Parameterless Harmony Search for image Multi-thresholding

Krishna Gopal Dhal
Midnapore College (Autonomous),Dept. of Computer Sc. & Application
Midnapore (West)-721101, India
krishnacse42@gmail.com

Iztok Fister Jr.
Faculty of Electrical Engineering and Computer Science, University of Maribor
Smetanova 17, 2000 Maribor
iztok.fister1@um.si

Sanjay Das
University of Kalyani, Dept. of Eng. & Technological Studies
Kalyani-741235, India
dassanjay0810@hotmail.com

ABSTRACT

The Harmony Search (HS) Algorithm is one of the efficient nature-inspired optimization algorithms which exhibits interesting search capability within less computational overhead. However, empirical studies showed that the main problem of this kind of algorithms is the proper setting of the associated parameters. HS associated with a few parameters and to find out the proper combination of the parameter values is time consuming. That's why a parameterless variant has been proposed here, which does not need the tuning over control parameters. The effect of different population size and stopping criterion has been considered in the experiment. The efficiency of the proposed HS is measured in Shannon's entropy based image multi-thresholding field.

Keywords

Harmony Search, Control parameters, multi-thresholding, optimization.

1. INTRODUCTION

Recently, several nature-inspired optimization algorithms have been developed which mimic the behavior of natural and biological systems [5]. These algorithms are very powerful and effective for solving the real world optimization problems within a reasonable time [12]. In this study, the Harmony Search (HS) Algorithm [8] has been taken into consideration, and its extension to a parameterless variant. HS proves its effective performance in different optimization fields. But, the efficiency of the original HS depends on the proper tuning of the associated three control parameters. The proper setting of the values of these three parameters is very difficult for different kinds of problems. In order to overcome that problem, one parameterless variant of HS (PLHS) is developed here. In literature, parameterless variants of some algorithms, such as the Bat Algorithm (BA) [4, 3], Genetic Algorithm [7] and Differential Evolution (DE) [6], have been developed and proved their significant performance over a mathematical optimization field. One parameterless variant of HS is reported in literature where the associated parameters initialized by constant values, including population size [11]. In [11], an experiment with population size and stopping criterion was not performed. In our research paper, these experiments have been performed, inspired by methodologies the same as in [4, 3]. The proposed PLHS has been employed in a multi-thresholding based image segmentation domain, which is one of the significant pre-processing steps in computer vision application. Shannon entropy is used here as an objective function that maximizes the entropy of different regions in the image. Therefore, the organization of this paper is as follows. Section 2 presents the discussion about the HS and the associated control parameters. In section 3, a parameterless variant of HS has been presented and Shannon entropy based multi-thresholding is also explained. Experimental results are discussed in section 4. The paper is concluded in section 5.

2. HARMONY SEARCH (HS) ALGORITHM

In the Harmony Search (HS) Algorithm [8], the individual algorithms are called a "harmony" and they are represented by a real vector whose dimension is n. Let $X_i = \{x_i(1), x_i(2), \ldots, x_i(n)\}$ represent i^{th} randomly generated harmony vector: $x_i(j) = l(j)+(u(j)-l(j))\times rand(0,1)\ for\ j = 1, 2, \ldots, n\ and\ i = 1, 2.., HMS$, where $l(j)$ and $u(j)$ denotes the upper bound and lower bound of the search space respectively and rand(0,1) is a uniform random number between 0 and 1. The HM memory is filled by the HMS harmony vector as follows:

$$HM = \begin{bmatrix} \mathbf{X_1} \\ \mathbf{X_2} \\ \vdots \\ \mathbf{X_{HMS}} \end{bmatrix} \tag{1}$$

2.1 Control Parameters in the HS Algorithm

The values of the control parameters affect the efficiency of the algorithm under experiment significantly.To control these parameters is the same as the controlling the exploration and exploitation efficiency of the considered algorithm. Therefore, the parameter tuning and control become an essential area in the nature-inspired optimization algorithms based research field. But, the setting of the control

DOI: https://doi.org/10.26493/978-961-7023-40-4.5-12

Algorithm 1 Harmony Search.

1: Set the parameters **HMS, HMCR, PAR, BW** and
2: **NI or MAX_FE**, which are discussed in sections 2.1 and 3.
3: Initialize the **HM** and calculate the objective function value
4: of each harmony vector.
5: Improvise a New Harmony X_{new}as follows:
6: **for** $j = 1$ **to** n **do**
7:   **if** $\mathbf{r_1 < HMCR}$ **then**
8:     $\mathbf{x_{new} = x_a(j)}$ where $\mathbf{a} \in (\mathbf{1, 2, \ldots, HMS})$
9:     **if** $\mathbf{r_2 < PAR}$ **then**
10:       $\mathbf{x_{new} = x_{new}(j) \pm r_3 \times BW}$
11:       where $\mathbf{r_1, r_2, r_3 \in rand(0, 1)}$
12:     **end if**
13:     **if** $\mathbf{x_{new}(j) < l(j)}$ **then**
14:       $\mathbf{x_{new}(j) = l(j)}$
15:     **end if**
16:     **if** $\mathbf{x_{new}(j) > u(j)}$ **then**
17:       $\mathbf{x_{new}(j) = u(j)}$
18:     **end if**
19:   **else**
20:     $\mathbf{x_{new}(j) = l(j) + r \times (u(j) - l(j)}$, where
21:     $\mathbf{r \in rand(0, 1)}$
22:   **end if**
23: **end for**
24: Update the HM as $\mathbf{X_w = X_{new}}$ if $\mathbf{f(X_{new}) < f(X_w)}$,
25: where $\mathbf{f}(\cdot)$ represents objective function value.
26: If stopping criterion is completed, the best harmony
27: vector $\mathbf{X_b}$ in the **HM** is returned; Otherwise go back to line 6.

parameters depends crucially on the type of problems. HS is guided by five parameters which are as follows:

a. Population size or Harmony Memory size (HM)

b. Harmony-Memory Consideration Rate (HMCR)

c. Pitch Adjusting Rate (PAR)

d. Distance Bandwidth (BW)

e. Number of iterations (NI)

In order to evade the tuning process, a parameterless variant of the HS has been proposed which will be introduced in the next section.

3. DESIGN OF A PARAMETERLESS HARMONY SEARCH ALGORITHM

The performance of HS is influenced strongly by the values assigned to parameters, i.e. HM, HMCR, PAR, BW and NI. In order to develop a new parameterless HS (PLHS), the influence of these algorithm dependent parameters was studied, which demonstrates that some algorithm parameters such as, in this case, Number of Iterations (NI) could be set wisely, while the optimal setting of other parameters are not so easy. In the memory consideration step (i.e. line no. 7), depending on the HMCR new solution, $(x_{new}(j))$ is generated by selecting randomly from a value in the current existing HM i.e. from the set of $\{x_1(j), x_2(j), \ldots, x_{HMS}(j)\}$. For this operation, one random number r_1 is generated within the range of [0,1] from uniform distribution. If r_1 is less than HMCR, the decision variable $x_{new}(j)$ is generated from memory consideration, otherwise, it is generated from random initialization between $[l(j), u(j)]$ (i.e. line no 20), which are the search boundaries. Therefore, HMCR controls the global search or exploration capability of the HS. Equation no. (2) represents the action of HMCR. Every component which is obtained by memory consideration is checked further to determine whether it should be pitch adjusted or not. The Pitch Adjustment Rate (PAR) is defined as assignment of the frequency adjustment and the bandwidth factor (BW) to control the local search of the HM. The pitch-adjustment decision is calculated by equation no. (3).

Table 1: Parameters' Setting

Alg.	NI	HMCR	PAR	BW	HM
HS	5000	0.75	0.5	0.5	100
PLHS	5000	0.8	0.5	0.55	[10, 1280]

$$x_{new}(j) = \begin{cases} x_i(j) \in \{x_1(j), x_2(j), \ldots, x_{HMS}(j)\} \\ \text{with probability HMCR,} \\ l(j) + (u(j) - l(j) \times rand(0,1) \\ \text{with probability 1-HMCR.} \end{cases} \tag{2}$$

$$x_{new}(j) = \begin{cases} x_{new}(j) = x_{new}(j) = x_{new}(j) \pm rand(0,1) \times BW \\ \text{with - probability PAR,} \\ x_{new}(j) \\ \text{with probability (1-PAR).} \end{cases} \tag{3}$$

Finding the Stopping Criterion is very crucial for different optimization algorithms. In the experiments, two Stopping Criteria (SC) have been considered, which are as follows:

1^{st} Stopping Criterion (SC1): First is the number of times in which the best fitness values remain unchanged. Therefore, if the fitness value for the best harmony remains the same in 10% of the total Number of Iterations (NI), then the HS is stopped.
2^{nd} Stopping Criterion (SC2): The other Stopping Criterion is the number of Fitness Evaluations (FEs), and the maximum number of FEs (i.e. MAX_FE) has been taken as 10,000.

The values of the parameters of the traditional HS are same as [8], but the parameters' values of the proposed PLHS are set from the experience that is given as Table 1.

The population size is also a crucial parameter in HS. It is also reported that an appropriate population size (i.e. HM) is significant to both run-time efficiency and effectiveness [10, 1]. A lower population size may suffer from lack of diversity, whereas the higher population size may affect the convergence speed. In traditional HS, it is set as 100, but, in PLHS, it is varied in the interval $HM \in [10, 1280]$ such that each population size is multiplied by two in each run starting with HM=10. Therefore, eight instances of PLHS have been executed (i.e. PL-1, ..., PL-8) and the best one is considered by the user.

3.1 Multi-Level Shannon Entropy

Let $P = (p_1, p_2, p_3, \ldots, p_n)\ in\delta_n$, where $\delta_n\{(p_1, p_2, \ldots, p_n) \mid p_i \geq 0,\ i = 1, 2, \ldots, n,\ n \geq 2,\ \sum_{i=1}^{n} p_i = 1\}$ is a set of discrete finite n-ary probability distributions. Then entropy of the total image can be defined as [9]:

$$H(P) = -\sum_{i=1}^{n} p_i log_2 p_i \tag{4}$$

I denotes an 8 bit gray level digital image of dimension $M \times N$. P is the normalized histogram for image with $L = 256$ gray levels. Now, if there are $n-1$ thresholds (t), partitioning the normalized histogram into n classes, then the entropy for each class may be computed as:

$$\begin{aligned} H_1(t) &= -\sum_{i=0}^{t_1} \frac{p_i}{P_1} ln \frac{p_i}{P_1}, \\ H_2(t) &= -\sum_{i=t_1+1}^{t_2} \frac{p_i}{P_2} ln \frac{p_i}{P_2}, \\ H_n(t) &= -\sum_{i=t_{n-1}+1}^{L-1} \frac{p_i}{P_n} ln \frac{p_i}{P_n}, \end{aligned} \tag{5}$$

where

$$P_1(t) = \sum_{i=0}^{t_1} p_i,\ P_2(t) = \sum_{i=t_1+1}^{t_2} p_i, \ldots,\ P_n(t) = \sum_{i=t_{n-1}+1}^{L-1} p_i, \tag{6}$$

and for ease of computation, two dummy thresholds $t_0 = 0$, and $t_n = L - 1$ are introduced with $t_0 < t_1 < \ldots < t_{n-1} < t_n$. Then the optimum threshold value can be found by:

$$\varphi(t_1, t_2, \ldots, t_n) = Arg\ max([H_1(t) + H_2(t) + \ldots + H_n(t)]) \tag{7}$$

4. EXPERIMENTAL RESULTS

The experiment has been performed over 50 benchmark images with MatlabR2009b with Windows-7 OS, x32-based PC, Intel(R) Pentium (R)-CPU, 2.20 GHz with 2 GB RAM. The purpose of our experiment is to prove how much the efficiency of the HS is affected by employing different population sizes (i.e. HM) and stopping criteria. In line with this, the traditional HA with 100 numbers of individuals is also compared with these eight PLHS. The traditional HS and PLHSs are run to solve the Shannon's entropy based multi-thresholding problem where optimal threshold values are found by solving equation no. 7. Both HS and PLHSs are stochastic in nature, and that's why each algorithm is run 30 times for each image. The number of thresholds used in this experiment is 2, 3, 4 and 5-level thresholding. The efficiency and consistency of the algorithms are evaluated and compared in terms of Computational Time (CT), Mean Fitness value (Fit$_m$) and Standard Deviation (Fit$_{std}$) for each problem. On the other hand, the image quality assessment metric, known as Peak-Signal to Noise Ratio (PSNR) [2] is computed to assess the similarity of the segmented image against the original image. It is actually a distortion metric,which depends crucially on Mean-Squared Error (MSE), which is defined as:

$$MSE(f, G) = \frac{\sum_{i=0}^{N-1} \sum_{j=0}^{M-1} [f(i,j) - G(i,j)]^2}{M \times N}, \tag{8}$$

where f and G and are the inputs and output image respectively. M and N are the numbers of rows and columns of the image. The PSNR is calculated as follows:

$$PSNR(f, G) = 10 log_{10} \frac{(L-1)^2}{MSE(f, G)}. \tag{9}$$

Greater values of PSNR represent better segmentation.

Table 2: Comparison and ranking based on Computational Time (CT), Mean Fitness value (Fit$_m$), Standard Deviation (Fit$_{std}$) and PSNR for 2-level multi-thresholding.

Alg.	CT	Fit$_m$	Fit$_{std}$	PSNR
HS	1.92 (5)	**18.8027 (1)**	**0 (1)**	**14.60 (1)**
PL-1	1.90 (4)	**18.8027 (1)**	**0 (1)**	**14.60 (1)**
PL-2	2.22 (8)	**18.8027 (1)**	**0 (1)**	**14.60 (1)**
PL-3	2.31 (9)	**18.8027 (1)**	**0 (1)**	**14.60 (1)**
PL-4	1.73 (3)	**18.8027 (1)**	**0 (1)**	**14.60 (1)**
PL-5	2.14 (7)	**18.8027 (1)**	**0 (1)**	**14.60 (1)**
PL-6	2.10 (6)	18.8010 (7)	1.1024e-13 (9)	14.38 (7)
PL-7	1.68 (2)	**18.7820 (9)**	1.0669e-13 (8)	14.31 (9)
PL-8	**1.02 (1)**	18.7985 (8)	5.3334e-16 (7)	14.38 (7)

4.1 Result section for 1st Stopping Criterion (SC1)

Tables numbers 2 to 5 represent the experimental results using the first Stopping Criterion (SC1), and the Tables demonstrate clearly that population size has a great impact over the performance of the HS. According to the Mean Fitness value (Fit$_m$) and Standard Deviation (Fit$_{std}$), less population size produces better output, and that's why PLHS with HM=10 outperforms others. But in terms of computational time, PLHSs with larger population size are better, but they fail to produce the best solution in terms of the objective function. Therefore, it could be said that the larger population performs premature convergence of the HS. Stability (i.e. Fit$_{std}$) also decreases when the size of the population increases. But, experimental study also indicates that when the number of threshold levels increases, stability of the PLHS with lower population size also decreases, but the values of the Fit$_{std}$ remains same for PLHS with larger population size. Therefore, larger population size may help to solve the more complex problems. To get an average performance of HS and PLHSs over different threshold levels, the sum of the algorithms' rankings of each problem has been presented in Table 10, and general ranking is also done based on the sum of the rankings. Table 10 also demonstrates that PLHS with HM=10 and 20 are the best variants in terms of Fit$_m$ and PLHS with HM=10 is the best variant depending on Fit$_{std}$ and PSNR. But, PLHS with HM=1280 is the best when considering the Computational Time (CT). It could be concluded that the average performance of the traditional HS is good as it gets middle ranks in Table 10 by considering all efficiency assessment metrics. Fig. 1 represents the thresholded images and histograms for PL-1, whereas Fig. 2 represents the convergence curves of PL-1 (HM=10) for 2, 3, 4 and 5 level thresholdings of Fig. 1(j).

Table 3: Comparison and ranking based on Computational Time (CT), Mean Fitness value (Fit_m), Standard Deviation (Fit_{std}) and PSNR for 3-level multi-thresholding.

Alg.	CT	Fit_m	Fit_{std}	PSNR
HS	2.33 (5)	**23.4286 (1)**	**0 (1)**	16.88 (2)
PL-1	2.17 (4)	**23.4286 (1)**	**0 (1)**	16.88 (2)
PL-2	2.58 (7)	**23.4286 (1)**	**0 (1)**	16.88 (2)
PL-3	2.49 (6)	**23.4286 (1)**	**0 (1)**	16.88 (2)
PL-4	2.97 (8)	**23.4286 (1)**	**0 (1)**	16.88 (2)
PL-5	3.01 (9)	**23.4286 (1)**	**0 (1)**	16.88 (2)
PL-6	1.70 (3)	23.4065 (7)	1.0092e-14 (8)	**16.92 (1)**
PL-7	1.15 (2)	23.4011 (8)	1.0262e-15 (7)	16.80 (9)
PL-8	**1.06 (1)**	23.3893 (9)	2.0334e-14 (9)	16.84 (8)

Table 4: Comparison and ranking based on Computational Time (CT), Mean Fitness value (Fit_m), Standard Deviation (Fit_{std}) and PSNR for 4-level multi-thresholding.

Alg.	CT	Fit_m	Fit_{std}	PSNR
HS	4.57 (5)	27.7252 (6)	2.0021e-15 (5)	19.03 (4)
PL-1	6.45 (9)	**27.7275 (1)**	**0 (1)**	**19.05 (1)**
PL-2	6.07 (8)	**27.7275 (1)**	**0 (1)**	**19.05 (1)**
PL-3	5.68 (6)	**27.7275 (1)**	**0 (1)**	**19.05 (1)**
PL-4	4.51 (4)	27.7272 (4)	2.0001e-15 (4)	19.02 (5)
PL-5	4.89 (7)	27.7256 (5)	1.0121e-14 (8)	18.88 (6)
PL-6	1.98 (3)	27.6518 (9)	1.0093e-14 (7)	18.61 (9)
PL-7	1.35 (2)	27.6692 (7)	1.0342e-14 (9)	18.80 (7)
PL-8	**1.16 (1)**	27.6687 (8)	2.0302e-15 (6)	18.78 (8)

Table 5: Comparison and ranking based on Computational Time (CT), Mean Fitness value (Fit_m), Standard Deviation (Fit_{std}) and PSNR for 5-level multi-thresholding.

Alg.	CT	Fit_m	Fit_{std}	PSNR
HS	5.07 (6)	31.6959 (4)	3.0225e-13 (5)	20.33 (4)
PL-1	8.23 (9)	**31.6975 (1)**	**2.0543e-14 (1)**	**20.40 (1)**
PL-2	7.89 (8)	**31.6975 (1)**	2.0888e-14 (2)	20.38 (2)
PL-3	6.03 (7)	31.6959 (4)	2.0786e-13 (4)	20.34 (3)
PL-4	4.97 (5)	31.6961 (3)	2.0031e-12 (8)	20.32 (5)
PL-5	4.94 (4)	31.6804 (6)	1.9021e-12 (7)	20.16 (6)
PL-6	2.78 (3)	31.6260 (7)	1.8763e-12 (6)	20.14 (7)
PL-7	1.75 (2)	31.5988 (9)	2.0042e-12 (9)	20.10 (8)
PL-8	**1.18 (1)**	31.6055 (8)	1.0030e-13 (3)	20.11 (9)

Table 6: Comparison and ranking based on Computational Time (CT), Mean Fitness value (Fit_m), Standard Deviation (Fit_{std}) and PSNR for 2-level multi-thresholding.

Alg.	CT	Fit_m	Fit_{std}	PSNR
PL-1	19.06 (6)	**18.8027 (1)**	**0 (1)**	**14.60 (1)**
PL-2	19.58 (8)	**18.8027 (1)**	**0 (1)**	**14.60 (1)**
PL-3	18.94 (3)	**18.8027 (1)**	**0 (1)**	**14.60 (1)**
PL-4	18.85 (2)	**18.8027 (1)**	**0 (1)**	**14.60 (1)**
PL-5	**18.43 (1)**	**18.8027 (1)**	**0 (1)**	**14.60 (1)**
PL-6	19.07 (7)	**18.8027 (1)**	**0 (1)**	**14.60 (1)**
PL-7	19.04 (5)	**18.8027 (1)**	**0 (1)**	**14.60 (1)**
PL-8	19.03 (4)	**18.8027 (1)**	**0 (1)**	**14.60 (1)**

Table 7: Comparison and ranking based on Computational Time (CT), Mean Fitness value (Fit_m), Standard Deviation (Fit_{std}) and PSNR for 3-level multi-thresholding.

Alg.	CT	Fit_m	Fit_{std}	PSNR
PL-1	20.70 (7)	**23.4286 (1)**	**0 (1)**	**16.88 (1)**
PL-2	20.31 (4)	**23.4286 (1)**	**0 (1)**	**16.88 (1)**
PL-3	20.86 (8)	**23.4286 (1)**	**0 (1)**	**16.88 (1)**
PL-4	20.47 (5)	**23.4286 (1)**	**0 (1)**	**16.88 (1)**
PL-5	20.63 (6)	**23.4286 (1)**	**0 (1)**	**16.88 (1)**
PL-6	19.08 (3)	**23.4286 (1)**	**0 (1)**	**16.88 (1)**
PL-7	19.04 (2)	**23.4286 (1)**	**0 (1)**	**16.88 (1)**
PL-8	**19.03 (1)**	**23.4286 (1)**	**0 (1)**	**16.88 (1)**

4.2 Result section for 2nd Stopping Criterion (SC2)

Tables 6- 9 demonstrate the results of the PLHSs using MAX_FE as the stopping criterion. From the analysis of the experimental results, it can be said easily that, when population size resides within 40 and 160, then the HS gives the best result i.e. PL-3, 4 and 5 are the best among all the PLHSs. According to Fit_m and Fit_{std}, PL-1, PL-2 and PL-3 are better than others. Large population size (i.e. 360, 640, 1280) are efficient in terms of CT only. Stability (Fit_{std}) and Fit_m decreases when population size resides within [360, 1280]. MAX_FE based Stopping Criterion helps to reduce the stability issue compared to the NI based Stopping Criterion, which could be verified easily from the values of the corresponding tables. But, MAX_FE increases the CT rapidly. In Table 10, average efficiency has been computed by summing the ranking over different levels of thresholding and again, ranking is done based on the total ranking. PL-1 gives the best average result by considering , and PSNR, whereas, PL-8 takes less time to converge compare to others. But the convergence may be premature convergence according to the values of . Fig. 3 represents the convergence curves of PL-1 using the MAX_FE based Stopping Criterion.

5. CONCLUSION

Efficiency of the population-based nature-inspired optimization algorithms are significantly depends on the proper tuning of algorithm's control parameters. But finding the proper combination of the values of these parameters is very tedious work and problem specific. In order to overcome that one parameterless variant of HS (PLHS) has been developed. The most of the parameters are set from the experimental study. But the population size has been varied in the interval ∈ [10, 1280] to evaluate the effect of the different population size over the efficiency of the HS. Two stopping

Table 8: Comparison and ranking based on Computational Time (CT), Mean Fitness value (Fit_m), Standard Deviation ((Fit_{std}) and PSNR for 4-level multi-thresholding.

Alg.	CT	Fit_m	Fit_{std}	PSNR
PL-1	21.22 (7)	**27.7275(1)**	**0 (1)**	**19.05 (1)**
PL-2	21.41 (8)	**27.7275(1)**	**0 (1)**	**19.05 (1)**
PL-3	21.02 (5)	**27.7275(1)**	**0 (1)**	**19.05 (1)**
PL-4	20.82 (4)	**27.7275(1)**	**0 (1)**	**19.05 (1)**
PL-5	21.04 (6)	27.7256(6)	2.0100e-12 (6)	18.85 (6)
PL-6	20.56 (3)	27.7259(5)	2.0093e-12 (5)	18.87 (5)
PL-7	20.54 (2)	27.7239(7)	3.0303e-12 (7)	18.83 (7)
PL-8	**19.91 (1)**	27.7194(8)	1.0001e-11 (8)	18.83 (7)

Table 9: Comparison and ranking based on Computational Time (CT), Mean Fitness value (Fit_m), Standard Deviation (Fit_{std}) and PSNR for 5-level multi-thresholding.

Alg.	CT	Fit_m	Fit_{std}	PSNR
PL-1	21.93 (7)	**31.6975 (1)**	**0 (1)**	**20.40 (1)**
PL-2	20.88 (2)	**31.6975 (1)**	**0 (1)**	**20.40 (1)**
PL-3	21.73 (6)	31.6959 (3)	2.1044e-15 (3)	20.34 (3)
PL-4	21.55 (5)	31.6957 (4)	1.0030e-13 (5)	20.34 (3)
PL-5	21.46 (4)	31.6952 (5)	1.9001e-13 (7)	20.29 (5)
PL-6	21.02 (3)	31.6937 (6)	1.7703e-13 (6)	20.24 (6)
PL-7	21.94 (8)	31.6930 (7)	2.0030e-13 (8)	20.23 (7)
PL-8	**20.45 (1)**	31.6862 (8)	2.0011e-14 (4)	20.20 (8)

criteria have been used here for analysis the efficiency of the PLHSs. Analysis of the experimental results prove that PLHSs with lower population size are better for maximizing the Shannon's entropy based objective function with less standard deviation but with more computational time when Iteration based stopping criterion is used. Larger population size helps to reduce the computational time, but may performs premature convergence. In the case of MAX_FE based stopping criterion, HS with population size $\in$ [40, 160] gives best and consistent output. Here large population size also affect the stability issue. But, MAX_FE based stopping condition assists to reduce the stability issue with large computational time compare to iteration based termination condition. Therefore, development of robust adaptive nature-inspired optimization algorithms algorithms where all parameters including population size and stopping criterion for a set of problems are automatically adapted is still a big problem in this optimization field. In the future, an extensive study and systematic analysis of the parameters of different nature-inspired optimization algorithms are needed over a different set of problems.

6. REFERENCES

[1] Thomas Back, David B. Fogel, and Zbigniew Michalewicz, editors. *Handbook of Evolutionary Computation*. IOP Publishing Ltd., Bristol, UK, UK, 1st edition, 1997.

[2] Ashish Kumar Bhandari, Anil Kumar, S Chaudhary, and Girish Kumar Singh. A novel color image multilevel thresholding based segmentation using nature inspired optimization algorithms. *Expert Systems with Applications*, 63:112–133, 2016.

[3] Iztok Fister Jr, Iztok Fister, and Xin-She Yang. Towards the development of a parameter-free bat algorithm. In *StuCoSReC: Proceedings of the 2015 2nd Student Computer Science Research Conference*, pages 31–34, 2015.

[4] Iztok Fister Jr, Uroš Mlakar, Xin-She Yang, and Iztok Fister. Parameterless bat algorithm and its performance study. In *Nature-Inspired Computation in Engineering*, pages 267–276. Springer, 2016.

[5] Iztok Fister Jr., Xin-She Yang, Iztok Fister, Janez Brest, and Dušan Fister. A brief review of nature-inspired algorithms for optimization. *Elektrotehniški vestnik*, 80(3):116–122, 2013.

[6] Teo J. and M.Y. Hamid. A parameterless differential evolution optimizer. In *5th International Conference on Systems Theory and Scientific Computation (ISTASC'05), Malta*, pages 330–335, 2005.

[7] Fernando G Lobo and David E Goldberg. An overview of the parameter-less genetic algorithm. *Urbana*, 51:61801, 2003.

[8] Diego Oliva, Erik Cuevas, Gonzalo Pajares, Daniel Zaldivar, and Marco Perez-Cisneros. Multilevel thresholding segmentation based on harmony search optimization. *Journal of Applied Mathematics*, 2013, 2013.

[9] Soham Sarkar, Sujoy Paul, Ritambhar Burman, Swagatam Das, and Sheli Sinha Chaudhuri. A fuzzy entropy based multi-level image thresholding using differential evolution. In *International Conference on Swarm, Evolutionary, and Memetic Computing*, pages 386–395. Springer, 2014.

[10] Ruhul Amin Sarker and MF Azam Kazi. Population size, search space and quality of solution: An experimental study. In *Evolutionary Computation, 2003. CEC'03. The 2003 Congress on*, volume 3, pages 2011–2018. IEEE, 2003.

[11] LA Silva, PB Ribeiro, GH Rosa, KAP Costa, and João Paulo Papa. Parameter setting-free harmony search optimization of restricted boltzmann machines and its applications to spam detection. In *12th International Conference Applied Computing*, pages 142–150, 2015.

[12] Xin-She Yang. *Nature-inspired optimization algorithms*. Elsevier, 2014.

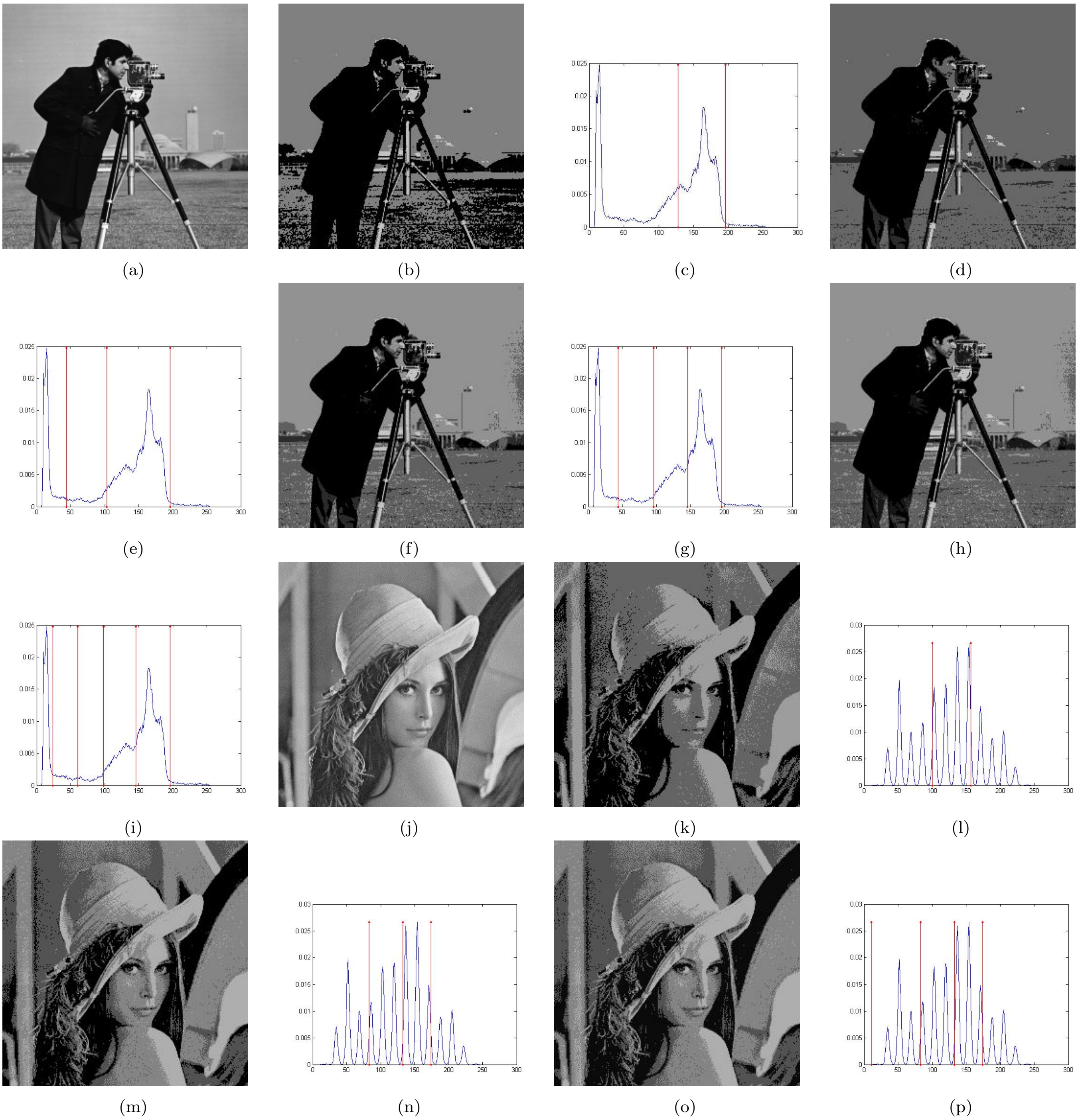


Figure 1: Results of PL-1 using SC1. (a) & (j) Original image; (b), (c) & (k), (l) are the result of 2-level thresholding; (d), (e) & (m), (n) are the result of 3-level thresholding; (f), (g) & (o), (p) are the result of 4-level thresholding; (h), (i) & (q), (r) are the result of 5-level thresholding. (Red lines point the thrershold values).

Table 10: The sum of ranking of each algorithm based on CT, Fit$_m$, Fit$_{std}$, PSNR and general ranking based on the total ranking.

Algo.	CT(SC1)		CT(SC2)		Fit$_m$(SC1)		Fit$_m$(SC2)		Fit$_{std}$(SC1)		Fit$_{std}$(SC2)		PSNR(SC1)		PSNR(SC2)	
	T.R	G.R	T.R	G.R	T.R	G.R	T.R	G.R	T.R	G.R	T.R	G.R	T.R	G.R	T.R	G.R
HS	21	5	-	-	12	5	-	-	12	4	-	-	11	4	-	-
PL-1	26	6	27	8	4	1	4	1	4	1	4	1	5	1	4	1
PL-2	31	9	22	6	4	1	4	1	5	2	4	1	6	2	4	1
PL-3	28	8	22	6	7	3	6	3	7	3	6	3	7	3	6	3
PL-4	20	4	16	3	9	4	7	4	14	5	8	4	13	5	6	3
PL-5	27	7	17	4	13	6	13	5	17	6	15	7	15	6	13	5
PL-6	15	3	16	2	30	7	13	5	30	8	13	5	24	7	13	5
PL-7	8	2	17	4	33	8	16	7	33	9	17	8	33	9	16	7
PL-8	4	1	7	1	33	8	18	8	25	7	14	6	32	8	17	8

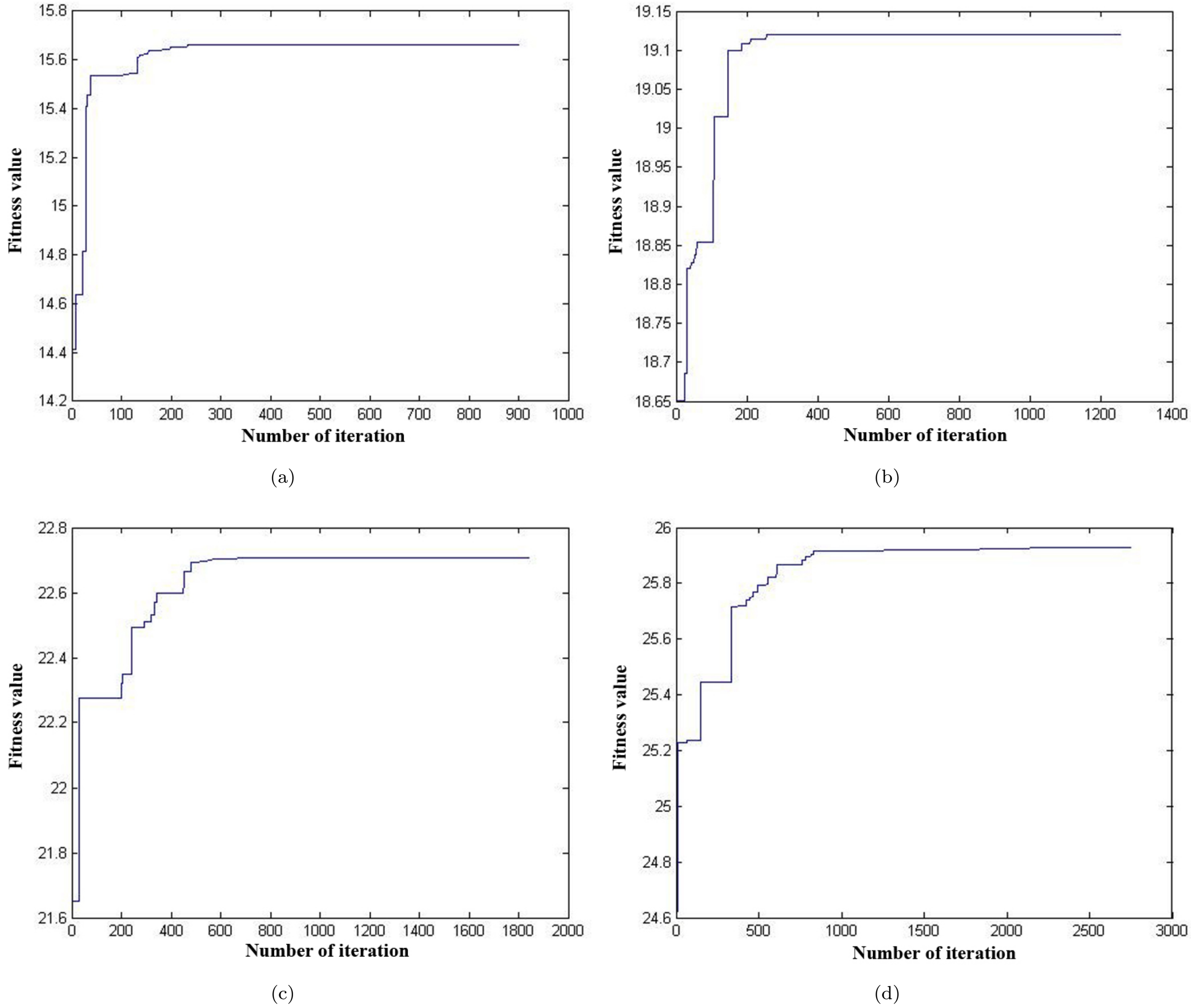
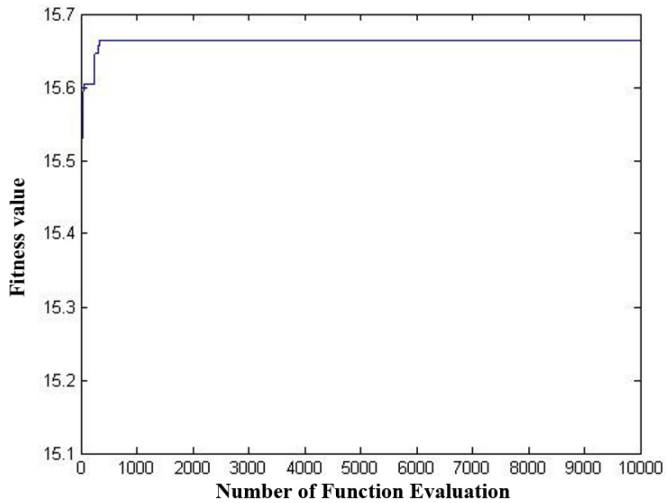
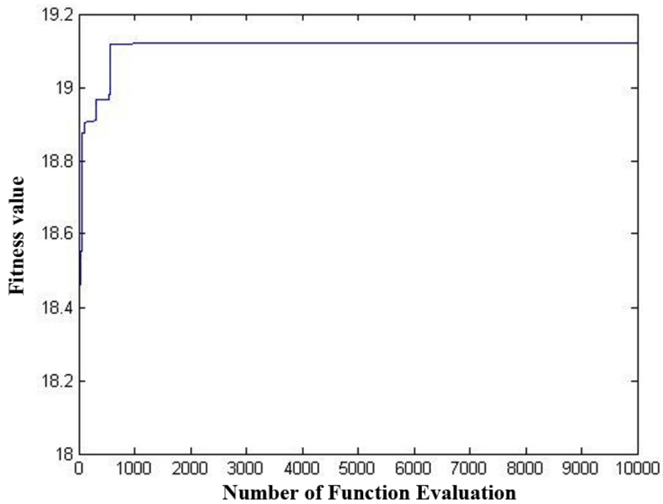


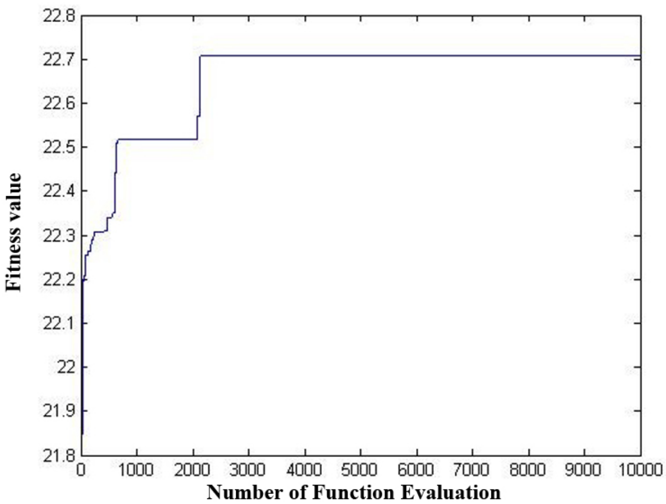
Figure 2: Convergence curves of PL-1 (HM-10) for Fig. 1(j) using SC1 (a) for 2-level thresholding (b) for 3-level thresholding (c) for 4-level thresholding (d) for 5-level thresholding.



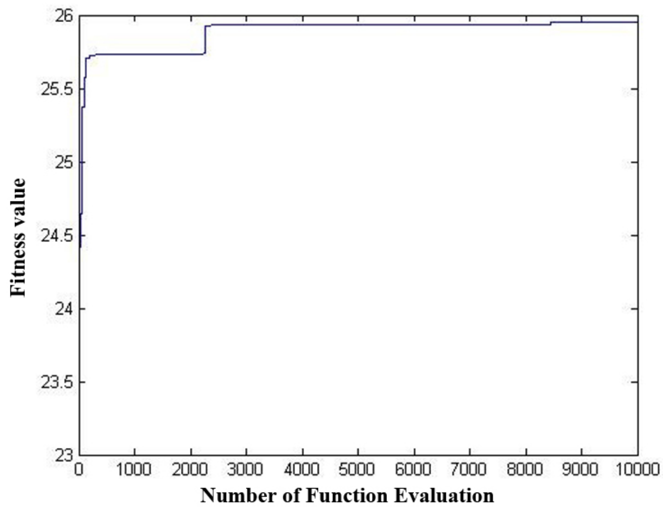
(a)



(b)



(c)



(d)

Figure 3: Convergence curves of PL-1 (HM-10) for Fig. 1(j) using SC1 (a) for 2-level thresholding (b) for 3-level thresholding (c) for 4-level thresholding (d) for 5-level thresholding.