Hindawi Publishing Corporation
Journal of Applied Mathematics
Volume 2014, Article ID 827206, 9 pages
http://dx.doi.org/10.1155/2014/827206

Research Article

A Comparison of Evolutionary Computation Techniques for IIR Model Identification

**Erik Cuevas,[1,2] Jorge Gálvez,[1] Salvador Hinojosa,[1] Omar Avalos,[1]
Daniel Zaldívar,[1,2] and Marco Pérez-Cisneros[3]**

[1] *Departamento de Electrónica, Universidad de Guadalajara, CUCEI, Avenida Revolución 1500, 44430 Guadalajara, JAL, Mexico*
[2] *Centro Universitario Azteca, Unidad de Investigación, Avenida Juárez 340, 44280 Guadalajara, JAL, Mexico*
[3] *CUTONALA, Avenida Nuevo Periférico 555, Ejido San José Tateposco, 48525 Tonalá, JAL, Mexico*

Correspondence should be addressed to Erik Cuevas; erik.cuevas@cucei.udg.mx

Received 18 August 2014; Accepted 30 September 2014; Published 25 December 2014

Academic Editor: Xin-She Yang

System identification is a complex optimization problem which has recently attracted the attention in the field of science and engineering. In particular, the use of infinite impulse response (IIR) models for identification is preferred over their equivalent FIR (finite impulse response) models since the former yield more accurate models of physical plants for real world applications. However, IIR structures tend to produce multimodal error surfaces whose cost functions are significantly difficult to minimize. Evolutionary computation techniques (ECT) are used to estimate the solution to complex optimization problems. They are often designed to meet the requirements of particular problems because no single optimization algorithm can solve all problems competitively. Therefore, when new algorithms are proposed, their relative efficacies must be appropriately evaluated. Several comparisons among ECT have been reported in the literature. Nevertheless, they suffer from one limitation: their conclusions are based on the performance of popular evolutionary approaches over a set of synthetic functions with exact solutions and well-known behaviors, without considering the application context or including recent developments. This study presents the comparison of various evolutionary computation optimization techniques applied to IIR model identification. Results over several models are presented and statistically validated.

1. Introduction

System identification is a complex optimization problem which has recently attracted the attention in the field of science and engineering. System identification is important in the disciplines of control systems [1], communication [2], signal processing [3], and image processing [4].

In a system identification configuration, an optimization algorithm attempts to iteratively determine the adaptive model parameters to get an optimal model for an unknown plant by minimizing some error function between the output of the candidate model and the output of the plant. The optimal model or solution is attained when such error function is effectively reduced. The adequacy of the estimated model depends on the adaptive model structure, the optimization algorithm, and also the characteristic and quality of the input-output data [5].

Systems or plants can be better modeled through infinite impulse response (IIR) models because they emulate physical plants more accurately than their equivalent FIR (finite impulse response) models [6]. In addition, IIR models are typically capable of meeting performance specifications using fewer model parameters. However, IIR structures tend to produce multimodal error surfaces whose cost functions are significantly difficult to minimize [7]. Hence, in order to identify IIR models, a practical, efficient, and robust global optimization algorithm is necessary to minimize the multimodal error function.

Traditionally, the least mean square (LMS) technique and its variants [8] have been extensively used as optimization tools for IIR model identification. The wide acceptance of such gradient based optimization techniques is due to the low complexity and simplicity of implementation. However, the error surface for the IIR model is mostly multimodal with respect to the filter coefficients. This may result in leading traditional gradient-descent approaches into local optima [9].

The difficulties associated with the use of gradient based optimization methods for solving several engineering problems have contributed to the development of alternative solutions. Evolutionary computation techniques (ECT) such as the particle swarm optimization (PSO) [10], artificial bee colony (ABC) [11], electromagnetism-like method (EM) [12], cuckoo search (CS) [13], and flower pollination algorithm (FPA) [14] have received much attention regarding their potential as global optimization methods in real-world applications. Inspired by the evolution process and survival of the fittest in the biological world, ECT are search methods that are different from traditional optimization methods. They are based on a collective learning process within a population of candidate solutions. The population in ECT is usually arbitrarily initialized, and each iteration (also called a generation) evolves towards better and better solution regions by means of randomized processes where several operators are applied to each candidate solution. ECT have been applied to many engineering optimization problems and have proven to be effective for solving some specific problems, including multimodal optimization, dynamic optimization, noisy optimization, and multiobjective optimization [15–17]. Hence, they are becoming increasingly popular tools to solve various hard optimization problems.

As an alternative to gradient based techniques, the problem of IIR modelling has also been handled through evolutionary computation techniques. In general, they have been demonstrated to yield better results than those based on gradient algorithms with respect to accuracy and robustness [9]. Such approaches have produced several robust IIR identification systems by using different evolutionary computation techniques such as PSO [18], ABC [19], EM [20], and CS [21], whose results have been individually reported.

ECT are often designed to meet the requirements of particular problems because no single optimization algorithm can solve all problems competitively [22]. Therefore, when new alternative algorithms are proposed, their relative efficiency must be appropriately evaluated. Many efforts [23–25] have also been devoted to comparing ECT to each other. Typically, such comparisons have been based on synthetic numerical benchmark problems with most studies verifying if one algorithm outperforms others over a given set of benchmarks functions overlooking any statistical test. However, few comparative studies of various ECT considering the application context are available in the literature. Therefore, it is very important to discuss and compare the performance of ECT methods from an application point of view.

This paper presents the comparison of various evolutionary computation optimization techniques that are applied to IIR model identification. In the comparison, special attention is paid to recently developed algorithms such as the cuckoo search (CS) and the flower pollination algorithm (FPA), including also popular approaches as the particle swarm optimization (PSO), the artificial bee colony (ABC) optimization, and the electromagnetism-like optimization (EM) algorithm. Results over several models with different ranges of complexity are presented and validated within a statistically significant framework.

The rest of this paper is organized as follows: Section 2 presents a review of the evolutionary computation techniques that are employed in the comparison whereas Section 3 discusses the IIR system identification problem. In Section 4 all experimental results are depicted with some concluding remarks being drawn in Section 5.

2. Evolutionary Computation Techniques (ECT)

In the real world, many optimization problems can be considered as black box challenges. Often, less information is available about an optimization problem itself unless the information emerges from function evaluations. In the worst case, nothing is known about the characteristics of the fitness function, for example, whether it is unimodal or multimodal.

On the other hand, ECT are used to estimate the solution to complex optimization problems since they adapt easily to black-box formulations and extremely ill-behaved functions. ECT are based on a collective learning process within a population of candidate solutions. The population in ECT is usually arbitrarily initialized while each iteration (also called a generation) evolves towards better solution regions by means of randomized processes with several operators being applied to each candidate solution. ECT have been applied to many engineering optimization problems ensuring an effective solution for some specific problems, including multimodal optimization, dynamic optimization, noisy optimization, multiobjective optimization, and others [15–17].

Therefore, ECT are becoming increasingly popular tools to solve various hard optimization problems. This section presents a brief description of five evolutionary computation techniques: swarm optimization (PSO), artificial bee colony (ABC) optimization and electromagnetism-like optimization (EM), cuckoo search (CS), and flower pollination algorithm (FPA), which have been all employed in our comparative study.

2.1. Particle Swarm Optimization (PSO). PSO, proposed by Kennedy and Eberhart in 1995 [10], is a population-based stochastic optimization technique that is inspired on the social behavior of bird flocking or fish schooling. The algorithm searches for the optimum using a group or swarm formed by possible solutions of the problem, which are called particles. From the implementation point of view, in the PSO operation, a population $\mathbf{P}^k$ $(\{\mathbf{p}_1^k, \mathbf{p}_2^k, \ldots, \mathbf{p}_N^k\})$ of N particles (individuals) evolves from the initial point ($k = 0$) to a total gen number of iterations ($k = \text{gen}$). Each particle $\mathbf{p}_i^k$ $(i \in [1, \ldots, N])$ represents a d-dimensional vector $\{p_{i,1}^k, p_{i,2}^k, \ldots, p_{i,d}^k\}$ where each dimension corresponds to a decision variable of the optimization problem at hand. The quality of each particle $\mathbf{p}_i^k$ (candidate solution) is evaluated by using an objective function $f(\mathbf{p}_i^k)$ whose final result represents the fitness value of $\mathbf{p}_i^k$. During the evolution process, the best global position $\mathbf{g}$ $(g_1, g_2, \ldots g_d)$ seen so far is stored with the best position $\mathbf{p}_i^*$ $(p_{i,1}^*, p_{i,2}^*, \dots, p_{i,d}^*)$ being reached by each particle. Such positions are computed by considering a minimization problem as follows:

$$\mathbf{g} = \underset{i\in\{1,2,\dots,N\}, a\in\{1,2,\dots,k\}}{\arg\min} \left(f\left(\mathbf{p}_i^a\right)\right), \qquad \mathbf{p}_i^* = \underset{a\in\{1,2,\dots,k\}}{\arg\min} \left(f\left(\mathbf{p}_i^a\right)\right). \tag{1}$$

In this work, the modified PSO version proposed by Lin et al. in [26] has been implemented. Under such approach, the new position $\mathbf{p}_i^{k+1}$ of each particle $\mathbf{p}_i^k$ is calculated by using the following equations:

$$v_{i,j}^{k+1} = w \cdot v_{i,j}^k + c_1 \cdot r_1 \cdot \left(p_{i,j}^* - p_{i,j}^k\right) + c_2 \cdot r_2 \cdot \left(g_j - p_{i,j}^k\right); \qquad p_{i,j}^{k+1} = p_{i,j}^k + v_{i,j}^{k+1}, \tag{2}$$

where w is called the inertia weight that controls the impact of the current velocity on the updated velocity $(i \in [1,\dots,N], j \in [1,\dots,d])$. c_1 and c_2 are the positive acceleration coefficients that rule the movement of each particle towards the positions $\mathbf{g}$ and $\mathbf{p}_i^*$, respectively. r_1 and r_2 are uniformly distributed random numbers that are chosen within the interval $[0,1]$.

2.2. Artificial Bee Colony (ABC). The artificial bee colony (ABC) algorithm, proposed by Karaboga [11], is an ECT inspired by the intelligent foraging behavior of a honeybee swarm. In the ABC operation, a population $\mathbf{L}^k$ $(\{\mathbf{l}_1^k, \mathbf{l}_2^k, \dots, \mathbf{l}_N^k\})$ of N food locations (individuals) is evolved from the initial point $(k = 0)$ to a total gen number of iterations $(k = \text{gen})$. Each food location $\mathbf{l}_i^k$ $(i \in [1,\dots,N])$ represents a d-dimensional vector $\{l_{i,1}^k, l_{i,2}^k, \dots, l_{i,d}^k\}$ where each dimension corresponds to a decision variable of the optimization problem to be solved. After initialization, an objective function evaluates each food location to assess whether it represent an acceptable solution (nectar-amount) or not. Guided by the values of such an objective function, the candidate solution $\mathbf{l}_i^k$ is evolved through different ABC operations (honeybee types). In the main operator, each food location $\mathbf{l}_i^k$ generates a new food source $\mathbf{t}_i$ in the neighborhood of its present position as follows:

$$\mathbf{t}_i = \mathbf{l}_i^k + \phi\left(\mathbf{l}_i^k - \mathbf{l}_r^k\right), \quad i, r \in (1, 2, \dots, N), \tag{3}$$

where $\mathbf{l}_r^k$ is a randomly chosen food location, satisfying the condition $r \neq i$. The scale factor ϕ is a random number between $[-1, 1]$. Once a new solution $\mathbf{t}_i$ is generated, a fitness value representing the profitability associated with a particular solution $\text{fit}(\mathbf{l}_i^k)$ is calculated. The fitness value for a minimization problem can be assigned to a candidate solution $\mathbf{l}_i^k$ by the following expression:

$$\text{fit}\left(\mathbf{l}_i^k\right) = \begin{cases} \dfrac{1}{1 + f\left(\mathbf{l}_i^k\right)}, & \text{if } f\left(\mathbf{l}_i^k\right) \geq 0, \\ 1 + \text{abs}\left(f\left(\mathbf{l}_i^k\right)\right), & \text{if } f\left(\mathbf{l}_i^k\right) < 0, \end{cases} \tag{4}$$

where $f(\cdot)$ represents the objective function to be minimized. Once the fitness values are calculated, a greedy selection process is applied between $\mathbf{t}_i$ and $\mathbf{l}_i^k$. If $\text{fit}(\mathbf{t}_i)$ is better than $\text{fit}(\mathbf{l}_i^k)$, then the candidate solution $\mathbf{l}_i^k$ is replaced by $\mathbf{t}_i$; otherwise, $\mathbf{l}_i^k$ remains.

2.3. Electromagnetism-Like (EM) Algorithm. The EM algorithm, proposed by İlker et al. [12] is a simple and population-based search algorithm which has been inspired by the electromagnetism phenomenon. In EM, individuals emulate charged particles which interact to each other based on the electromagnetism laws of repulsion and attraction. The method utilizes N, d-dimensional points $\mathbf{x}_i^k$, $i = 1, 2, \dots, N$, where each point $\mathbf{x}_i^k$ is a d-dimensional vector containing the parameter values to be optimized $(\mathbf{x}_i^k = \{x_{i,1}^k, \dots, x_{i,d}^k\})$ whereas k denotes the iteration (or generation) number. The initial population $\mathbf{X}^k = \{\mathbf{x}_1^k, \mathbf{x}_2^k, \dots, \mathbf{x}_N^k\}$ (being $k = 0$) is taken from uniformly distributed samples of the search space. We denote the population set at the kth generation by $\mathbf{X}^k$, because members of $\mathbf{X}^k$ change with k. After the initialization of $\mathbf{X}^0$, EM continues its iterative process until a stopping condition (e.g., the maximum number of generations, $k = \text{gen}$) is met. An iteration of EM consists of three steps. In the first step each point in $\mathbf{X}^k$ moves to a different location by using the attraction-repulsion mechanism of the electromagnetism theory. In the second step, points moved by the electromagnetism principle are further moved locally by a local search procedure. Finally, in the third step, in order to generate the new population $\mathbf{X}^{k+1}$, a greedy selection process selects best points between those produced by the local search procedure and the originals. Both the attraction-repulsion mechanism and the local search in EM are responsible for driving the members $\mathbf{x}_i^k$ of $\mathbf{X}^k$ to the proximity of the global optimum.

2.4. Cuckoo Search (CS) Method. CS is one of the latest nature-inspired algorithms that has been developed by Yang and Deb [13]. CS is based on the brood parasitism of some cuckoo species. In addition, this algorithm is enhanced by the so-called Lévy flights [27], rather than by simple isotropic random walks. From the implementation point of view of the CS operation, a population $\mathbf{E}^k$ $(\{\mathbf{e}_1^k, \mathbf{e}_2^k, \dots, \mathbf{e}_N^k\})$ of N eggs (individuals) is evolved from the initial point $(k = 0)$ to a total gen number of iterations $(k = 2 \cdot \text{gen})$. Each egg $\mathbf{e}_i^k$ $(i \in [1,\dots,N])$ represents a d-dimensional vector $\{e_{i,1}^k, e_{i,2}^k, \dots, e_{i,d}^k\}$ where each dimension corresponds to a decision variable of the optimization problem to be solved. The quality of each egg $\mathbf{e}_i^k$ (candidate solution) is evaluated by using an objective function $f(\mathbf{e}_i^k)$ whose final result represents the fitness value of $\mathbf{e}_i^k$. Three different operators define the evolution process of CS: (A) Lévy flight, (B) the replacing of nests operator for constructing new solutions, and (C) the elitist selection strategy.

(A) The Lévy Flight. One of the most powerful features of cuckoo search is the use of Lévy flights to generate

new candidate solutions (eggs). Under this approach, a new candidate solution $\mathbf{e}_i^{k+1}$ ($i \in [1,\ldots,N]$) is produced by perturbing the current $\mathbf{e}_i^k$ with a change of position $\mathbf{c}_i$. In order to obtain $\mathbf{c}_i$, a random step $\mathbf{s}_i$ is generated by a symmetric Lévy distribution. For producing $\mathbf{s}_i$, Mantegna's algorithm [28] is employed as follows:

$$\mathbf{s}_i = \frac{\mathbf{u}}{|\mathbf{v}|^{1/\beta}}, \tag{5}$$

where $\mathbf{u}$ ($\{u_1,\ldots,u_d\}$) and $\mathbf{v}$ ($\{v_1,\ldots,v_d\}$) are n-dimensional vectors and $\beta = 3/2$. Each element of $\mathbf{u}$ and $\mathbf{v}$ is calculated by considering the following normal distributions:

$$u \sim N\left(0,\sigma_u^2\right), \quad v \sim N\left(0,\sigma_v^2\right),$$
$$\sigma_u = \left(\frac{\Gamma(1+\beta)\cdot\sin(\pi\cdot\beta/2)}{\Gamma((1+\beta)/2)\cdot\beta\cdot 2^{(\beta-1)/2}}\right)^{1/\beta}, \qquad \sigma_v = 1, \tag{6}$$

where $\Gamma(\cdot)$ represents the Gamma distribution. Once $\mathbf{s}_i$ has been calculated, the required change of position $\mathbf{c}_i$ is computed as follows:

$$\mathbf{c}_i = 0.01\cdot\mathbf{s}_i \oplus \left(\mathbf{e}_i^k - \mathbf{e}^{\text{best}}\right), \tag{7}$$

where the product $\oplus$ denotes entrywise multiplications whereas $\mathbf{e}^{\text{best}}$ is the best solution (egg) seen so far in terms of its fitness value. Finally, the new candidate solution $\mathbf{e}_i^{k+1}$ is calculated by using

$$\mathbf{e}_i^{k+1} = \mathbf{e}_i^k + \mathbf{c}_i. \tag{8}$$

(B) Replacing Some Nests by Constructing New Solutions. Under this operation, a set of individuals (eggs) are probabilistically selected and replaced with a new value. Each individual $\mathbf{e}_i^k$ ($i \in [1,\ldots,N]$) can be selected with a probability $p_a \in [0,1]$. In order to implement this operation, a uniform random number r_1 is generated within the range $[0,1]$. If r_1 is less than p_a, the individual $\mathbf{e}_i^k$ is selected and modified according to (5); otherwise $\mathbf{e}_i^k$ remains with no change. This operation can be resumed by the following model:

$$\mathbf{e}_i^{k+1} = \begin{cases} \mathbf{e}_i^k + \text{rand}\cdot\left(\mathbf{e}_j^k - \mathbf{e}_h^k\right), & \text{with probability } p_a, \\ \mathbf{e}_i^k, & \text{with probability } (1-p_a), \end{cases} \tag{9}$$

where rand is a random number normally distributed whereas j and h are random integers from 1 to N.

(C) The Elitist Selection Strategy. After producing $\mathbf{e}_i^{k+1}$ either by the operator A or by the operator B, it must be compared with its past value $\mathbf{e}_i^k$. If the fitness value of $\mathbf{e}_i^{k+1}$ is better than $\mathbf{e}_i^k$, then $\mathbf{e}_i^{k+1}$ is accepted as the final solution; otherwise, $\mathbf{e}_i^k$ is retained. This procedure can be resumed by the following statement:

$$\mathbf{e}_i^{k+1} = \begin{cases} \mathbf{e}_i^{k+1}, & \text{if } f\left(\mathbf{e}_i^{k+1}\right) < f\left(\mathbf{e}_i^k\right), \\ \mathbf{e}_i^k, & \text{otherwise.} \end{cases} \tag{10}$$

The elitist selection strategy denotes that only high-quality eggs (best solutions near to the optimal value) which are the most similar to the host bird's eggs have the opportunity to develop (next generation) and become mature cuckoos.

2.5. Flower Pollination Algorithm (FPA). The flower pollination algorithm (FPA), proposed by Yang [14], is an ECT inspired by the pollination process of flowers. In FPA, individuals emulate a set of flowers or pollen gametes which behaves based on biological laws of the pollination process. From the implementation point of view, in the FPA operation, a population $\mathbf{F}^k$ ($\{\mathbf{f}_1^k, \mathbf{f}_2^k, \ldots, \mathbf{f}_N^k\}$) of N flower positions (individuals) is evolved from the initial point ($k = 0$) to a total gen number of iterations (k = gen). Each flower $\mathbf{f}_i^k$ ($i \in [1,\ldots,N]$) represents a d-dimensional vector $\{f_{i,1}^k, f_{i,2}^k, \ldots, f_{i,d}^k\}$ where each dimension corresponds to a decision variable of the optimization problem to be solved. In FPA, a new population $\mathbf{F}^{k+1}$ is produced by considering two operators: local and global pollination. A probabilistic global pollination factor p is associated with such operators. In order to decide which operator should be applied to each current flower position $\mathbf{f}_i^k$, a uniform random number r_p is generated within the range $[0,1]$. If r_p is less than p, the global pollination operator is applied to $\mathbf{f}_i^k$. Otherwise, the local pollination operator is considered.

Global Pollination Operator. Under this operator, the original position $\mathbf{f}_i^k$ is displaced to a new position $\mathbf{f}_i^{k+1}$ according to the following model:

$$\mathbf{f}_i^{k+1} = \mathbf{f}_i^k + s_i\cdot\left(\mathbf{f}_i^k - \mathbf{g}\right), \tag{11}$$

where $\mathbf{g}$ is the global best position seen so far whereas s_i controls the length of the displacement. The s_i value is generated by a symmetric Lévy distribution according to (5)-(6).

Local Pollination Operator. In the local pollination operator, the current position $\mathbf{f}_i^k$ is perturbed to a new position $\mathbf{f}_i^{k+1}$ as follows:

$$\mathbf{f}_i^{k+1} = \mathbf{f}_i^k + \varepsilon\cdot\left(\mathbf{f}_j^k - \mathbf{f}_h^k\right); \quad i,j,h \in (1,2,\ldots,N), \tag{12}$$

where $\mathbf{f}_j^k$ and $\mathbf{f}_h^k$ are two randomly chosen flower positions, satisfying the condition $j \neq h \neq i$. The scale factor ε is a random number between $[-1,1]$.

3. IIR Model Identification (Problem Formulation)

System identification is the mathematical representation of an unknown system by using only input-output data. In a system identification configuration, an optimization algorithm attempts to iteratively determine the adaptive model parameters to get an optimal model for the unknown plant based on minimizing some error function between the output of the candidate model and the actual output of the plant.

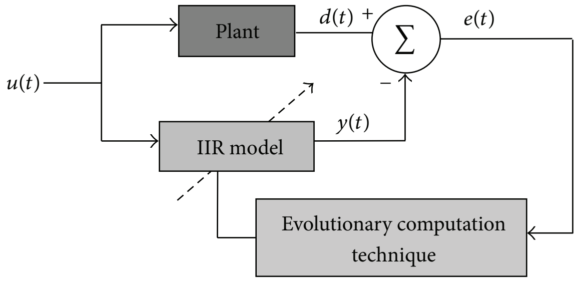


FIGURE 1: Adaptive IIR model for system identification.

The use of infinite impulse response (IIR) models for identification is preferred over their equivalent FIR (finite impulse response) models since the former produce more accurate models of physical plants for real world applications [6]. In addition, IIR models are typically capable of meeting performance specifications using fewer model parameters. Figure 1 represents an IIR identification model of any arbitrary system.

An IIR system can be represented by the transfer function:

$$\frac{Y(z)}{X(z)} = \frac{b_0 + b_1 z^{-1} + b_2 z^{-2} + \cdots + b_m z^{-m}}{1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_n z^{-n}}, \tag{13}$$

where m and n are the number of numerator and denominator coefficients of the transfer function, respectively, and, a_i and b_j are the pole and zero parameters of the IIR model ($i \in [1, \ldots, n]$, $j \in [1, \ldots, m]$). Equation (13) can be written as difference equation of the form:

$$y(t) = \sum_{i=1}^{n} a_i \cdot y(t-i) + \sum_{j=0}^{m} b_j \cdot x(t-j), \tag{14}$$

where $u(t)$ and $y(t)$ represent the tth input and output of the system, respectively. Therefore, the set of unknown parameters that models the IIR system is represented by $\theta = \{a_1, \ldots, a_n, b_0, \ldots, b_m\}$. Considering that the number of unknown parameters of θ is $(n+m+1)$, the search space $\mathbf{S}$ of feasible values for θ is $\mathfrak{R}^{(n+m+1)}$.

According to the block diagram of Figure 1, the output of the plant is $d(t)$ whereas the output of the IIR filter is $y(t)$. The output difference between the actual system and its model yields the error $e(t) = d(t) - y(t)$. Hence, the problem of IIR model identification can be considered as a minimization problem of the function $f(\theta)$ stated as the following:

$$f(\theta) = \frac{1}{W} \sum_{t=1}^{W} (d(t) - y(t))^2, \tag{15}$$

where W is the number of samples used in the simulation.

The aim is to minimize the cost function $f(\theta)$ by adjusting θ. The optimal model θ^* or solution is attained when the error function $f(\theta)$ reaches its minimum value, as follows:

$$\theta^* = \arg\min_{\theta \in \mathbf{S}} (f(\theta)). \tag{16}$$

4. Experimental Results

In the comparison study, a comprehensive set of experiments has been used to test the performance of each evolutionary computation technique. The set considers the use of IIR models with different orders. Such experimental set has been carefully selected to assure compatibility between similar works reported in the literature [18–21]. In the comparison, five ETC have been considered: PSO, ABC, EM, CS, and FPA.

The parameter setting for each evolutionary computation algorithm that is used in the comparison is described as follows.

(1) PSO: the parameters are set to $c_1 = 2$, $c_2 = 2$; besides, the weight factor decreases linearly from 0.9 to 0.2 [18].

(2) ABC: the algorithm has been implemented using the guidelines provided by its own reference [19], using the parameter *limit* = 100.

(3) EM: particle number = 50, $\delta = 0.001$, *LISTER* = 4, *MaxIter* = 300. Such values, according to [12, 20] represent the best possible configuration.

(4) CS: according to [13, 21], the parameters are set to $p_a = 0.25$ and the number of generations gen = 500.

(5) FPA: the probabilistic global pollination factor p is set to 0.8. Under such value, the algorithm presents the best performance according to [14].

For all algorithms, the population size has been set to 25 ($N = 25$) whereas the maximum iteration number has been configured to 3000 generations (gen = 3000).

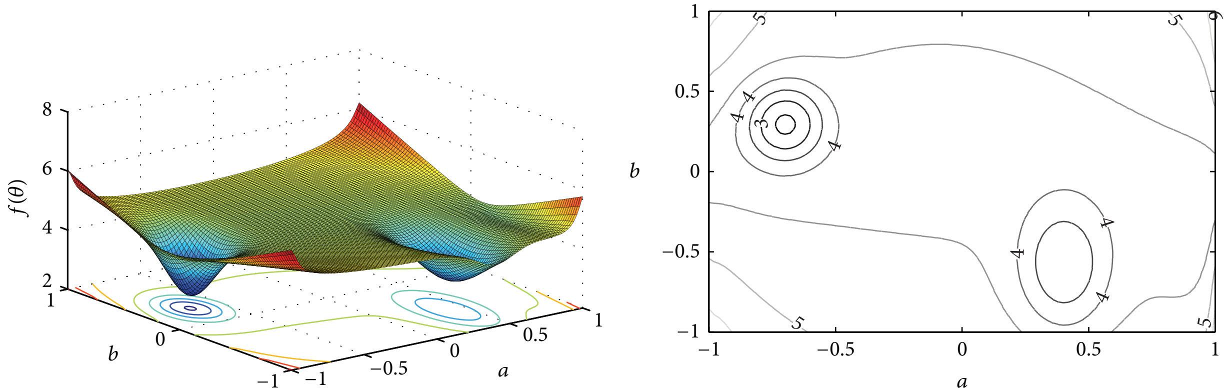
The results are divided into two sections. In the first set, the performance of each ETC for each identification experiment is presented. In the second set, the results are analyzed from a statistical point of view by using the Wilcoxon test.

4.1. IIR Model Identification Results. The results are reported considering three experiments that include (1) a second-order plant with a first-order IIR model; (2) a second-order plant with a second-order IIR model; and finally, (3) a high-order plant with a high-order model. Each case is discussed below.

(1) A Plant with a Second-Order System and a First-Order IIR Model (First Experiment). In this experiment, each algorithm is applied to identify a second-order plant through a first-order IIR model. Under such conditions, the unknown plant H_P and the IIR model H_M hold the following transfer functions:

$$H_P(z^{-1}) = \frac{0.05 - 0.4z^{-1}}{1 - 1.1314z^{-1} + 0.25z^{-2}},$$
$$H_M(z^{-1}) = \frac{b}{1 - az^{-1}}. \tag{17}$$

In the simulations, it has been considered a white sequence of 100 samples ($W = 100$) for the input $u(t)$. Since a reduced order model is employed to identify a plant of a superior



(a) (b)

Figure 2: Multimodal error surface $f(\theta)$ for the first experiment: (a) tridimensional figure and (b) contour.

order, $f(\theta)$ is multimodal [19]. The error surface $f(\theta)$ is shown in Figure 2.

The performance evaluation over 30 different executions is reported in Table 1 considering the following indexes: the best parameter values (ABP), the average $f(\theta)$ value (AV), and the standard deviation (SD). The best parameter values (ABP) report the best model parameters obtained during the 30 executions while the average $f(\theta)$ value (AV) indicates the average minimum value of $f(\theta)$, considering the same number of executions. Finally, the standard deviation (SD) reports the dispersion from the average $f(\theta)$ value regarding 30 executions.

According to Table 1, the CS algorithm provides better results than PSO, ABC, and EM. In particular, the results show that CS maintains a considerable precision (the lowest AV value) and more robustness (smallest SD value). Nevertheless, the CS performance is similar to the FPA algorithm. On the other hand, the worst performance is reached by the PSO algorithm. Such a fact corresponds to its difficulty (premature convergence) to overcome local minima in multimodal functions.

(2) A Plant with Second-Order System and Second-Order IIR Model (Second Experiment). In the second experiment, the performance for each algorithm is evaluated at the identification of a second-order plant through a second-order IIR model. Therefore, the unknown plant H_P and the IIR model H_M hold the following transfer functions:

$$H_P\left(z^{-1}\right)=\frac{1}{1-1.4z^{-1}+0.49z^{-2}},$$
$$H_M\left(z^{-1}\right)=\frac{b}{1+a_1z^{-1}+a_2z^{-2}}. \tag{18}$$

For the simulations, the input $u(t)$ that is applied to the system and to the IIR model simultaneously has been configured as a white sequence with 100 samples. Since the order of the model H_M is equal to the order of the to-be-identified system H_P, only one global minimum exists in $f(\theta)$ [19]. The results of this experiment over 30 different executions are reported in Table 2.

Table 1: Performance results of the first experiment.

Algorithms	ABP		AV	SD
	a	b		
PSO	0.9125	−0.3012	0.0284	0.0105
ABC	0.1420	−0.3525	0.0197	0.0015
EM	0.9034	0.3030	0.0165	0.0012
CS	0.9173	−0.2382	0.0101	3.118e − 004
FPA	0.9364	−0.2001	0.0105	5.103e − 004

The results in Table 2 show that PSO, ABC, EM, CS, and FPA have similar values in their performance. The evidence shows that evolutionary algorithms maintain a similar average performance when they face unimodal low-dimensional functions [29, 30]. In particular, the test remarks that the small difference in performance is directly related to a better exploitation mechanism included in CS and FPA.

(3) A Superior-Order Plant and a High-Order Model (Third Experiment). Finally, the performance for each algorithm is evaluated at the identification of a superior-order plant through a high-order IIR model. Therefore, the unknown plant H_P and the IIR model H_M hold the following transfer functions:

$$H_P\left(z^{-1}\right)=\frac{1-0.4z^{-2}-0.65z^{-4}+0.26z^{-6}}{1-0.77z^{-2}-0.8498z^{-4}+0.6486z^{-6}},$$
$$H_M\left(z^{-1}\right)=\frac{b_0+b_1z^{-1}+b_2z^{-2}+b_3z^{-3}+b_4z^{-4}}{1+a_1z^{-1}+a_2z^{-2}+a_3z^{-3}+a_4z^{-4}}. \tag{19}$$

Since the plant is a sixth-order system and the IIR model a fourth-order system, the error surface $f(\theta)$ is multimodal just as it is in the first experiment. A white sequence with 100 samples has been used as input. The results of this experiment over 30 different executions are reported in Tables 3 and 4. Table 3 presents the best parameter values (ABP) whereas Table 4 shows the average $f(\theta)$ value (AV) and its standard deviation (SD).

According to the AV and SD indexes in Table 4, the CS algorithm finds better results than PSO, ABC, EM, and FPA.

Table 2: Performance results of the second experiment.

Algorithms	ABP			AV	SD
	a_1	a_2	b		
PSO	−1.4024	0.4925	0.9706	$4.0035e-005$	$1.3970e-005$
ABC	−1.2138	0.6850	0.2736	0.3584	0.1987
EM	−1.0301	0.4802	1.0091	$3.9648e-005$	$8.7077e-005$
CS	−1.400	0.4900	1.000	0.000	0.000
FPA	−1.400	0.4900	1.000	$4.6246e-32$	$2.7360e-31$

Table 3: The best parameter values (ABP) for the second experiment.

Algorithms	ABP								
	a_1	a_2	a_3	a_4	b_0	b_1	b_2	b_3	b_4
PSO	0.3683	−0.7043	0.2807	0.3818	0.9939	−0.6601	−0.8520	0.2275	−1.4990
ABC	−1.1634	−0.6354	−1.5182	0.6923	0.5214	−1.2703	0.3520	1.1816	−1.9411
EM	−0.4950	−0.7049	0.5656	−0.2691	1.0335	−0.6670	−0.4682	0.6961	−0.0673
CS	0.9599	0.0248	0.0368	−0.0002	−0.2377	0.0031	−0.3579	0.0011	−0.5330
FPA	0.0328	−0.1059	−0.0243	−0.7619	1.0171	0.0038	0.2374	0.0259	−0.3365

Table 4: The average $f(\theta)$ value (AV) and the standard deviation (SD).

Algorithms	AV	SD
PSO	5.8843	3.4812
ABC	7.3067	4.3194
EM	0.0140	0.0064
CS	$6.7515e-004$	$4.1451e-004$
FPA	0.0018	0.0020

Table 5: P-values produced by Wilcoxon's test comparing CS vs PSO, ABC, EM and FPA over the "The average $f(\theta)$ values (AV)" from Tables 1, 2 and 4.

CS vs	PSO	ABC	EM	FPA
First experiment	$6.5455e-13$	$8.4673e-13$	$3.8593e-08$	0.7870
Second experiment	$1.5346e-14$	$1.5346e-14$	$1.5346e-14$	0.3313
Third experiment	$6.5455e-13$	$1.5346e-14$	$4.3234e-13$	0.1011

The results show that CS presents better precision (AV value) and robustness (SD value). These results also indicate that CS, FPA, and EM are able to identify the sixth-order plant under different accuracy levels. On the other hand, PSO and ABC obtain suboptimal solutions whose parameters weakly model the unknown system.

4.2. Statistical Analysis. In order to statistically validate the results, a nonparametric statistical significance-proof which is known as Wilcoxon's rank sum test for independent samples [31, 32] has been conducted over the "the average $f(\theta)$ value" (AV) data of Tables 1, 2, and 4 with a 5% significance level. The test has been conducted considering 30 different executions for each algorithm. Table 5 reports the P values produced by Wilcoxon's test for the pairwise comparison of the "the average $f(\theta)$ value" of four groups. Such groups are formed by CS versus PSO, CS versus ABC, CS versus EM, and CS versus FPA. As a null hypothesis, it is assumed that there is no significant difference between averaged values of the two algorithms. The alternative hypothesis considers a significant difference between the AV values of both approaches.

For the case of PSO, ABC, and EM, all P values reported in Table 5 are less than 0.05 (5% significance level) which is a strong evidence against the null hypothesis. Therefore, such evidence indicates that CS results are statistically significant and that it has not occurred by coincidence (i.e., due to common noise contained in the process). On the other hand, since the P values for the case of CS versus FPA are more than 0.05, there is not statistical difference between both. Therefore, it can be concluded that the CS algorithm is better than PSO, ABC, and EM in the application of IIR modeling for system identification. However, CS presents the same performance as FPA and therefore there is not statistical evidence that CS surpasses the FPA algorithm.

5. Conclusions

This paper presents a comparison study between five evolutionary algorithms for the IIR-based model identification. Under this research, the identification task is considered as an optimization problem. In the comparison, special attention is paid to recently developed algorithms such as the cuckoo search (CS) and the flower pollination algorithm (FPA), also including popular approaches such as the particle swarm optimization (PSO), the artificial Bee colony optimization (ABC), and the electromagnetism-like (EM) optimization algorithm.

The comparison has been experimentally evaluated over a test suite of three benchmark experiments that produce multimodal functions. The experiment results have demonstrated that CS outperforms PSO, ABC, and EM in terms of both the accuracy (AV values) and robustness (SD values), within a statistically significant framework (Wilcoxon test). However, there is not statistical evidence that CS surpasses the FPA performance.

The remarkable performance of CS and FPA is explained by two different features: (i) operators (such as Lévy flight) that allow a better exploration of the search space, increasing the capacity to find multiple optima, and (ii) their exploitation operators that allow a better precision of previously found solutions.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] X. Zhou, C. Yang, and W. Gui, "Nonlinear system identification and control using state transition algorithm," *Applied Mathematics and Computation*, vol. 226, pp. 169–179, 2014.

[2] M. Albaghdadi, B. Briley, and M. Evens, "Event storm detection and identification in communication systems," *Reliability Engineering and System Safety*, vol. 91, no. 5, pp. 602–613, 2006.

[3] P. Frank Pai, B.-A. Nguyen, and M. J. Sundaresan, "Nonlinearity identification by time-domain-only signal processing," *International Journal of Non-Linear Mechanics*, vol. 54, pp. 85–98, 2013.

[4] H.-C. Chung, J. Liang, S. Kushiyama, and M. Shinozuka, "Digital image processing for non-linear system identification," *International Journal of Non-Linear Mechanics*, vol. 39, no. 5, pp. 691–707, 2004.

[5] J. Na, X. Ren, and Y. Xia, "Adaptive parameter identification of linear SISO systems with unknown time-delay," *Systems & Control Letters*, vol. 66, no. 1, pp. 43–50, 2014.

[6] O. Kukrer, "Analysis of the dynamics of a memoryless nonlinear gradient IIR adaptive notch filter," *Signal Processing*, vol. 91, no. 10, pp. 2379–2394, 2011.

[7] T. Mostajabi, J. Poshtan, and Z. Mostajabi, "IIR model identification via evolutionary algorithms—a comparative study," *Artificial Intelligence Review*, 2013.

[8] C. Dai, W. Chen, and Y. Zhu, "Seeker optimization algorithm for digital IIR filter design," *IEEE Transactions on Industrial Electronics*, vol. 57, no. 5, pp. 1710–1718, 2010.

[9] W. Fang, J. Sun, and W. Xu, "A new mutated quantum-behaved particle swarm optimizer for digital IIR filter design," *Eurasip Journal on Advances in Signal Processing*, vol. 2009, Article ID 367465, 7 pages, 2009.

[10] J. Kennedy and R. C. Eberhart, "Particle swarm optimization," in *Proceedings of the IEEE International Conference on Neural Networks*, vol. 4, pp. 1942–1948, December 1995.

[11] D. Karaboga, "An idea based on honey bee swarm for numerical optimization," Tech. Rep. TR06, Computer Engineering Department, Engineering Faculty, Erciyes University, 2005.

[12] B. İlker, S. Birbil, and F. Shu-Cherng, "An electromagnetism-like mechanism for global optimization," *Journal of Global Optimization*, vol. 25, no. 3, pp. 263–282, 2003.

[13] X.-S. Yang and S. Deb, "Cuckoo search via Lévy flights," in *Proceedings of the World Congress on Nature and Biologically Inspired Computing (NABIC '09)*, pp. 210–214, December 2009.

[14] X. S. Yang, "Flower pollination algorithm for global optimization," in *Unconventional Computation and Natural Computation*, vol. 7445 of *Lecture Notes in Computer Science*, pp. 240–249, Springer, Berlin, Germany, 2012.

[15] C. Ahn, *Advances in Evolutionary Algorithms: Theory, Design and Practice*, Springer, New York, NY, USA, 2006.

[16] R. Chiong, T. Weise, and Z. Michalewicz, *Variants of Evolutionary Algorithms for Real-World Applications*, Springer, New York, NY, USA, 2012.

[17] M. Oltean, "Evolving evolutionary algorithms with patterns," *Soft Computing*, vol. 11, no. 6, pp. 503–518, 2007.

[18] S. Chen and B. L. Luk, "Digital IIR filter design using particle swarm optimisation," *International Journal of Modelling, Identification and Control*, vol. 9, no. 4, pp. 327–335, 2010.

[19] N. Karaboga, "A new design method based on artificial bee colony algorithm for digital IIR filters," *Journal of the Franklin Institute*, vol. 346, no. 4, pp. 328–348, 2009.

[20] E. Cuevas and D. Oliva, "IIR filter modeling using an algorithm inspired on electromagnetism," *Ingeniería Investigación y Tecnología*, vol. 14, no. 1, pp. 125–138, 2013.

[21] A. P. Patwardhan, R. Patidar, and N. V. George, "On a cuckoo search optimization approach towards feedback system identification," *Digital Signal Processing*, vol. 32, pp. 156–163, 2014.

[22] D. H. Wolpert and W. G. Macready, "No free lunch theorems for optimization," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 1, pp. 67–82, 1997.

[23] E. Elbeltagi, T. Hegazy, and D. Grierson, "Comparison among five evolutionary-based optimization algorithms," *Advanced Engineering Informatics*, vol. 19, no. 1, pp. 43–53, 2005.

[24] D. Shilane, J. Martikainen, S. Dudoit, and S. J. Ovaska, "A general framework for statistical performance comparison of evolutionary computation algorithms," *Information Sciences*, vol. 178, no. 14, pp. 2870–2879, 2008.

[25] V. Osuna-Enciso, E. Cuevas, and H. Sossa, "A comparison of nature inspired algorithms for multi-threshold image segmentation," *Expert Systems with Applications*, vol. 40, no. 4, pp. 1213–1219, 2013.

[26] Y.-L. Lin, W.-D. Chang, and J.-G. Hsieh, "A particle swarm optimization approach to nonlinear rational filter modeling," *Expert Systems with Applications*, vol. 34, no. 2, pp. 1194–1199, 2008.

[27] I. Pavlyukevich, "Lévy flights, non-local search and simulated annealing," *Journal of Computational Physics*, vol. 226, no. 2, pp. 1830–1844, 2007.

[28] R. N. Mantegna, "Fast, accurate algorithm for numerical simulation of Lévy stable stochastic processes," *Physical Review E*, vol. 49, no. 5, pp. 4677–4683, 2007.

[29] E. Cuevas, M. González, D. Zaldivar, M. Pérez-Cisneros, and G. García, "An algorithm for global optimization inspired by collective animal behavior," *Discrete Dynamics in Nature and Society*, vol. 2012, Article ID 638275, 24 pages, 2012.

[30] E. Cuevas, M. Cienfuegos, D. Zaldívar, and M. Pérez-Cisneros, "A swarm optimization algorithm inspired in the behavior of the social-spider," *Expert Systems with Applications*, vol. 40, no. 16, pp. 6374–6384, 2013.

[31] S. García, D. Molina, M. Lozano, and F. Herrera, "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*, vol. 15, no. 6, pp. 617–644, 2009.

[32] D. Shilane, J. Martikainen, S. Dudoit, and S. J. Ovaska, "A general framework for statistical performance comparison of evolutionary computation algorithms," *Information Sciences*, vol. 178, no. 14, pp. 2870–2879, 2008.