IOP Conf. Series: Journal of Physics: Conf. Series **803** (2017) 012111 doi:10.1088/1742-6596/803/1/012111

Engineering applications of metaheuristics: an introduction

Diego Oliva [1,3,4], Salvador Hinojosa [2,3], M V Demeshko[4]

[1] Departamento de Ciencias Computacionales, Tecnológico de Monterrey, Campus Guadalajara, Av. Gral. Ramón Corona 2514, Zapopan, Jal, México
[2] Dpto. Ingeniería del Software e Inteligencia Artificial, Facultad Informática, Universidad Complutense de Madrid, 28040 Madrid, Spain
[3] Departamento de Electrónica, Universidad de Guadalajara, CUCEI, Av. Revolución 1500, Guadalajara, Jal, México
[4] Tomsk Polytechnic University, Lenin Avenue 30, Tomsk, Russian Federation

E-mail: diego.oliva@itesm.mx

Abstract. Metaheuristic algorithms are important tools that in recent years have been used extensively in several fields. In engineering, there is a big amount of problems that can be solved from an optimization point of view. This paper is an introduction of how metaheuristics can be used to solve complex problems of engineering. Their use produces accurate results in problems that are computationally expensive. Experimental results support the performance obtained by the selected algorithms in such specific problems as digital filter design, image processing and solar cells design.

1. Introduction

In engineering, practically everything can be optimized. Optimization can be included in different subfields, for example, quality control, design, surveillance, etc. The aim is to find the best values that solve the problem. For example, less energy consumption. The optimization methods can be divided into two big groups: deterministic and stochastic. The deterministic techniques include classical approaches and most of them use gradient operations. On the other hand, the stochastic approaches include the use of heuristics and metaheuristics that imitates the behavior of different processes in nature. Recently, several metaheuristic methods have been proposed with interesting results. Such approaches use our scientific understanding of biological, natural or social systems, which at some level of abstraction can be represented as optimization processes, as inspiration. These methods include the social behavior of bird flocking and fish schooling such as the Particle Swarm Optimization (PSO) algorithm [1], the cooperative behavior of bee colonies such as the Artificial Bee Colony (ABC) technique [2], the improvisation process that occurs when a musician searches for a better state of harmony such as the Harmony Search (HS) [3], the emulation of the bat behavior such as the Bat Algorithm (BA) method [4], the mating behavior of firefly insects such as the Firefly (FF) method [5], the social-spider behavior such as the Social Spider Optimization (SSO) [6], the simulation of the animal behavior in a group such as the Collective Animal Behavior [6], the emulation of immunological systems as the clonal selection algorithm (CSA) [7], the simulation of the electromagnetism phenomenon as the electromagnetism-Like algorithm [8], and the emulation of the differential and conventional evolution in species such as the Differential Evolution (DE) [9] and Genetic Algorithms (GA) [10], respectively. They have been growing very fast. For that reason, there is the necessity to test them in real world problems. The drawback of the use of metaheuristics is the necessity to adapt the problem to be solved from an optimization point of view. In other words, it is necessary to define the objective function.

This paper presents a study of a selected group of problems solved using metaheuristics. The methods have been selected considering their importance in their own areas. All their areas are adapted and defined as an optimization problem. Different and popular algorithms have been used to search the best solutions. These techniques are Electromagnetism-Like Optimization (EMO) [8], Harmony Search (HS) [3] and Artificial Bee Colony (ABC) [2].

The rest of the paper is organized as follows. Section 2 introduces the basics of metaheuristics and explains the concepts of EMO, HS and ABC. In section 3, the optimization problems are formulated. Section 4 presents the experimental results and, in section 5, there are conclusions presented.

2. Metaheuristic optimization algorithms

From a conventional point of view, a metaheuristic method is an algorithm that simulates a biological, natural or social system at some level of abstraction. To be more specific, a standard EC algorithm includes:

1. One or more populations of candidate solutions are considered.
2. These populations change dynamically due to the production of new solutions.
3. A fitness function reflects the ability of a solution to survive and reproduce.
4. Several operators are employed in order to explore an exploit appropriately the space of solutions.

The methodology of metaheuristics suggest that, on average, candidate solutions improve their fitness over generations (i. e., their capability of solving the optimization problem). A simulation of the evolution process, based on a set of candidate solutions whose fitness is properly correlated to the objective function to optimize, will, on average, lead to an improvement of their fitness and thus steer the simulated population towards the global solution.

Most of the optimization methods have been designed to solve the problem of finding a global solution of a nonlinear optimization problem with box constraints in the following form:

$$\begin{aligned} &\text{maximize} \quad f(\mathbf{x}), \quad \mathbf{x}=(x_1,\ldots,x_d)\in\mathbb{R}^d \\ &\text{subject to} \quad \mathbf{x}\in\mathbf{X} \end{aligned} \tag{1}$$

where $f:\mathbb{R}^d \to \mathbb{R}$ is a nonlinear function, whereas $\mathbf{X}=\left\{\mathbf{x}\in\mathbb{R}^d \,\middle|\, l_i \le x_i \le u_i, i=1,\ldots,d\right\}$ is a bounded feasible search space, constrained by the lower (l_i) and upper (u_i) limits.

In its initial point, the algorithm begins by initializing the set of N candidate solutions with values that are randomly and uniformly distributed between the pre-specified lower (l_i) and upper (u_i) limits. In each iteration, a set of operators is applied over population $\mathbf{P}^k$ to build new population $\mathbf{P}^{k+1}$. Each candidate solution $\mathbf{p}_i^k$ ($i\in[1,\ldots,N]$) represents d-dimensional vector $\left\{p_{i,1}^k, p_{i,2}^k,\ldots,p_{i,d}^k\right\}$ where each dimension corresponds to a decision variable of the optimization problem at hand. The quality of each candidate solution $\mathbf{p}_i^k$ is evaluated by using objective function $f\left(\mathbf{p}_i^k\right)$ whose final result represents the fitness value of $\mathbf{p}_i^k$. During the iterative optimization process, best candidate solution $\mathbf{g}$ ($g_1, g_2,\ldots g_d$) seen so-far is preserved considering that it represents the best available solution. Fig. 1 presents a graphical representation of a basic cycle of a metaheuristic method.

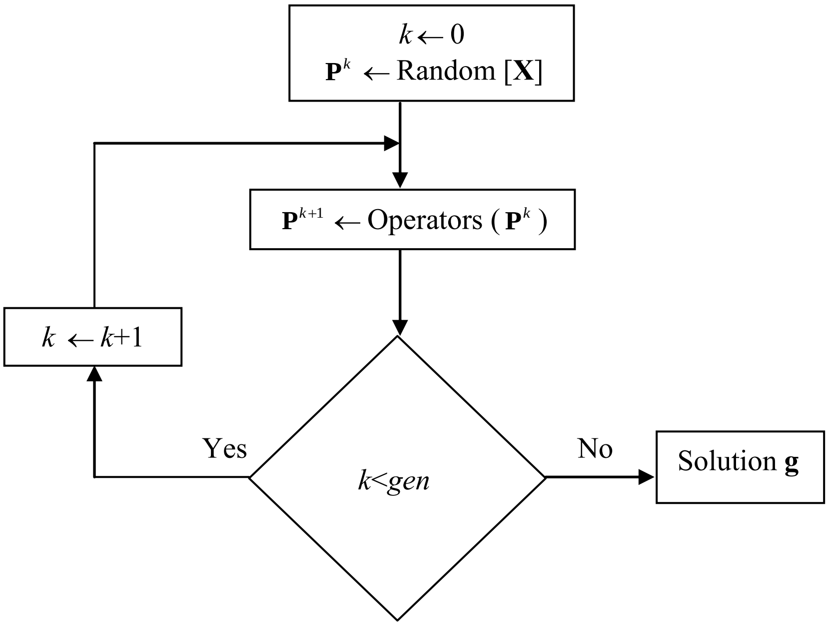


Figure 1. The basic flowchart of the metaheuristic algorithm.

2.1. Electromagnetism-Like Optimization (EMO)

The EMO method, initially designed for bound constrained optimization problems, [8] utilizes N , n-dimensional points $x_{i,t}$, $i=1,2,...,n$, as a population for searching feasible set $\mathbf{X}=\left\{x\in\mathbb{R}^n \middle| l_i \le x \le u_i\right\}$, where t denotes the iteration (or generation) number in the algorithm. Initial population $\mathbf{S}_t=\left\{x_{1,t},x_{2,t},...,x_{N,t}\right\}$ (being $t=1$) is taken of uniformly distributed samples of search region, $\mathbf{X}$. We denote the population set at the t-th iteration by $\mathbf{S}_t$, as the members of $\mathbf{S}_t$ changes with t. After the initialization of $\mathbf{S}_t$, EMO continues its iterative process until a stopping condition (e.g. the maximum number of iterations) is met. An iteration of EMO consists of two steps. In the first step, each point in $\mathbf{S}_t$ moves to a different location by using the attraction-repulsion mechanism of the electromagnetism theory [11]. In the second step, points, moved by the electromagnetism principle, are further moved locally by a local search and then become members of $\mathbf{S}_{t+1}$ in the $(t+1)$-th iteration. Both the attraction-repulsion mechanism and the local search in EMO are responsible for driving the members, $x_{i,t}$, of $\mathbf{S}_t$ to the proximity of the global optimizer.

2.2. Harmony Search (HS) Optimization

The Harmony Search Algorithm (HSA) introduced by Geem, Kim, and Loganathan [12] is an evolutionary optimization algorithm which is based on the metaphor of the improvisation process that occurs when a musician searches for a better state of harmony. The standard HSA, considers each solution as a ‘‘harmony” and is represented by an n-dimension real vector. An initial population of harmony vectors are randomly generated in the search space and stored within a Harmony Memory (HM). A new candidate harmony is thus generated from the elements in the HM by using a memory consideration operation either by a random re-initialization or a pitch adjustment operation. Finally, the HM is updated by comparing the new candidate harmony and the worst harmony vector in the HM. The worst harmony vector is replaced by the new candidate vector in case it is better than the worst harmony vector in the HM. The above process is repeated until a certain termination criterion is met. The basic HS algorithm consists of three main phases: HM initialization, improvisation of new harmony vectors and updating of the HM.

2.3. Artificial Bee Colony (ABC) Optimization

The ABC is an evolutionary algorithm inspired by the intelligent behavior of honey bees [2]. ABC consists of three essential components: food source positions, nectar amount, and several honey-bee classes. Each food source position represents a feasible solution for the problem under consideration. The nectar amount for a food source represents the quality of such a solution (represented by a fitness value). Each bee class symbolizes one particular operation for generating new candidate food source positions (i.e., candidate solutions). The ABC algorithm starts by producing a randomly distributed initial population (food source locations). After initialization, an objective function evaluates whether such candidates represent an acceptable solution (nectar amount) or not. Guided by the values of such an objective function, candidate solutions are evolved through different ABC operations (honey-bee types) until a termination criterion is met.

3. Problems formulation

Metaheuristics can be applied to several problems not only in engineering. However both optimization algorithms and problems, must be adapted according to the problem to solve. In this context, three problems are selected and are solved using one of the methods presented in Section 2. Each problem has been selected from different areas considering their importance. The selected problems are: Digital filters design, Image thresholding and Solar cells design.

3.1 Digital Filters Design

In the filter design process nowadays, the most efficient way to perform is by using metaheuristic algorithms which find the parameters that shape the filter. The main objective of filter design is to approximate the response of the filter to a desired signal by minimizing the objective function while adjusting the parameters of the filter coefficients (see Fig. 2). This paper studies the Infinite Impulse Response (IIR) filters; they are defined in Eq. (2).

$$\frac{Y(z)}{X(z)}=\frac{b_0+b_1z^{-1}+b_2z^{-2}+\ldots+b_rz^{-r}}{a_0+a_1z^{-1}+b_2z^{-2}+\ldots+a_cz^{-c}} \tag{2}$$

Here Y(z) and X(z) are the output and input of the filter. The parameters to be estimates are $\mathbf{A}\cup\mathbf{B}=\{b_0,b_1,b_2,\ldots,b_r,a_0,a_1,a_2,\ldots,a_c\}$. The objective function used to measure the quality of the solutions is the Mean Squared Error (MSE) and is presented in Eq. (3).

$$J=\frac{1}{T}\sum_{t=1}^{T}(Y_p(t)-Y_{fil}(t))^2 , \tag{3}$$

where Y_{fil} is the output generated by the filter using the values estimated by the optimization algorithm. Meanwhile Y_p is the desired output.

3.2 Multilevel Thresholding

One important process in image processing is segmentation. This tasks consist in classifying the pixels of the image. Thresholding is the mostly used process in segmentation, in which the pixels of a gray scale image are divided in sets or classes depending on their intensity level (*L*). For this classification, it is necessary to select a threshold value (*th*) and follow the simple rule of Eq. (4).

$$\begin{aligned} C_1 &\leftarrow p \quad \text{if} \quad 0\le p<th \\ C_2 &\leftarrow p \quad \text{if} \quad th\le p<L-1 \end{aligned} , \tag{4}$$

where *p* is one of the $m\times n$ pixels of gray scale image I_g that can be represented in *L* gray scale levels $L=\{0,1,2,...,L-1\}$. C_1 and C_2 are the classes in which pixel *p* can be located, finally *th* is the threshold. The rule in Eq. (5) corresponds to bi-level thresholding and can be easily extended for multiple sets. The objective function used for this problem is the between class variance proposed by

Otsu [13] that employs the maximum variance value of the different classes as a criterion to segment the image.The objective function is defined as follows:

$$J(\mathbf{TH}) = \max(\sigma^2(\mathbf{TH})), \quad 0 \le th_i \le L-1, \quad i = 1,2,...,k \tag{5}$$

where $\mathbf{TH} = [th_1, th_2, ..., th_{k-1}]$, is a vector that contains multiple thresholds, and the variances are computed using $\sigma^{2^c} = \sum_{i=1}^{k} \sigma_i = \sum_{i=1}^{k} \omega_i \left(\mu_i - \mu_T\right)^2$. Here, i represents a specific class. ω_i and μ_j are, respectively, the probability of occurrence and the mean of a class. For MT, such values are obtained as:

$$\omega_0^c(th) = \sum_{i=1}^{th_1} Ph_i^c, \ \ \omega_1^c(th) = \sum_{i=th_1+1}^{th_2} Ph_i^c, \ldots, \omega_{k-1}^c(th) = \sum_{i=th_k+1}^{L} Ph_i^c \tag{6}$$

and for the mean values,

$$\mu_0^c = \sum_{i=1}^{th_1} \frac{iPh_i^c}{\omega_0^c(th_1)}, \ \mu_1^c = \sum_{i=th_1+1}^{th_2} \frac{iPh_i^c}{\omega_0^c(th_2)}, \ldots, \mu_{k-1}^c = \sum_{i=th_k+1}^{L} \frac{iPh_i^c}{\omega_1^c(th_k)} \tag{7}$$

3.3 Solar Cells Design

Solar cell (SC) accurate modeling has received significant attention in recent years [14]. The modeling of PV cells consists of two steps: the mathematical model formulation and the accurate estimation of their parameter values. For the mathematical model, the Current vs. Voltage (I-V) characteristics that rule the behavior of a solar cell are considered. One of the most used methods for parameter estimation is the use of an equivalent circuit. The Single Diode model presented in Fig. 2 is an easy approximation widely used to obtain the best design of an SC.

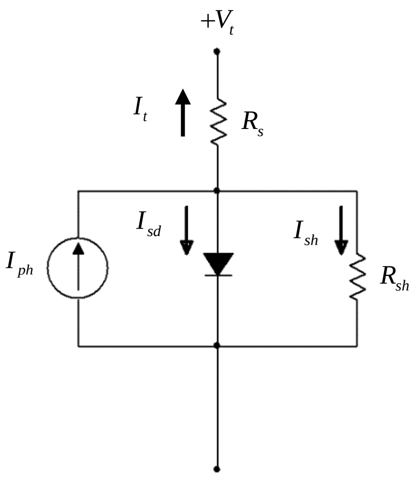


Figure 2. The single diode model of Solar Cells.

From Fig. 2, the mathematical model for the circuit is defined as:

$$I_t = I_{ph} - I_{sd}\left[\exp\left(\frac{q(V_t + R_s \cdot I_t)}{n \cdot k \cdot T}\right) - 1\right] - \frac{V_t + R_s \cdot I_t}{R_{sh}} \tag{8}$$

In Table 1, the range of values for each parameter is presented.

Table 1. The upper and lower range of the solar cell parameters

Parameter	Lower value	Upper value
$R_s(\Omega)$	0	0.5
$R_{sh}(\Omega)$	0	100
$I_{ph}(A)$	0	1
$I_{sd}(\mu A)$	0	1
n	1	2

For solar cell design, the objective function is the Root Mean Square Error (*RMSE*) that exists between measurements and the estimated parameters (Eq. (8) and Eq. (9)).

$$f_{\mathrm{SD}}(V_t, I_t, \mathbf{x}) = I_t - I_{ph} + I_{sd}\left[\exp\left(\frac{q(V_t + R_s \cdot I_t)}{n \cdot k \cdot T}\right) - 1\right] + \frac{V_t + R_s \cdot I_t}{R_{sh}} \tag{8}$$

$$RMSE(\mathbf{x}) = \sqrt{\frac{1}{N}\sum_{c=1}^{N}\left(f^c(V_t^c, I_t^c, \mathbf{x})\right)^2}, \tag{9}$$

4. Experimental results

4.1. Digital Filter Design Using EMO

The problem of designing IIR filters has been solved using the Electromagnetism-Like Optimization algorithm. A second order system using a first order model has been selected as an example. For this problem the transfer function of the system is defined as:

$$H_s(z^{-1}) = \frac{y_p(z^{-1})}{u(z^{-1})} = \frac{0.05 - 0.4z^{-1}}{1 - 1.1314z^{-1} + 0.25z^{-2}} \tag{10}$$

An IIR filter can model the system of Eq. (10). In this case, it has the following transfer function:

$$H_m(z^{-1}) = \frac{y_m(z^{-1})}{u(z^{-1})} = \frac{b}{1 - az^{-1}} \tag{11}$$

Here $H_s(z^{-1})$ and $H_m(z^{-1})$ correspond to the input-output relation of the system and the proposed model, respectively. A noisy signal is applied to the system and the filter and defined by $u(z^{-1})$. From Eq. (11), the EMO algorithm estimates the values of a and b using the objective function defined in Eq. (3). The results obtained by EMO have been compared with PSO and are described in Table 2 and Fig. 3.

Table 2. Comparison between the values obtained by EMO and PSO for IIR filter design.

Values	a	b	J
EMO	0.9034	-0.3030	0.0107
PSO	0.8950	-0.3142	0.0121
Real	0.906	-0.311	-

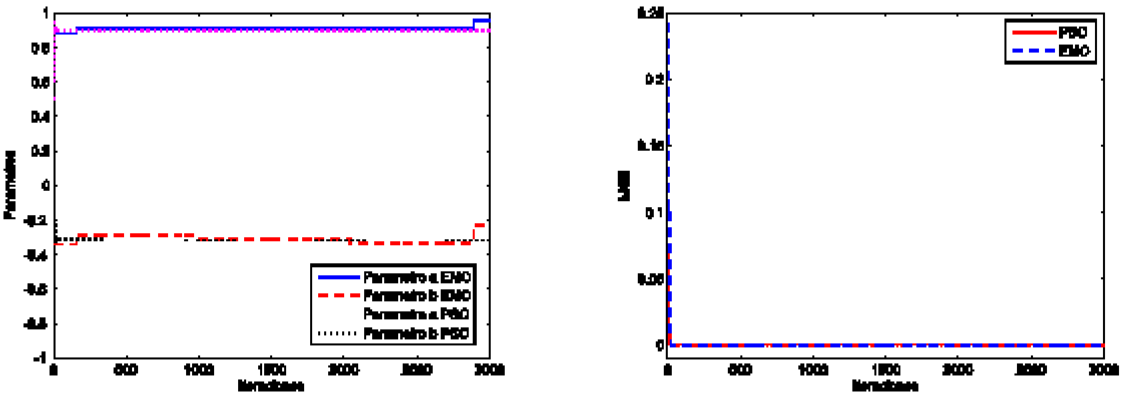


Figure 3. Comparison of parameters for EMO and PSO for IIR filter design.

4.2. Multilevel Thresholding Using HS

A Harmony Search algorithm has been applied to find the best thresholds that can segment an image. The approach is applied over the complete set of benchmark images whereas the results of a selected images are registered in Table 3. Such results present the best threshold values obtained after testing the proposed method, considering four different threshold points $th = 2,3,4,5$. In the Table, there are also the *PSNR* and the *STD* values. According to the results, it is evident that the *PSNR* and *STD* values increment their magnitude as the number of threshold points increases.

Table 3. Results after applying the HSMA using Otsu's function to the set of benchmark images.

Image	k	Thresholds x_{best}	*PSNR*	*STD*
Camera man	2	70, 144	17.247	2.30 E-12
	3	59, 119, 156	20.211	1.55 E-02
	4	42, 95, 140, 170	21.533	2.76 E-12
	5	36, 82, 122, 149, 173	23.282	5.30 E-03
Lena	2	91, 150	15.401	9.22 E-13
	3	79, 125, 170	17.427	2.99 E-02
	4	73, 112, 144, 179	18.763	2.77 E-01
	5	71, 107, 134, 158, 186	19.443	3.04 E-01
Baboon	2	97, 149	15.422	6.92 E-13
	3	85, 125, 161	17.709	1.92 E-02
	4	71, 105, 136, 167	20.289	5.82 E-02
	5	66, 97, 123, 147, 173	21.713	4.40 E-01

The results obtained after applying the HS to each of the images from Table 3 are presented in Fig. 4 as an example. The results show the segmented image, the histogram with the thresholds (best solutions) and the objective function evolution.

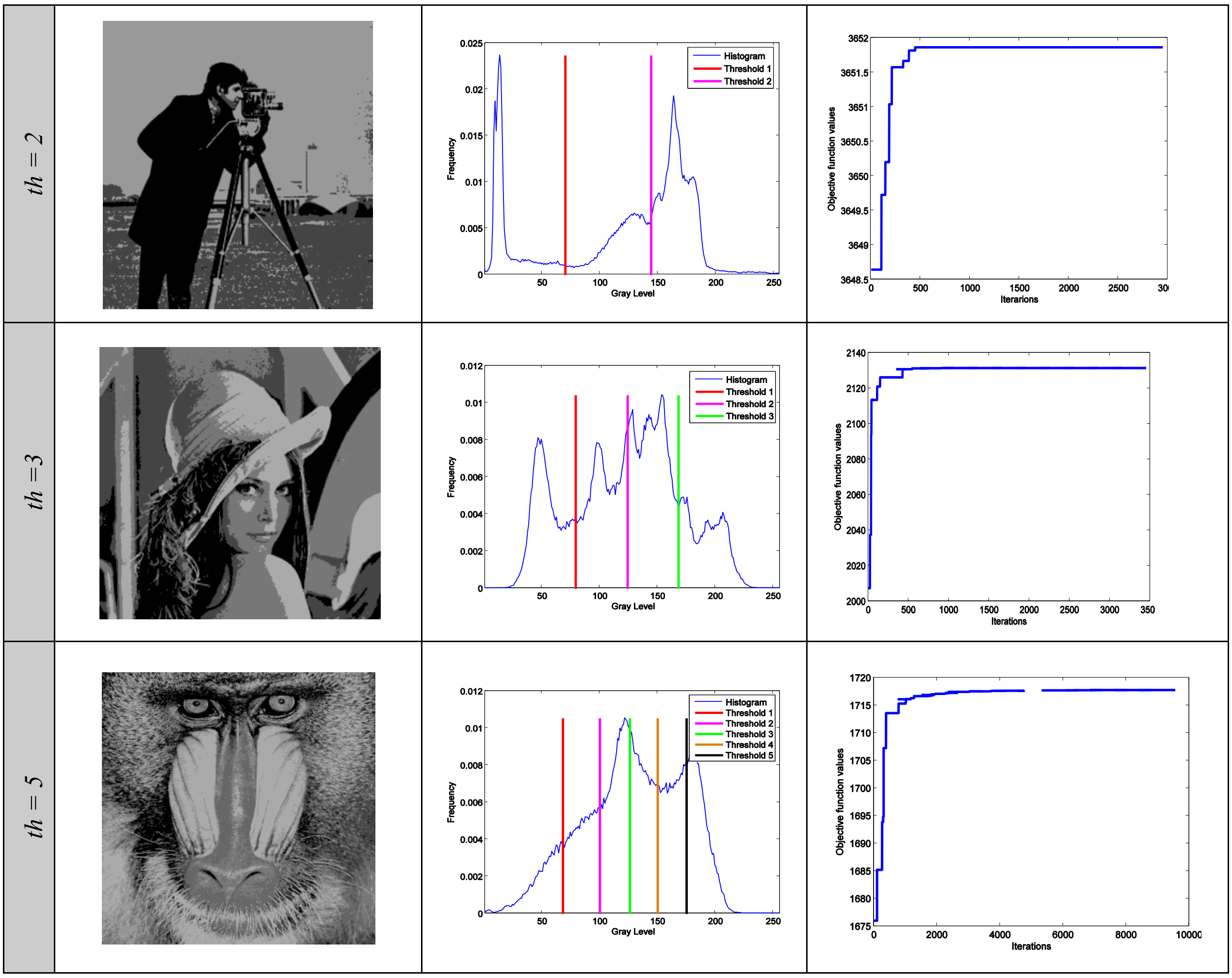


Figure 4. Results of different thresholds over obtained with HS in the selected test images.

4.3. Solar Cells Design Using ABC

For the solar cells design, the Artificial Bee Colony method has been used to estimate the best parameters of the single diode mode. The experiments have been performed using a commercial silicon solar cell (from the R.T.C. Company of France) under the standard test conditions (STC), with a diameter of 57mm. During the data collection process, it is considered that the solar cell operates under the following operating conditions: 1 sun $(1000W/m^2)$ at *T=33°C*. In order to test the performance of the ABC method, four more temperatures have been included: $T=25°C$, $T=50°C$, $T=75°C$. The experiments were performed using 26 measurements form the physical SC. Such samples are shown in Table 5. Table 4 shows the values of the SC estimated by ABC; they are also compared with PSO.

Table 4. Comparative results of ABC and PSO applied over the SD model.

Parameter	**ABC**	**PSO**
$R_s(\Omega)$	0.0364	0.0354
$R_{sh}(\Omega)$	53.6433	59.0120
$I_{ph}(A)$	0.7608	0.7607
$I_{sd}(\mu A)$	0.3251	0.4000

n	1.4817	1.5033
RMSE	9.862 E-04	0.0013
Mean	0.0010	0.2544
STD	1.497 E-05	0.0289

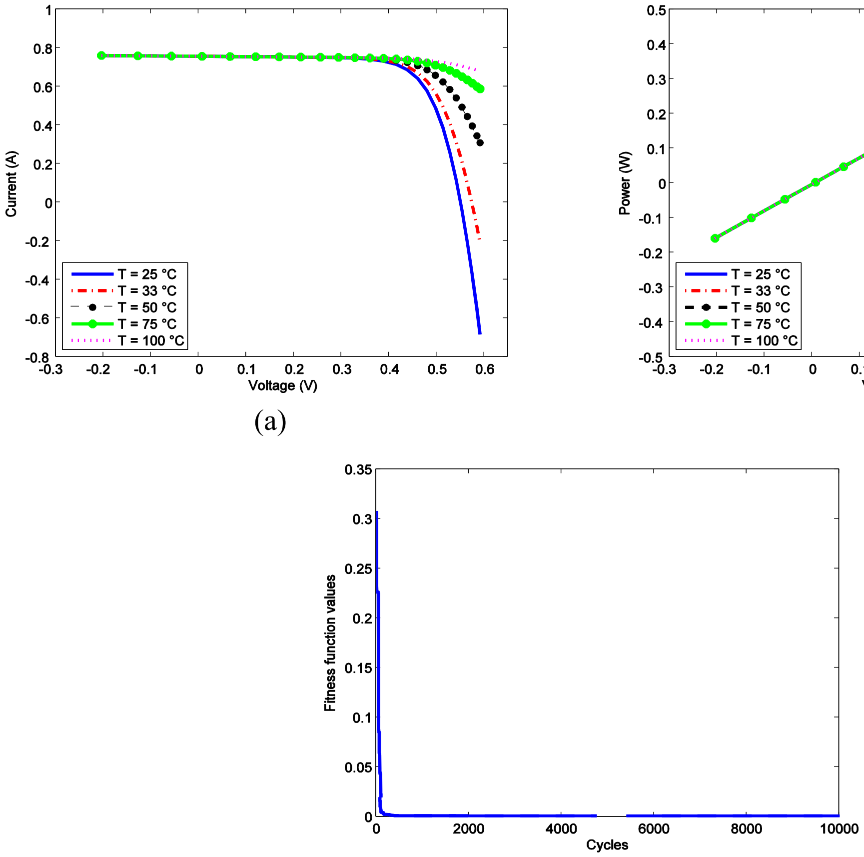
On the other hand, the estimated data obtained with the best result of the ABC algorithm are also show in Table 5.

Table 5. Terminal ($V_t - I_t$) measurements and relative error values for single diode models.

Data	$V_t(V)$ **Measured**	$I_t(A)$ **Measured**	$I_{t-calculated}(A)$ **ABC single diode model**	R_{error} **ABC single diode model**	**Normalized** $NR_{error}(\%)$ **ABC single diode model**
1	-0.2057	0.7640	0.7641	-0.0001	36.6608
2	-0.1291	0.7620	0.7626	-0.0006	22.6153
3	-0.0588	0.7605	0.7613	-0.0008	17.95165
4	0.0057	0.7605	0.7601	0.0003	47.4202
5	0.0646	0.7600	0.7590	0.0009	62.1258
6	0.1185	0.7590	0.7580	0.0009	62.4765
7	0.1678	0.7570	0.7571	-0.0001	36.8073
8	0.2132	0.7570	0.7561	0.0008	60.1225
9	0.2545	0.7555	0.7550	0.0004	49.2543
10	0.2924	0.7540	0.7536	0.0003	47.4304
11	0.3269	0.7505	.7513	-0.0008	17.4642
12	0.3585	0.7465	0.7473	-0.0008	18.5101
13	0.3873	0.7385	0.7401	-0.0016	0
14	0.4137	0.7280	0.7273	0.0006	54.9849
15	0.4373	0.7065	0.7069	-0.0004	28.5214
16	0.4590	0.6755	0.6752	0.0002	45.6933
17	0.4784	0.6320	0.6307	0.0012	70.8228
18	0.4960	0.5730	0.5718	0.0011	66.6002
19	0.5119	0.4990	0.4995	-0.0005	25.2670
20	0.5265	0.4130	0.4136	-0.0006	23.8922
21	0.5398	0.3165	0.3175	-0.0010	14.6298
22	0.5521	0.2120	0.2121	-0.0001	35.1718
23	0.5633	0.1035	0.1022	0.0012	69.2431
24	0.5736	-0.0100	-0.0086	-0.0013	7.13579
25	0.5833	-0.1230	-0.1254	0.0024	100
26	0.5900	-0.2100	-0.2084	-0.0015	1.41935

Figure 5 shows the graphs of current vs. voltage at different temperatures ($T = 25°C$, $T = 50°C$, $T = 75°C$ and $T = 100°C$), the power, and the fitness values for a single diode model.

IOP Conf. Series: Journal of Physics: Conf. Series **803** (2017) 012111 doi:10.1088/1742-6596/803/1/012111



(a) (b)

(c)

Figure 5. SD model results using ABC: (a) Measured voltage vs. ABC computed current for different temperatures, (b) Measured voltage vs. ABC-power at different temperatures, (c) *RMSE* evolution.

5. Conclusions

This work presents the implementation of three different algorithms applied over selected problems from engineering. The selected methods are the EMO, HS and ABC; they have been used to solve the problems of digital filter design, image segmentation and solar cells design. In all cases, the objective function depends directly on the problem and helps the algorithm to find solutions in the search space. In all cases, the proposed method obtains better results in comparison, for example, with Particles Swarm Optimization. It is important to mention that the aim of this paper is to propose metaheuristic algorithms as an alternative for classical approaches.

References

[1] Kennedy J., Eberhart R. 1995 Particle swarm optimization, Neural Networks *Proceedings., IEEE Int. Conf.* **4** 1942–1948

[2] Karaboga D. 2005 An idea based on honey bee swarm for numerical optimization, Tech. Rep. TR06, Comput. Eng. Dep. Eng. Fac. Erciyes Univ

[3] Geem Z.W. ed. 2009 *Harmony Search Algorithms for Structural Design Optimization* (Springer Berlin Heidelberg)

[4] Yang X.-S. 2010 A New Metaheuristic Bat-Inspired Algorithm, *Nat. Inspired Coop. Strateg. Optim. (NICSO 2010).* **284** 65–74

[5] Yang X.-S. 2014 *Cuckoo Search and Firefly Algorithm: Overview and Analysis, in: Cuckoo Search and Firefly Algorithm* (Springer Berlin Heidelberg) pp. 1–26

[6] Cuevas E., Diaz Cortés M.A., Oliva Navarro D.A. 2016 Advances of Evolutionary Computation: Methods and Operators (1st ed., Springer International Publishing) doi:10.1007/978-3-319-28503-0

[7] De Castro L.N., Von Zuben F.J. 2002 Learning and optimization using the clonal selection principle, *IEEE Trans. Evol. Comput.* **6** 239–251

[8] Birbil Ş.I., Fang S.C. 2003 An electromagnetism-like mechanism for global optimization, *J. Glob. Optim.* **25** 263–282

[9] Storn R., Price K., Differential Evolution – A Simple and Efficient Heuristic for global Optimization over Continuous Spaces, *J. Glob. Optim.* **11 (n.d.)** 341–359

[10] David G. 1989 *Genetic Algorithms in Search, Optimization, and Machine Learning* (1st ed., Addison-Wesley, Boston, MA, USA)

[11] Cowan E.W. 1968 *Basic Electromagnetism* (Academic Press, New York)

[12] Geem Z.W., Kim J.H., Loganathan G.V. 2001 A New Heuristic Optimization Algorithm: Harmony Search, *Simulation.* **76** 60–68

[13] Otsu N. 1979 A Threshold Selection Method from Gray-Level Histograms, *IEEE Trans. Syst. Man. Cybern.* **9** 62–66

[14] AlRashidi M.R., AlHajri M.F., El-Naggar K.M., Al-Othman A.K. 2011 A new estimation approach for determining the I-V characteristics of solar cells, *Sol. Energy*. **85** 1543–1550