSNR} = 20\log_{10}\left(\frac{255}{\text{RMSE}}\right) \text{ (dB)} \quad (15)$$



## Algorithm 2

```

1: Input:  $Np$ =population size,  $K$ =the number of thresholds
2: Initialize a population of  $Np$  flowers in random positions using Eq. (6)
3: Evaluate each solution  $x_i$  ( $i=1\dots Np$ ) using Eq. (2) or Eq. (3)
4: Find the best solution  $best$  in the initial population
5: Define the switch probability  $p \in [0, 1]$ 
6: Max_gen=maximum number of generation
7:  $G=0$ 
8: while  $G < \text{Max\_Generation}$  do
9:   for  $i=1 : Np$  do
10:    if  $\text{rand} < p$  then
11:     Draw a d-dimensional step vector  $L$  which obeys a Lévy distribution
12:     Do global pollination using Eq. (13)
13:    else
14:     Draw  $\varepsilon$  from a uniform distribution in  $[0,1]$ 
15:     Do local pollination using Eq. (14)
16:    end if
17:   Evaluate the newly generated solution  $x_i(G+1)$  using Eq. (2) or Eq. (3)
18:   if the newly generated solution is better, replace  $x_i(G)$  by  $x_i(G+1)$ 
19:   end for
20: Update the current best solution  $best$ 
21:  $G=G+1$ 
22: end while
23: Output the best recorded solution  $best$  with the highest fitness value of the population

```

where the RMSE is the root mean-squared error, defined as:

$$\text{RMSE} = \sqrt{\frac{\sum_{i=1}^X \sum_{j=1}^Y (I(i, j) - \text{Seg}(i, j))^2}{X \cdot Y}} \quad (16)$$

Here  $I$  and  $\text{Seg}$  are original and segmented images of size  $X, Y$ , respectively.

The SSIM given by Eq. (17) evaluates the visual similarity between the original image and the reconstructed image. The SSIM index combines luminance comparison, contrast comparison and structure comparison and satisfies symmetry, boundedness and unique maximum properties (Wang et al., 2004):

$$\text{SSIM}(I, \text{Seg}) = \frac{(2\mu_I \mu_{\text{Seg}} + c_1)(2\sigma_{I, \text{Seg}} + c_2)}{(\mu_I^2 + \mu_{\text{Seg}}^2 + c_1)(\sigma_I^2 + \sigma_{\text{Seg}}^2 + c_2)} \quad (17)$$

where  $\mu_I$  is the mean intensity of the image  $I$ ,  $\mu_{\text{Seg}}$  is the mean of the image  $\text{Seg}$ ,  $\sigma_I^2$  represents the variance of  $I$ ,  $\sigma_{\text{Seg}}^2$  represents the variance of  $\text{Seg}$ ,  $\sigma_{I, \text{Seg}}$  is the covariance of  $I$  and  $\text{Seg}$ . According to (Wang et al., 2004),  $c_1$  and  $c_2$  are two constants, such that  $c_1 = (k_1 L)^2$  and  $c_2 = (k_2 L)^2$ ,  $k_1 = 0.01$ ,  $k_2 = 0.03$  and  $L$  is the number of gray levels in the image. A higher value of SSIM shows better performance.

Tables 5 and 6 give the PSNR and SSIM values corresponding to those given in Tables 3 and 4 respectively. We can see that as the number of threshold increases, the PSNR and the SSIM values are increased as well for all algorithms.

The four algorithms have been also compared based on their time complexity. For this purpose, the CPU time of the best solution over 30 runs, has been recorded in Tables 4 and 5. From the above tables, it comes out that Otsu-based methods are faster than Kapur-based ones. Furthermore, the results show that PSO is faster than SSO, FP and BAT algorithms and that SSO exhibits slightly higher CPU time. If a better time span is needed, an optimized coding or parallel implementations of the algorithm should be adopted.

For a visual evaluation, segmented images and their gray level histograms labeled with thresholds found by the SSO algorithm on the Otsu's method and on Kapur's entropy have been given in Figs. 2 and 3 for Bird, Starfish, Lake and Woman images.

On the other hand, another experiment has been conducted to compare the two proposed methods. Fig. 4 shows the evolution of the objective function obtained for Bird, Lake and Woman images.

From Fig. 4, it is possible to see that the SSO algorithm reaches the maximum fitness values using Otsu's method and Kapur's entropy in less iterations than the FP method.

To analyze the efficiency and the stability of the algorithms, the mean and standard deviations over the 30 runs have been presented in Tables 7 and 8 when the Otsu's method and the Kapur's entropy are used as objective function respectively.

The mean  $\mu$  and standard deviation  $\sigma$  are defined as:

$$\mu = \frac{\sum_{i=1}^k \sigma_i}{k}, \sigma = \sqrt{\frac{1}{k} \sum_{i=1}^k (\sigma_i - \mu)^2} \quad (18)$$

where  $k$  is the number of runs for each stochastic algorithm ( $k=30$ ),  $\sigma_i$  is the best objective value obtained by the  $i$ th run of the algorithm.

From Tables 7 and 8, we can see that SSO and FP algorithms give good results both in terms of accuracy (mean fitness) and robustness against PSO and BAT algorithms and have the same performance when the number of thresholds  $K$  is equal to 2. When the value of  $K$  is greater than 2, the SSO algorithm is more robust (high mean values) and more stable than FP algorithm since its standard deviation is very low in all cases.

The SSO performance can be viewed as a consequence of two facts (Cuevas et al., 2013): (i) splitting the population into different individual types (male and female) provides the use of different rates between exploration and exploitation during the evolution process (ii) the use of specialized operators allow a better particle distribution in the search space, increasing the algorithm's ability to find the global optima.

The FP limitations are probably due to its poor exploration process. In fact, in the global pollination equation, the best solution is used to guide the search; this would cause the premature convergence of the algorithm.

A non-parametric statistical test known as the Wilcoxon's rank sum test for independent samples (García, Molina, Lozano, & Herrera, 2009; Wilcoxon, 1945) has been conducted, with a 5% significance level to test the significance of the difference between algorithms.

In the experiments with Otsu's criterion and the entropy values, the SSO algorithm was chosen as the control algorithm and was compared with the FP, BAT and PSO algorithms in terms of the mean value.

Table 9 reports the  $p$ -values produced by Wilcoxon's test for the pairwise comparison of three groups. Such groups are constituted

**Table 10**The best, mean and Standard Deviation of fitness function values obtained by between-class variance and Kapur-based optimization algorithms for  $M = 10, 15, 20$ .

Image	K	Between-class variance based method						Kapur's entropy					
		SSO			FP			SSO			FP		
		Best	Mean	Std	Best	Mean	Std	Best	Mean	Std	Best	Mean	Std
101085	10	3205.46475033	3205.19918675	3.0884e-01	3205.18531918	3204.72998343	3.2530e-01	33.58702435	33.49993605	1.2806e-02	33.54236020	33.44846300	4.1715e-02
	15	3225.79703647	3224.99348451	1.3628e-01	3225.02948620	3224.51066539	3.6589e-01	43.16627884	43.04659273	1.0107e-01	42.88815043	42.67430192	1.0866e-01
	20	3233.08464887	3232.33924116	1.7400e-01	3232.31921228	3231.51594988	3.6871e-01	51.28665574	51.02914798	1.3693e-01	50.36913858	50.12115592	1.4546e-01
Wherry	10	3477.55699340	3476.41468714	1.2307e+00	3477.34420897	3476.68708082	4.2912e-01	32.63086947	32.50938975	1.1500e-02	32.56378711	32.39000593	6.3667e-02
	15	3492.43206949	3491.67536742	2.1058e-01	3492.03195975	3491.07299747	3.3535e-01	41.89415529	41.70217771	1.1084e-01	41.48513678	41.16059045	1.3262e-01
	20	3497.96910447	3497.35505796	1.6576e-01	3497.01940581	3496.49595188	3.0414e-01	49.58901435	49.02975020	1.5882e-01	48.85017408	48.11073838	2.5239e-01
Zebra	10	1035.51824716	1035.45578521	1.1041e-01	1035.12953329	1034.47533216	3.8783e-01	32.30499789	32.19190037	1.8514e-02	32.23104897	32.11906638	4.7114e-02
	15	1046.86948952	1046.20689184	1.8844e-01	1045.99859564	1045.43613871	2.9518e-01	41.17107727	40.94449528	1.7851e-02	40.74135244	40.59807921	9.7061e-02
	20	1051.07659556	1050.57066903	1.8746e-01	1050.15734342	1049.68707446	2.5801e-01	48.41403933	48.06924097	1.3403e-01	47.67041625	47.10631390	2.0141e-01
Butterfly	10	3832.61848989	3831.75464222	1.4414e-01	3832.22624617	3831.82928974	3.3123e-01	33.15027837	33.08088196	1.2371e-02	33.06987428	32.99403523	4.4210e-02
	15	3850.07998218	3849.48102734	1.1240e-01	3849.33892725	3848.79211412	3.2527e-01	42.46318928	42.33271826	1.0829e-01	42.16565448	41.93624173	1.1452e-01
	20	3856.76854393	3856.19973619	1.8306e-01	3855.81912282	3855.32959910	2.5490e-01	50.19957573	49.91426342	1.2586e-01	49.32289967	48.98959966	1.8619e-01
Bird	10	2708.20833458	2707.69938637	1.9175e-01	2708.01420162	2707.42165593	2.7990e-01	31.91089105	31.76068144	1.8289e-02	31.77573169	31.71541755	4.4604e-02
	15	2717.41993593	2716.88005583	1.7623e-01	2716.80299264	2716.54851327	1.7812e-01	40.90639804	40.56039475	1.0733e-01	40.41632330	40.22079879	1.1077e-01
	20	2721.31104979	2720.80651518	1.7133e-01	2720.73644466	2720.15077209	1.9663e-01	48.35833463	47.96357654	2.0456e-01	47.41106014	46.95712135	2.4026e-01
Landscape	10	4857.57353705	4855.86530751	2.3021e-01	4857.40845424	4856.99324242	3.2134e-01	31.10670083	30.82676563	2.1722e-02	30.98030240	30.78494913	7.8593e-02
	15	4866.53079955	4865.98029986	1.4766e-01	4866.07654865	4865.59257263	2.2501e-01	40.06264154	39.57518656	1.1124e-01	39.50318423	39.03514910	1.5228e-01
	20	4870.42108614	4869.62595834	1.7811e-01	4869.73284407	4869.31795421	1.8383e-01	47.21733975	46.71575786	2.1436e-01	46.15867286	45.51822315	2.4805e-01
Ostrish	10	1210.38207190	1209.54189722	2.1981e-01	1209.86618547	1209.16381090	5.6139e-01	32.28622990	32.15535102	3.3082e-02	32.18061647	32.04606064	5.0819e-02
	15	1220.90421109	1219.36205304	2.2811e-01	1219.94359005	1218.97378314	3.9735e-01	41.19354992	40.99918301	1.0051e-01	40.74962003	40.48983623	1.0887e-01
	20	1224.64755538	1223.78106948	3.3026e-01	1223.80044841	1223.04371650	3.7909e-01	48.71456644	48.18226572	1.6341e-01	47.51751534	47.17761082	2.3307e-01
86016	10	1364.84087881	1364.61675242	2.3415e-01	1364.60580628	1364.26064914	2.5491e-01	29.08252335	29.00412180	1.7405e-02	28.98981948	28.88719746	5.1911e-02
	15	1371.60181846	1371.15824463	2.1831e-01	1371.23319216	1370.70576594	2.2469e-01	36.75864093	36.53088039	1.0324e-01	36.37401031	36.10509760	1.1080e-01
	20	1374.26921512	1373.66257514	1.3016e-01	1373.50700746	1373.31210807	1.3769e-01	42.79644313	42.21277958	1.8124e-01	41.57821028	41.25302168	2.3385e-01
Starfish	10	3307.48963675	3306.24442198	2.3413e-01	3307.31289662	3306.77764454	3.7828e-01	33.53869498	33.52205443	1.0840e-02	33.51546638	33.49374210	1.4117e-02
	15	3314.43186921	3313.85544563	3.0656e-01	3314.05139363	3313.34814934	3.0935e-01	42.88827241	42.75666318	1.5428e-02	42.75129606	42.64953748	5.8667e-02
	20	3317.13588292	3316.52745828	1.3478e-01	3316.43110139	3316.15373656	1.7602e-01	50.61197398	50.29974338	4.1651e-02	50.20302670	49.99213139	9.8016e-02
House	10	2982.36997228	2982.24774594	2.3091e-01	2982.02093788	2981.59833933	2.4960e-01	30.63107328	30.46038630	3.1955e-02	30.58395725	30.45437171	8.5795e-02
	15	2999.33495364	2998.96111308	3.4241e-01	2998.57502574	2997.96516451	3.5747e-01	40.05652169	39.62862006	1.1369e-01	39.86928597	39.60006793	1.2229e-01
	20	3005.78988125	3005.17103617	2.2514e-01	3004.66662756	3004.25308672	2.7294e-01	47.94156809	47.36623249	1.2978e-01	47.28554089	46.88203659	1.7505e-01
Lake	10	3776.30576013	3776.10607509	1.3057e-01	3776.12889854	3775.78339076	1.9194e-01	32.08293033	32.01307209	1.8660e-02	32.03590219	31.94856013	5.3066e-02
	15	3789.55971231	3789.24139476	1.0948e-01	3789.09181301	3788.77823882	1.4895e-01	41.00700877	40.87202369	1.1488e-01	40.59442994	40.42732030	1.1787e-01
	20	3794.64180764	3794.26009034	1.7039e-01	3793.86997525	3793.59638473	1.7227e-01	48.46962412	48.07799682	1.7469e-01	47.49403104	47.09494128	2.4689e-01
Woman	10	4030.70602114	4030.52391240	2.3633e-01	4030.43873674	4030.00475388	2.4285e-01	32.46317650	32.45467349	1.7203e-02	32.44178725	32.42083720	1.8580e-02
	15	4043.69933627	4043.18204542	3.1274e-01	4042.87766770	4042.38218037	3.2877e-01	41.31769111	41.21740733	2.2673e-02	41.19161935	41.10209082	5.1008e-02
	20	4048.58039864	4048.07162620	2.1797e-01	4047.71017738	4047.24817735	2.4083e-01	48.57838379	48.37268501	1.0894e-01	48.12972855	47.92795552	1.1078e-01

by SSO vs. FP, SSO vs. BAT and SSO vs. PSO. As a null hypothesis, it is assumed that there is no significant difference between mean values of the four algorithms. The alternative hypothesis considers a significant difference between the four algorithms.

In the experiments using Otsu's criterion, the SSO algorithm produced better results in 29 cases out of 48 total cases when it is compared with the FP algorithm and produced better results in 48 cases out of 48 total cases when it is compared with the PSO and the BAT algorithms.

In the experiments using the entropy criterion, the SSO algorithm produced better results in 28 cases out of 48 total cases when it is compared with the FP algorithm. SSO algorithm produced better results in 47 cases out of 48 total cases when it is compared with PSO and BAT algorithms.

From Table 9, the Wilcoxon rank sum test states that there is a significant difference between the median values of the SSO algorithm and the FP, BAT and PSO algorithms; in most cases, the SSO algorithm is better than the three other nature-inspired algorithms.

To further verify the searching ability of each population-based method on high dimensionality, we test SSO and FP metaheuristic on each image with  $K=10, 15, 20$ . Table 10 shows the best, the average fitness value and standard deviation obtained by each population-based algorithm. From Table 10, we can observe that still the SSO algorithm shows the best performance and stability on a higher dimension of all images compared to the FP algorithm. The numerical results reported in Table 10 confirm that FP algorithm is more sensitive to the increase of the number of thresholds than the SSO algorithm since its solution quality (mean value) is smaller and its variance is higher in all cases.

## 5. Conclusion and future works

In this study, the problem of image segmentation through multilevel thresholding has been cast as a continuous optimization problem where two objective functions namely the Otsu's method and the Kapur's entropy have been considered. Consequently, two new nature inspired algorithms have been proposed by tailoring the FP and SSO metaheuristics to the problem at hand. Basically, the task achieved by each algorithm is to find the best values of a given number of thresholds that maximize the used objective function. The adaptation scheme include the use of an appropriate solution encoding and the adoption of the general dynamics that underlies the used metaheuristic. A thorough experimental study has been conducted with the aim to assess the ability of the proposed algorithms to solve efficiently the tackled problem and to compare their performance to other state of the art population-based algorithms such as PSO and BAT algorithms. The validity and stability of all the methods have been evaluated both qualitatively and quantitatively using the fitness value and the two popular performance measures PSNR and SSIM. The obtained experimental results using several benchmark test images have shown that all algorithms achieve the same results and accuracy for bi-level thresholding for both objective functions. This is because the problem is relatively easy to solve in this case. In case of three-level thresholding, the SSO and FP algorithms compete with each other and outperform PSO and BAT algorithms on almost all images. For a higher number of thresholds, the SSO algorithm distinguishes itself by its stability and accuracy as the number of thresholds increases. In all cases and for both objective functions, results have shown that the SSO algorithm outperforms all the other three algorithms in terms of achieved fitness values and stability. However, in terms of time complexity, analysis of recorded CPU time required by each algorithm has shown that the SSO algorithm exhibits higher CPU time but scales linearly with the problem dimension. This drawback can be removed with an optimized coding or a parallel implementation of the algorithm.

Consequently, the experimental results suggest that the SSO algorithm for image multilevel thresholding is the most successful and most robust while the FP algorithm gives good results for a small number of thresholds but its performance decreases for higher numbers. This can be explained by the fact that in one hand, the SSO algorithm maintains a good balance between exploration and exploitation while the FP algorithm tends to converge prematurely at early stages of the optimization process. More specifically, the good search capabilities of the SSO algorithm could be seen as a consequence of the use of two types of search agents with different rates between exploration and exploitation ensured by the different specialized operators. The FP limitations are likely due to its poor exploration process. In fact, in the global pollination equation, the best solution is used to guide the search; this would cause a premature convergence that leads the algorithm to get stuck in local optima.

Encouraged by the very promising results of the SSO algorithm, our future attempt is to further investigate its extension to other complex image processing applications like the dynamic multilevel thresholding problem. In fact, when dealing with multilevel thresholding, in addition to the need for accurate and fast estimation of the optimal threshold values, finding the adequate number of levels that allows keeping most of the relevant details of the original image needs to be efficiently handled as well. The SSO algorithm can be also adapted for dealing the multi-objective optimization approach with some additional mechanisms in order to achieve better segmentation. Another future research direction includes the application of the SSO algorithm to color multilevel thresholding as well as to noisy images by considering the two-dimensional histogram without excessive consumption of CPU time.

## Acknowledgments

The authors would like to thank the anonymous reviewers whose valuable comments have greatly improved the presentation of the paper. The authors express also their sincere thanks to Professor Souham Meshoul for its careful reading of the revised paper.

## Reference

- Abdel-Raouf, O., Abdel-Baset, M., & El-Henawy, I. (2014). An improved flower pollination algorithm with chaos. *International Journal of Education and Management Engineering*, 4(2), 1–8.
- Abdel-Raouf, O., Abdel-Baset, M., & El-Henawy, I. (2014). A novel hybrid flower pollination algorithm with chaotic harmony search for solving Sudoku puzzles. *International Journal of Engineering Trends and Technology*, 7(3), 126–132.
- Akay, B. (2013). A study on particle swarm optimization and artificial bee colony algorithms for multilevel thresholding. *Applied Soft Computing*, 13(6), 3066–3091.
- Alihodzic, A., & Tuba, M. (2013). Bat algorithm (BA) for image thresholding. In *Recent researches in telecommunications, informatics, Electronics and Signal Processing* (pp. 364–369).
- Bakhshali, M. A., & Shamsi, M. (2014). Segmentation of color lip images by optimal thresholding using bacterial foraging optimization (BFO). *Journal of Computational Science*, 5(2), 251–257.
- Bayraktar, Z., Komurcu, M., Bossard, J. A., & Werner, D. H. (2013). The wind driven optimization technique and its application in electromagnetics. *IEEE Transactions on Antennas and Propagation*, 61(5), 2745–2757.
- Bayraktar, Z., Turpin, J. P., & Werner, D. H. (2011). Nature-inspired optimization of high impedance metasurfaces with ultrasmall interwoven unit cells. *IEEE Letters on Antennas and Wireless Propagation*, 10, 1563–1566.
- Bekdaş, G., Nigdeli, S. M., & Yang, X.-S. (2015). Sizing optimization of truss structures using flower pollination algorithm. *Applied Soft Computing*, 37, 322–331.
- Bhandari, A. K., Kumar, A., & Singh, G. K. (2015). Modified artificial bee colony based computationally efficient multilevel thresholding for satellite image segmentation using Kapur's, Otsu and Tsallis functions. *Expert Systems with Applications*, 42(3), 1573–1601.
- Bhandari, A. K., Singh, V. K., Kumar, A., & Singh, G. K. (2014). Cuckoo search algorithm and wind driven optimization based study of satellite image segmentation for multilevel thresholding using Kapur's entropy. *Expert Systems with Applications*, 41(7), 3538–3560.
- Brajevic, I., Tuba, M., & Bacanin, N. (2012). Multilevel image thresholding selection based on the cuckoo search algorithm. In *Advances in sensors, signals, visualization, imaging and simulation* (pp. 217–222).

- Chakraborty, D., Saha, S., & Dutta, O. (2014). DE-FPA, a hybrid differential evolution-flower pollination algorithm for function minimization, high perform. In *Conference on high performances computing and applications (ICHPCA'14)* (pp. 1–6).
- Cuevas, E., Cienfuegos, M., Zaldivar, D., & Perez, M. (2013). A swarm optimization algorithm inspired in the behavior of the social-spider. *Expert Systems with Applications*, 40(16), 6374–6384.
- Cheng, H. D., Chen, Y. H., & Sun, Y. (1999). A novel fuzzy entropy approach to image enhancement and thresholding. *Signal Processing*, 75, 277–301.
- Cuevas, E., Sención, F., Zaldivar, D., Pérez-Cisneros, M., & Sossa, H. (2012). A multithreshold segmentation approach based on artificial bee colony optimization. *Applied Intelligence*, 37(3), 321–336.
- Cuevas, E., Zaldivar, D., & Pérez-Cisneros, M. (2010). A novel multi-threshold segmentation approach based on differential evolution optimization. *Expert Systems with Applications*, 37(7), 5265–5271.
- Dirami, A., Hammouche, K., Diaf, M., & Siarry, P. (2013). Fast multilevel thresholding for image segmentation through a multiphase level set method. *Signal Processing*, 104(6), 139–153.
- Fan, S., & Lin, Y. (2007). A multi-level thresholding approach using a hybrid optimal estimation algorithm. *Pattern Recognition Letters*, 28, 662–669.
- Gao, H., Xu, W., Sun, J., & Tang, Y. (2013). Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm. *IEEE Transactions on Instrumentation and Measurement*, 59(4), 934–946.
- Gao, H., Xu, W. B., Sun, J., & Tang, Y. L. (2010). Multilevel thresholding for image segmentation through an improved quantum-behaved particle swarm algorithm. *IEEE Transactions on Instrumentation and Measurement*, 59(4), 934–946.
- García, S., Molina, D., Lozano, M., & Herrera, F. (2009). A study on the use of non-parametric tests for analyzing the evolutionary algorithms' behaviour: a case study on the CEC'2005 special session on real parameter optimization. *Journal of Heuristics*, 15(6), 617–644.
- Ghamisi, P., Couceiro, M. S., Martins, F. M., & Benediktsson, J. A. (2014). Multilevel image segmentation based on fractional-order Darwinian particle swarm optimization. *IEEE Transactions on Geoscience and Remote Sensing*, 52(5), 2382–2394.
- Hammouche, K., Diaf, M., & Siarry, P. (2008). A multilevel automatic thresholding method based on a genetic algorithm for a fast image segmentation. *Computer Vision Image Understanding*, 109(2), 163–175.
- Hammouche, K., Diaf, M., & Siarry, P. (2010). A comparative study of various meta-heuristic techniques applied to multilevel thresholding problem. *Engineering Applications of Artificial Intelligence*, 23, 667–688.
- Hornig, M. H. (2010). A multilevel image thresholding using the honey bee mating optimization. *Applied Mathematics and Computation*, 215(9), 3302–3310.
- Hornig, M. H. (2010). Multilevel minimum cross entropy threshold selection based on the honey bee mating optimization. *Expert Systems with Applications*, 37(6), 4580–4592.
- Hornig, M. H. (2011). Multilevel thresholding selection based on the artificial bee colony algorithm for image segmentation. *Expert Systems with Applications*, 38, 13785–13791.
- Hornig, M. H., & Liou, R. J. (2011). Multilevel minimum cross entropy threshold selection based on the firefly algorithm. *Expert Systems with Applications*, 38(12), 14805–11076.
- Ibrahim, El-H., & Mahmoud, I. (2014). An improved chaotic flower pollination algorithm for solving large integer programming problems. *International Journal of Digital Content Technology and Its Applications*, 8(3), 72–81.
- Kapur, J. N., Sahoo, P. K., & Wong, A. K. C. (1985). A new method for gray-level picture thresholding using the entropy of the histogram. *Computer Vision Graphics Image Processing*, 29, 273–285.
- Kennedy, J., & Eberhart, R. (1995). Particle swarm optimization. In *Proceedings of the IEEE international conference on neural networks: Vol. 4* (pp. 1942–1948).
- Kumar, S., Kumar, P., Sharma, T. K., & Pant, M. (2013). Bi-level thresholding using PSO, artificial bee colony and MRLDE embedded with Otsu method. *Memetic Computing*, 5(4), 323–334.
- Lukasik, S., & Kowalski, P. (2015). Study of flower pollination algorithm for continuous optimization, intelligent systems'2014. *Advances in Intelligent Systems and Computing*, 322, 451–459.
- Ma, M., Liang, J., Guo, M., Fan, Y., & Yin, Y. (2011). SAR image segmentation based on artificial bee colony algorithm. *Applied Soft Computing*, 11(8), 5205–5214.
- Maitra, M., & Chatterjee, A. (2008). A hybrid cooperative-comprehensive learning based PSO algorithm for image segmentation using multilevel thresholding. *Expert Systems with Applications*, 34, 1341–1350.
- Manikandan, S., Ramar, K., Willjuice, I. M., & Srinivasagan, K. G. (2014). Multilevel thresholding for segmentation of medical brain images using real coded genetic algorithm. *Measurement*, 47, 558–568.
- Mulya, M. M., Hadi, S. P., & Ali, H. R. (2014). Flower pollination algorithm for optimal control in multi-machine system with GUPFC. In *6th international conference on information technology and electrical engineering (ICITEE)* (pp. 1–6).
- Oliva, D., Cuevas, E., Pajares, G., Zaldivar, D., & Osuna, V. (2014). A multilevel thresholding algorithm using electromagnetism optimization. *Neurocomputing*, 139, 357–1118.
- Oliva, D., Cuevas, E., Pajares, G., Zaldivar, D., & Perez-Cisneros, M. (2013). Multilevel thresholding segmentation based on harmony search optimization. *Journal of Applied Mathematics*, 2013, 1–24.
- Otsu, N. (1979). A threshold selection method from gray level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 62–66.
- Ouadfel, S., & Meshoul, S. (2014). Bio-inspired algorithms for multilevel image thresholding. *International Journal of Computer Applications in Technology, Special Issue on Computational Optimization and Engineering Applications*, 49(3/4), 207–226.
- Pal, N. R., & Pal, S. K. (1993). Pattern Recognition. *Expert Systems with Applications*, 26(9), 1274–1294.
- Panda, R., Agrawal, S., & Bhuyan, S. (2013). Edgemagnitude based multilevel thresholding using Cuckoo search technique. *Expert Systems with Applications*, 40(18), 7617–7628.
- Pavlyukevich, I. (2007). Levy flights, non-local search and simulated annealing. *Journal of Computational Physics*, 226, 1830–1844.
- Peng, B., Zhang, L., & Zhang, D. (2013). A survey of graph theoretical approaches to image segmentation. *Pattern Recognition*, 46(3), 1020–1038.
- Pereira, D. R., Delpiano, J., & Papa, J. P. (2014). Evolutionary optimization applied for fine-tuning parameter estimation in optical flow-based environments. In *2014 27th SIBGRAPI conference on graphics, patterns and images (SIBGRAPI)* (pp. 125–132). IEEE.
- Platt, G. M. (2014). Application of flower pollination algorithm in nonlinear algebraic systems with multiple solutions. In *Proceedings of the international conference on engineering optimization (ENGOPT 2014), Lisbon, Portugal, 8–11 September 2014*.
- Prathiba, R., Balasingh, M., & Moses, S. S. (2014). Flower pollination algorithm applied for different economic load dispatch problems. *International Journal of Engineering and Technology (IJET)*, 6(April–May (2)), 1009–1016.
- Priyadharshini, V., Divya, P., Preethi, D., Pazhaniraja, N., & Paul, P. V. (2015). A novel Web service publishing model based on social spider optimization technique. *Computation of Power, Energy Information and Communication (ICCPEIC)*, 2015, 0373–0387.
- Pun, T. (1981). Entropy thresholding: A new approach. *Computer Vision Graphics and Image Processing*, 16, 210–239.
- Rodrigues, D., Yang, X. S., de Souza, A. N., & Papa, J. P. (2015). Binary flower pollination algorithm and its application to feature selection. *Recent advances in swarm intelligence and evolutionary computation studies in computational intelligence*: 585 (pp. 85–100). Springer.
- Sahoo, P. K., Soltani, S., & Wong, A. K. C. (1988). A survey of thresholding techniques. *Computer Vision Graphics Image Processing*, 41, 233–260.
- Sanjay, A., Rutuparna, P., Sudipta, B., & Panigrahi, B. K. (2013). Tsallis entropy based optimal multilevel thresholding using cuckoo search algorithm. *Swarm and Evolutionary Computation*, 11(August), 16–30.
- Sanyal, N., Chatterjee, A., & Munshi, S. (2011). An adaptive bacterial foraging algorithm for fuzzy entropy based image segmentation. *Expert Systems with Applications*, 38(12), 15489–15498.
- Sarkar, S., RanjanPatra, G., & Das, S. A. (2011). Differential evolution based approach for multilevel image segmentation using minimum cross entropy thresholding. *SEMCCO*, 1, 51–58.
- Sathya, P. D., & Kayalvizhi, R. (2011). Optimal multilevel thresholding using bacterial foraging algorithm. *Expert Systems with Applications*, 38(12), 15549–15564.
- Sathya, P. D., & Kayalvizhi, R. (2011b). Modified bacterial foraging algorithm based multilevel thresholding for image segmentation. *Engineering Applications of Artificial Intelligence*, 24(4), 65–69.
- Sezgin, M., & Sankur, B. (2004). Survey over image thresholding techniques and quantitative performance. *Journal of Electronic Imaging*, 13(January), 146–165.
- Shapiro, L. G., & Stockman, C. (2002). *Computer vision*. Upper Saddle River, NJ, USA: Prentice Hall.
- Raja, N. S. M., Rajinikanth, V., & Latha, K. (2014). Otsu based optimal multilevel image thresholding using firefly algorithm. *Modelling and Simulation in Engineering*, 2014, 1–17.
- Tang, K., Yuan, X., Sun, T., Yang, J., & Gao, S. (2011). An improved scheme for minimum cross entropy threshold selection based on genetic algorithm. *Knowledge Based Systems*, 24(8), 1131–1138.
- Tao, W., Jin, H., & Liu, L. (2007). Object segmentation using ant colony optimization algorithm and fuzzy entropy. *Pattern Recognition Letters*, 28(7), 788–796.
- Wang, R., & Zhou, Y. (2014). Flower pollination algorithm with dimension by dimension improvement. *Mathematical Problems in Engineering*, 2014, 1–9.
- Wang, Z., Bovik, A. C., Sheikh, H. R., & Simoncelli, E. P. (2004). Image quality assessment: From error measurement to structural similarity. *IEEE Transactions on Image Processing*, 13(1), 600–612.
- Wilcoxon, F. (1945). Individual comparisons by ranking methods. *Biometrics*, 1, 80–83.
- Xue, J. H., & Titterton, D. M. (2011). Median-based image thresholding. *Image and Vision Computing*, 29(9), 631–637.
- Yang, X. S. (2009). Firefly algorithms for multimodal optimization. *Stochastic algorithms: Foundations and applications* (pp. 169–178). Berlin, Heidelberg: Springer.
- Yang, X.-S. (2011). Bat algorithm for multi-objective optimisation. *International Journal of Bio-Inspired Computation*, 3(5), 267–274.
- Yang, X.-S. (2012). Flower pollination algorithm for global optimization. In *Unconventional computation and natural computation 2012: 7445* (pp. 240–249). Lecture notes in computer science.
- Yang, X. S., Karamanoglu, M., & He, X. S. (2014). Flower Pollination Algorithm: A Novel Approach for Multiobjective Optimization. *Engineering Optimization*, 46(9), 1222–1237.
- Yin, P. I. (2007). Multilevel minimum cross entropy threshold selection based on particle swarm optimization. *Applied Mathematics and Computation*, 184, 503–513.
- Yin, P.-Y. (1999). A fast scheme for optimal thresholding using genetic algorithms. *Signal Processing*, 72, 85–95.
- Zhang, J., Li, H., Tang, Z., Lu, Q., Zheng, X., & Zhou, J. (2014). An improved quantum inspired genetic algorithm for image multilevel thresholding segmentation. *Mathematical Problems in Engineering*, 2014, 1–12.
- Zhang, Y., & Wu, L. (2011). Optimal multi-level thresholding based on maximum Tsallis entropy via an artificial bee colony approach. *Entropy*, 13(4), 841–859.