Parameter estimation in S-System Models using Firefly Algorithm with Decoupling Method

Raquel C. Cajayon*[1], Carlene P.C. Pilar-Arceo[2], Eduardo R. Mendoza[3]

[1]Division of Physical Sciences and Mathematics, College of Arts and Sciences, University of the Philippines Visayas, Miagao, Iloilo (5023)
[2] Institute of Mathematics, University of the Philippines Diliman, Diliman Quezon City (1101)
[3] Physics Department, Ludwig Maximilians University, Munich, Germany
Max Planck Institute of Biochemistry, Munich, Germany

ABSTRACT

An essential element for describing and predicting the dynamic behavior of interacting components within a biochemical system is the formulation of mathematical models. One of the widely used mathematical models is the power-law formalism-based S-systems. However, parameter estimation in S-systems continues to be a difficult challenge because of the nonlinearity and high dimensionality of the underlying coupled systems of ordinary differential equations. Hence it is important to provide efficient and effective novel methods to tackle parameter estimation problem. In this paper, we use decoupling technique on the S-system and then Xin-She Yang's Firefly Algorithm (FA), a metaheuristic algorithm based on the bioluminescence process which characterizes fireflies, as optimization algorithm in the parameter estimation. FA's automatic subdivision of the whole population into subgroups and natural capability of dealing with multi-modal optimization are the two major advantages of FA over other algorithms, making FA a good algorithm choice. Using three S-systems of increasing complexity from the MADMan Benchmarking Framework, we assess the performance of the method. Simulation results show that FA-decoupling method is applicable in estimating parameters of the 3 S-system models using noise-free data with concentration error lower than 10^{-3} and better approximations were recorded for kinetic orders than rate constants. Since FA-decoupling method worked on parameter estimation of S-system models using noise-free data, it is now reasonable to implement the method using datasets with different noise levels to check how the method is affected by the presence of noise.

INTRODUCTION

One of the modeling techniques for biochemical systems is Biochemical Systems Theory (BST). BST was introduced by M. Savageau in 1969 and it provides a good framework for constructing mathematical models of biological systems (Voit 2013). One of the canonical formats used in BST is the S-system format.

S-systems are characterized by a set of rate constants and kinetic orders, which are the control parameters of the model. Some of these parameters usually cannot be experimentally determined. This leads to the need to estimate these parameters using computational methods.

A common practice to estimate S-systems' parameters is to transform the estimation problem into an optimization problem. In the optimization version, the cost function is the difference between the model-based prediction and the experimental data, and we find the combination of model parameters that will minimize this value. We can deal with two types of cost function: one based on the concentration error and the other on the slope error.

The concentration error-based cost function is a straightforward representation that measures the model's prediction consistency with the data and is therefore widely used. However, one big disadvantage in using the concentration error-based cost function is it is too computationally expensive. Each time the optimization algorithm tries a new parameter value, the system of ODEs in the S-system model must be solved and numerical integration takes up to more than 95% of the search time (Voit et. al. 2004).

However, the use of differential equations may be avoided by replacing the derivatives with slopes of the response curve. Using the slope error-based cost function counters the problem of the time-consuming

*Corresponding Author:
Division of Physical Sciences and Mathematics, University of the Philippines Visayas, Miagao, Iloilo (5023); email address: rccajayon@up.edu.ph; Tel. No. (033) 513-8785; loc. 239

numerical integration process, and thus shows superiority in speed of optimization.

There are recent methods that have been proposed to estimate parameters in S-system and General Mass Action (GMA) models that adopt slope error-based cost function. One of them is the method developed by Voit and Almeida based on artificial neural networks to decouple the differential equations in an ODE system (Almeida et. al. 2003; Voit et. al. 2004). This paper makes use of the same decoupling process in computing the cost functions.

Increasing interest in the BST models has led to the development of parameter estimation methods proposed for such models. An excellent systematic review is provided in (Chou et. al. 2009). Among the novel methods discussed are the metaheuristic methods.

Some of the nature-inspired metaheuristic algorithms that have already been developed for BST models are the Genetic Algorithm (Kikuchi et. al. 2003), Simulated Annealing (Echavez unpubl. data; Gonzalez et. al. 2007), Particle Swarm Optimization (Naval et. al. 2010), and Ant Colony Optimization (Zuniga 2011).

In 2008, Yang introduced the Firefly Algorithm (FA), another nature-inspired metaheuristic algorithm (Yang 2008). FA literature has expanded in the last 11 years with several hundred papers claiming its efficiency in the field of optimization (Fister et. al. 2013; Yang et. al. 2018). Since a number of metaheuristic algorithms have already been used to do parameter estimation, the authors want to see how the nature-inspired algorithm FA will be applicable in the problem. It is important in the computing community to learn how well an algorithm work, or even the circumstances where it does not work well.

This paper describes FA with the decoupling technique as a new method for parameter estimation in S-systems using noise-free data. Since there is no theoretical mathematical guarantee that FA will always work, we will test the applicability of FA on three different S-system models. The models are: a generic branched pathway (VA04), a simple genetic network (HS96) and fermentation in *S. cerevisiae* (CC06). These models are part of the benchmarking framework described in the MADMan User's Guide (MUG), a technical manual for parameter estimation in BST models, developed by researchers at LMU in Munich, Georgia Tech in Atlanta, UP Diliman and UP Manila (del Rosario et. al. 2008).

MATERIALS AND METHODS

1. The S-system Model

An S-system has the form (1),

$$\dot{X}_i = \alpha_i \prod_{j=1}^{n} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n} X_j^{h_{ij}} \quad \text{for } i = 1,2,\dots,n \tag{1}$$

where X_i denotes the concentration of metabolite i and X_i is the corresponding rate of change. Coefficients α_i and β_i are nonnegative rate constants for the production and degradation terms, respectively. The kinetic orders g_{ij} and h_{ij} quantify the regulation effect of X_j on the activation or inhibition of X_i.

The variables α_i , β_i , g_{ij} and h_{ij} are called parameters. These are the values that need to be estimated. In the biochemical engineering context, the rate constants are nonnegative, and the kinetic orders are real-valued exponents.

The rate constants describe the turnover rate of the process while the kinetic orders determine the structure of the biochemical network. In the case where, $g_{ij} > 0$, X_j induces the synthesis of X_i. If $g_{ij} < 0$, X_j , inhibits the synthesis of X_i. Similarly, a positive (negative) value of h_{ij} indicates that X_j induces (suppresses) the degradation of X_i. Variables (X_j) that do not directly affect X_i have kinetic orders of 0.

The rate constants and the kinetic orders are the parameters of the model. The S-system model allows for the inclusion of independent variables, but since the values of these variables are known, they can be merged with the rate constants. An overview of the numerous examples of S-system models of biological systems that have been constructed is given in (Voit 2000) and (Voit 2013).

Note that the signs of the kinetic orders are not explicitly indicated in the S-system model. However, when a map of the biochemical system is available, we will know which of the kinetic orders are positive and which are negative.

Restricting the differential equations into S-system format may seem incapable of capturing all the complicated behaviors in a biochemical system, but fortunately it has been shown that any nonlinear differential equation composed of elementary functions could be recast exactly as an S-system

(Savageau et al. 1987).

The 3 S-system models considered in this paper from MUG were chosen according to their complexity (number or dependent and independent variables, number of parameters, etc.).

The simplest networks we considered are VA04 and HS96. VA04 is a generic branched pathway and HS96 is a synthetic model of a small gene regulatory system which contains most essential features of a real pathway. The third network considered is a moderately complex network CC06.

The Generic Branched Pathway (VA04)

VA04 is a generic branched pathway with four dependent variables and one independent variable, a map of which is shown in Figure 1. The S-system model of the pathway is described in (2), with a total of 18 rate constants and kinetic orders to estimate. The independent variable X_0 however has constant value 0.9 and g_{10} has true value 1.0 and can be merged with rate constant α_1, hence we are left with 17 parameters to estimate for VA04.

$$\begin{aligned}\dot{X}_1 &= \alpha_1 X_3^{g_{13}} X_0^{g_{10}} - \beta_1 X_1^{h_{11}} \\ \dot{X}_2 &= \alpha_2 X_1^{g_{21}} - \beta_2 X_2^{h_{22}} \\ \dot{X}_3 &= \alpha_3 X_2^{g_{32}} - \beta_3 X_3^{h_{33}} X_4^{h_{34}} \\ \dot{X}_4 &= \alpha_4 X_1^{g_{41}} - \beta_4 X_4^{h_{44}}\end{aligned} \tag{2}$$

The Simple Gene Regulatory Network (HS96)

HS96 is a synthetic gene regulation network given in Figure 2, by Hlavacek and Savageau. The S-system model of the network is described in (3) having 5 independent variables. It has 23 unknown parameters.

$$\begin{aligned}\dot{X}_1 &= \alpha_1 X_3^{g_{13}} X_5^{g_{15}} - \beta_1 X_1^{h_{11}} \\ \dot{X}_2 &= \alpha_2 X_1^{g_{21}} - \beta_2 X_2^{h_{22}} \\ \dot{X}_3 &= \alpha_3 X_2^{g_{32}} - \beta_3 X_2^{h_{32}} X_3^{h_{33}} \\ \dot{X}_4 &= \alpha_4 X_3^{g_{43}} X_5^{g_{45}} - \beta_4 X_4^{h_{44}} \\ \dot{X}_5 &= \alpha_5 X_4^{g_{54}} - \beta_5 X_5^{h_{55}}\end{aligned} \tag{3}$$

The Yeast Fermentation Network (CC06)

The CC06 network, which is a model of ethanol production in yeast (*Saccharomyces cerevisiae*) with 5 dependent variables and 8 independent variables. The map of the yeast fermentation pathway is shown in Figure 3. The S-system model of this network is described in (4). The model has a total of 49 parameters to estimate.

$$\begin{aligned}\dot{X}_1 &= \alpha_1 X_2^{g_{12}} X_6^{g_{16}} - \beta_1 X_1^{h_{11}} X_5^{h_{15}} X_7^{h_{17}} \\ \dot{X}_2 &= \alpha_2 X_1^{g_{21}} X_5^{g_{25}} X_7^{g_{27}} - \beta_2 X_2^{h_{22}} X_5^{h_{25}} X_8^{h_{28}} X_{11}^{h_{2,11}} \\ \dot{X}_3 &= \alpha_3 X_2^{g_{32}} X_5^{g_{35}} X_8^{g_{38}} - \beta_3 X_3^{h_{33}} X_4^{h_{34}} X_5^{h_{35}} X_9^{h_{39}} X_{12}^{h_{3,12}} \\ \dot{X}_4 &= \alpha_4 X_3^{g_{43}} X_5^{g_{45}} X_9^{g_{49}} - \beta_4 X_3^{h_{43}} X_4^{h_{44}} X_5^{h_{45}} X_{10}^{h_{4,10}} \\ \dot{X}_5 &= \alpha_5 X_3^{g_{53}} X_4^{g_{54}} X_5^{g_{55}} X_9^{g_{59}} X_{10}^{g_{5,10}} - \beta_5 X_1^{h_{51}} X_2^{h_{52}} X_5^{h_{55}} X_7^{h_{57}} X_8^{h_{58}} X_{11}^{h_{5,11}} X_{13}^{h_{5,13}}\end{aligned} \tag{4}$$

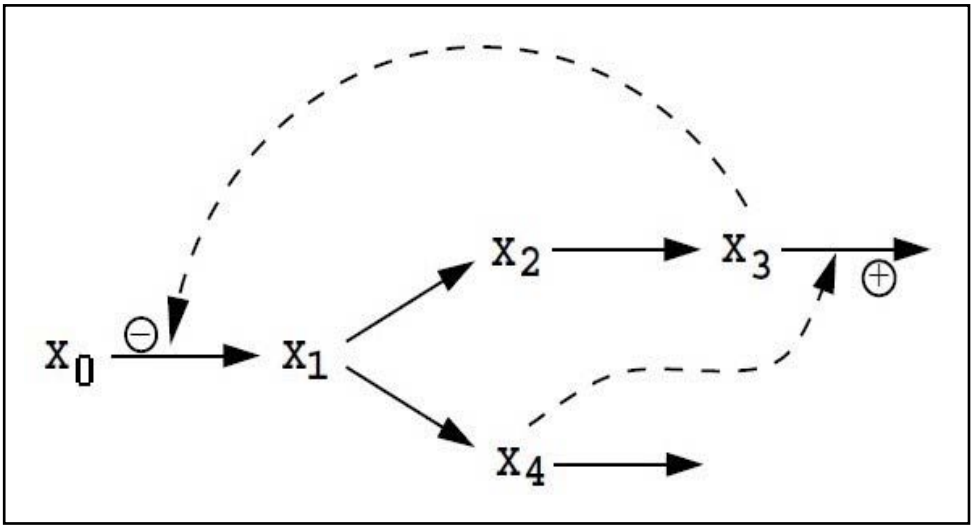

Figure 1. The Generic Branched Pathway (VA04)

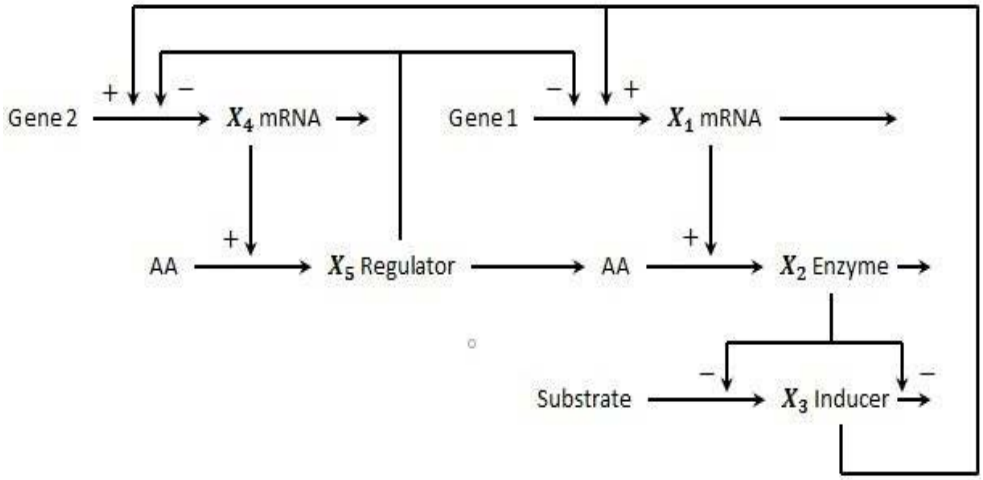

Figure 2. The Simple Gene Regulation Network (HS96)

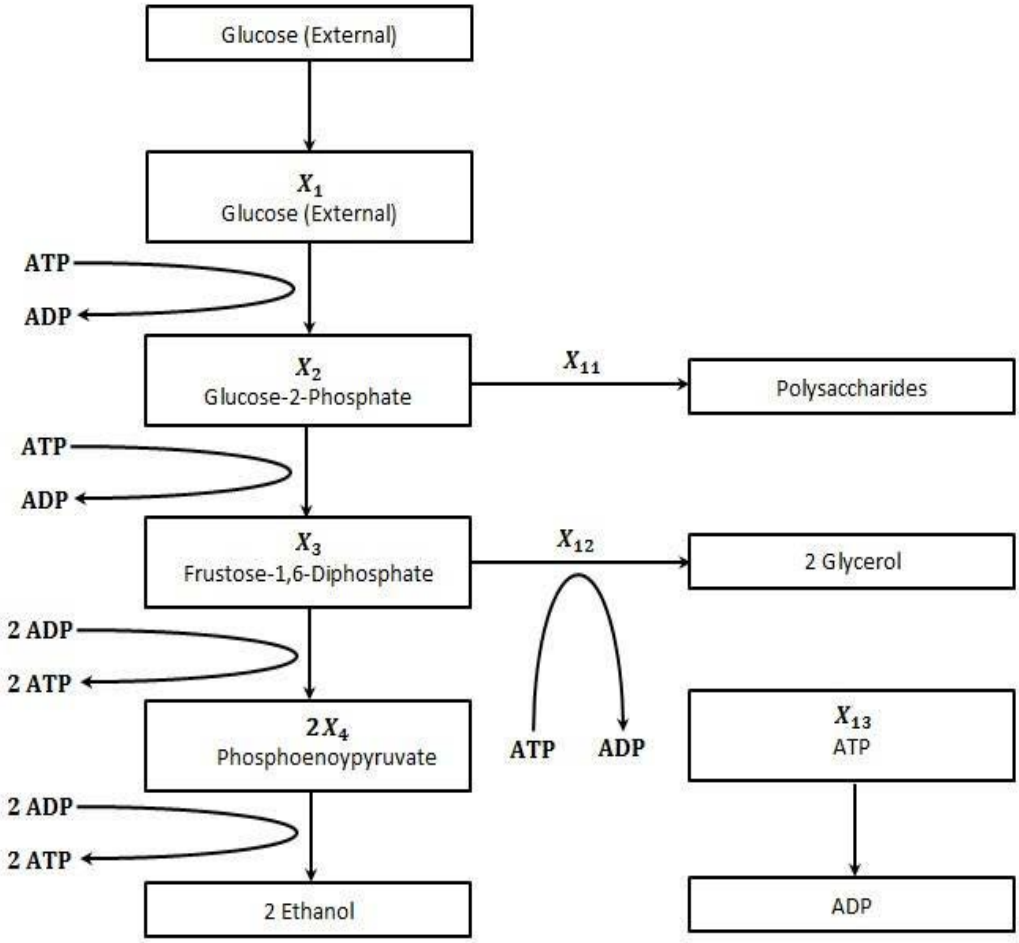

Figure 3. The Yeast Fermentation Network (CC06)

2. Parameter Estimation and the Cost Function

A flowchart for parameter estimation is provided in Figure 4. The steps were mainly suggested by MADMan User Guide (del Rosario et al. 2008). The parameter estimation begins with a prediction from the model and reference data obtained from experiments. In this study, the predictions are generated from an ODE solver using the parameter guesses, while reference data are generated using the true S-system parameters. The difference between the predicted and the expected data is computed in an iteration. The iteration is repeated until an optimal parameter set is found by minimizing the difference. The optimal set of S-system parameters is then recorded.

For post-optimization, the error of the computed parameters relative to the true parameters is computed. The final cost function value and the running time for this process are recorded. Lastly, the prediction of the model using the computed parameters are plotted against the reference data sets to see the goodness of fit.

One way to set up the cost function for parameter estimation is by (5)

$$C_1 = \frac{1}{nN}\sum_{i=1}^{n}\sum_{k=1}^{N}\left(X_{e_i}(t_k) - X_i(t_k)\right)^2, \qquad (5)$$

where $X_{e_i}(t_k)$ is the measured data for the i^{th} component at sampling time t_k, $X_i(t_k)$ denotes the computed concentration for the i^{th} component at sampling time t_k, n is the number of equations (also the number of dependent variables) and N is the number of time points.

Obtaining the profiles $X_i(t_k)$ by numerically integrating the S-system differential equations in (1) using the parameter guesses is a computationally expensive process. Thus, researches have geared towards finding parameters that fit the right hand of the ODEs in (1) with the differentials replaced by estimated slopes. This technique is called *decoupling*.

3. The Decoupling Technique

In decoupling, consider function

$$\dot{X}_i = f(X,p), \quad X(t_0) = X_0, \qquad (6)$$

where

$$f(X,p) = \begin{bmatrix} f_1(X,p) = \alpha_1 \prod_{j=1}^{n+m} X_j^{g_{1j}} - \beta_1 \prod_{j=1}^{n+m} X_j^{h_{1j}} \\ \vdots \\ f_i(X,p) = \alpha_i \prod_{j=1}^{n+m} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{n+m} X_j^{h_{ij}} \\ \vdots \\ f_n(X,p) = \alpha_n \prod_{j=1}^{n+m} X_j^{g_{nj}} - \beta_n \prod_{j=1}^{n+m} X_j^{h_{nj}} \end{bmatrix} \qquad (7)$$

and X_0 represents the initial condition(s) at the starting time t_0. The instantaneous rate of change of X_i at any given time t_k is equal to $f(X,p)$ at t_k, i.e., $\dot{X}_i(t_k) = f(X(t_k), p)$ and the time courses of $X(t)$ can be computed. If we can measure the slope $S_i(t_k)$ for each variable X_i at each time point $t_1, t_2, \ldots, t_k, \ldots, t_N$, then the new formulation of (6) is:

$$\begin{aligned} S_1(t_1) &\approx \dot{X}_1(t_1) = \alpha_1 \prod_{j=1}^{n+m} X_j^{g_{1j}}(t_1) - \beta_1 \prod_{j=1}^{n+m} X_j^{h_{1j}}(t_1) \\ S_1(t_2) &\approx \dot{X}_1(t_2) = \alpha_1 \prod_{j=1}^{n+m} X_j^{g_{1j}}(t_2) - \beta_1 \prod_{j=1}^{n+m} X_j^{h_{1j}}(t_2) \\ &\vdots \\ S_1(t_N) &\approx \dot{X}_1(t_N) = \alpha_1 \prod_{j=1}^{n+m} X_j^{g_{1j}}(t_N) - \beta_1 \prod_{j=1}^{n+m} X_j^{h_{1j}}(t_N) \\ &\vdots \\ S_i(t_k) &\approx \dot{X}_i(t_k) = \alpha_i \prod_{j=1}^{n+m} X_j^{g_{ij}}(t_k) - \beta_i \prod_{j=1}^{n+m} X_j^{h_{ij}}(t_k) \\ &\vdots \\ S_n(t_N) &\approx \dot{X}_n(t_N) = \alpha_n \prod_{j=1}^{n+m} X_j^{g_{nj}}(t_N) - \beta_n \prod_{j=1}^{n+m} X_j^{h_{nj}}(t_N). \end{aligned} \qquad (8)$$

Thus, assuming that time-series concentration data of all metabolites $X_i(t_k)$ are available, the slopes $S_i(t_k)$ can be calculated and the estimation process simplifies to solving a set of n x N (nonlinear) algebraic equations, where N is the number of time points and n is the number equations (equal to the number of dependent variables). Each equation now will then be treated as an equation independent from the other equations, and parameter estimation can now be done per equation, and not for the entire system.

We can set up a cost function for equation i for $i = 1,2,\ldots, n$:

$$C_{2_i} = \frac{1}{N}\sum_{k=1}^{N}\left[Xdot_{e_i}(t_k) - S_i(t_k)\right]^2, \qquad (9)$$

where $X\mathrm{dot}_{e_i}(t_k)$ is the approximated experimental

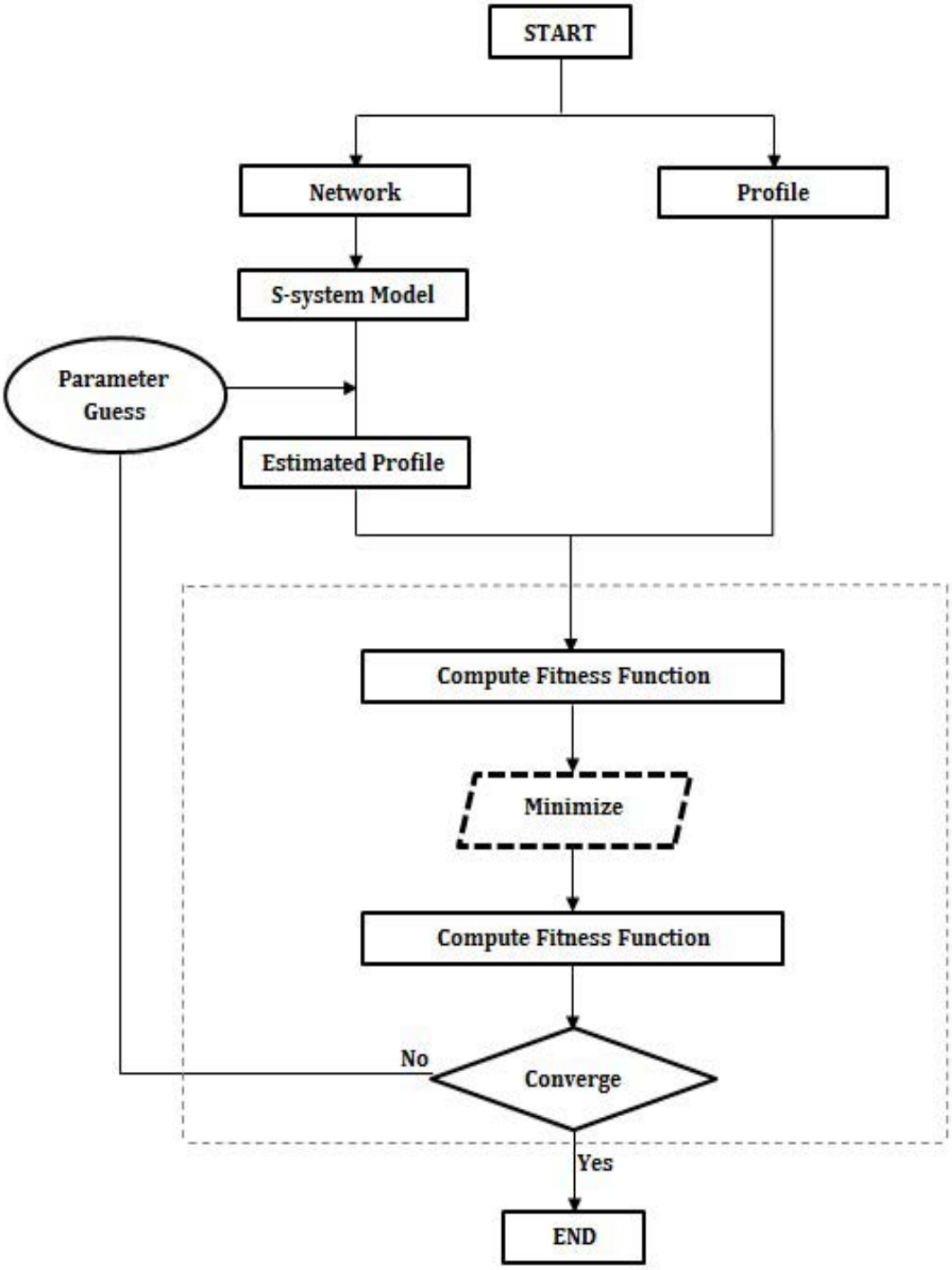

Figure 4. The Parameter Estimation Flowchart

slope for the i^{th} component at sampling time t_k, $S_i(t_k)$ denotes the computed slope for the i^{th} component at sampling time t_k, and N is the number of time points.

Note that since there is no integration of the ODEs, the minimization of the difference between slopes and $f(X, p)$ is computationally efficient, even for a large number of parameters. Decoupling eliminates issues of stiffness and speeds up computational time.

In decoupling however, we must always keep in mind that we are looking for a set of parameters which will allow the slope or derivative of the numerical solution to fit *the derivative of the data*. Hence, there is additional search bias since it is well established in mathematics that "If $f'' = g'$, then $f = g$." does not always hold.

4. The Firefly Algorithm as Optimization Method

The Firefly Algorithm (FA) is a metaheuristic algorithm proposed by Xin-She Yang (2008) based on the flashing patterns and behavior of fireflies.

Yang (2008) idealized some of the flashing characteristics of fireflies in order to formulate the FA, as follows:

> *(1) All fireflies are unisex, so one firefly is attracted to all other fireflies regardless of their sex; (2) The attractiveness is proportional to their light intensity, so, for any two fireflies, the less bright one will move towards the brighter one; (3) Attractiveness and brightness both decrease as distance increases; and (4) The light intensity of a firefly is affected or determined by the landscape of the objective function.*

The basic rule in Yang's FA is that a population of n fireflies (possible solutions) of dimension d, scattered randomly through a search space, is evaluated based on their light intensity. We compare the light intensities and move a firefly accordingly towards the more attractive firefly(ies).

In FA, attractiveness of a firefly is determined by its brightness, which is associated with the objective function (Yang 2010). The objective function used in FA is the function that we want to optimize, say $f(x)$. In minimization, firefly j is brighter than firefly i if $f(j) < f(i)$. Due to exponential decay of light absorption and inverse-square law of light variation with distance, the movement of a firefly i if attracted to another more attractive firefly j is determined by

$$x_i^{t+1} = x_i^t + \beta_0 e^{-\gamma r_{ij}^2}\left(x_j^t - x_i^t\right) + \alpha_t \epsilon_i^t, \quad (10)$$

where r_{ij} is the Euclidean distance between any two fireflies i and j at x_i and x_j, respectively and β_0 is the attractiveness at $r_{ij} = 0$. The second term is due to the attraction, the third term is randomization, α_t being the randomization parameter, and ϵ_i^t is a vector of random numbers drawn from a Gaussian distribution at time *t*.

The parameters γ and β_0 in the second term of equation (10) are important scaling parameters (Yang 2008; Yang 2010). Differential evolution (DE), accelerated particle swarm optimization (APSO), simulated annealing (SA) and the harmony search (HS) algorithm are special cases of the standard FA (Yang et. al. 2018). FA can now be considered as combined APSO, HS, SA and DE enhanced in a nonlinear system. Hence, with optimal FA parameter setting, FA can outperform these algorithms in many applications (Yang et. al. 2018).

The main steps of FA are summarized in the pseudo-code shown in Figure 5.

Simulation Environment

The FA code was implemented in using a desktop computer running on **Intel Core**™ i5 CPU 650 @320GHz with 1.86GB of RAM. The ordinary differential equation solver **ode** of **type = "stiff"** was used whenever we need to solve for the solution of the S-system model. For the S-system Model, initial conditions, initial time and the times at which the solution is computed will be provided for each S-system from the MADMan User's Guide (del Rosario et. al. 2008).

Initial guesses for S-system parameters to be estimated were set to 2.0 for production rate constant α, 1.0 for degradation rate constant β, and -0.2 and 0.5 for the inhibition and activation kinetic orders, respectively. Initial values for kinetic orders were taken from the range given by MUG (del Rosario et. al. 2008) and Voit et. al. (2004). Furthermore, we constrained the rate constants to be within [0,20] and the kinetic orders for inhibition to within [-2,0] and activation to within [0,2].

Aside from the maximum number of iterations as the basic FA's stopping criterion, we also allowed termination of the optimization process when the objective function value became less than our preset

```
Objective function f(x),  x = (x_1, ..., x_d)^T
Define Light Absorption Coefficient γ, number of fireflies n,
maximum number of iterations MaxGen, attractiveness at
zero distance β_0, and randomization parameter α.
for (i = 1 : n) do
    Generate initial solution x_i, (i = 1, ..., n).
    Define light intensity I_i for each x_i, determined by f(x_i)
end for
while (t < MaxGen) do
    for (i = 1 : n ) do
        for (j = 1 : n ) do
            if (I_i < I_j) then
                Move firefly i towards firefly j
            end if
        end for
    end for
    Vary attractiveness with distance r via e^(-γr)
    Evaluate new set of solution.
    Update the light intensity I_i for each x_i
    Rank the fireflies based on their light intensity and find
the current best x_i
    Define t = t + 1
end while
Post process results and visualization
```

Figure 5. Pseudocode for Firefly Algorithm

relative tolerance of 10^{-5}.

Also, the optimal setting of parameters in FA largely depends on the problem, and there is no unique best setting for all problems. Hence, we find the combination of algorithm parameters that will give the best performance of FA in the estimation of S-system parameters. We need to take note however that algorithm-dependent parameter tuning is also a complicated optimization problem and is not the focus of this paper. For simplicity, we just considered specific sets of algorithm-dependent parameter values, and from those sets, the ones that give the best result were chosen for final implementation.

The algorithm-dependent parameters in the FA are the number of fireflies (n), attractiveness at zero distance (β_0), randomization (α), and the light absorption coefficient (γ) .

Earlier studies for FA revealed that β_0 = 1.0, $n \in [25, 45]$ and $\gamma = 1.0$, work for most applications (Yang 2013, Yang 2014, Mo et al. 2013). The randomization parameter $\alpha \in [0, 1]$, however, is a sensitive parameter and needs more tuning (Yang 2013, Yang 2014).

We used for FA parameters $n = 30$, β_0 = 1.0, and γ = 1.0 from the results of earlier studies for FA (Yang 2013, Yang 2014, Mo et al. 2013). We only tuned the sensitive parameter, α , for each S-system model.

FA's performance was tested first for each S-system with different values of algorithm-dependent parameter α, where α= 0.05: 0.05: 1.0. On the otherhand, the remaining parameters were set at fixed values: β = 1.0, $n = 30$, MaxGeneration= 1000 and γ =1.0. For each S-system, we find the optimal setting of FA-dependent parameters (considering only the values we selected).

In addition to the final S-system parameter values, cost function values and CPU time, we also computed for the relative estimation error percentage (we will refer to this value as REE),

$$\text{REE} = \left\| \frac{\text{Estimate} - \text{True}}{\text{True}} \right\| \times 100\ \%, \tag{11}$$

for each parameter estimate. The mean was then recorded.

RESULTS

A test for the optimal value of α for each S-system was conducted. What were recorded as cost function values are the averages from the 10 independent runs using the corresponding α value. Results of the test for the optimal value of α in VA04, HS96 and CC06 are summarized in Table 1. The respective optimal values of α are 0.85 for VA04, 0.70 for HS96, and 0.15 for CC06.

VA04 Biochemical Network

Table 2 compares the FA parameter estimates of VA04 for different maximum number of iteration values (100; 1000; 2000; 10000), generated using noiseless datasets. Corresponding REE were recorded in Table 3.

Comparing the outcomes of FA using different values for the maximum number of iterations showed that we can obtain improved accuracy of the parameter estimates by increasing the number of iterations, as shown by the decrease in the mean REE in Table 3. This was also supported by the values in Table 4, where mean REEs both for rate constants and kinetic orders and both concentration error-based and slope-error-based cost functions improved with the increase in the number of iterations.

We can also observe from Table 4, that the REEs for kinetic orders are slightly lower (better) than the REEs for the rate constants. This could mean that FA

Table 1. Slope-Error Based Cost Function Value corresponding to each alpha value tested, for each S-system Model.

	COST FUNCTION VALUE		
α	VA04	HS96	CC06
0.05	0.5976	2.7198	0.6056
0.10	0.2140	0.7219	0.4532
0.15	0.0947	0.1054	**0.3308**
0.20	0.0530	0.0623	0.3538
0.25	0.0334	0.0515	0.3942
0.30	0.0281	0.0399	0.3394
0.35	0.0248	0.0401	0.3480
0.40	0.0202	0.0269	0.3393
0.45	0.0213	0.0240	0.4799
0.50	0.0215	0.0230	0.4170
0.55	0.0324	0.0163	0.4185
0.60	0.0188	0.0232	0.6084
0.65	0.0204	0.0231	0.8947
0.70	0.0186	**0.0113**	1.0328
0.75	0.0352	0.0351	1.2091
0.80	0.0206	0.0276	1.4183
0.85	**0.0152**	0.0289	1.3930
0.90	0.0196	0.0278	1.5590
0.95	0.0156	0.0354	1.4897
1.00	0.0330	0.0423	1.8106

Table 2. Parameter Estimates for VA04 using different maximum number of iterations

PARAMETERS	INITIAL	MAXIMUM ITERATIONS				TRUE
		100	1000	2000	10000	
α_1	2.0	11.9214	16.7220	17.9280	18.0000	18.0
g_{13}	-0.2	-0.9941	-0.8003	-0.8000	-0.8000	-0.8
β_1	1.0	5.9015	9.9910	9.9994	10.0000	10.0
h_{11}	0.5	0.6618	0.5003	0.5000	0.5000	0.5
α_2	2.0	6.9456	7.9981	7.9996	8.0003	8.0
g_{21}	0.5	0.5534	0.5001	0.5000	0.5000	0.5
β_2	1.0	2.3069	2.9986	2.9996	3.0003	3.0
h_{22}	0.5	0.8485	0.7502	0.7501	0.7500	0.8
α_3	2.0	2.3514	2.7442	2.9337	2.9976	3.0
g_{32}	0.5	0.8400	0.7826	0.7581	0.7503	0.8
β_3	1.0	4.1134	4.6601	4.9126	4.9967	5.0
h_{33}	0.5	0.5691	0.5241	0.5060	0.5002	0.5
h_{34}	0.5	0.2214	0.2078	0.2020	0.2001	0.2
α_4	2.0	1.6167	2.0002	2.0001	1.9999	2.0
g_{41}	0.5	0.5437	0.5000	0.5000	0.5000	0.5
β_4	1.0	5.3567	6.0002	6.0002	5.9997	6.0
h_{44}	0.5	0.8612	0.8000	0.8000	0.8000	0.8

Table 3. REE of Parameter Estimates for VA04 using different maximum number of iterations.

PARAMETERS	MAXIMUM ITERATIONS			
	100	1000	2000	10000
α_1	33.770 %	0.071 %	0.004 %	0.000 %
g_{13}	24.259 %	0.040 %	0.003 %	0.000 %
β_1	40.985 %	0.090 %	0.006 %	0.000 %
h_{11}	32.369 %	0.055 %	0.003 %	0.001 %
α_2	13.180 %	0.024 %	0.006 %	0.003 %
g_{21}	10.674 %	0.018 %	0.005 %	0.004 %
β_2	23.105 %	0.045 %	0.012 %	0.009 %
h_{22}	13.129 %	0.023 %	0.007 %	0.006 %
α_3	21.621 %	8.527 %	2.209 %	0.081 %
g_{32}	12.002 %	4.343 %	1.083 %	0.039 %
β_3	17.732 %	6.799 %	1.747 %	0.067 %
h_{33}	13.816 %	4.810 %	1.197 %	0.046 %
h_{34}	10.683 %	3.913 %	0.981 %	0.033 %
α_4	19.166 %	0.011 %	0.005 %	0.005 %
g_{41}	8.749 %	0.008 %	0.006 %	0.000 %
β_4	10.722 %	0.003 %	0.003 %	0.005 %
h_{44}	7.644 %	0.004 %	0.005 %	0.003 %

can more accurately estimate the numerical values of kinetic orders than those of the rate constants. However, the first row of Table 4 indicates that an increased number of iterations also means longer computation times.

Figure 6 shows how the VA04 model predictions using the estimated parameter values fit with the synthetic data derived from the model using the true parameter values. Note that there is an almost perfect fit with the model prediction and the synthetic data.

2. HS96 Biochemical Network

Table 5 compares the FA parameter estimates of HS96 for different maximum number of iteration values, generated using noiseless datasets. Corresponding REE were recorded in Table 6.

Similar to the computational results presented in VA04, we can obtain improved accuracy of the parameter estimates by increasing the number of iterations, as supported also by the values from Table 7 for mean REEs both of rate constants and kinetic

Table 4. Quality of Parameter Estimates for VA04 using different maximum number of iterations.

	MAXIMUM ITERATIONS			
	100	**1000**	**2000**	**10000**
CPU Time (sec)	67.61	632.08	1222.06	5477.25
Mean REE for Rate Constants	22.535%	1.946%	0.499%	0.021%
Mean REE for Kinetic Orders	14.814%	1.468%	0.365%	0.015%
Slope Error-Based Cost Function Value	0.2465	0.0063	0.0016	0.0001
Concentration Error-Based Cost Function Value	0.2500	0.0206	0.0050	0.0002

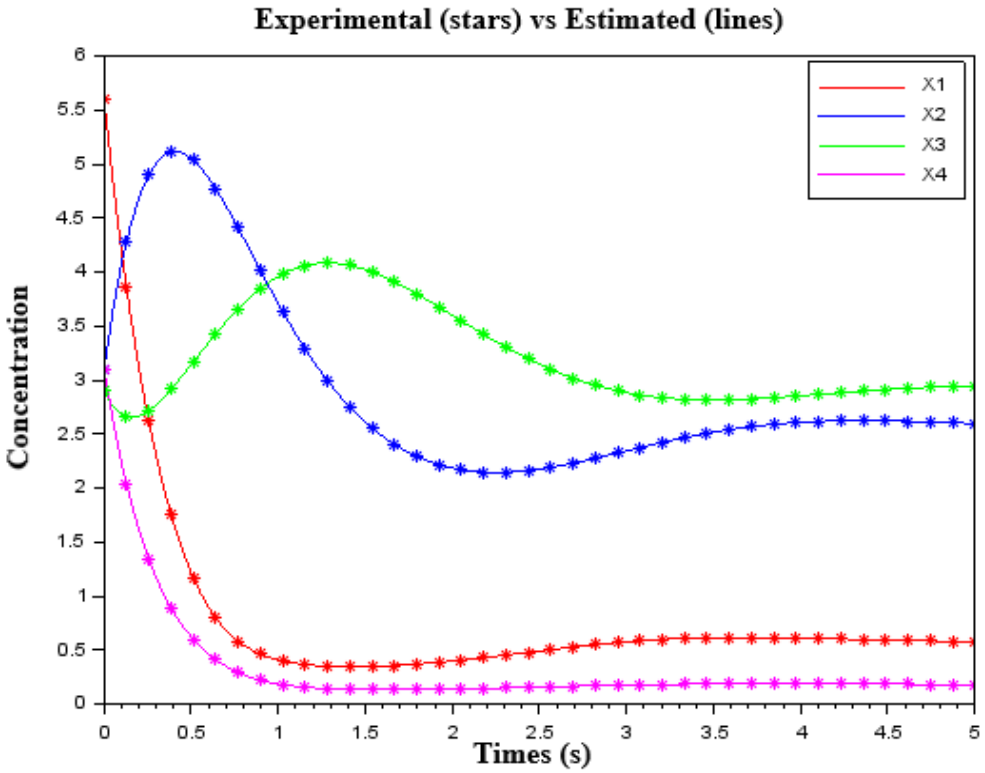

Figure 6. Plot of VA04 synthetic data (points) and the model prediction using the estimated parameters.

Table 5. Parameter Estimates for HS96 using different maximum number of iterations.

PARAMETERS	INITIAL	MAXIMUM ITERATIONS				TRUE
		100	1000	2000	10000	
α_1	2.0	4.3256	4.9999	5.0000	4.9997	5.0
g_{13}	0.5	0.9038	1.0000	1.0000	1.0000	1.0
g_{15}	-2.0	-1.0623	-1.0000	-1.0000	-1.0000	-1.0
β_1	1.0	6.9106	9.9997	9.9996	9.9995	10.0
h_{11}	0.5	1.2093	2.0000	1.9999	2.0000	2.0
α_2	2.0	8.3078	10.0002	10.0000	9.9999	10.0
g_{21}	0.5	1.8213	2.0000	2.0000	2.0000	2.0
β_2	1.0	7.4680	10.0003	10.0001	9.9998	10.0
h_{22}	0.5	1.6179	2.0000	2.0000	2.0000	2.0
α_3	2.0	5.6233	10.0954	9.9993	10.0004	10.0
g_{32}	-0.2	-1.2794	-0.9956	-1.0000	-1.0000	-1.0
β_3	1.0	5.5144	10.0991	9.9993	10.0004	10.0
h_{32}	-0.2	-1.2861	-0.9954	-1.0000	-1.0000	-1.0
h_{33}	0.5	1.8024	1.9958	2.0000	2.0000	2.0
α_4	2.0	4.6991	7.9998	8.0003	8.0003	8.0
g_{43}	0.5	1.9309	2.0000	1.9999	2.0000	2.0
g_{45}	-0.2	-1.2969	-1.0000	-1.0000	-1.0000	-1.0
β_4	1.0	6.0641	9.9998	10.0004	10.0002	10.0
h_{44}	0.5	2.0000	2.0000	2.0000	2.0000	2.0
α_5	2.0	7.8712	10.0000	10.0000	10.0003	10.0
g_{54}	0.5	2.0000	2.0000	2.0000	2.0000	2.0
β_5	1.0	7.5383	10.0001	10.0001	10.0003	10.0
h_{55}	0.5	2.0000	2.0000	2.0000	2.0000	2.0

Table 6. REE of Parameter Estimates for HS96 using different maximum number of iterations.

PARAMETERS	MAXIMUM ITERATIONS			
	100	1000	2000	10000
α_1	13.488%	0.002%	0.001%	0.006%
g_{13}	9.618%	0.005%	0.005%	0.004%
g_{15}	6.235%	0.000%	0.001%	0.004%
β_1	30.894%	0.003%	0.004%	0.005%
h_{11}	39.537%	0.000%	0.008%	0.000%
α_2	16.922%	0.002%	0.000%	0.001%
g_{21}	8.935%	0.000%	0.000%	0.000%
β_2	25.320%	0.003%	0.001%	0.002%
h_{22}	19.106%	0.000%	0.000%	0.000%
α_3	43.767%	0.954%	0.007%	0.004%
g_{32}	27.944%	0.437%	0.003%	0.002%
β_3	44.856%	0.991%	0.007%	0.004%
h_{32}	28.609%	0.464%	0.003%	0.002%
h_{33}	9.881%	0.210%	0.000%	0.001%
α_4	41.261%	0.003%	0.004%	0.004%
g_{43}	3.453%	0.000%	0.003%	0.000%
g_{45}	29.693%	0.001%	0.003%	0.002%
β_4	39.359%	0.002%	0.004%	0.002%
h_{44}	0.001%	0.000%	0.000%	0.000%
α_5	21.288%	0.000%	0.000%	0.003%
g_{54}	0.000%	0.000%	0.000%	0.000%
β_5	24.617%	0.001%	0.001%	0.003%
h_{55}	0.001%	0.000%	0.000%	0.000%

Table 7. Quality of Parameter Estimates for HS96 using different maximum number of iterations.

	MAXIMUM ITERATIONS			
	100	1000	2000	10000
CPU Time (sec)	85.03	599.06	912.59	2451.59
Mean REE for Rate Constants	30.177%	0.196%	0.003%	0.003%
Mean REE for Kinetic Orders	14.078%	0.086%	0.002%	0.001%
Slope Error-Based Cost Function Value	0.6230	0.0032	0.0001	0.0001
Concentration Error-Based Cost Function Value	0.1182	0.0007	0.0000	0.0000

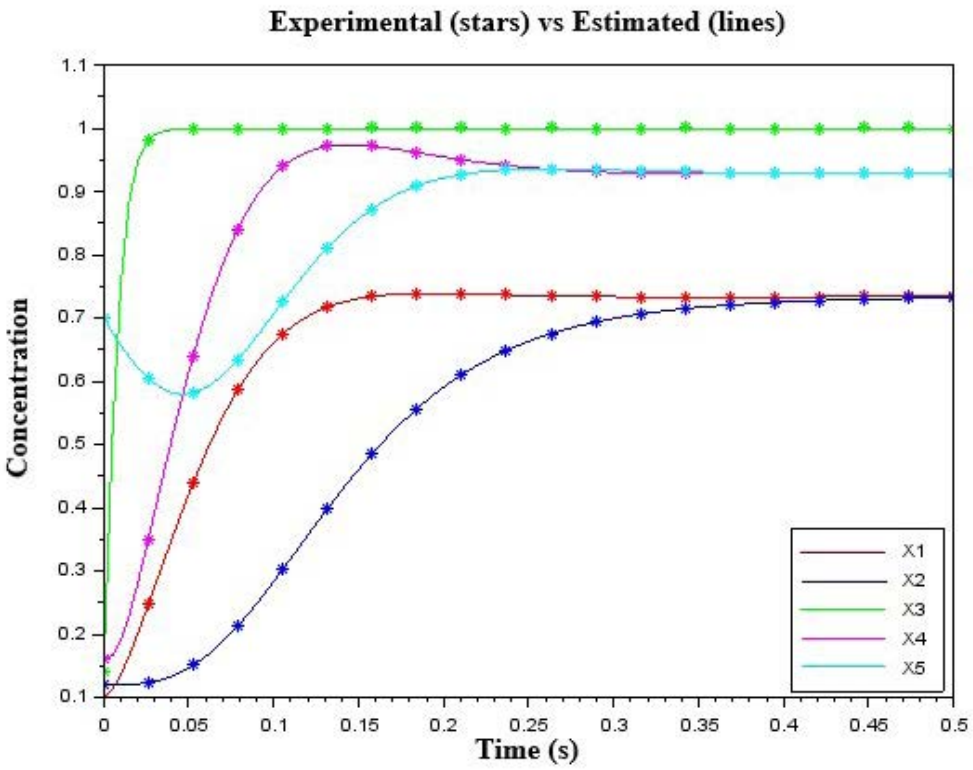

Figure 7. Plot of HS96 synthetic data (points) and the model prediction using the estimated parameters.

orders and for concentration error-based and slope-error-based cost function values. Still in Table 7, the REEs for kinetic orders are better than those for the rate constants.

This could mean that for HS96, like in VA04, FA can more accurately estimate the numerical values of kinetic orders than those of the rate constants.

Similar to VA04, Figure 7 shows how the HS96 model predictions using the estimated parameter values fit with the synthetic data derived from the model using the true parameter values. Note that there is also an almost perfect fit with the model prediction and the synthetic data.

3. CC06 Biochemical Network

Table 8 compares the FA parameters estimates of CC06 for different maximum numbers of iteration values, generated using noise-free datasets. Corresponding REEs were recorded in Table 9.

Similar to the computational results presented in VA04 and HS96, we can obtain improved accuracy of the parameter estimates by increasing the number of iterations, as supported also by the values from Table 10 for mean REEs both of rate constants and kinetic orders, and for concentration error-based and slope-error-based cost function values. Still in Table 10, the REEs for kinetic orders are better than those for the rate constants. This could mean that for CC06, like in the first two S-systems, FA can more accurately estimate the numerical values of kinetic orders than those of the rate constants. It should be noted, however, that the estimated values for CC06 parameters are significantly different from the true parameters as reflected by large REE values.

Figure 8 shows the fit of CC06 models predictions with the synthetic data. CC06 model predictions using the estimated parameter values still shows good fit with the synthetic data, though this is expected since concentration-based error is still minimal despite significant differences in parameter values.

Table 8. Parameter Estimates for CC06 using different maximum number of iterations.

PARAMETERS	INITIAL	MAXIMUM ITERATIONS				TRUE
		100	1000	2000	10000	
α_1	2.0	2.3688	2.0181	2.8162	1.8016	1.0000
g_{12}	-0.2	-0.1259	-0.0543	-0.0493	-0.0492	-0.0492
g_{16}	0.5	0.5123	0.8099	0.7315	0.8475	1.0000
β_1	1.0	1.6723	1.5452	2.0846	2.3119	1.6497
h_{11}	0.5	0.9789	0.5711	0.5586	0.5582	0.5582
h_{15}	0.5	0.1270	0.0477	0.0465	0.0465	0.0465
h_{17}	0.5	0.6136	1.0078	0.9261	0.8940	1.0000
α_2	2.0	3.4905	2.6020	3.8257	2.9818	1.6497
g_{21}	0.5	0.6571	0.5879	0.5638	0.5582	0.5582
g_{25}	0.5	0.1488	0.0630	0.0505	0.0465	0.0465
g_{29}	0.5	0.6231	0.8210	0.7283	0.8140	1.0000
β_2	1.0	0.8718	0.5113	0.4875	1.6618	0.5793
h_{22}	0.5	0.5819	0.5370	0.5145	0.5098	0.5097
h_{25}	-0.2	-0.1267	-0.2165	-0.2192	-0.2219	-0.2218
h_{28}	0.5	0.5080	0.2678	0.7957	0.4161	0.8322
$h_{2,11}$	0.5	0.2570	0.6775	0.2339	0.3048	0.1678
α_3	2.0	1.2116	2.8130	2.1322	2.3532	0.4536
g_{32}	0.5	0.4030	0.4831	0.4604	0.4418	0.4407
g_{35}	-0.2	-0.1757	-0.2487	-0.2790	-0.2675	-0.2665
g_{38}	0.5	0.5523	0.4849	0.5951	0.5862	1.0000
β_3	1.0	0.6547	0.6738	1.1055	0.9747	0.2456
h_{33}	0.5	0.4527	0.4964	0.4687	0.4516	0.4506
h_{34}	0.5	0.1037	0.0464	0.0457	0.0442	0.0441
h_{35}	0.5	0.3675	0.1397	0.0962	0.0920	0.0920
h_{39}	0.5	0.2147	0.3131	0.5305	0.4751	0.8547
$h_{3,12}$	0.5	0.3639	0.4108	0.1209	0.2186	0.1453
α_4	2.0	0.5562	2.0137	2.4624	1.3812	0.2365
g_{43}	0.5	0.4677	0.5144	0.5266	0.5287	0.5285
g_{45}	0.5	0.4586	0.1344	0.1021	0.0992	0.0994
g_{49}	0.5	0.4924	0.5182	0.4823	0.6090	1.0000
β_4	1.0	1.7541	1.3799	1.6138	1.8336	2.0892
h_{43}	-0.2	-0.1234	-0.0077	-0.0067	-0.0076	-0.0075
h_{44}	0.5	0.4437	0.3006	0.3025	0.3041	0.3040
h_{45}	0.5	0.8819	0.5076	0.4846	0.4840	0.4840
$h_{4,10}$	0.5	0.6706	1.1300	1.0902	1.0449	1.0000
α_5	2.0	2.2831	1.2161	2.6584	1.5504	1.4060
g_{53}	0.5	0.4062	0.3117	0.2712	0.2606	0.2605
g_{54}	0.5	0.2481	0.1773	0.1583	0.1521	0.1520
g_{55}	0.5	0.0332	0.1590	0.0907	0.0741	0.0739
g_{59}	0.5	0.2068	0.3427	0.4100	0.5052	0.5000
$g_{5,10}$	0.5	0.5690	0.6556	0.3891	0.4578	0.5000
β_5	1.0	1.2951	1.5256	0.6619	0.5821	2.9437
h_{51}	0.5	0.3295	0.2198	0.2022	0.1963	0.1962
h_{52}	0.5	0.2905	0.2071	0.1853	0.1792	0.1791
h_{55}	0.5	0.2867	0.3651	0.2628	0.2357	0.2354
h_{57}	0.5	0.1580	0.4217	0.4237	0.6201	0.3514
h_{58}	0.5	0.4316	0.1666	0.4560	0.1484	0.2925
$h_{5,11}$	0.5	0.0579	0.1628	0.1723	0.3620	0.0589
$h_{5,13}$	0.5	0.4980	1.0313	0.5138	0.5587	0.2970

Table 9. REE of Parameter Estimates for CC06 using different maximum number of iterations.

PARAMETERS	MAXIMUM ITERATIONS			
	100	1000	2000	10000
α_1	136.882%	101.806%	181.623%	80.160%
g_{12}	155.905%	10.411%	0.279%	0.007%
g_{16}	48.774%	19.006%	26.845%	15.247%
β_1	1.370%	6.335%	26.360%	40.141%
h_{11}	75.372%	2.308%	0.070%	0.002%
h_{15}	173.020%	2.501%	0.082%	0.005%
h_{17}	38.637%	0.778%	7.392%	10.604%
α_2	111.585%	57.728%	131.905%	80.746%
g_{21}	17.726%	5.329%	1.007%	0.007%
g_{25}	220.016%	35.408%	8.560%	0.039%
g_{29}	37.693%	17.897%	27.170%	18.604%
β_2	50.496%	11.744%	15.839%	186.859%
h_{22}	14.167%	5.349%	0.946%	0.010%
h_{25}	42.877%	2.377%	1.167%	0.025%
h_{28}	38.954%	67.824%	4.382%	50.001%
$h_{2,11}$	53.145%	303.767%	39.409%	81.636%
α_3	167.112%	520.139%	370.069%	418.785%
g_{32}	8.557%	9.632%	4.480%	0.255%
g_{35}	34.066%	6.683%	4.701%	0.364%
g_{38}	44.772%	51.511%	40.494%	41.376%
β_3	166.577%	174.368%	350.130%	296.877%
h_{33}	0.465%	10.168%	4.024%	0.226%
h_{34}	135.139%	5.266%	3.598%	0.212%
h_{35}	299.494%	51.816%	4.526%	0.021%
h_{39}	74.881%	63.369%	37.929%	44.409%
$h_{3,12}$	150.417%	182.705%	16.781%	50.448%
α_4	135.180%	751.447%	941.186%	484.027%
g_{43}	11.498%	2.677%	0.350%	0.034%
g_{45}	361.412%	35.234%	2.732%	0.243%
g_{49}	50.758%	48.180%	51.770%	39.098%
β_4	16.040%	33.952%	22.754%	12.234%
h_{43}	1545.601%	3.208%	11.284%	1.115%
h_{44}	45.948%	1.107%	0.479%	0.047%
h_{45}	82.205%	4.870%	0.115%	0.006%
$h_{4,10}$	32.939%	13.004%	9.017%	4.494%
α_5	62.384%	13.508%	89.076%	10.274%
g_{53}	55.938%	19.668%	4.097%	0.037%
g_{54}	63.218%	16.671%	4.149%	0.043%
g_{55}	55.119%	115.141%	22.693%	0.314%
g_{59}	58.640%	31.467%	18.005%	1.040%
$g_{5,10}$	13.796%	31.123%	22.175%	8.445%
β_5	56.005%	48.176%	77.513%	80.227%
h_{51}	67.917%	12.040%	3.042%	0.027%
h_{52}	62.179%	15.622%	3.443%	0.031%
h_{55}	21.802%	55.082%	11.637%	0.144%
h_{57}	55.039%	20.012%	20.564%	76.470%
h_{58}	47.564%	43.053%	55.907%	49.258%
$h_{5,11}$	1.685%	176.388%	192.564%	514.598%
$h_{5,13}$	67.693%	247.227%	72.996%	88.108%

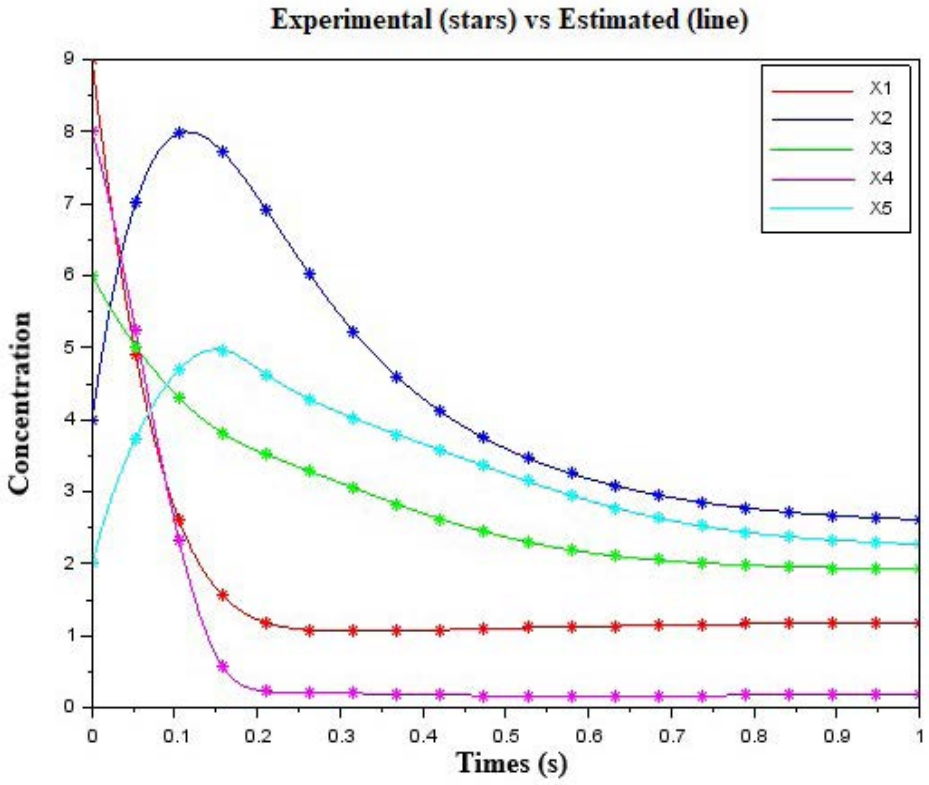

Figure 8. Plot of CC06 synthetic data (points) and the model prediction using the estimated parameters.

Table 10. Quality of Parameter Estimates for CC06 using different maximum number of iterations.

	MAXIMUM ITERATIONS			
	100	**1000**	**2000**	**10000**
CPU Time (sec)	79.46	772.44	1638.21	7766.27
Mean REE for Rate Constants	90.363%	171.920%	220.646%	169.033%
Mean REE for Kinetic Orders	111.924%	44.766%	18.997%	28.130%
Slope Error-Based Cost Function Value	1.8430	0.2388	0.0511	0.0013
Concentration Error-Based Cost Function Value	0.5459	0.0543	0.0134	0.0003

DISCUSSION

The Firefly Algorithm (FA), implemented with the decoupling method, was successful in the parameter estimation of a generic branched-pathway (VA04), a simple genetic network (HS96), and a yeast fermentation network (CC06), using specified parameter estimation steps from MUG (del Rosario et. al. 2008).

Complexities of the given biochemical networks affected the performance of FA with decoupling method in estimating the parameters of their respective S-system models. For VA04 and HS96, FA produced approximations of the S-system parameters which are close to the true values. However, in CC06, FA generated results which are significantly different from the true values, even though it provides a good fit. The difference in the results may be due to the presence of independent variables and the number or parameters to be estimated. We note that VA04 only has 1 independent variable with 17 parameters to be estimated, HS96 does not have independent variables but has 23 parameters to be estimated, while CC06 has 8 independent variables and 49 parameters to be estimated. Moreover, it seems that in these S-systems, the FA-decoupling method can more accurately estimate numerical values of the kinetic orders than rate constants.

For all 3 S-system models, we only tuned the FA-parameter α. As pointed out in the paper of Yang et. al. (2013), the algorithm-dependent parameter tuning is also a complicated optimization problem. Since such optimization per se is not the goal of this study, we just considered specific sets of algorithm-dependent parameter values. The results could probably be improved if we properly test FA-parameters through all their possible values and through all their possible combinations.

This paper only focused on parameter estimation using noise-free data. To check how the method is affected by the presence of noise, the authors of this paper also plan to do parameter estimation using datasets with different noise levels.

ACKNOWLEDGMENTS

The codes used in this study were developed at the University of the Philippines –Diliman under the supervision of Dr. Cherryl O. Talaue.

This study is supported by grants from the Department of Science and Technology – Science Education Institute (DOST–SEI) under the Accelerated Science and Technology Human Resource Development Program (ASTHRDP).

LITERATURE CITED

Almeida JS, Voit EO. 2003. Neural-network-based parameter estimation in s-system models of biological networks in genome informatics. GENOME INFORM 14: 114-123.

Chou IC, Voit EO. 2009. Recent developments in parameter estimation and structure identification of biochemical and genomic systems. MATH BIOSCI 219: 57–83.

del Rosario RCH, Echavez M, de Paz M, Zuniga PC, Bargo MC, Arellano C, Pasia JM, Naval JRP, Voit EO, Mendoza ER. 2008. MADMan: a benchmarking framework for parameter estimation in biochemical systems theory models. In: International Conference of Molecular Systems.

Fister I, Fister I, Yang XS, Brest J. 2013. A comprehensive review of firefly algorithms. SWARM EVOL COMPUT 13: 34-46.

Gonzalez OR, Küper C, Jung K, Naval Jr PC, Mendoza ER. 2007. Parameter estimation using simulated annealing for s-system models of biochemical networks. BIOINFORMATICS 23: 480-486.

Kikuchi S, Tominaga D, Arita M, Takahashi K, Tomita M. 2003. Dynamic modeling of genetic networks using genetic algorithm and s-system. BIOINFORMATICS 19: 643-650.

Mo YB, Ma YZ, Zheng QY. 2013. Optimal Choice of Parameters for Firefly Algorithm. In: 4th International Conference on Digital Manufacturing and Automation (ICDMA). 887-892.

Naval Jr PC, Sison LG, Mendoza ER. 2010. Parameter estimation with term-wise decomposition in biochemical network GMA models by hybrid regularized least squares-particle swarm optimization. In: IEEE C EVOL COMPUTAT.

Savageau MA, Voit EO. 1987. Recasting nonlinear differential equations as s-systems: a canonical nonlinear form. MATH BIOSCI 87:83-115

Voit EO. 2000. Computational analysis of biochemical systems: a practical guide for biochemists and molecular biologists. New York: Cambridge University Press. 532 p

Voit EO. 2013. Biochemical systems theory: a review. ISRN Biomathematics; 2013:1-53.

Voit EO, Almeida JS. 2004. Decoupling dynamical systems for pathway identification from metabolic profiles. BIOINFORMATICS 20:1670-1681.

Yang XS. 2008. Firefly algorithm. In: Nature-inspired metaheuristic algorithms. Bristol, UK: Luniver Press: 79-89.

Yang XS. 2010. Firefly algorithm. In: Nature-inspired metaheuristic algorithms. 2. Bristol, UK: Luniver Press: 81-89.

Yang XS (ed). 2014. Cuckoo search and firefly algorithm: overview and analysis. Studies in Computational Intelligence 516. 1. USA: Springer:1-26.

Yang XS, Deb S, Loomes M, Karamanoglu M. 2013. A framework for self-tuning optimization algorithm, neural computing and applications. NEURAL COMPUT APPL 23:2051-2057.

Yang XS, He XS. 2018. Why the firefly algorithm works?. In: Yang XS, editors. Nature-Inspired Algorithms and Applied Optimization. USA: Springer, pp. 245-259.

Zuniga PC. 2011. Using the ant colony optimization algorithm in the network inference and parameter estimation of biochemical systems. J COMPUT BIOL BIOINFORM 3(5):53-62.