Abstract — Time series is an ordered sequence of observations in an equal interval space; this ordering is generated through time or other dimensions such as space. Time series occur in a variety of fields such as engineering. In this paper, Differential Evolution (DE) algorithm is used to optimize the maximum likelihood function \( \ln(L(\theta_1, \theta_2)) \) of the mixed \( ARMA(1,1) \) model. Simulation results based on MSE were compared with those obtained from the sDA, showed that the Differential Evolution can give good estimator of \( (\theta_1, \theta_2) \) for the \( ARMA(1,1) \) model better than that of sDA and Genetic Algorithm.

Keywords — Differential Evolution (DE), Genetic Algorithm (GA), Moving Average (MA), Likelihood Function, Moment Estimation Method, Mean Square Error (MSE).

I. INTRODUCTION

One of the most famous procedures for the solution of optimization problems is Genetic Algorithms (GAs). GA is composed mainly of three steps: recombination, crossover and mutation. By maintaining a population of solutions, GA can be viewed as implicitly modeling of the solutions seen in the search process. In the standard GA, new solutions are generated by applying randomized recombination operators on two or more high-quality individuals of the current population [1]. These recombination operators, such as one-point, two-point or uniform crossover, randomly selected non-overlapping subsets of two "parent" solutions to form "children" solutions.

The poor behavior of genetic algorithms in some problems, sometimes attributed to designed operators and the utilization of binary encoding have led to the development of other types of algorithms such as Differential Evolution (DE) algorithm. It is a class of Evolutionary Algorithms which has been developed recently to solve the problems related with the standard GA [2-3]. The key principal concept in this technique is to increase the exploitation and exploration ability of the search process by utilizing a continuous encoding for the problem (individuals) in hand and applying floating point operators (mutation and crossover) on these individuals. As a case study to investigate the relative performance of DE for optimizing the solution of estimation problem, we have utilized DE for optimizing the maximum likelihood function \( \ln(L(\theta_1, \theta_2)) \) of the mixed \( ARMA(1,1) \) model.

Formally, time series is represented in a stochastic model known as mixed autoregressive-moving average model, \( ARMA(1,1) \) [4]

\[
\begin{align*}
    z_t &= a_t - \theta_1 a_{t-1} - \cdots - \theta_q a_{t-q} \\
    \theta(B) &= 1 - \theta_1 B - \cdots - \theta_q B^{q-1}
\end{align*}
\]

where \( a_t \sim \text{idd } N(0, \sigma_a^2) \), meaning that the \( a_t \) are identically, independently distributed, each with a normal distribution having mean 0 and the same variance. This model employs \( (p + q + 2) \) unknown parameters \( \mu, \theta_1, \theta_2, \ldots, \theta_p, \theta_1, \theta_2, \ldots, \theta_q, \sigma_a^2 \) that are estimated from the data. The moving average process is stationary for any value of \( \theta_1, \theta_2, \ldots, \theta_q \), that is, the mean and the variance of the underlying process are constant and autocovariance’s depends only on the time lag. But many economic and business time series are sometimes considered to be non-stationary. Non-stationary time series can occur in many different ways. Sometimes the series has a non-stationary behavior about a fix mean, and hence its behavior can be represented by a model which calls for \( d^{th} \) difference of the process to be stationary. In practice \( d \) is usually 0, 1, or at most 2. This behavior can be represented by \( (p, d, q) \) model for the \( d^{th} \) difference to be stationary [4]. Then the model takes the formula,

\[
w_t = \phi_1 w_{t-1} + \cdots + \phi_p w_{t-p} + a_t - \theta_1 a_{t-1}
\]

or in its operator form,

\[
\phi(B)w_t = \theta(B)a_t
\]

where \( w_t = \nabla^d z_t \) and the moving average operator,

\[
\phi(B)(1-B)^d z_t = \theta(B)a_t
\]

Eq. 4 provides a powerful model for describing stationary and non-stationary time series and it is called an integrated moving average model (IMA) process or \( ARIMA \) of order \( (0,d,q) \). Mixed model of first order is known as first order autoregressive-moving average model and it is denoted by \( ARMA(1,1) \). Then Eq. 1-2 can be changed to be,

\[
z_t = \phi_1 z_{t-1} + a_t - \theta_1 a_{t-1}
\]

or

\[
(1 - \phi_1 B)z_t = (1 - \theta_1 B)a_t
\]

This model will be invertible and stationary if the root of \( (1 - \phi_1 B) = 0 \) lies outside the unit circle; and invertible MA is time reversible, so we can get that \( |\phi_1| < 1 \) and \( |\theta_1| < 1 \).
Estimation is the second step in analysis of the time series. It indicates an efficient use of the data to make inferences about the parameters conditional to the adequacy of the entrained model. ARMA models can be difficult to estimate if the parameter estimates are not within the appropriate range, a moving average model’s residual terms will grow exponentially. The calculated residuals for later observations can be very large or can overflow. This can happen either because of improper starting values being used or because the iterations moved away from reasonable values. Moreover, our model is nonlinear because \( a_t = \frac{1-\phi_1\theta_1}{1-\theta_1\theta_2} \) so there is no direct method that can handle these limitations, but all suitable methods are indirect methods (Iterative method) which start with an initial value and then this value is modified iteratively by using some numerical algorithms. The numerical method gives an approximate estimator with some accuracy.

Recently, B. Hussain [5-6] proposed the use of Canonical Genetic Algorithm (CGA) for optimizing the maximum likelihood function \( L(\theta, \sigma^2_\theta) \) of the first order moving average MA(1) model. And its results were compared with the results obtained by the moment estimator method.

In this paper we introduce a new evolutionary way to estimate the same model by using DE algorithm. We also presented the behavior of the DE for non-linear functions in comparison with GA according to the number of function evaluations taken, solution quality, the percentage of the search space searched until convergence and the convergence speed.

The rest of the paper is organized as follows. Section II explains the problem formulation of maximum likelihood estimator. Section III briefly describes the standard DE. Section IV presents the simulation results obtained from applying CGA, cGA and moment method for optimizing the model under study. Section V concludes the paper.

II. PROBLEM FORMULATION

Maximum likelihood estimator (MLE) is a standard approach to parameter estimation and inference in statistics; it is a method that finds the most likely value for the parameter based on the data set collected, in particular in non-linear modeling with non-normal data. MLE has many optimal properties in estimation: sufficiency (complete information about the parameter of interest contained in its MLE estimator); consistency (true parameter value that generated the data recovered asymptotically, i.e. for data of sufficiently large samples); efficiency (lowest-possible variance of parameter estimates achieved asymptotically); and parameterization invariance (same MLE solution obtained independent of the parameterization used) [14].

In order to study the \( \text{ARMA}(1,1) \) model, let us assume that a time series which is denoted by \( z_{-d+1}, \ldots, z_0, z_1, z_2, \ldots, z_n \) is generated by Equation 6 model over \( N = n + d \) original observations \( z \). Then, the stationary mixed \( \text{ARMA}(0,1) \) model in Equation 6 is rewritten as [7-8],

\[
a_t = w_t + \varphi_1 w_{t-1} + \theta_1 a_{t-1}
\]  

where \( E(w_t) = 0 \). Suppose that \( \{a_t\} \) has the normal distribution with zero mean and constant variance equal to \( \sigma^2_a \), then the likelihood function can be written as follows [14-15],

\[
L = (2\pi \sigma^2)^{-\frac{n}{2}} |M^{(1,1)}|^{\frac{1}{2}} \exp \left(-\frac{s(\theta_1, \theta_1)}{2\sigma^2_\theta} \right)
\]  

where

\[
M^{(1,1)} = \text{var} - \text{cov}(\theta_1, \theta_1) = I^{-1}(\theta_1, \theta_1) = \frac{1}{\text{adj}(I(\theta_1, \theta_1))}
\]

If \( M^{(1,1)} = I^{-1}(\theta_1, \theta_1) \), then the logarithmic likelihood function will be given by,

\[
\ln(L) = -\frac{n}{2} (2\pi \sigma^2) + \frac{1}{2} \ln(|M^{(1,1)}|) - \frac{s(\theta_1, \theta_1)}{2\sigma^2_\theta}
\]

where

\[
I(\theta_1, \theta_1) = \frac{n}{\sigma^2_\theta} \left[ \begin{array}{cc}
\sigma^2_\theta & \sigma^2_\theta \\
\sigma^2_\theta & 1-\theta_1\theta_2
\end{array} \right]^{-1}
\]

and

\[
S(\theta_1, \theta_1) = \sum^n_{t=0} (a_t \mid \varphi, \theta_1, w)^2
\]

is the sum squares errors and \( \{a_t\} \mid \varphi, \theta_1, w \) denotes the expectation of \( a_t \) conditional on \( \varphi, \theta_1 \) and \( w \). The sum squares of errors can be found by unconditional calculation of the \( \{a_t\} \) which are computed recursively by taking expectations in Equation 13, it is also called Least Square Estimate (LSE) in which the parameter estimated is obtained by minimizing the sum of square in Equation 13, it usually provides very close approximation to the maximum likelihood estimator. Back-forecasting is a popular technique, it estimates the parameters which are crudely put into the model and run backwards in time; a back-calculation provides the values \( w_{-j} \), \( j = 0, 1, 2, \ldots \) this back-forecasting is needed to start off the forward recursion. For moderate and large values of \( n \), Equation 13 is dominated by \( S(\theta_1, \theta_1)/2\sigma^2_\theta \) and thus the contours of the unconditional sum squares function in the space of the parameters \( (\theta_1, \theta_1) \) are very nearly contours of likelihood and log-likelihood.

III. DIFFERENTIAL EVOLUTION (DE)

The Differential Evolution (DE) is similar to the GA but requires fewer steps, fewer parameters and less of a gene sample, as well as less computational complexity as the individual genes are encoded directly as floating point numbers, not binary [9-10]. Fig 1 shows flowchart of the DE.

A. Individual Initialization and Encoding:

To apply the DE to real-values parameters optimization problems of the form \( f : \prod[u_t, v_t] \rightarrow R \) (\( u < v \)), the individual is logically divided into \( n \) segments of (in most cases) equal length \( l_t (t = nl_i) \) and each segment is interpreted as the floating-point code of the corresponding
object variable $x_i \in [u_i, v_i]$. A segment decoding function $\Gamma^i: [0,1]^l \rightarrow [u_i, v_i]$ typically looks like

$$\Gamma^i(a_{i1}a_{i2}...a_{in}) = u_i + \frac{v_i - u_i}{2^i - 1} \sum a_{ij}2^{j-1}.$$ 

where $(a_{i1}a_{i2}...a_{in})$ denotes the $i^{th}$-segment of an individual $\vec{a} = (a_{i1}, ..., a_{in}) \in \Gamma^n = \Gamma^i$. These individuals are called the target vectors.

**B. Fitness Evaluation:**

A fitness function is a numerical value associated with each individual to measure the goodness of the solution. The individual with higher fitness value represents better solution, while lower fitness value is attributed to the individual whose bit-string represents inferior solution. Combining the segment-wise decoding function to individual-decoding function $\Gamma = \Gamma^1 \times ... \times \Gamma^n$, [11] fitness values are obtained by setting

$$\Phi(\vec{a}) = \delta \left( f(\Gamma(\vec{a})) \right)$$

where $\delta$ denotes a scaling function ensuring positive fitness values such that the best individual receives largest fitness.

**C. Mutation**

Mutation is a differential operation between two randomly selected individuals $r_1$ and $r_2$ from the current population and should be different from the current individual $i$; to generate the donor vector ($V_i$) using the following equation,

$$v_i^{\theta+1} = x_{r_1}^\theta + F(x_{r_2}^\theta - x_{r_3}^\theta)$$

where $F$ is the mutation amplification factor with value $\in [0, 2]$.

**D. Crossover**

After the mutation operation, a crossover operation called binomial crossover ($bin$) is performed between the donor vectors ($V_i$) and the target vectors ($A_i$), using the equation,

$$u_{ij}^\theta = \begin{cases} v_{ij}^\theta & \text{if } \alpha_{ij} \leq CR \text{ or } (j = j_{random}) \\ x_{ij}^\theta & \text{otherwise } j = 1, 2, ..., l \end{cases}$$

where $CR$ is the crossover probability, $\alpha_{ij}$ is a random uniform real number. A new individual will be produce from this operation which called the trial vector ($U_i$).

**E. Selection**

A competition between the fitness values of the target and trial vector is held during the selection operation and the better will survive to the next generation as follows,

$$A_{i}^{\theta+1} = \begin{cases} U_{i}^{\theta} & \text{if } f(U_{i}^{\theta}) \leq f(A_{i}^{\theta}) \\ A_{i}^{\theta} & \text{otherwise} \end{cases}$$

IV. Experimental Results

This section presents the simulation results obtained and the comparison conducted between the DE and GA in terms of solution quality for the likelihood estimator of ARMA(1,1) [5-6]. All simulation results are attained by triggering 5 distinct runs; each run has 60 generations, then averaging the results data. Furthermore, the results of these former methods have been compared with those obtained by steepest descent method sDA based on initial values obtained by moment method for the same value of $(\theta, \phi)$ with 1000 runs. The simulation results performed are based on different sample size (i.e. $n = 25, 50, 100$), $\theta$ is set to $(\mp 0.1, \mp 0.4, \pm 0.3, \mp 0.6)$ and $\phi$ is set to $(\pm 0.2, \pm 0.4, \pm 0.5, \pm 0.8)$. The random variables $a_i$’s are generated by using Box-Muller formula and sample of size $n$ generated by Equation 3. The comparison has been based on Mean Square Error $MSE = var(\theta) + bias$.

The DE parameter settings used $F = 0.5$ and $CR = 0.7$. GA used binary tournament selection without replacement, and uniform crossover with exchange probability $P_c = 0.75$. Inversion mutation is used with probability $P_m = 0.005$. The population size is set to 50. All runs end when the population fully converged that is when the individuals have the same alleles at each gene position.

![DE general flow-chart][12]
the model parameters \((\theta_1, \phi_1)\) take positive values is smaller than when these parameters are assigned to a negative ones. Formally speaking, there is also evidence that the two algorithms are quite different, while GA has a memory requirement of \((P_x \times l)\) bits, the DE requires only \((P_x \times l)\) bits.

Formally speaking, there is also evidence that the two algorithms are quite different, while GA has a memory requirement of \((P_x \times l)\) bits, the DE requires only \((P_x \times l)\) bits.

<table>
<thead>
<tr>
<th>Table I</th>
<th>MSE For sDA, GA And DE For Different Sample Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>(n)</td>
<td>(\theta)</td>
</tr>
<tr>
<td>25</td>
<td>0.6 0.8</td>
</tr>
<tr>
<td></td>
<td>0.4 0.5</td>
</tr>
<tr>
<td></td>
<td>-0.1 -0.2</td>
</tr>
<tr>
<td></td>
<td>-0.3 -0.4</td>
</tr>
<tr>
<td></td>
<td>0.6 0.8</td>
</tr>
<tr>
<td></td>
<td>0.4 0.5</td>
</tr>
<tr>
<td>75</td>
<td>-0.1 0.2</td>
</tr>
<tr>
<td></td>
<td>-0.3 -0.4</td>
</tr>
<tr>
<td></td>
<td>0.6 0.8</td>
</tr>
<tr>
<td></td>
<td>0.4 0.5</td>
</tr>
<tr>
<td></td>
<td>-0.1 0.2</td>
</tr>
<tr>
<td>125</td>
<td>-0.3 -0.4</td>
</tr>
</tbody>
</table>

From Fig. 2, it is clear that the quality of solutions and convergence speed found by the DE is better than these obtained by GA. Ultimately, the results suggest that the DE performs the best and the GA performs the worst.

Fig. 2 Best Objective Function (MSE) of 60 generations for DE and GA algorithms

V. CONCLUSION

In this paper, we investigate the performance of the Differential Evolution algorithm for estimating the parameter of log-likelihood function of first order moving average model \(ARMA(1,1)\). Based on MSE, DE provides effective results for three random samples with different sizes \((n = 25, 75, 125)\) with \((\theta, \phi)\) that are set to \((\pm 0.1, \pm 0.4, \pm 0.3, \pm 0.6)\) and \((\pm 0.2, \pm 0.4, \pm 0.5, \pm 0.8)\), respectively; in comparison with the GA and sDA methods. Simulation results also show that the DE has a higher optimal precision or at least the same as that obtained from the GA, at same time, the DE needs minimum computational complexity than that of the GA.

REFERENCES


Prof. Dr. Mohd Sapiyan bin Baba - received the B.Sc. degree in Mathematical Computation from Essex University, Essex, UK, in 1977, the Master of Numerical Analysis and Programming from Dundee University, Dundee, Scotland in 1979, and the PhD degree from Keele University, Keele, UK in 1997. He is currently professor at Gulf University of Science and Technology- Kuwait. Prof. Dr. Sapiyan is the author and coauthor of more than 120 publications in international journals and proceedings. His research interest includes: Information- Computer Communication and Artificial Intelligence - cognitive science (e.g. Cognitive Model Robotics way finding, Bioinformatics: Data Mining and Image Processing, Multimedia Visualization).

Rawaa Dawoud Al-Dabbagh - received the B.Sc. from University of Al-Nahrain, Baghdad, Iraq in 1996 and M.Sc. Degree in Computer Science from the University of Baghdad, Baghdad, Iraq in 2000. Currently, she is a lecturer at the Computer Science Department - College of Science-University of Baghdad, and doing her PhD at University of Malaya-Artificial Intelligence Department, Kuala Lumpur, Malaysia. Rawaa is the author and coauthor of more than 6 publications in international journals and proceedings. Her main research interest includes: Soft Computing (Genetic Algorithm, Differential Evolution, Neural Network, and Fuzzy Logic), Security, and Natural Language Processing.