Parameter Estimation in Neutral Delay Differential Equations Using Genetic Algorithm With Multi-Parent Crossover

CRISTETA U. JAMILLA, RENIER G. MENDOZA, AND VICTORIA MAY P. MENDOZA

Institute of Mathematics, University of the Philippines Diliman, Quezon City 1101, Philippines
Natural Sciences Research Institute, University of the Philippines Diliman, Quezon City 1101, Philippines

Corresponding author: Renier G. Mendoza (rmendoza@math.upd.edu.ph)

This work was supported by the Natural Sciences Research Institute, University of the Philippines Diliman, under Grant MAT-21-1-02.

ABSTRACT Neutral delay differential equations (NDDEs) are differential equations containing time lags not only in the states but also in the state derivatives. NDDEs have applications in modeling physical and biological systems. An NDDE model may consist of several parameters, some of which can be determined using available data. Various numerical techniques have been studied to estimate parameters of mathematical models. In this study, parameter estimation is posed as an optimization problem. The use of heuristic algorithms in parameter estimation has gained popularity because of its ease of implementation, requiring only function evaluations. But to our knowledge, heuristic algorithms have never been employed in estimating parameters in NDDE models. In this work, we apply Genetic Algorithm with Multi-Parent Crossover (GA-MPC) to obtain parameter estimates of three NDDE models with a discrete delay. We compare the estimates to those obtained using standard heuristic algorithms. Results show that GA-MPC is capable of consistently identifying model parameters that provide a good fit of the model to the data.

INDEX TERMS Neutral delay differential equations, parameter estimation, genetic algorithm, bootstrap

I. INTRODUCTION

Differential equations, where the derivatives at time $t$ depend on the solution and its derivatives at a previous time, are called delay differential equations (DDEs). A general form of a DDE is given by

$$y^{(k)}(t) = f(t, y(t), \ldots, y^{(k-1)}(t), y(d_0), \ldots, y^{(L)}(d_L); \mathbf{p}), \quad (1)$$

where $d_j = d_j(t, y(t)), \ j = 0, \ldots, k$ are the delays which satisfy $d_j \leq t \forall t \in [t_0, t_1]$ for some $t_0, t_1 \in \mathbb{R}^+$. The function $f$ is parametrized by $\mathbf{p} \in \mathbb{R}^L$, where $L$ is the number of parameters. The solution to (1) is dependent on some initial function $\phi(t)$ defined at a previous time. This function $\phi(t)$ is referred to as the history or preshape function. A DDE is of retarded type (RDDE) when there is no delay in the derivative terms of the DDE. On the other hand, it is of neutral type (NDDE) if there are delays in the derivatives of the state variables.

The associate editor coordinating the review of this manuscript and approving it for publication was Sun-Yuan Hsieh.

Several studies have been done on the numerical approximations and asymptotic properties of solutions to NDDEs [1]–[7]. NDDEs have been used to develop models in engineering [8]–[12], biosciences [1], [13]–[17], and economics [18], [19]. A review of the applications of NDDEs in various fields is discussed in [20].

Here, we consider a single constant delay $\tau$. That is,

$$d_j(t, y(t)) = t - \tau, \quad \forall j.$$  

Note that the solution $y(t)$ to (1) depends on the parameter $\mathbf{p}$ and the delay $\tau$. Because $\tau$ can also be treated as an additional parameter, we let

$$\theta := [\mathbf{p}; \tau],$$

be the vector of parameters. We also denote by $y_0(t)$ the solution to (1) to make the dependence of $y(t)$ on the parameters and delay more obvious.

The values of the parameters in a model are often unknown. While some parameters are found in the literature, others need to be estimated. Ideally, we want the experimental data...
recorded at time $t_i$, denoted by $D_i$, to coincide with the solution to the model at $t_i$. That is,

$$D_i \approx y_{\theta}(t_i), \quad i = 1, 2, \ldots, N,$$

(2)

where $N$ is the number of data points. Hence, we look for the value of $\theta$ that will make the approximation in (2) as close as possible. This can be done by minimizing a least-squares formulation expressed as

$$\min_{\theta} \frac{1}{M} \sum_{i=1}^{N} (D_i - y_{\theta}(t_i))^2,$$

(3)

where $M$ is a normalizing constant. We denote the cost function to be minimized by

$$J(\theta) := \frac{1}{M} \sum_{i=1}^{N} (D_i - y_{\theta}(t_i))^2.$$  

(4)

There are many numerical optimization algorithms in the literature that can be used to solve (3). Classical techniques use the gradient of the cost function $J$ in (4) to estimate the minimizer. Gradient-based methods are fast but can be stuck to a local minimum or a saddle point [21]. Using derivative discontinuity tracking theory for DDEs, it was shown in [22] that jumps can arise in the gradient of $J$. Hence, to solve (3), we rely on a derivative-free method.

Numerical algorithms that only use function values of $J$ are either deterministic or heuristic. Deterministic algorithms are computationally inexpensive and fast but tend to converge to local minimizers [21], [23]. Heuristic algorithms are often nature-inspired [24] and are computationally more expensive than deterministic algorithms but can converge to a global minimizer given enough number of function evaluations [24]–[26].

In [27], a modified particle swarm optimization (PSO) was used to estimate the parameters of dynamical systems involving ordinary differential equations. It was shown in [28] how a hybrid adaptive cuckoo search with simulated annealing could be effective in identifying parameters in chaotic systems. A hybrid Taguchi-differential evolution algorithm was proposed in [29] to estimate the parameters of an HIV model. Heuristic algorithms have also been shown to be effective in identifying parameters of S-system models [30]–[32]. Other recent works on the application of various heuristic optimization methods on estimating parameters of engineering systems are found in [33]–[41].

Heuristic algorithms such as genetic algorithm (GA) [42], which is a hybrid of an evolutionary search strategy with a local multiple-shooting approach [43], and differential evolution [44] have been applied to parameter estimation in RDDE models. In [45], Ant Colony Optimization was used to determine parameter values in RDDE models on human health. To determine the unknown parameters in chaotic systems with time-delays, a hybrid bio-geography optimization was used in [46], PSO was used in [47], and a hybrid cuckoo search algorithm was used in [48].

To the best of our knowledge, heuristic algorithms have never been employed in estimating parameters of models involving NDDEs. In this paper, we solve the least-squares formulation in (3) using Genetic Algorithm with Multi-Parent Crossover (GA-MPC), and compare the results to those obtained using standard global optimization algorithms: GA, pattern search algorithm (PS), simulated annealing (SA), and PSO. GA-MPC is a modified GA that proposes a new crossover method [49]. It has been used to effectively solve an economic dispatch problem [50] and an optimal power flow problem [51]. It has also been applied to solve an inverse problem in RDDEs [52].

In the next section, we discuss GA-MPC and a parameter bootstrapping method. In Section III, parameter estimation is done on three NDDE models arising from applications using the heuristic algorithms mentioned above. The first model describes the population growth of E. coli using a first-order NDDE. The second one is an age-structured model involving a system of first-order NDDEs. The third model is a second-order NDDE describing a pendulum-mass-spring-damper (PMSD) system. Parameter bootstrapping is done to assess uncertainty in the estimates. Furthermore, since the only stopping criterion is the maximum number of function evaluations, we investigate the effect of varying this quantity to the cost function value. In the last section, we state our conclusions and recommendations.

II. METHODS

A. GENETIC ALGORITHM WITH MULTI-PARENT CROSSOVER

Genetic Algorithm was formulated by John Holland in 1975 based on Darwin’s theory of evolution [53]. It has gained popularity for its ease of use and applicability in various fields of study. Some applications of GA can be found in [25], [26], [42], [54]–[59].

The general process of GA starts with a pool of randomly generated individuals that serve as members of the first generation. The members of the population act as estimates of the minimizer. The best members of the population are selected based on their survival fitness. In the case of minimization, the lower the function value of an individual, the higher is its chance of survival. The members of the population who did not survive will be replaced by the offspring of the surviving population. The offspring are created using a crossover method. To make sure that the solution does not converge to a local minimum, a mutation step is done at the end of the generation. In this step, a portion of the population is modified randomly. The process is repeated until a stopping criterion is satisfied. A detailed discussion on GA, as well as the MATLAB codes, are found in [26].

GA-MPC proposes a new crossover method that uses three parents from the selection pool to generate three offspring (refer to Algorithm 1). Two of the offspring ($o_1$ and $o_3$ in Algorithm 1) are formulated in such a way that they are generated in the direction with less function value. These
Algorithm 1 Genetic Algorithm With Multi-Parent Crossover (GA-MPC) [49]

Require: Cost function \( f(\mathbf{x}), \mathbf{x} \in \mathbb{R}^n \), \( \text{PopSize}, \mathbf{x}_{LB}, \mathbf{x}_{UB}, M, \beta, cr, p, \text{MaxEval} \).

Ensure: \( f \) is optimal at \( \mathbf{x}^* \)

1: Randomly generate the initial population size \( \text{PopSize} \).
2: while FuncEvals < MaxEval do
3: Sort the population by their cost function value.
4: Generate tournament selection size \( TC \) randomly between 2 and 3. Do a tournament selection and fill the selection pool.
5: Randomly generate a number \( \tilde{u} \in [0, 1] \).
6: for each three consecutive individuals in the selection pool do
7: if one of the selected individual is the same as another individual then
8: Replace one by a randomly-selected individual in the selection pool.
9: end if
10: if \( \tilde{u} < cr \) then
11: Sort the three individuals \( x_i, x_{i+1}, x_{i+2} \) so that \( f(x_i) \leq f(x_{i+1}) \leq f(x_{i+2}) \).
12: Compute \( \beta = N(\tilde{u}, \tilde{\sigma}) \).
13: The three offspring are generated from the three parents using: \( o_1 = x_1 + \beta(x_2 - x_3), o_2 = x_2 + \beta(x_3 - x_1), o_3 = x_3 + \beta(x_1 - x_2) \).
14: end if
15: for \( i = 1 : 3 \) do
16: Randomly generate \( \tilde{u} \in [0, 1] \).
17: if \( \tilde{u} < p \) then
18: The offspring undergoes mutation: \( d'_i = x'_{\text{mut}} \),
19: where \( x'_{\text{mut}} \in \tilde{A} \) and \( \text{mut} \in [1, m] \).
20: end if
21: end for
22: if a duplicate individual \( x'_i \) exists then
23: Change the duplicate with \( x'_i = x'_i + N(0.5 \times \tilde{u}, 0.25 \times \tilde{u}) \), \( \tilde{u} \in [0, 1] \).
24: end if
25: end while

In the following simulations, the parameter values of GA-MPC are set to their default values as stated in [49]. That is, the population size \( \text{PopSize} = 90 \), selection rate \( M = 0.50 \times \text{PopSize} \), crossover rate \( cr = 1 \), crossover factor \( \beta \sim N(0.5, 0.3) \), and mutation rate \( p = 0.1 \).

The results obtained using GA-MPC are compared to those obtained using standard heuristic algorithms: GA, PS, SA, and PSO, which are implemented using the built-in functions \text{ga}, \text{patternsearch}, \text{simulannealbnd}, and \text{particleswarm}, respectively, in the global optimization toolbox of MATLAB. For consistency of comparison, all the parameters in these built-in functions are set to their default values. The only stopping criterion in all the algorithms is the maximum number of function evaluations. In the parameter estimation results in Section III, the maximum number of function evaluations is set to 2000 times the number of parameters to be estimated. Because heuristic algorithms are probabilistic, we run each of the algorithms 100 times and compute for the mean of the estimates. Box plots are used to illustrate the distribution of the parameter estimates. Furthermore, we explore the effect of the maximum number of function evaluations to the cost function value by setting this maximum number to the number of parameters to be estimated times 500, 1000, 1500, or 2000.

B. PARAMETER UNCERTAINTY ANALYSIS

In [61], a parameter bootstrapping approach is presented to quantify the standard error in the parameter estimates in models involving RDDEs. We adopt this strategy to analyze the uncertainty in parameter estimates in the NDDE models. The method is summarized in the following steps:

1) Solve the parameter estimation problem in (3) and denote the estimated parameter by \( \hat{\theta} \).
2) Let \( y_\theta(t) \) be the solution of the NDDE model. Set the maximum number \( L \) of realizations to 100. Initialize the number of realizations \( r \) to 1.
3) Using \( (y_\theta(t_i), \sigma^2), i = 1, \ldots, N \), generate a noisy data \( \{\hat{D}_i\}_{i=1}^N \) based on an assumed probability distribution. Here, the noisy data are generated using the normal distribution \( \text{Normal}(y_\theta(t_i), \sigma^2) \), \( i = 1, \ldots, N \), where \( \sigma \) is the noise set to 10% of the standard deviation of \( \{y_\theta(t_i)\}_{i=1}^N \) (compare with [61]).
4) Get a new set of parameter estimates using the simulated noisy data \( \{\hat{D}_i\}_{i=1}^N \). This is done by replacing \( \{D_i\}_{i=1}^N \) by \( \{\hat{D}_i\}_{i=1}^N \) in (4) and minimizing the modified cost function. Store this parameter as \( \hat{\theta}_r \).
5) If \( r < L \), set \( r \leftarrow r + 1 \) and go back to step 3. Otherwise, go to the next step.
6) The standard error of the parameter \( \hat{\theta} \), denoted by \( Err(\hat{\theta}) \), is equal to the standard deviation of \( \{\hat{\theta}_r\}_{r=1}^{100} \). The 95% confidence interval for the parameter \( \theta \) is \( [\hat{\theta} - 1.96 \times Err(\hat{\theta}), \hat{\theta} + 1.96 \times Err(\hat{\theta})] \). The standard error and 95% confidence interval of the solution to the NDDE are computed similarly.
C. U. Jamilla et al.: Parameter Estimation in NDDEs Using GA-MPC

FIGURE 1. Box plots of the parameter estimates for the *E. coli* population growth model using GA-MPC, GA, PS, SA, and PSO. The outliers, i.e., the estimates that fall outside the lower and upper quartiles, are plotted in red (+). The median and mean are represented by a red line across the box plots and black dot, respectively.

TABLE 1. Bounds for the parameters in the NDDE model of *E. coli* population growth.

| Parameter | $\rho_0$ | $\rho_1$ | $\rho_2$ | $\tau$ | $\beta$ |
|-----------|---------|---------|---------|-------|-------|
| min       | -0.5    | 0       | 1       | 10    | 0     |
| max       | 0       | 0.5     | 2       | 30    | 1     |

III. RESULTS AND DISCUSSION

A. E. COLI POPULATION GROWTH MODEL

In [16], an NDDE model of *E. coli* population growth is proposed. The model assumes that all cells have the same division time and that they divide simultaneously. Furthermore, it is assumed that there is a prolonged initial step-like growth. The model is given by

$$y'(t) = \rho_0 y(t) + \rho_1 y(t - \tau) + \rho_2 y'(t - \tau), \quad (5)$$

where $y(t)$ denotes the number of colonies at time $t$, and $\rho_0$, $\rho_1$, $\rho_2$, and $\tau$ are the model parameters. The parameter $\rho_0$ is the rate of cell death, $\rho_1$ is the rate of commitment to cell division, $\rho_2$ represents the gradual dispersal of synchronization, and $\tau$ is the average cell-division time [62]. The history function for this problem is given by

$$\phi(t) = y_0 + \beta \Psi(t), \quad t \in [-\tau, 0],$$

where

$$\Psi(t) = \frac{2y_0 \beta \gamma}{\rho_1 \tau} E \left( \frac{2t}{\tau} + 1 \right),$$

and

$$E(t) = \begin{cases} \exp[-1/(1 - t^2)], & \text{for } |t| < 1, \\ 0, & \text{otherwise.} \end{cases}$$

The function $\phi(t)$ follows a bell-shaped distribution that corresponds to non-monotone cell growth [16]. The value of $\gamma$ is set to 2.25 and the initial population $y_0$ to 99 [16].

We wish to estimate the parameters $\rho_0$, $\rho_1$, $\rho_2$, $\tau$, and $\beta$ by comparing the numerical solution to (5) with the number of colonies $\{y_i\}_{i=1}^{28}$ based on experimental data [16]. For brevity, we define

$$\theta := [\rho_0, \rho_1, \rho_2, \tau, \beta].$$
Thus, the minimization problem to estimate the parameters of the NDDE model of *E. coli* population growth is given by

\[
\min_{\theta \in \mathbb{R}^5} \sum_{i=1}^{28} (y^*_i - y_\theta(t_i))^2 \div \sum_{i=1}^{28} (y^*_i)^2.
\]

Here, \(y_\theta(t)\) is the solution to (5) given an arbitrary parameter vector \(\theta\). To compute for \(y_\theta(t)\) numerically, we use the MATLAB built-in function `ddensd` with step size 0.01. The maximum and minimum values of the parameters used in the simulations are shown in Table 1.

Figure 1 shows the box plots of the parameter estimates of the *E. coli* population growth model obtained using GA-MPC, GA, PS, SA, and PSO. Compared to the other algorithms, most of the estimates obtained using GA-MPC fall within the first and third quartiles of the 100 estimates. Furthermore, the narrow interquartile ranges suggest small variations among estimates. The cyan curves in Figure 2
TABLE 2. Standard error $\text{Err}(\hat{\theta})$, 95% confidence interval, and mean of the 100 realizations using GA-MPC for the NDDE model of $E. coli$ population growth.

| Parameter | $\text{Err}(\hat{\theta})$ | 95% Confidence Interval | Mean |
|-----------|---------------------|---------------------|------|
| $\rho_0$  | 0.004              | [-0.010, 0.006]     | -0.004|
| $\rho_1$  | 0.004              | [-0.001, 0.016]     | 0.010 |
| $\rho_2$  | 0.101              | [1.647, 2.031]      | 1.869 |
| $\tau$    | 1.623              | [17.666, 24.027]    | 20.617|
| $\beta$   | 0.011              | [0.024, 0.067]      | 0.044 |

FIGURE 3. The numerical solution $y(t)$ of the NDDE model of $E. coli$ population growth using the mean parameter estimates of GA-MPC is shown as the blue curve. The black asterisks are the experimental data points. The 95% confidence interval of the solutions is shown in gray and is bounded by the red curves.

are the plots of the solutions corresponding to the 100 sets of estimates. The data points are represented by the asterisks. Among the five algorithms, plots of the solution using GA-MPC produce the best fit to the experimental data.

Next, we apply the bootstrapping method presented in Subsection II-B to assess the uncertainty in the parameter estimates obtained using GA-MPC. Table 2 shows the standard errors and mean of the estimated parameters, which all fall within the 95% confidence interval. In Figure 3, we see that the plot of the solution using the mean of the estimated parameters (blue curve) fits the experimental data (asterisks) well. The bounds for the 95% confidence interval of the solution $y_3(t)$ are represented by the red curves.

TABLE 3. True values and bounds for the parameters in the age-structured NDDE model.

| Parameter | $\tau$ | $b_0$ | $b_1$ | $b_2$ | $\mu_0$ | $\mu_1$ | $\mu_2$ |
|-----------|-------|------|------|------|--------|--------|--------|
| value     | 36    | 0.0075 | 0.0133 | 0.045 | 0.0011 | 0.00056 | 0.005 |
| min       | 30    | 0     | 0     | 0     | 0      | 0       | 0       |
| max       | 40    | 0.5   | 0.5   | 0.5   | 0.5    | 0       | 0       |

$\hat{y}(t) = \begin{cases} n_0(\tau - t)\mu_0 \tau - V(t), & 0 \leq t \leq \tau, \\ a_1 V(t - \tau) + a_2 \hat{y}(t - \tau) - a_3 V(t), & t > \tau, \end{cases}$

where $U(t)$ is the juvenile population, $V(t)$ is the adult population,

$c_0 = b_0 - \mu_0,$
$c_1 = b_1,$
$c_2 = (b_2 - 1) b_0 e^{-\mu_0 \tau},$
$c_3 = (b_2 - 1) (b_1 e^{-\mu_2} + b_2 \mu_1) e^{\mu_2 - \mu_0 \tau},$
$c_4 = (b_2 - 1) b_2 e^{\mu_2 - \mu_0 \tau},$
$a_0 = b_0 e^{-\mu_2 - \mu_0 \tau},$
$a_1 = b_1 e^{-\mu_2} + b_2 \mu_1 e^{-\mu_0 \tau},$
$a_2 = b_2 e^{-\mu_0 \tau},$
$a_3 = \mu_1,$

are different birth and death rates, $b_j$, and $\mu_j$, $j = 0, 1, 2$, are non-negative and that $a_1 \geq a_2 a_3$.

We set the initial values for $U(t)$ and $V(t)$ as (compare with [15])

$\begin{bmatrix} U(0) \\ V(0) \end{bmatrix} = \begin{bmatrix} \int_{0}^{36} n_0(a)da \\ \int_{36}^{50} n_0(a)da \end{bmatrix} = \begin{bmatrix} \int_{0}^{36} 1da \\ \int_{36}^{50} 1da \end{bmatrix} = \begin{bmatrix} 36 \\ 14 \end{bmatrix},$

where the initial age distribution is given by

$n_0(a) = \begin{cases} 1, & 0 \leq a \leq 50, \\ 0, & \text{otherwise}. \end{cases}$

In this example, we use generated population data for $U(t)$ and $V(t)$ at $n$ different time points. To generate noisy data, we follow the method used in DDE models presented in [61]. First, clean data are generated by solving the model given the true parameters (see Table 3). The solution of the NDDE model is computed numerically using the MATLAB function $\text{ddensd}$. Then noisy data are generated following a normal distribution, with standard deviation set to 10% of the standard deviation of the NDDE solution. We generate data for $n = 20, 50$, and 100 time points over a fixed interval. The goal is to estimate the parameters of the age-structured NDDE model from the generated noisy data $\{U^*, V^*(t_i)\}, i = 1, 2, \ldots, n$.

The minimization problem for the parameter estimation of this model is given by

$\min_{\theta \in \mathbb{R}^7} \sum_{i=1}^{n} \left( \frac{|U^*(t_i) - U_\theta(t_i)|^2}{|U^*(t_i)|^2} + \frac{|V^*(t_i) - V_\theta(t_i)|^2}{|V^*(t_i)|^2} \right) + \xi g,$

where $U_\theta(t)$ and $V_\theta(t)$ are the plots of the solutions corresponding to the 100 sets of estimates. The data points are represented by the asterisks. Among the five algorithms, plots of the solution using GA-MPC produce the best fit to the experimental data.

Next, we apply the bootstrapping method presented in Subsection II-B to assess the uncertainty in the parameter estimates obtained using GA-MPC. Table 2 shows the standard errors and mean of the estimated parameters, which all fall within the 95% confidence interval. In Figure 3, we see that the plot of the solution using the mean of the estimated parameters (blue curve) fits the experimental data (asterisks) well. The bounds for the 95% confidence interval of the solution $y_3(t)$ are represented by the red curves.

B. AGE-STRUCTURED MODEL

Next, we consider a population divided into two age groups—juveniles and adults. This can be described by the following age-structured population growth model from [14]:

$\dot{U}(t) = \begin{cases} -\mu_0 U(t) + b_1 V(t), & 0 \leq t \leq \tau, \\ c_0 U(t) + c_1 V(t), & \tau < t, \end{cases}$

$\dot{V}(t) = \begin{cases} (b_2 - 1) n_0(\tau - t) e^{-\mu_0 \tau}, & 0 \leq t \leq \tau, \\ c_3 V(t - \tau) + c_4 \dot{V}(t - \tau), & t > \tau, \end{cases}$
FIGURE 4. Box plots of the estimates for $\tau$, $b_0$, $b_1$, and $b_2$ in the age-structured NDDE model using GA-MPC, GA, PS, SA, and PSO. The outliers, i.e., the estimates that fall outside the lower and upper quartiles, are plotted in red (+). The median and mean are represented by a red line across the box plots and black dot, respectively. The true values are shown as a black dotted line. The estimates are computed from a generated data set containing 20, 50, and 100 sample points.
FIGURE 5. Box plots of the estimates for $\mu_0$, $\mu_1$, and $\mu_2$ in the age-structured NDDE model using GA-MPC, GA, PS, SA, and PSO. The outliers, i.e., the estimates that fall outside the lower and upper quartiles, are plotted in red (+). The median and mean are represented by a red line across the box plots and black dot, respectively. The true values are shown as a black dotted line. The estimates are computed from a generated data set containing 20, 50, and 100 sample points.

\[
\theta = [\tau, b_0, b_1, b_2, \mu_0, \mu_1, \mu_2],
\]

$\zeta$ is a penalty parameter, $n$ is the number of data points, \{\(U_\theta(t_i), V_\theta(t_i)\}\} are the solutions to the model at time $t_i$ given an arbitrary parameter vector $\theta$, and

\[
g = -\min[a_1 - a_2 a_3, 0]
\]

\[
= -\min[(b_1 e^{-\mu_2} + b_2 \mu_1) e^{-\mu_0 \tau} - \mu_1 b_2 e^{-\mu_0 \tau}, 0]
\]

is a penalty function to ensure that the parameter values satisfy the condition $a_1 \geq a_2 a_3$. The parameter $\zeta$ is set to 1000. We solve the minimization problem using GA-MPC, GA, PS, SA, and PSO. The bounds for the parameters are given in Table 3.

We see in Figures 4 to 5 that in most cases, GA-MPC and PSO result in parameter estimates which are close to the true parameter values (black dotted line) and vary within small interquartile ranges. However, GA-MPC gives better...
estimates than PSO for the parameters $b_2$, $\mu_0$, $\mu_1$, and $\mu_2$. Using PSO, some estimates for the death rates $\mu_0$ and $\mu_1$ are zero, which fall outside the meaningful range for the death rate. Some parameter estimates using GA, PS, and SA vary on a larger range and a significant number of estimates fall outside the first and third quartiles. There is no notable difference in the parameter estimation results when $n = 20$, 50, or 100.

Figure 6 shows the plots of the solutions $U(t)$ and $V(t)$ (cyan and magenta curves, respectively) using the estimates from the 100 runs of GA-MPC. We see that the plots have a good fit to the $n$ data points (asterisks) and that the value of $n$ does not affect the performance of GA-MPC. Figures 12 to 14 in the Appendix show the plots of the solutions to the age-structured NDDE model using the estimates from GA, PS, SA, and PSO.

To analyze the uncertainty in the estimates, we apply the bootstrapping method to the data with $n = 50$. Results in Table 4 show small standard errors of all the parameters. Furthermore, the mean values of the estimates all fall within
FIGURE 8. Box plots of the parameter estimates in the PMSD NDDE model using GA-MPC, GA, PS, SA, and PSO. The outliers, i.e., the estimates that fall outside the lower and upper quartiles, are plotted in red (+). The median and mean are represented by a red line across the box plots and black dot, respectively. The true values are shown as a black dotted line. The estimates are computed from a generated data set containing 20, 50, and 100 sample points.
FIGURE 9. Plots of the solutions $y(t)$ (cyan curves) to the PMSD NDDE model using the estimates from the 100 runs of GA-MPC. The generated data sets with 20, 50, and 100 sample points are represented by asterisks.

FIGURE 10. The numerical solution $y(t)$ of the PMSD NDDE model using the estimated parameter values of GA-MPC is shown as the blue curve. The black asterisks are the 50 generated data points. The 95% confidence interval for the NDDE solutions is illustrated in gray and is bounded by the red curves.

the 95% confidence intervals. Figure 7 shows the good fit of the solutions to the data set using the re-estimates.

C. COUPLED PENDULUM-MASS-SPRING-DAMPER SYSTEM

Consider a mechanical system consisting of a mass $M$ mounted on a linear spring, where a pendulum of mass $m$ is attached to it via a hinged rod of length $l$ [9]. We assume that the angular deflection of the pendulum from the downward position is very small, and is thus negligible. Moreover, we assume that there is a linear, viscous damping $C$ on the mass $M$ and no external forces are acting on the system. The equation of motion is given by the second-order NDDE

$$M\ddot{x}(t) + C\dot{x}(t) + Kx(t) + m\ddot{x}(t - \tau) = 0,$$

where $C$ and $K$ are the damping and stiffness coefficients, respectively. This equation describes the vertical motion of a PMSD system, and the terms $x(t), \dot{x}(t)$ and $\ddot{x}(t)$ are the position, velocity, and acceleration of the PMSD at time $t$.

In the following discussion, we consider the nondimensionalized version of (7) given by

$$\ddot{y} + 2\zeta\dot{y} + y + p\dot{y}(t - \tau) = 0,$$

with the preshape function $\phi(t) = \cos(t/2)$ [9]. Note that if $|p| < 1$ and $\zeta > 1/\sqrt{2}$, then the steady state of (8) is locally asymptotically stable [9]. In particular, we are interested in the case when $p > 2\zeta\sqrt{1 - \zeta^2}$, so that there is a succession of stability switches as the value of $\tau$ increases.

We estimate the parameters of (8) from noisy data generated in the same way as in the age-structured NDDE model. First, clean data is obtained by numerically solving (8)
TABLE 5. True values and bounds for the parameters in the PMSD NDDE model.

| Parameter | τ     | ζ    | p    |
|-----------|-------|------|------|
| true value| 1     | 0.05 | 0.2  |
| min       | 0.5   | 0    | 0    |
| max       | 5     | 1    | 1    |

using the MATLAB function `ddensd`. We use the following parameter values from [9]: τ = 1, ζ = 0.05, and p = 0. Then noisy data y∗(ti), i = 1, 2, . . . , n, are generated following a normal distribution, with the standard deviation set to 10% of the standard deviation of the NDDE solution. We generate n = 20, 50, and 100 time points divided over a fixed interval. We minimize the objective function given by

\[ \min_{\theta \in \mathbb{R}^3} \frac{1}{n} \sum_{i=1}^{n} (y_i^* - y_\theta(t_i))^2, \]

where

\[ \theta := [\tau, \zeta, p], \]

yθ(t_i) is the solution to (8) at time t_i using an arbitrary parameter vector θ. The bounds for the parameters are shown in Table 5.

Figure 8 shows the zoomed-in box plots of the parameter estimates of the PMSD NDDE model. Aside from PS, the heuristic algorithms performed well in the parameter estimation. This might be attributed to having only 3 parameters being estimated. We note that there are many outliers in the estimates using PS with n = 50, which are not seen in Figure 8 but are seen in the plots of the solutions (see Figures 15 to 17 in the Appendix). The algorithms GA-MPC and GA both give good approximations to the true parameter values (black dotted line) in all 100 runs, but GA-MPC has fewer outliers. The plots of the solutions using parameter estimates of GA-MPC are shown in Figure 9. The number of data points used in the estimation does not appear to significantly affect the cost function value and the fit of the solutions to the data points.

Results of the parameter bootstrapping method show small standard errors of the parameter estimates in the PMSD NDDE model with mean estimates lying within the 95% confidence intervals (Table 6). Figure 10 shows a good fit to the data of most of the solutions using the re-estimated parameters.

### D. VARYING MAXIMUM NUMBER OF FUNCTION EVALUATIONS

The only stopping criterion used in the previous simulations is the maximum number of function evaluations and this is set to 2000 times the number of parameters to be estimated. Results show that GA-MPC consistently obtained good parameter estimates in the three models. Now, we look at how varying the maximum number of function evaluations affect the

![Figure 11](image-url)
FIGURE 12. Plots of the solutions $U(t)$ and $V(t)$ (cyan and magenta curves, respectively) to the age-structured NDDE model using the estimates from 100 runs of GA, PS, SA, and PSO. The generated data sets with 20 data points are represented by asterisks.

FIGURE 13. Plots of the solutions $U(t)$ and $V(t)$ (cyan and magenta curves, respectively) to the age-structured NDDE model using the estimates from 100 runs of GA, PS, SA, and PSO. The generated data set with 50 data points are represented by asterisks.

performance of GA-MPC, GA, PS, SA, and PSO in parameter estimation.

We consider setting the stopping criterion to 500, 1000, 1500, or 2000 times the number of parameters to be estimated, which we denote by $d$. Figure 11 shows the mean cost function value of 20 independent runs performed using GA-MPC, GA, PS, SA, and PSO. In Model 1, we see that GA-MPC produces the least cost function values across the different maximum numbers of function evaluations. In Models 2 and 3, we observe that GA-MPC and PSO give parameter estimates with the lowest cost function values, even when the maximum numbers of function evaluations are...
500d or 1000d. In all three models, GA-MPC consistently provides parameter estimates with low cost function values even for smaller numbers of maximum function evaluations. This is advantageous in instances when modeling using NDDEs requires parameter tuning at a faster time.

IV. CONCLUSION

In this work, we examine the use of a modified genetic algorithm called GA-MPC in solving parameter estimation problems involving NDDEs. We test our method on three different NDDE models – a first-order NDDE, a system of first-order NDDEs, and a second-order NDDE. We look at
FIGURE 16. Plots of the solutions $y(t)$ (cyan curves) to the PMSD NDDE model using the estimates from 100 runs of GA, PS, SA, and PSO. The generated data set with 50 sample points are represented by asterisks.

FIGURE 17. Plots of the solutions $y(t)$ (cyan curves) to the PMSD NDDE model using the estimates from 100 runs of GA, PS, SA, and PSO. The generated data set with 100 sample points are represented by asterisks.

the performance of GA-MPC in comparison to other heuristic algorithms and incorporate parameter uncertainty analysis to examine the extent of uncertainties in the estimates. For consistency of comparison, the default values of the parameters in the algorithms used are unchanged and the only stopping criterion is the maximum number of function evaluations. The simulations show that heuristic algorithms, particularly GA-MPC, can provide good estimates for the parameter values and a good fit to experimental data. The high accuracy and good consistency of the parameter estimates using GA-MPC are observed in all three different types of NDDE models we presented.

An advantage of using GA-MPC in solving the least-squares formulation arising from the parameter
estimation problem is that it does not depend on the explicit formulation or the derivatives of the solution to the NDDE. Although heuristic algorithms are computationally expensive, they are robust and are easy to implement.

Many factors contribute to the performance of heuristic algorithms in solving optimization problems. For future research, one can explore modifying the default parameter values or the stopping criteria in the algorithms. One can also consider other recent heuristic algorithms and hybrids of heuristic algorithms in solving inverse problems in NDDEs.

In this study, we have varied the number of data points and maximum number of function evaluations. For future work, one can use other metrics for comparison such as varying the noise level in the simulated data and adding different types of noise.

**APPENDIX A**

**ADDITIONAL FIGURES**

Figures 12 to 14 show the plots of the solutions to the noise. In this study, we have varied the number of data points and maximum number of function evaluations. For future work, one can use other metrics for comparison such as varying the noise level in the simulated data and adding different types of noise.

**REFERENCES**

[1] C. Jamilla, R. Mendoza, and I. Mezo, “Solutions of neutral delay differential equations using a generalized Lambert w function,” *Appl. Math. Comput.*, vol. 382, Oct. 2020, Art. no. 125334, doi: 10.1016/j.amc.2020.125334.

[2] H. Qin, Q. Zhang, and S. Wan, “The continuous Galerkin finite element methods for linear neutral delay differential equations,” *Appl. Math. Comput.*, vol. 346, pp. 76–85, Apr. 2019, doi: 10.1016/j.amc.2018.10.056.

[3] L. Berezansky and E. Braverman, “Solution estimates and stability tests for linear neutral differential equations,” *Appl. Math. Lett.*, vol. 108, Oct. 2020, Art. no. 106515, doi: 10.1016/j.aml.2020.106515.

[4] A. Alireza, “PSO with adaptive mutation and inertia weight and its applications in parameter estimation for chaotic systems using a hybrid adaptive cuckoo search with simulated annealing algorithm,” *Chaos, Interdiscip. J. Nonlinear Sci.*, vol. 24, no. 1, 2014, Art. no. 013133, doi: 10.1063/1.4867989.

[5] M. Faheem, A. Raza, and A. Khan, “Collocation methods based on Gegenbauer and Bernoulli wavelets for solving neutral delay differential equations,” *Math. Comput. Simul.*, vol. 180, pp. 72–92, Feb. 2021, doi: 10.1016/j.matcom.2020.08.018.

[6] H. Li, X. Liu, Z. Huang, C. Zeng, P. Zou, Z. Chu, and J. Yi, “Newly emerging nature-inspired optimization–algorithm review, unified framework, evaluation, and behavioural parameter optimization,” *IEEE Access*, vol. 8, pp. 72620–72649, Apr. 2020, doi: 10.1109/ACCESS.2020.2987689.

[7] S.-Y. Yang, *Nature-Inspired Optimization Algorithms*, 1st ed. Amsterdam, The Netherlands: Elsevier, 2014.

[8] R.L. Haupt and S.E. Haupt, *Practical Genetic Algorithms*, 2nd ed. Hoboken, NJ, USA: Wiley, 2004.

[9] A. Alireza, “PSO with adaptive mutation and inertia weight and its applications in parameter estimation of dynamic systems,” *Acta Automatica Sinica*, vol. 37, no. 5, pp. 541–549, 2011, doi: 10.1016/S1874-1029(11)60205-X.

[10] Z. Sheng, J. Wang, S. Zhou, and B. Zhou, “Parameter estimation for chaotic systems using a hybrid adaptive cuckoo search with simulated annealing algorithm,” *Chaos, Interdiscip. J. Nonlinear Sci.*, vol. 24, no. 1, 2014, Art. no. 013133, doi: 10.1063/1.4867989.

[11] W.-H. Ho and A. L.-F. Chan, “Hybrid taguchi-differential evolution algorithm for parameter estimation of differential equation models with application to HIV dynamics,” *Math. Problems Eng.*, vol. 2011, pp. 1–14, 2011, doi: 10.1155/2011/514756.

[12] J. H. Lin, C. W. Chou, C. H. Yang, and H.-L. Tsai, “A chaotic Levy flight algorithm for parameter estimation in nonlinear dynamic biological systems,” *Comput. Inf. Technol.*, vol. 2, no. 2, pp. 56–63, 2012.

[13] J. P. Gentile, R. Mendoza, and V. M. P. Mendoza, “Parameter estimation of an S-system model using hybrid genetic algorithm with the aid of sensitivity analysis,” in *Proc. North American Fuzzy Information Processing Society Annual Conf.*, 2009, pp. 94–102.

[14] R. C. Cajavay, J. A. Lucilo, C. P. Pilar-Arceo, and E. R. Mendoza, “Comparison of two nature-inspired algorithms for parameter estimation of system models,” *Philipp. J. Sci.*, vol. 149, no. 1, pp. 63–78, 2020.

[15] L. Lu, Z. Guo, W. Zhang, Z. Huang, and M. Liu, “Parameter estimation for a capacitive coupling communication channel within a metal cabinet based on a modified artificial immune algorithm,” *IEEE Access*, vol. 9, pp. 75683–75698, 2021, doi: 10.1109/ACCESS.2021.3082629.

[16] A. G. A. Muthalif, M. K. M. Razali, N. H. D. Nordin, and S. B. A. Hamid, “Parameter estimation from empirical data using particle swarm optimization method for different magnetoreheological damper models,” *IEEE Access*, vol. 9, pp. 77602–77613, 2021, doi: 10.1109/access.2021.3080432.
[35] D. Matajira-Rueda, J. M. Cruz-Duarte, J. G. Avina-Cervantes, M. A. Ibarra-Manzano, and R. Correa, “Optimization-based strategies for optimal inverse parameters estimation for heat transfer systems,” IEEE Access, vol. 9, pp. 71328–71343, 2021, doi: 10.1109/access.2021.3075370.

[36] A. Ramadan, S. Kamel, M. M. Hussein, and M. H. Hassan, “A new application of chaos game optimization algorithm for parameters extraction of three diode photovoltaic model,” IEEE Access, vol. 9, pp. 51582–51594, 2021, doi: 10.1109/access.2021.3069939.

[37] M. A. Soliman, A. Al-Durra, and H. M. Hasanien, “Electrical parameters identification of three-diode photovoltaic model based on equilibrium optimization,” IEEE Access, vol. 9, pp. 41891–41901, 2021, doi: 10.1109/access.2021.3065386.

[38] O. Rodriguez-Albero, J. M. Hernandez-Paredes, A. F. Rangel, C. Fuentes-Silva, and F. A. C. Velasquez, “Parameter identification of motors by cuckoo search using steady-state relations,” IEEE Access, vol. 9, pp. 72017–72024, 2021, doi: 10.1109/access.2021.3078578.

[39] D. S. Abdelalimnaa, M. Said, and E. H. Houssein, “Turbulent flow of water-based optimization using new objective function for parameter extraction of six photovoltaic models,” IEEE Access, vol. 9, pp. 35382–35398, 2021, doi: 10.1109/access.2021.3061529.

[40] A. A. Z. Diab, H. M. Sultan, R. Aljendy, A. S. Al-Sumaiti, M. Shoyama, and Z. M. Ali, “Tree growth based optimization algorithm for parameter extraction of different models of photovoltaic cells and modules,” IEEE Access, vol. 8, pp. 119668–119687, 2020, doi: 10.1109/access.2020.3005236.

[41] Z. Liao, Q. Gu, S. Li, Z. Hu, and B. Ning, “An improved differential evolution to extract photovoltaic cell parameters,” IEEE Access, vol. 8, pp. 177838–177850, 2020, doi: 10.1109/access.2020.3024975.

[42] K. S. Tang, K. F. Man, S. Kwong, and Q. He, “Genetic algorithms and their applications,” IEEE Signal Process. Mag., vol. 13, no. 6, pp. 22–37, Nov. 1996, doi: 10.1109/79.543973.

[43] E. Balsa-Canto, M. Peifer, J. R. Banga, J. Timmer, and C. Fleck, “Hybrid optimization method with general switching strategy for parameter estimation,” BMC Syst. Biol., vol. 2, no. 1, p. 26, Mar. 2008, doi: 10.1186/1752-0509-2-26.

[44] C. Zhan, W. Situ, L. F. Yeung, P. W.-M. Tsang, and G. Yang, “A parameter estimation method for biological systems modelled by ODE/DDDE models using spline approximation and differential evolution algorithm,” IEEE/ACM Trans. Comput. Biol. Bioinf., vol. 11, no. 6, pp. 1066–1076, Nov. 2014, doi: 10.1145/2589057.2589068.

[45] M. Yoldiz and H. Kanze, “Inverse problems for delay differential equations using the collage theorem,” J. Math. Anal. Appl., vol. 436, no. 2, pp. 1284–1296, Apr. 2016, doi: 10.1016/j.jmaa.2015.11.070.

[46] J. Lin, “Parameter estimation for time-delay chaotic systems by hybrid biogeography-based optimization,” Nonlinear Dyn., vol. 77, no. 3, pp. 983–992, 2014, doi: 10.1007/s11071-014-1356-7.

[47] Y. Tang and X. Guan, “Parameter estimation for time-delay chaotic system by particle swarm optimization,” Chaos, Solitons Fractals, vol. 40, no. 3, pp. 1391–1398, 2009, doi: 10.1016/j.chaos.2007.09.055.

[48] J. Wei and Y. Yu, “An effective hybrid cuckoo search algorithm for unknown parameters and time delays estimation of chaotic systems,” IEEE Access, vol. 6, pp. 6560–6571, 2017, doi: 10.1109/access.2017.2738006.

[49] S. M. Elsayed, A. R. Sarkar, and D. L. Essam, “GA with a new multi-parent crossover for solving IEEE-CEC2011 competition problems,” in Proc. IEEE Congr. Evol. Comput. (CEC), Jun. 2011, pp. 1034–1040, doi: 10.1109/CEC.2011.5949731.

[50] H. Pulluri, M. Vyshnavi, P. Shradhha, B. S. Priya, and T. S. Hari, “Genetic algorithm with multi-parent crossover solution for economic dispatch with valve point loading effects,” in Innovations in Electrical and Electronics Engineering (Lecture Notes in Electrical Engineering), Singapore: Springer, 2020, pp. 429–438, doi: 10.1007/978-981-15-2256-7_41.

[51] T. Srijari, M. Boppa, S. Anil Kumar, and H. Pulluri, “The application of genetic algorithm with multi-parent crossover to optimal power flow problem,” in Innovations in Electrical and Electronics Engineering (Lecture Notes in Electrical Engineering), Singapore: Springer, 2020, pp. 417–427, doi: 10.1007/978-981-15-2256-7_40.

[52] A. L. Caro, R. Mendoza, and V. M. P. Mendoza, “Application of genetic algorithm with multi-parent crossover on an inverse problem in delay differential equations,” in Proc. AIP Conf., 2021, pp. 1–7.

[53] J. Holland, Adaptation in Natural and Artificial Systems, 1st ed. Ann Arbor, MI, USA: Univ. Michigan Press, 1975.

[54] J. McCall, “Genetic algorithms for modelling and optimisation,” J. Comput. Appl. Math., vol. 184, no. 1, pp. 205–222, Dec. 2005, doi: 10.1016/j.cam.2004.07.034.