Estimations of Parameters of Delay Differential Equation Using Genetic Algorithm

Yoso Armando¹, Kie Van Ivanky Saputra¹*, Lina Cahyadi¹

¹Department of Mathematics, Universitas Pelita Harapan

*For correspondence: kie.saputra@uph.edu

Abstract In this paper, the numerical problem of estimating unknown parameters in systems of delay differential equations from complete data is treated. The numerical method, based on the genetic algorithm method with a fitness function to minimize the difference between the complete data and the numerical solution, is presented. The method is implemented in the framework of an epidemic model. Several models from epidemic applications such as SIR, SI, SIS and SIRS with time delay are treated. The genetic algorithm performs very well in estimating some parameters but is quite poor in estimating time delay parameter.

Keywords: delay differential equations, epidemic models, SIR, SIRS, SIS, SI, genetic algorithm, parameter estimation.

DOI: https://doi.org/xxxxxx

Introduction

This paper is intended to be an application of delay differential equations. Delay differential equations (DDEs) require a bit more mathematics than its ordinary differential equations (ODEs) counterparts. Various delay differential equations came naturally from applications; most of them range from population biology, physiology, epidemiology, economics, and neural networks, to control mechanical systems. Delay differential equations in some senses are more realistic because they include some of the past states of these systems. Like it or not, time delays occur so often in almost every situation, that to ignore them is to ignore reality.

Dynamics of ODEs and DDEs could have significant differences with just a little modification. Consider the following initial value problem in ODE:

\[
\frac{dx}{dt} = -x, x(0) = 1.
\] (1)

Solution of the above system, which is \(x(t) = e^{-t}\), will converge to zero as \(t \to \infty\). Let's modify the above initial value problem into a delay differential equation by introducing a time delay \(\tau\):

\[
\frac{dx}{dt} = -x(t - \tau), x(t) = 1, -\tau \leq t \leq 0.
\] (2)

The solution of the above DDE can be found analytically and is given by:

\[
x(t) = 1 + \sum_{k=1}^{n} \frac{(-1)^k(t - (k-1)\tau)^k}{k!}, (n - 1)\tau \leq t < n\tau, n \in \mathbb{N}.
\] (3)

Therefore, the solution of (2) is a polynomial of degree \(n\) on each subinterval \([((n-1)\tau, n\tau)\). When \(\tau > \pi/2\), the solution is unstable, in contrast to the solution of (1) which is stable and converge to zero as time goes to infinity. This is illustrated in Figure 1.

Another motivating example has been analyzed by Rusdi et al. [1], who investigated the effect of time delay in an epidemic model that includes both susceptible class and exposed class. The model is called SEIR model with time delay and given by

\[
\begin{align*}
\dot{S} &= \lambda - \beta S(t)I(t - \tau) - \mu S(t) \\
\dot{E} &= \beta S(t)I(t - \tau) - (\sigma + \mu + \epsilon)E(t) \\
\dot{I} &= \sigma E(t) - (\gamma + \mu + \eta)I(t) \\
\dot{R} &= \gamma I(t) - \mu R(t)
\end{align*}
\] (4)
Figure 1. Plot of solutions of differential equations without delay (left) and with delay (right). The solution of differential equation without delay converge to zero as time goes to infinity while the solution of differential equations with delays oscillates and is unstable.

Their results can be seen in Figure 2. They concluded that time delay does not change the stability of the equilibrium but only affects the dynamics in the short term (see Figure 2). When $\beta = 0.1$, the short-term dynamics of each class differ when the time delay is varied. Interesting dynamics takes place in class E and class I. When there is no time delay, the population of these classes increases intensely and at some time it decreases and finally it converges to its equilibrium. However, when there is a time delay, we do not have the same situation. In the beginning, the population of class I increases but it begins oscillating near the value of equilibrium.

DDEs play a very important role in application as it occurs naturally in just about every interaction of the real world. When we suspect an application following a DDE model, naturally we want to find parameters that fit the data. In this paper, we will derive a method to predict the parameters of DDE based on the data using a genetic algorithm.

Genetic algorithm is a search algorithm that is based on the Charles Darwin’s natural evolution theory. There are plenty features in this algorithm such as natural selection, reproduction, and mutation which will be used to find the fittest individuals. Genetic algorithms are frequently used to find solutions to optimization problems, optimizing decision trees for better performance, automatically solve sudoku puzzles, hyperparameter optimization, and many more. Five stages are considered in a genetic algorithm, which are (1) determining the initial population; (2) defining fitness function; (3) selection of individuals; (4) crossover; and (5) mutation. There are a lot of applications of genetic algorithm and one of them is to estimate parameters in mathematical model [2, 3]. In this paper, we will use genetic algorithm to find the best parameters for our model that fits the data.

Epidemic models with delay

Infectious diseases are characteristically categorized as either acute or chronic. When an infectious disease is called acute, it means the infection spreads fast. In this case, individuals will develop immune response that will remove pathogens after a short period of time (days or weeks). Examples of acute infections are influenza, chickenpox, and rubella. On the other hand, chronic infections can last for longer periods (months or years). Examples of chronic infections are herpes and chlamydia.

The acute infections that cause illness for a short period of time and are followed by immunity can be described by the so-called SIR models [4]. The model categorizes each individual into three different classes which are Susceptible (if previously unexposed to the pathogen), Infected (if currently occupied by the pathogen), and Recovered (if they have successfully cleared the infection). In this model, we have the transition $S \rightarrow I \rightarrow R$ and commonly expressed by the following model:

$$\dot{S} = -\beta S(t)I(t)$$
$$\dot{I} = \beta S(t)I(t) - \gamma I(t)$$
$$\dot{R} = \gamma I(t)$$

where $\beta$ is called the infection/contact rates and $\gamma$ is called the recovery rate. The previous model represents cases where the infection class is infectious for a short period of time. There are, however, numerous examples of animal and plant pathogens that are once infectious, it will be infectious forever. This scenario is called the SI model that is commonly expresses as:

$$\dot{S} = -\beta S(t)I(t)$$
$$\dot{I} = \beta S(t)I(t)$$

The previous model represents cases where the infection class is infectious for a short period of time. There are, however, numerous examples of animal and plant pathogens that are once infectious, it will be infectious forever. This scenario is called the SI model that is commonly expresses as:
Both SIR and SI models describe dynamics of acute infections that either give lifelong immunity once recovered or remain infectious forever. However, there are also infectious diseases that happen to have no long-lasting immunity, such as sexually transmitted infections, and bacterial infections. For such diseases, an individual can be infected multiple times and it is called the SIS model. The mathematical equation of this model is given by:

\[
\begin{align*}
\dot{S} &= -\beta S(t)I(t) + \gamma I(t) \\
\dot{I} &= \beta S(t)I(t) - \gamma I(t)
\end{align*}
\]  

(7)

Finally, there is another mathematical model that describes an intermediate assumption between short term immunity before waning such that, the individual is once again susceptible. It is called the SIRS model and is given by:

\[
\begin{align*}
\dot{S} &= -\beta S(t)I(t) + \alpha R(t) \\
\dot{I} &= \beta S(t)I(t) - \gamma I(t) \\
\dot{R} &= \gamma I(t) - \alpha R(t)
\end{align*}
\]  

(8)

where \(\alpha\) denotes the rate at which immunity is lost and recovered individuals move to susceptible class.

![Figure 2. Plot of solutions of differential equations (4) with different time delays. The solution of differential equation without delay converges to its equilibrium as time goes to infinity.](image)

In this paper, we shall incorporate time delays into the four ordinary differential equation models introduced before. Time delays are usually used to model the fact that an individual is not infectious until sometime after being infected. In the context of the epidemiology [5], this delay can be caused by a few factors such as (i) the latency of the infection in a vector, and (ii) the latency of the infection in an infected host. In these cases, sometimes should elapse before transmitting the infection even further. In this paper, we introduce a time delay \(\tau\), which describes the time needed for infectious individuals before transmitting the diseases. Note that under this assumption the incidence rate depends on the number of the susceptible host at this given moment \(t\), and on the number of the infected host at the moment \(t - \tau\). We obtain the following time-delay generalization for the above models:

1. SIR Model with time delay

\[
\begin{align*}
\dot{S} &= -\beta S(t)I(t - \tau) \\
\dot{I} &= \beta S(t)I(t - \tau) - \gamma I(t) \\
\dot{R} &= \gamma I(t)
\end{align*}
\]  

(8)
2. SI Model with time delay
\[ \dot{S} = -\beta S(t)I(t - \tau) \]
\[ \dot{I} = \beta S(t)I(t - \tau) \]  
(9)

3. SIS Model with time delay, and
\[ \dot{S} = -\beta S(t)I(t - \tau) + \gamma I(t) \]
\[ \dot{I} = \beta S(t)I(t - \tau) - \gamma I(t) \]  
(10)

4. SIRS Model with time delay
\[ \dot{S} = -\beta S(t)I(t - \tau) + \alpha R(t) \]
\[ \dot{I} = \beta S(t)I(t - \tau) - \gamma I(t) \]
\[ \dot{R} = \gamma I(t) - \alpha R(t) \]  
(11)

The meaning of variables and parameters (and their units) is given in the following:
- \( \beta \) : rate of infection/transmission (per months)
- \( \gamma \) : rate of recovery (per months)
- \( \alpha \) : rate at which immunity is lost (per months)
- \( \tau \) : time delay (months)
- \( S(t) \) : proportion of population that are susceptible
- \( I(t) \) : proportion of population that are infectious
- \( R(t) \) : proportion of population that recover

We also assume that that the population is closed which means there is no demographics (no births, deaths nor migrations). We assume that when an infectious agent is introduced in the beginning and the resulting epidemic occurs sufficiently quickly such that demographic processes are not influential. This implies that \( S(t) + I(t) + R(t) = 1 \) for SIR model and SIRS model and \( S(t) + I(t) = 1 \) for SI and SIS models.

Note that the analytical solution of non-linear delay differential equations is very difficult to find, therefore we will find the numerical solution instead. We will use DDE23 procedure [6] to find the numerical solution of each delay differential equation.

Parametric estimations
In this section we shall derive the process of parameter estimations using genetic algorithm. The general process of genetic algorithm is as follows: 1 randomly generating population of parameters; 2 defining a fitness function by comparing the data with the numerical solution generated by the chosen parameters; 3 selection process; 4 crossover process; and 5 mutation process. Let’s suppose we want to fit parameters of SIR model with time delay and suppose that we have data of variable S, I and R for every time unit. As parameters of the SIR model with time delay are \( \beta, \gamma, \) and \( \tau \), we shall generate a population of these parameters. We will define three vectors; each of these vectors represents the parameter we are estimating. Each vector consists of \( n \) chromosomes which represents possible values of each parameter. Ideally, we need to generate as many chromosomes as possible, which means \( n \) is supposed to be large.

In the next step, we will define the fitness function for our problem. For each chromosome of these variables, we can obtain the solution of DDE which will be compared to data. When the solution is close to data the fitness function will be large and when the solution is far from data, the fitness function will be very small. We will evaluate every possible chromosome and we will select some chromosomes which have higher fitness function values. We will eliminate chromosomes that give low fitness function values, and we will do crossover and mutation processes for selected chromosomes. We will repeat the selection, crossover, and mutation processes repeatedly until we find the best chromosome that gives the highest fitness function.

We will also do the same process for the other models to find the best parameters.

Computational results
The research in this paper is still in preliminary steps and the goal of this paper is to evaluate whether the genetic algorithm parameter estimation is able to estimate the parameters of our models. Therefore, in this research we will not use real data, instead we will generate solution of models (8)-(11) using some chosen values of parameters and we will perform genetic algorithm to estimate parameters of these models using the data generated in the previous step.
Consider model (8) with parameters $\beta = 0.05$, $\gamma = 1/30$, and time delay $\tau = 14/30$. The solution of this model is illustrated in Figure 3. The numerical solution is generated from time 0 until time 1000. Using this data, we are going to estimate these parameters using genetic algorithm. Because there is randomness in the genetic algorithm process, we repeat the process of estimating parameters 5 times.

To run genetic algorithm, we need to specify some numerical parameters. The first parameter is $sval$ which represents the proportion of chromosomes that will be selected after the selection process. The next parameter is called $cval$ which represents the proportion of chromosomes that will undergo crossover process. Finally, the final parameter is call $mval$ which represents the proportion of chromosomes that will experience mutation process. Numerical parameters used in this research are $sval = 0.7$, $cval = 0.7$, $mval = 0.7$ with total $n = 40$ chromosomes for initial population.

The result of our genetic algorithm parameter estimation is summarized in Table 1.

<table>
<thead>
<tr>
<th>Simulation</th>
<th>$\hat{\beta}$</th>
<th>$\hat{\gamma}$</th>
<th>$\hat{\tau}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.051101</td>
<td>0.033162</td>
<td>0.6168</td>
</tr>
<tr>
<td>2</td>
<td>0.050644</td>
<td>0.033247</td>
<td>0.526011</td>
</tr>
<tr>
<td>3</td>
<td>0.050764</td>
<td>0.033064</td>
<td>0.574724</td>
</tr>
<tr>
<td>4</td>
<td>0.049079</td>
<td>0.033648</td>
<td>0.341978</td>
</tr>
<tr>
<td>5</td>
<td>0.050778</td>
<td>0.033171</td>
<td>0.536195</td>
</tr>
</tbody>
</table>

From Table 1, we obtained that the predictions for $\beta$ and $\gamma$ are not far from the exact values. However, the estimation for the time delay coefficient $\tau$ is relatively far from the exact value. All genetic algorithm simulations gave different values of time delay, and it clearly shows that our algorithm cannot estimate this parameter well.

Next, let us consider the SI model (9) with parameters $\beta = 0.05$, and time delay $\tau = 14/30$. The solution is generated from time 0 until time 1000 and is illustrated in Figure 4. Using genetic algorithm, we are going to estimate these parameters and we will also repeat the process 5 times to see whether this algorithm can perform well for every iteration. We also use the same numerical parameters for the genetic algorithm which are $sval = 0.7$, $cval = 0.7$, $mval = 0.7$ with total $n = 40$ chromosomes for initial population.

The result of our genetic algorithm is shown in Table 2.
Table 2. Parameters of SI Model obtained from genetic algorithm

<table>
<thead>
<tr>
<th>Simulation</th>
<th>$\beta$</th>
<th>$\tau$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.048852</td>
<td>0.430904</td>
</tr>
<tr>
<td>2</td>
<td>0.049549</td>
<td>0.455423</td>
</tr>
<tr>
<td>3</td>
<td>0.050562</td>
<td>0.487537</td>
</tr>
<tr>
<td>4</td>
<td>0.049204</td>
<td>0.440113</td>
</tr>
<tr>
<td>5</td>
<td>0.050824</td>
<td>0.4921</td>
</tr>
</tbody>
</table>

From Table 2, we obtained that the prediction for $\beta$ is not far from the exact values. Moreover, in contrast to the estimation for the time delay coefficient $\tau$ in the previous case, the prediction of $\tau$ is better as it can predict the true value of the time delay.

Figure 4. Plot of numerical solutions of SI differential equations with time delay. Initial condition of this problem $S(0) = 0.98, I(0) = 0.02$. Parameters of this model are $\beta = 0.05$, and time delay $\tau = 14/30$.

**SIS model**

In the next step, we shall consider model (10) with parameters $\beta = 0.05, \gamma = 1/30$, and time delay $\tau = 14/30$. Again, the solution is generated from time 0 until time 1000. In Figure 5, we can see the solution of this model. Using data generated and using our genetic algorithm, we are going to estimate these parameters. The process will be repeated 5 times due to randomness of the procedure. We also use the numerical parameters for the genetic algorithm which are $sval = 0.7, cval = 0.7, mval = 0.7$ with total n = 40 chromosomes for initial population. The result of our genetic algorithm is shown in Table 3.

Table 3. Parameters of SIS Model obtained from genetic algorithm

<table>
<thead>
<tr>
<th>Simulation</th>
<th>$\hat{\beta}$</th>
<th>$\hat{\gamma}$</th>
<th>$\hat{\tau}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.374777</td>
<td>0.339414</td>
<td>0.157385</td>
</tr>
<tr>
<td>2</td>
<td>0.108405</td>
<td>0.090692</td>
<td>0.169483</td>
</tr>
<tr>
<td>3</td>
<td>0.616886</td>
<td>0.561141</td>
<td>0.183532</td>
</tr>
<tr>
<td>4</td>
<td>0.941819</td>
<td>0.867383</td>
<td>0.159347</td>
</tr>
<tr>
<td>5</td>
<td>0.421978</td>
<td>0.388920</td>
<td>0.103117</td>
</tr>
</tbody>
</table>

As one can see from Table 3, the prediction of parameters $\beta, \gamma$ and $\tau$ are quite far from the exact values. Moreover, every simulation produces different predictions which is quite surprising as the same algorithm gave us a pretty good result in the SIR and the SI models.
Figure 5. Plot of numerical solutions of SIS differential equations with time delay. Initial condition of this problem $S(0) = 0.98, I(0) = 0.02$. Parameters of this model are $\beta = 0.05, \gamma = 1/30$, and time delay $\tau = 14/30$.

SIRS model

Finally, let us consider the final model which is the SIRS model (11). Parameters used for this model are $\beta = 0.05, \gamma = 1/30, \alpha = 1/30$, and time delay $\tau = 14/30$, the solution is generated from time 0 until time 1000 (see Figure 6). We are going to estimate these parameters using data generated and the genetic algorithm. The process will be repeated 5 times due to randomness. We also use the numerical parameters for the genetic algorithm which are $sval = 0.7, cval = 0.7, mval = 0.7$ with total $n = 40$ chromosomes for initial population. The result of our genetic algorithm is shown in Table 4. From the result, we obtained that the predictions for $\beta$, $\gamma$, and $\alpha$ are not far from the exact values. However, the estimation for the time delay coefficient $\tau$ is relatively far from the exact value. All genetic algorithm simulations gave different values of time delay, and it clearly shows that our algorithm cannot estimate this parameter well.

Table 4. Parameters of SIRS Model obtained from genetic algorithm

<table>
<thead>
<tr>
<th>Simulation</th>
<th>$\hat{\beta}$</th>
<th>$\hat{\gamma}$</th>
<th>$\hat{\alpha}$</th>
<th>$\hat{\tau}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.048821</td>
<td>0.035150</td>
<td>0.036775</td>
<td>0.173193</td>
</tr>
<tr>
<td>2</td>
<td>0.050646</td>
<td>0.035214</td>
<td>0.036977</td>
<td>0.328961</td>
</tr>
<tr>
<td>3</td>
<td>0.050061</td>
<td>0.035796</td>
<td>0.038309</td>
<td>0.215528</td>
</tr>
<tr>
<td>4</td>
<td>0.049731</td>
<td>0.033678</td>
<td>0.032795</td>
<td>0.426350</td>
</tr>
<tr>
<td>5</td>
<td>0.048653</td>
<td>0.034051</td>
<td>0.034388</td>
<td>0.253655</td>
</tr>
</tbody>
</table>

Figure 6. Plot of numerical solutions of SIRS differential equations with time delay. Initial condition of this problem $S(0) = 0.98, I(0) = 0.02$ and $R(0) = 0$. Parameters of this model are $\beta = 0.05, \gamma = 1/30, \alpha = 1/30$, and time delay $\tau = 14/30$. 


Discussions
In this paper we have investigated the performance of genetic algorithm in predicting parameters of epidemic models with time delay.

There are four epidemic models with time delays considered in this paper which are SIR model, SI model, SIS model and SIRS model. In summary, parameter estimations for the SIR model with time delay produce a good result as it can estimate parameter $\beta$ and $\gamma$ well but it is not doing a good job estimating the time delay coefficient $\tau$. The genetic algorithm is doing very well in estimating parameters of SI model with time delay as all parameters were estimated with very good accuracy. For SIRS model with time delay, genetic algorithm also gives a good result as all parameters except the time delay were well predicted. However, a poor result was obtained in estimating the parameters the SIS model with time delay.

There are a few reasons why genetic algorithm did not do a very good performance in did not estimating parameters of models with time delay. The first reason could be the fact that the solution of the model with time delay was produced numerically not analytically. We use a DDE23 procedures which is actually a good algorithm to solve common delay differential equations. We suspect that the genetic algorithm is sensitive to the numerical solution produced as we cannot evaluate how good or how accurate the difference between the numerical solution and the true solution of our DDEs.

The second reason could be the numerical parameters of the genetic algorithm which are $n$, sval, cval and mval. In this paper we fixed these parameters, and we did not conduct an analysis when these parameters are varied. There could be some values such that the genetic algorithm would produce a good result perhaps by increasing the value of $n$ or tuning the value of the other parameters.

The final reason could be that the genetics algorithm or the DDE23 procedures depends heavily on the stability of the equilibrium of the model with time delays. In this paper, we did not investigate the stability analysis of equilibrium. We did some preliminary calculation and we obtained that all equilibrium of the four models analyzed in this paper have zero eigenvalue. Perhaps, this is one of the reasons why our genetic algorithm doesn’t perform very well in estimating the parameters of the models.

To conclude our discussion, perhaps in the future we could use real data to estimate the parameters of the model suspected. Even though our procedure is not great, but it is a good start to estimate parameters. To our knowledge there is a very little discussion on how to estimate parameters of delay differential equations

Conflict of interest
The author(s) declare(s) that there is no conflict of interest regarding the publication of this paper.

References