Estimating parameters of S-systems by an auxiliary function guided coordinate descent method

Li-Zhi Liu, Fang-Xiang Wu* and Wen-Jun Zhang

Department of Mechanical Engineering, Division of Biomedical Engineering, University of Saskatchewan, Saskatoon, SK, Canada

(Received 20 November 2013; final version received 20 January 2014)

The S-system, a set of nonlinear ordinary differential equations and derived from the generalized mass action law, is an effective model to describe various biological systems. Parameters in S-systems have significant biological meanings, yet difficult to be estimated because of the nonlinearity and complexity of the model. Given time series biological data, its parameter estimation turns out to be a nonlinear optimization problem. A novel method, auxiliary function guided coordinate descent, is proposed in this paper to solve the optimization problem by cyclically optimizing every parameter. In each iteration, only one parameter value is updated and it proves that the objective function keeps nonincreasing during the iterations. The updating rules in each iteration is simple and efficient. Based on this idea, two algorithms are developed to estimate the S-systems for two different constraint situations. The performances of algorithms are studied in several simulation examples. The results demonstrate the effectiveness of the proposed method.

Keywords: parameter estimation; nonlinear programming; optimization; S-system; coordinate descent

1. Introduction

There are many various molecules and productions in a cell. Some of them can regulate others via some mechanisms to achieve specific cellular functions, based on which the cells adapt to the changing environments. These components and their interactions constitute biological systems, such as metabolic pathways and genetic regulatory networks. One task of systems biology is to reveal the interactions and the biological functions those interactions may result in Klipp, Herwig, Kowald, Wierling, and Lehrach (2005). Instead of focusing on individual components, systems biology applies system engineering methods and principles to studying all components and their interactions as parts of a biological system. Such a systematic view provides an insight into the control and optimization of parts of the systems while taking the effects those may have on the whole system into account. It is of help to the discovery of new properties of biological systems and may provide valuable clues and new ideas in practical areas such as disease treatment and drug design (Gardner, di Bernardo, Lorenz, and Collins 2003).

One effective way to study the biological system is using mathematical or computational methods. Many mathematical models have been developed to describe the molecular biological systems based on biochemical principles. Most of them are nonlinear in both parameters and state variables (Klipp et al. 2005; Voit 2000). Nonlinear optimization problems are usually formulated for estimating the parameters in those models. However, analytical solutions to those problems are hardly available. One of the most popular models is the S-system which is highly nonlinear and derived from the generalized mass action law (Voit, 2000).

The S-system is an effective mathematical framework to characterize and analyse the molecular biological systems. An S-system consisting of \(N\) components is type of power-law formalism and typically a group of nonlinear ordinary differential equations (ODEs),

\[
\dot{X}_i = \alpha_i \prod_{j=1}^{N} X_j^{g_{ij}} - \beta_i \prod_{j=1}^{N} X_j^{h_{ij}}, \quad i = 1, \ldots, N, \tag{1}
\]

where \(X_i\) represents the concentration of molecular species \(i\) measured at time \(t\), whose changes are the difference between production and degradation. \(\alpha_i\’s\) and \(\beta_i\’s\) are non-negative rate constants, and \(g_{ij}\’s\) and \(h_{ij}\’s\) are real-valued kinetic orders that reflect the interaction intensity of \(X_i\) to \(X_j\). If \(g_{ij} > 0\), \(X_j\) activates the production of \(X_i\); if \(g_{ij} < 0\), \(X_j\) inhibits the production of \(X_i\); \(h_{ij}\) has the same effects but on the degradation. A zero-valued kinetic order indicates that \(X_j\) has no such effect on \(X_i\).

The parameter estimation and structure identification of the S-systems are extremely difficult tasks. The parameter estimation usually occurs after or in the process of structure identification. As the parameter estimation of the S-systems is essentially a nonlinear problem, in principle, all nonlinear...
optimization algorithms can be used, e.g. Gauss–Newton method and its variants, such as Box–Kanemasu interpolation method, Levenberg damped least-squares method, and Marquardt’s method (Beck & Arnold, 1977). However, most of these methods are initial-sensitive and need to calculate the inverse of the Hessian which is computationally expensive.

Several numerical methods have been proposed to estimate the parameters in S-systems. Most of them are based on heuristic methods. For example, Kikuchi, Tominaga, Arita, Takahashi, and Tomita (2003) employ a genetic algorithm to infer the S-systems. The effectiveness of the simulated annealing is studied in Gonzalez, Kper, Jung, Naval, and Mendoza (2007). Voit and Almeida (2004) develop an artificial neural network (ANN)-based method to identify and estimate the parameters of S-systems. Ho, Hsieh, Yu, and Huang (2007) develop an intelligent two-stage evolutionary algorithm that combines the genetic algorithm and the simulated annealing. An unified approach has been proposed in Wang, Qian, and Dougherty (2010). Most of these methods are computational expensive and do not sufficiently take advantage of special model structure of the S-systems.

Several methods taking the model structure of S-systems into account have been proposed. Wu and Mu (2009) introduce a separable parameter estimation method which divides the parameters into two groups: one group is linear in model while the other nonlinear. This method has been extended with a genetic algorithm to the case when system topology is unavailable in Liu, Wu, and Zhang (2012b). One can observe that if parameters in one term on the right-hand side of Equation (1) is known, moving it to the left and taking logarithm of both sides, a linear model is obtained. Using this observation, an alternating regression method is proposed by Chou, Martens, and Voit (2006), which reduces the nonlinear optimization into iterative procedures of linear regression. However, the convergence of the iterations cannot be guaranteed. Similarly, an alternating weighted least-squares method is proposed by Liu, Wu, and Zhang (2012a), in which the objective function to be optimized is approximated by a weighted linear regression in each iteration. However, this approximation may fail in some cases.

This study focus on the parameter estimation of S-systems when system topology is known. A novel method, auxiliary function guided coordinate descent (AFGCD), is proposed. After decoupling the S-systems by replacing the derivatives with numerical slops, the parameter estimation is formulated as a nonlinear optimization problem. AFGCD iteratively solves this problem and in each iteration only one parameter is optimized with other parameters fixed. Instead of directly optimizing the objective function, AFGCD takes advantage of a property of the exponential function and constructs an auxiliary function for each kinetic order parameter to optimize. It shows that optimizing the auxiliary function makes the objective function keep nonincreasing. Because the auxiliary function is simple and its optimization has an analytical solution, the parameter updating in each iteration is simple and efficient. Based on this idea, two algorithms are developed to estimate the parameters in S-systems under two different situations. One algorithm is developed for the case when the range of each parameter is known and the other algorithm for the case that the only constraint is the non-negativity of the rate constants.

The remaining of this paper is organized as follows. In Section 2, two algorithms are developed based on the idea of AFGCD and their descent properties are proven. In Section 3, the effectivenesses of the proposed algorithms are studied by several simulation examples. Finally, Section 4 concludes this study and points out some future works along this research.

2. Auxiliary function guided coordinate descent

2.1. Problem statement

Consider a biological system of \( N \) molecular species, described by an S-system as Equation (1), and assume that for each molecular species \( X_i \), a time series concentration data measured at \( n \) equally spaced time points, \( x_{1i}, \ldots, x_{ni} \) are obtained. This study assumes that the topology of the system is available, i.e. the zero-values kinetic orders are known. The purpose is to estimate the parameters of nonzero-valued kinetic orders and rate constants. We substitute the derivative of \( X_i \) at time \( t \) with the estimated slop, \( S_{it} \), so that the original coupled ordinary differential equations are decoupled into \( n \times N \) uncoupled algebraic equations (Vilela et al., 2008; Voit & Almeida, 2004):

\[
S_{it} = \alpha_i \prod_{j=1}^{N} x_{jt}^{gij} - \beta_i \prod_{j=1}^{N} x_{jt}^{hij} + \epsilon_{it},
\]

where \( i = 1, \ldots, N, t = 1, \ldots, n \) and \( \epsilon_{it} \)'s are errors or noises. The estimation of slopes is a crucial step and may affect the final results. To increase the accuracy, the five-point numerical derivative method is used in this study, i.e.

\[
S_{it} = \frac{-x_{i, t+2} + 8x_{i, t+1} - 8x_{i, t-1} + x_{i, t-2}}{12\Delta t},
\]

where \( \Delta t \) is the length of the sampling step.

To determine the parameter values, the sum of least squares is usually employed as an objective function to be minimized, i.e. the parameters of each ODE equation \( i \) in Equation (1) is obtained by solving

\[
\begin{align*}
\text{minimize} & \; J_i = \frac{1}{2} \sum_{t=1}^{n} \left[ S_{it} - \alpha_i \prod_{j \in G_i} x_{jt}^{gij} + \beta_i \prod_{j \in H_i} x_{jt}^{hij} \right]^2 \\
\text{subject to} & \; \alpha_i \geq \delta, \quad \beta_i \geq \delta
\end{align*}
\]
where $\delta$ is a predefined small positive number; $G_i$ and $H_i$ are the sets of indices of molecular species which have effects on the production and degradation of molecular species $i$, respectively; $g_i = \{g_{ij} | j \in G_i\}$ and $h_i = \{h_{ij} | j \in H_i\}$. The minimization of Equation (4) is non-trivial, as it is highly nonlinear and contains a lot of parameters.

2.2. Proposed method

The optimization problem (4) has no analytical solutions and therefore, iterative methods are considered. A coordinate descent strategy is applied in this study, i.e. we cyclically optimize one parameter in one iteration. More specifically, denote by $\theta^{(\ell)} = (\theta_1^{(\ell)}, \ldots, \theta_p^{(\ell)})$ the parameters of an optimization problem (e.g. for Equation (4), $\theta = (\alpha, g_i, \beta_i, h_i)$) at iteration $\ell$. After one iteration, the first parameter is updated, $\theta_1^{(\ell+1)} = (\theta_1^{(\ell+1)}, \theta_2^{(\ell)}, \ldots, \theta_p^{(\ell)})$ and after $k$ iterations, the first $k$ parameters are updated, $\theta_k^{(\ell+1)} = (\theta_1^{(\ell+1)}, \ldots, \theta_k^{(\ell+1)}, \theta_{k+1}^{(\ell)}, \ldots, \theta_p^{(\ell)})$. Consider the problem (4) and suppose the parameter $h_{ik}$, for some $i \in H_i$, needs to be updated at the current iteration, one simple way to make sure that the objective function keeps nonincreasing is to use the classical gradient descent method:

$$h_{ik}^{(\ell+1)} = h_{ik}^{(\ell)} - d_{\ell} \frac{\partial J_i(h_{ik}^{(\ell)})}{\partial h_{ik}}, \quad (5)$$

where values of other parameters are fixed and $d_{\ell}$ is the stepsize taken along the negative gradient direction. Some methods can be used to search and select the stepsize, such as minimization rule and Armijo rule (Bazaraa, Sherali, & Shetty, 2006; Bertsekas, 1999). However, these methods may either depend on a line search algorithm or introduce an extra iteration loop.

In this paper, a smart stepsize is derived. The introduced stepsize is computational efficient and the non-increase of the objective function in each iteration is guaranteed, which is proved in the following subsection. The algorithm for solving the problem (4) is illustrated in Algorithm 1, in which the updates for each parameter are shown in Equations (6)–(9). To check the convergence, the following stopping criteria is used

$$\frac{\|\theta^{(\ell+1)} - \theta^{(\ell)}\|_2}{\|\theta^{(\ell)}\|_2} < \eta,$$

here $\eta$ is a preset threshold. In this paper, we set $\eta = 10^{-5}$.

In some situations, from experiments, the range of each parameter is known. Then, more constraints can be added to the problem (4) as follows:

$$\text{subject to} \quad r_{\min} \leq \alpha_i \leq r_{\max}, \quad r_{\min} \leq \beta_i \leq r_{\max},$$

$$k_{\min} \leq g_{ij} \leq k_{\max}, \quad j \in G_i,$$

$$k_{\min} \leq h_{ij} \leq k_{\max}, \quad j \in H_i, \quad (10)$$

that is, we require the rate constants stay in the range $[r_{\min}, r_{\max}]$ and kinetic orders in the range $[k_{\min}, k_{\max}]$. Simple adaptations to Algorithm 1 lead to Algorithm 2 for solving the problem (10). The update rules (11) and (13) are slight modifications to Equations (6) and (8). They efficiently make the objective function keep nonincreasing, which is proved in the following subsection.

2.3. Descent property

Similar to Seung and Lee (2001) and Li, Wu, Zhang, and Wu (2013), we make use of an auxiliary function to prove the nonincrease of the objective function in each iteration.

DEFINITION 1 $F(h, h')$ is an auxiliary function of $J(h)$ if there exists $\sigma(h')$, such that, if $|h - h'| \leq \sigma(h')$

$$F(h, h') \geq J(h) \quad \text{and} \quad F(h, h) = J(h) \quad (15)$$

are satisfied.

The auxiliary function is useful because of the following lemma.

LEMA 1 If $F$ is an auxiliary function, then $J$ is nonincreasing under the update

$$h^{(\ell+1)} = \arg \min_{h \in \mathbb{R}} F(h, h^{(\ell)}). \quad (16)$$

Proof

$$J(h^{(\ell+1)}) \leq F(h^{(\ell+1)}, h^{(\ell)}) \leq F(h^{(\ell)}, h^{(\ell)}) = J(h^{(\ell)})$$

Lemma 2 shows a property of the exponential function, which is useful for the following discussions.

LEMA 2 For some point $h'$, the exponential function $e^h$ satisfies

$$e^h \leq e^{h'} + e^{h'}(h - h') + e^{h'}(h - h')^2 \Delta(\sigma)$$

if $|h - h'| \leq \sigma, \quad (17)$

where

$$\Delta(\sigma) = \frac{1}{\sigma^2}(e^\sigma - 1 - \sigma). \quad (18)$$
Algorithm 1 AFGCD for problem (4).

1. Assign initial values to $\theta^{(0)} = (\alpha^{(0)}_i, g^{(0)}_i, \beta^{(0)}_i, h^{(0)}_i)$ and set $\ell = 0$.
2. Let $y_t = -S_t - \beta^{(t)}_i \prod_{j \in H_i} x_{jt}^{h^{(t)}_j}$, $t = 1, \ldots, n$ and cyclically update $\theta^{(t)}_{ik}$ to $\theta^{(t+1)}_{ik}$ for each $k \in G_i$ as follows:

$$
\theta^{(t+1)}_{ik} = \theta^{(t)}_{ik} - \frac{1}{2\tau_g(\sigma^{(t)}_g)} \frac{\partial J_g(\theta^{(t)}_{ik})}{\partial \theta^{(t)}_{ik}},
$$

where

$$
\tau_g(\sigma^{(t)}_g) = \sum_t a_t |y_t| x_{it}^{g^{(t)}_i} \alpha(\theta^{(t)}_{ik}) \beta(\theta^{(t)}_{ik}) \sum_{\sigma(\theta^{(t)}_{ik})} \sigma(\theta^{(t)}_{ik}) \log^2 x_{it} + 2 \sum_t a_t^2 x_{it}^{g^{(t)}_i} \log^2 x_{it}
$$

and the function $\Delta(\cdot)$ is defined in Equation (18).

3. Compute

$$
\alpha^{(t+1)}_i = \max \left\{ -\sum_t y_t p_t, \frac{\sum_t p_t}{\sum_t p_t^2}, \delta \right\}, \quad \text{where } p_t = \prod_{j \in G_i} x_{jt}^{g^{(t)}_j}
$$

4. Let $y_t = S_t - \alpha^{(t+1)}_i \prod_{j \in H_i} x_{jt}^{g^{(t+1)}_j}$, $t = 1, \ldots, n$ and cyclically update $h^{(t)}_ik$ to $h^{(t+1)}_ik$ for each $k \in H_i$ as follows

$$
h^{(t+1)}_ik = h^{(t)}_ik - \frac{1}{2\tau_h(\sigma^{(t)}_g)} \frac{\partial J_h(h^{(t)}_ik)}{\partial h^{(t)}_ik},
$$

where

$$
\tau_h(\sigma^{(t)}_h) = \sum_t b_t |y_t| x_{it}^{h^{(t)}_i} \alpha(\theta^{(t)}_{ik}) \beta(\theta^{(t)}_{ik}) \sum_{\sigma(\theta^{(t)}_{ik})} \sigma(\theta^{(t)}_{ik}) \log^2 x_{it} + 2 \sum_t b_t^2 x_{it}^{h^{(t)}_i} \log^2 x_{it}
$$

and

$$
b_t = \beta^{(t)}_i \prod_{j \in H_i, j \neq k} x_{jt}^{g^{(t+1)/j=\phi+1}(j=\phi+2)}
$$

5. Compute

$$
\beta^{(t+1)}_i = \max \left\{ -\sum_t y_t d_t, \frac{\sum_t d_t}{\sum_t d_t^2}, \delta \right\}, \quad \text{where } d_t = \prod_{j \in H_i} x_{jt}^{g^{(t+1)}_j}
$$

6. Let $\theta^{(t+1)} = (\alpha^{(t+1)}_i, g^{(t+1)}_i, \beta^{(t+1)}_i, h^{(t+1)}_i)$ and check the convergence. If converged, output $\theta^{(t+1)}$, otherwise, $\ell \leftarrow \ell + 1$ and go to 2.
Algorithm 2 AFGCD for problem \((10)\)

1. Assign initial values to \(\theta^{(0)} = (\alpha^{(0)}_t, g^{(0)}_i, \beta^{(0)}_t, h^{(0)}_k)\) and set \(\ell = 0\).

2. Let \(y_t = -S_t - \beta^{(\ell)}_t \prod_{j \in H_i} \frac{h^{(\ell)}_{ik}}{y_{jt}^{(\ell)}}, t = 1, \ldots, n\) and cyclically update \(g^{(\ell)}_{ik}\) to \(g^{(\ell+1)}_{ik}\) for each \(k \in G_i\) as follows

\[
g^{(\ell+1)}_{ik} = \min \left( \max \left( g^{(\ell)}_{ik} - \frac{1}{2} \frac{\partial J_i(g^{(\ell)}_{ik})}{\partial g_{ik}} \cdot L, R \right), \frac{\partial J_i(g^{(\ell)}_{ik})}{\partial g_{ik}} \right),
\]

where \(L = \max(g^{(\ell)}_{ik} - \sigma^{(\ell)}_e, \kappa_{\text{min}})\), \(R = \min(g^{(\ell)}_{ik} + \sigma^{(\ell)}_e, \kappa_{\text{max}})\) and \(\tau_e(\sigma^{(\ell)}_e)\) is calculated in the same way as in Algorithm 1.

3. Compute

\[
\alpha^{(\ell+1)}_t = \max \left\{ -\frac{\sum_j y_{jt} p_{jt}}{\sum_i p_{it}}, \delta \right\}, \quad \text{where} \quad p_{it} = \prod_{j \in G_i} x_{jt}^{(\ell+1)}
\]

4. Let \(y_t = S_t - \alpha^{(\ell+1)}_t \prod_{j \in G_i} \frac{y_{jt}^{(\ell+1)}}{h^{(\ell)}_{ik}}, t = 1, \ldots, n\) and cyclically update \(h^{(\ell)}_{ik}\) to \(h^{(\ell+1)}_{ik}\) for each \(k \in H_i\) as follows

\[
h^{(\ell+1)}_{ik} = \min \left( \max \left( h^{(\ell)}_{ik} - \frac{1}{2} \frac{\partial J_i(h^{(\ell)}_{ik})}{\partial h_{ik}} \cdot L, R \right), \frac{\partial J_i(h^{(\ell)}_{ik})}{\partial h_{ik}} \right),
\]

where \(L = \max(h^{(\ell)}_{ik} - \sigma^{(\ell)}_h, \kappa_{\text{min}})\) and \(R = \min(h^{(\ell)}_{ik} + \sigma^{(\ell)}_h, \kappa_{\text{max}})\).

5. Compute

\[
\beta^{(\ell+1)}_t = \max \left\{ -\frac{\sum_j y_{jt} d_{jt}}{\sum_i d_{it}}, \delta \right\}, \quad \text{where} \quad d_{it} = \prod_{j \in H_i} x_{jt}^{(\ell+1)}
\]

6. Let \(\theta^{(\ell+1)} = (\alpha^{(\ell+1)}_t, g^{(\ell+1)}_i, \beta^{(\ell+1)}_t, h^{(\ell+1)}_{ik})\) and check the convergence. If converged, output \(\theta^{(\ell+1)}\), otherwise, \(\ell \leftarrow \ell + 1\) and go to Step 2.

**Proof**

\[
e^\ell - e^{\ell-1} = e^{\ell-1} \left( 1 + (h - h') + \frac{1}{2!} (h - h')^2 + \cdots + \frac{1}{n!} (h - h')^n + \cdots \right)
\]

\[
= e^{\ell-1} \left( \left( \frac{1}{2!} \cdots + \frac{1}{n!} (h - h')^n + \cdots \right) \right)
\]

\[
= e^{\ell-1} \left( (h - h')^2 + \frac{1}{\sigma^2} (e^\sigma - 1 - \sigma) \right).
\]

Then, the descent property of Algorithm 1 is shown in the following theorem.

**Theorem 1** The updating rules (6)–(9) in Algorithm 1 keep the objective function in problem (4) nonincreasing.

**Proof** First, we show that the updating rules (6) and (8) can be obtained by minimizing constructed auxiliary functions. Here, we only prove that the rule (8) and the rule (6) can be proved similarly.

Suppose in one iteration, we need to update the parameter \(h^{(\ell)}_{ik}\) to \(h^{(\ell+1)}_{ik}\) with all the other parameters fixed. Using the notations in Algorithm 1, the objective function is

\[
J_i(h_{ik}) = \frac{1}{2} \sum_i (y_i + b_i h^{(\ell)}_{ik})^2.
\]

Because other parameters except \(h_{ik}\) are fixed, \(J_i\) can be considered only depending on \(h_{ik}\). The following auxiliary function is proposed:

\[
F(h_{ik}, h^{(\ell)}_{ik}) = J_i(h^{(\ell)}_{ik}) + J_i(h^{(\ell)}_{ik})(h - h^{(\ell)}_{ik}) + \tau_h(h^{(\ell)}_{ik}) (h - h^{(\ell)}_{ik})^2.
\]

The definitions of \(\tau_h(\cdot)\) and \(\alpha^{(\ell)}_{h_{ik}}\) can be found in Algorithm 1. Obviously, \(F(h_{ik}, h^{(\ell)}_{ik}) = J_i(h_{ik})\). We only need to verify the first condition in Definition 1. Using Lemma 2, we have,
if \(|h_{ik} - \hat{h}_{ik}^{(\ell)}| \leq \sigma_h^{(\ell)}\)

\[ J_{\ell}(h_{ik}) = \frac{1}{2} \sum_t y_t^2 + \sum_t b_t y_t e^{\theta_h x_t} + \frac{1}{2} \sum_t b_t^2 e^{2\theta_h x_t} \]

\[ \leq \frac{1}{2} \sum_t y_t^2 + \sum_t b_t y_t e^{\theta_h x_t} + \frac{1}{2} \sum_t b_t^2 e^{2\theta_h x_t} \]

\[ + \sum_t b_t y_t e^{\theta_h x_t} (\log x_t)(h_{ik} - \hat{h}_{ik}^{(\ell)}) \]

\[ + \sum_t b_t^2 e^{2\theta_h x_t} (\log x_t)(h_{ik} - \hat{h}_{ik}^{(\ell)}) \]

\[ + \sum_t b_t |y_t| e^{\theta_h x_t} (\log^2 x_t)(h_{ik} - \hat{h}_{ik}^{(\ell)})^2 \]

\[ \Delta(\sigma_h^{(\ell)} | \log x_t) | \]

\[ + 2 \sum_t b_t^2 e^{2\theta_h x_t} (\log^2 x_t)(h_{ik} - \hat{h}_{ik}^{(\ell)})^2 \]

\[ \Delta(2\sigma_h^{(\ell)} | \log x_t) | \]

\[ = J(h_{ik}^{(\ell)}) + J'(h_{ik}^{(\ell)})(h - h_{ik}^{(\ell)}) + \tau_h(\sigma_h^{(\ell)}) \]

\[ \cdot (h_{ik} - h_{ik}^{(\ell)})^2, \]

where the function \(\Delta(\cdot)\) is defined in Equation (18). Therefore, \(F\) defined in Equation (20) is an auxiliary function. By Lemma 1, the minimization of \(F\) with respect to \(h_{ik}\) in the neighborhood of \(h_{ik}^{(\ell)}\) makes the auxiliary function \(J\) keep nonincreasing during the iterations. Thus, the updating rule is obtained by solving

\[ \arg\min_{h_{ik}} F(h_{ik}, h_{ik}^{(\ell)}). \quad (21) \]

Note that the auxiliary function (20) is quadratic and its global minimum is attained at

\[ h_{ik}^* = h_{ik}^{(\ell)} - \frac{1}{2\tau_h(\sigma_h^{(\ell)})} J'(h_{ik}^{(\ell)}), \quad (22) \]

Also note that

\[ |h_{ik}^* - h_{ik}^{(\ell)}| = \frac{|J'(h_{ik}^{(\ell)})|}{2\tau_h(\sigma_h^{(\ell)})} \]

\[ \leq \frac{|J'(h_{ik}^{(\ell)})|}{\sum_t b_t |y_t| x_t e^{\theta_h x_t} \log^2 x_t + 2 \sum_t b_t^2 x_t e^{2\theta_h x_t} \log x_t} \]

\[ = \sigma_h^{(\ell)}, \quad (23) \]

the inequality is due to that \(\Delta(\sigma) \geq \frac{1}{2}\) for all \(\sigma > 0\). Therefore, the updating rule is (22) or (8) and the objective function keeps nonincreasing during the iterations.

The updating steps (7) and (9) are obtained by simply solving an univariate least-squares problem with a simple constraint, whose objective function is quadratic and thus has an analytical solution. The derivation is simple and we omit it here. Therefore, the objective function keeps nonincreasing in every iterations of Algorithm 1.

\[ \blacksquare \]

For Algorithm 2, we have the same descent property.

**Theorem 2** The updating rules (11)–(14) in Algorithm 2 keep the objective function in problem (10) nonincreasing.

**Proof** Similar to Theorem 1, considering the updating of \(h_{ik}^{(\ell)}\), the auxiliary function is defined as in Equation (20). The updating rule is obtained by solving

\[ \arg\min_{h_{ik}} F(h_{ik}, h_{ik}^{(\ell)}) \]

\[ \text{subject to } h_{ik}^{(\ell)} - \sigma_h^{(\ell)} \leq h_{ik} \leq h_{ik}^{(\ell)} + \sigma_h^{(\ell)}, \quad (24) \]

\[ k_{\min} \leq h_{ik} \leq k_{\max}. \]

Since \(F\) is quadratic in \(h_{ik}\) and the constraints are very simple, analytical solution (13) exists. The remaining proof is similar to Theorem 1 and we omit it here.

\[ \blacksquare \]

### 3. Simulation examples

To study the performances of the proposed methods, we apply the algorithms to simulated data and compare the estimated parameters with the corresponding true ones.

#### 3.1. Performances of Algorithm 1

**3.1.1. Four-dimensional model**

Consider the following S-system of four molecular species (Voit, 2000):

\[
\begin{align*}
\dot{X}_1 &= 12X_4^{0.8} - 10X_1^{0.5}, \\
\dot{X}_2 &= 8X_1^{0.5} - 3X_2^{0.75}, \\
\dot{X}_3 &= 3X_2^{0.75} - 5X_3^{0.5}X_4^{0.2}, \\
\dot{X}_4 &= 2X_1^{0.5} - 6X_4^{0.9}.
\end{align*}
\]

The noise-free time series data are obtained by numerically solving the S-system with an initial condition \(X(0) = [X_{10}, X_{20}, X_{30}, X_{40}]^T\). The data are sampled at time points in the interval \([0, 5]\) with \(\Delta t = 0.1\).

In this example, the data are generated with \(X(0) = [10, 1, 2, 3]^T\). The time series data are shown in Figure 1, from which we can see all states of \(X_i\)'s are eventually in the steady states. Algorithm 1 is applied to estimating the parameters from these data with initial values for all parameters chosen by

\[ \theta_{\text{initial}} = \theta_{\text{true}}(1 + s\epsilon) \quad \text{for any } \theta \in \{\alpha_i, g_i, \beta_i, h_i\}, \quad (26) \]

where \(\epsilon\) is a standard Gaussian random variable and \(s\) is a positive constant. Since Equation (25) is a nonlinear optimization problem, to avoid falling into the local optimum,
we apply Algorithm 1 for 100 times initiated with different values and choose the one with minimum objective value as the final solution. Here, we set \( s = 90\% \). The objective values \( J_i \) of the first 100 iterations in one run are illustrated in Figure 2. It can be seen that the objective values decrease with the increase of iteration steps. Table 1 shows the estimated results, from which we can see that the estimated values are quite close to their true values and the optimal objective values are all very small.

### 3.1.2. Five-dimensional model

A benchmark five-dimensional model (Vilela et al., 2008; Yang, Dent, & Nardini, 2012) is considered,

\[
\begin{align*}
\dot{X}_1 &= 5X_3X_5^{-1} - 10X_1^2, \\
\dot{X}_2 &= 10X_1^2 - 10X_2^2, \\
\dot{X}_3 &= 10X_2^{-1} - 10X_2^{-1}X_3^2, \\
\dot{X}_4 &= 8X_3^2X_5^{-1} - 10X_4^2, \\
\dot{X}_5 &= 10X_4^2 - 10X_5^2.
\end{align*}
\] (27)

The data used in this example are generated with the initial condition \( X(0) = [0.1, 0.7, 0.7, 0.16, 0.18]^T \), the same as in Yang et al. (2012). Figure 3 shows the time series data that are sampled in the interval \([0, 0.5]\) with \( \Delta t = 0.01 \). Note that the states of all variables quickly converge to the steady state. Hence, only limited information on the dynamics of the system is contained in the data. We run Algorithm 1 for 100 times with different initial values obtained from

---

Table 1. Estimated results of the four-dimensional model from Algorithm 1.

| Parameter | True value | Estimation | Relative error (%) | Objective value |
|-----------|------------|------------|-------------------|-----------------|
| \( \alpha_1 \) | 12         | 11.9856    | 0.12              | \( 6.809 \times 10^{-4} \) |
| \( \beta_1 \) | 10         | 9.9819     | 0.18              | \( 3.650 \times 10^{-4} \) |
| \( g_{13} \) | -0.8       | -0.8022    | 0.27              |                 |
| \( h_{11} \) | 0.5        | 0.5013     | 0.25              |                 |
| \( \alpha_2 \) | 8          | 7.9857     | 0.18              |                 |
| \( \beta_2 \) | 3          | 2.9912     | 0.29              |                 |
| \( g_{21} \) | 0.5        | 0.5005     | 0.11              |                 |
| \( h_{22} \) | 0.75       | 0.7511     | 0.14              |                 |
| \( \alpha_3 \) | 3          | 3.0377     | 1.26              | \( 5.8635 \times 10^{-5} \) |
| \( \beta_3 \) | 5          | 5.0375     | 0.75              |                 |
| \( g_{32} \) | 0.75       | 0.7444     | 0.74              |                 |
| \( h_{33} \) | 0.5        | 0.4957     | 0.86              |                 |
| \( h_{34} \) | 0.2        | 0.1978     | 1.11              |                 |
| \( \alpha_4 \) | 2          | 1.9965     | 0.17              | \( 4.0889 \times 10^{-5} \) |
| \( \beta_4 \) | 6          | 5.9977     | 0.04              |                 |
| \( g_{41} \) | 0.5        | 0.5014     | 0.29              |                 |
| \( h_{44} \) | 0.8        | 0.8014     | 0.18              |                 |

---

Figure 2. Objective values over various numbers of iterations in Algorithm 1.

---

Figure 3. Time series data of the five-dimensional model.
In this example, Algorithm 1 is applied to estimate the parameters in the following six-dimensional S-system Yang et al. (2012):

\[
\begin{align*}
\dot{X}_1 &= 10X_1^2X_5 - 5X_1^{0.5}, \\
\dot{X}_2 &= 5X_1^{0.5} - 10X_2^{0.5}, \\
\dot{X}_3 &= 2X_2^{0.5} - 1.25X_3^{0.5}, \\
\dot{X}_4 &= 8X_2^{0.5} - 5X_4^{0.5}, \\
\dot{X}_5 &= 0.5 - X_6, \\
\dot{X}_6 &= X_5 - 0.5.
\end{align*}
\]  

Equation (26) with \(s = 80\%\) and select the one with minimum objective value as the solution. The results in Table 2 show the effectiveness of this algorithm: estimated values of parameters are close to the true values and the optimal objective values are all very small.

### 3.1.3. Six-dimensional model

In this example, Algorithm 1 is applied to estimate the parameters in the following six-dimensional S-system Yang et al. (2012):

\[
\begin{align*}
\dot{X}_1 &= 10X_1^2X_5 - 5X_1^{0.5}, \\
\dot{X}_2 &= 5X_1^{0.5} - 10X_2^{0.5}, \\
\dot{X}_3 &= 2X_2^{0.5} - 1.25X_3^{0.5}, \\
\dot{X}_4 &= 8X_2^{0.5} - 5X_4^{0.5}, \\
\dot{X}_5 &= 0.5 - X_6, \\
\dot{X}_6 &= X_5 - 0.5.
\end{align*}
\]  

The noise-free time series data are generated with the initial condition \(X(0) = [1.1, 0.5, 0.9, 0.75, 0.5, 0.75]^T\) which matches that in Yang et al. (2012). Figure 4 illustrates the time series data which are sampled in the interval \([0, 10]\) with \(\Delta t = 0.1\). Figure 4 also shows the periodic oscillating behaviour of the data. The initial values are also chosen by Equation (26) with \(s = 50\%\). The solution shown in Table 3 is the best one among 100 runs of Algorithm 1 with different initial values. The results in Table 3 indicate that the estimated parameters are close to their true values and the optimal objective values are all very small.

### 3.2. Performances of Algorithm 2

The performances of Algorithm 2 is studied in this example. We apply the algorithm to the data generated from the four-dimensional, five-dimensional, and six-dimensional models, respectively. The time series data we use are exactly the same as above-mentioned examples. Since Algorithm 2 solves the problem (10) which includes a range constraint for each parameter, the initial values for the algorithm are randomly generated in those ranges. In this study, the rate
Table 4. Estimated results of the four-dimensional model from Algorithm 2.

| Parameter | True value | Estimation | Relative error (%) | Objective value |
|-----------|------------|------------|--------------------|-----------------
| $\alpha_1$ | 10         | 9.9288     | 0.7119             | $8.2968 \times 10^{-4}$ |
| $g_{13}$  | -0.8       | -0.807     | 0.8789%            |                  |
| $h_{11}$  | 0.5        | 0.5034     | 0.6858             |                  |
| $\beta_2$ | 3          | 2.9912     | 0.2931             |                  |
| $g_{21}$  | 0.5        | 0.5005     | 0.1093             |                  |
| $h_{22}$  | 0.75       | 0.7511     | 0.1411             |                  |
| $\alpha_3$| 3          | 3.0337     | 1.1233             | $7.9833 \times 10^{-5}$ |
| $\beta_3$ | 5          | 5.0487     | 0.8479             |                  |
| $g_{32}$  | 0.75       | 0.7435     | 0.8691             |                  |
| $h_{33}$  | 0.5        | 0.4937     | 1.26               |                  |
| $h_{34}$  | 0.2        | 0.1964     | 1.7893             |                  |
| $\alpha_4$| 2          | 2.0051     | 0.2528             | $1.991 \times 10^{-4}$ |
| $\beta_4$ | 6          | 5.9847     | 0.2551             |                  |
| $g_{41}$  | 0.5        | 0.4927     | 1.457              |                  |
| $h_{44}$  | 0.8        | 0.7938     | 0.7776             |                  |

Table 5. Estimated results of the five-dimensional model from Algorithm 2.

| Parameter | True value | Estimation | Relative error (%) | Objective value |
|-----------|------------|------------|--------------------|-----------------|
| $\alpha_1$| 5          | 4.7704     | 4.5918             | $3.1738 \times 10^{-4}$ |
| $\beta_1$ | 10         | 9.7686     | 2.3138             |                  |
| $g_{13}$  | 1          | 1.1535     | 15.3495            |                  |
| $g_{15}$  | -1         | -1.0424    | 4.2402             |                  |
| $h_{11}$  | 2          | 2.0652     | 3.2612             |                  |
| $\alpha_2$| 10         | 10.0133    | 0.1333             | $3.9712 \times 10^{-4}$ |
| $\beta_2$ | 10         | 9.9644     | 0.3563             |                  |
| $g_{21}$  | 2          | 2.0022     | 0.1112             |                  |
| $h_{22}$  | 2          | 1.9845     | 0.7763             |                  |
| $\alpha_3$| 10         | 8.1857     | 18.1435            | $9.5442 \times 10^{-4}$ |
| $\beta_3$ | 10         | 8.1839     | 18.1609            |                  |
| $g_{32}$  | -1         | -1.2361    | 23.6145            |                  |
| $h_{33}$  | -1         | -1.2363    | 23.6264            |                  |
| $h_{34}$  | 2          | 2.0232     | 11.6124            |                  |
| $\alpha_4$| 8          | 7.5565     | 5.5442             | $6.7221 \times 10^{-4}$ |
| $\beta_4$ | 10         | 9.5398     | 4.6023             |                  |
| $g_{43}$  | 2          | 2.1688     | 8.4413             |                  |
| $g_{45}$  | -1         | -1.0476    | 4.7607             |                  |
| $h_{44}$  | 2          | 2.0827     | 4.1342             |                  |
| $\alpha_5$| 10         | 9.9649     | 0.3505             | $5.6977 \times 10^{-4}$ |
| $\beta_5$ | 10         | 9.958      | 0.4204             |                  |
| $g_{54}$  | 2          | 2.0152     | 0.7598             |                  |
| $h_{55}$  | 2          | 2.0047     | 0.2345             |                  |

Table 6. Estimated results of the six-dimensional model from Algorithm 2.

| Parameter | True value | Estimation | Relative error (%) | Objective value |
|-----------|------------|------------|--------------------|-----------------|
| $\alpha_1$| 10         | 9.7512     | 2.4881             | $4.3511 \times 10^{-4}$ |
| $\beta_1$ | 5          | 4.9344     | 1.3118             |                  |
| $g_{13}$  | -2         | -1.9891    | 0.5461             |                  |
| $g_{15}$  | 1          | 0.9872     | 1.2845             |                  |
| $h_{11}$  | 0.5        | 0.4889     | 2.0172             |                  |
| $\alpha_2$| 5          | 4.6623     | 6.7549             | $3.3703 \times 10^{-4}$ |
| $\beta_2$ | 10         | 9.6115     | 3.8853             |                  |
| $g_{21}$  | 0.5        | 0.5221     | 4.4167             |                  |
| $h_{22}$  | 0.5        | 0.5218     | 4.3684             |                  |
| $\alpha_3$| 2          | 1.9983     | 0.084              | $4.0053 \times 10^{-6}$ |
| $\beta_3$ | 1.25       | 1.2484     | 0.1248             |                  |
| $g_{32}$  | 0.5        | 0.5005     | 0.0925             |                  |
| $h_{33}$  | 0.5        | 0.5005     | 0.0912             |                  |
| $\alpha_4$| 8          | 7.9848     | 0.1896             | $1.2578 \times 10^{-4}$ |
| $\beta_4$ | 5          | 4.9926     | 0.1487             |                  |
| $g_{42}$  | 0.5        | 0.4996     | 0.0864             |                  |
| $h_{44}$  | 0.5        | 0.4995     | 0.0953             |                  |
| $\alpha_5$| 0.5        | 0.4991     | 0.1864             | $1.2022 \times 10^{-7}$ |
| $\beta_5$ | 1          | 0.9994     | 0.0609             |                  |
| $g_{56}$  | 1          | 1.002      | 0.1989             |                  |
| $\alpha_6$| 1          | 0.9977     | 0.2326             | $2.2004 \times 10^{-6}$ |
| $\beta_6$ | 0.5        | 0.496      | 0.8035             |                  |
| $g_{65}$  | 1          | 1.0091     | 0.9094             |                  |

constants are restricted in the range $[0.1, 10]$ and kinetic orders are in the range $[-2, 3]$. For each model, we run the algorithm for 100 times with different initial values and choose the one with minimum objective value as the final result. The estimated results from Algorithm 2 for each model are reported in the Tables 4–6. From the results, it can be seen that estimated parameters for the four-dimensional and six-dimensional models have very small relative errors compared with their corresponding true values. For the five-dimensional model, most parameters are estimated with low errors, while the estimations of the parameters in the third ODE have relatively large errors. This may be due to the limited information contained in the time series data. We can also see that the estimated objective function values for these three models are all very small. All these results show the effectiveness of the proposed algorithm.

4. Conclusions

The S-system is an effective mathematical model to characterize and analyse the molecular biological systems. The parameters in S-systems have significant biological meanings. However, the estimation of these parameters from time series biological data is non-trivial, because of the nonlinearity and complexity of this model. A novel method, the AFGCD, is proposed in this study to estimate the parameters of S-systems. To solve the nonlinear optimization problem involved in the parameter estimation, the proposed method optimizes one parameter at a time. Taking advantage of a property of the exponential function, an auxiliary function is constructed for each kinetic order parameter. We prove that updating the parameter value by optimizing the auxiliary function keeps the objective function nonincreasing. As the auxiliary function is quadratic, the analytical solution exists and therefore the updating rule for each parameter is simple and efficient. Based on this idea, two algorithms are developed: one estimates the parameters in S-system with the
only non-negative constraints on rate constants; the other puts constraints both on rate constants and kinetic orders. Their performances are studied in simulation examples, which show that the estimated parameter values from the proposed algorithms are very close to the corresponding true values and the optimized objective functions has very low values. All of these results demonstrate the effectivenesses of the proposed algorithms.

In this study, the topology information of the system is assumed to be known. One direction of the future work is to extend the current method with structure identification method to infer the S-system without knowing the system topology.

Acknowledgements
This study is supported by Natural Science and Engineering Research Council of Canada (NSERC).

References
Bazaraa, M. S., Sherali, H. D., & Shetty, C. M. (2006). *Nonlinear programming: Theory and algorithms*. Hoboken, NJ: Wiley-Interscience. ISBN 0471486000.

Beck, J. V., & Arnold, K. J. (1977). *Parameter estimation in engineering and science*. New York: Wiley.

Bertsekas, D. P. (1999). *Nonlinear programming*. Belmont, MA: Athena Scientific. ISBN 1886529000.

Chou, I.-C., Martens, H., & Voit, E. (2006). Parameter estimation in biochemical systems models with alternating regression. *Theoretical Biology and Medical Modelling, 3*(1), 25. ISSN 1742-4682. doi:10.1186/1742-4682-3-25

Gardner, T. S., di Bernardo, D., Lorenz, D., & Collins, J. J. (2003). Inferring genetic networks and identifying compound mode of action via expression profiling. *Science, 301*(5629), 102–105. doi:10.1126/science.1081900

Gonzalez, O. R., Kper, C., Jung, K., Naval, P. C., & Mendoza, E. (2007). Parameter estimation using simulated annealing for S-system models of biochemical networks. *Bioinformatics, 23*(4), 480–486. doi:10.1093/bioinformatics/btl522

Ho, S.-Y., Hsieh, C.-H., Yu, F.-C., & Huang, H.-L. (2007). An intelligent two-stage evolutionary algorithm for dynamic pathway identification from gene expression profiles. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on, 4*(4), 648–704. ISSN 1545-5963. doi:10.1109/tcbb.2007.1051

Kikuchi, S., Tominaga, D., Arita, M., Takahashi, K., & Tomita, M. (2003). Dynamic modeling of genetic networks using genetic algorithm and S-system. *Bioinformatics, 19*(5), 643–650. doi:10.1093/bioinformatics/btg027

Klipp, E., Herwig, R., Kowald, A., Wierling, C., & Lehrach, H. (2005, May). *Systems biology in practice: Concepts, implementation and application*. Weinheim: Wiley-VCH.

Li, L.-X., Wu, L., Zhang, H.-S., & Wu, F.-X. (2013). A fast algorithm for nonnegative matrix factorization and its convergence. *Neural Networks and Learning Systems, IEEE Transactions, PP*(99), 1. doi:10.1109/TNNLS.2013.2296627

Liu, L.-Z., Wu, F.-X., & Zhang, W.-J. (2012a). Alternating weighted least squares parameter estimation for biological s-systems. 2012 IEEE 6th international conference on systems biology (ISB), Xi’an, China, pp. 6–11. doi:10.1109/ISB.2012.6314104

Liu, L.-Z., Wu, F.-X., & Zhang, W. J. (2012b). Inference of biological s-system using the separable estimation method and the genetic algorithm. *IEEE/ACM Transactions on Computational Biology and Bioinformatics, 9*(4), 955–965. ISSN 1545-5963. doi:10.1109/TCBB.2011.126

Seung, D., & Lee, L. (2001). Algorithms for non-negative matrix factorization. *Advances in Neural Information Processing Systems, 13*, 556–562.

Voit, E. O. (2000, September). Computational analysis of biochemical systems: A practical guide for biochemists and molecular biologists. Cambridge, UK: Cambridge University Press. ISBN 0521785790.

Voit, E. O., & Almeida, J. (2004). Decoupling dynamical systems for pathway identification from metabolic profiles. *Bioinformatics, 20*(11), 1670–1681. doi:10.1093/bioinformatics/bth140

Wang, H., Qian, L., & Dougherty, E. (2010, March). Inference of gene regulatory networks using S-system: A unified approach. *Systems Biology, IET, 4*(2), 145–156. ISSN 1751-8849. doi:10.1049/iet-syb.2008.0175

Wu, F.-X., & Mu, L. (2009, June). Separable parameter estimation method for nonlinear biological systems. ICBBE 2009. 3rd international conference on bioinformatics and biomedical engineering, Beijing, China, 2009, pp. 1–4. doi:10.1109/ICBBE.2009.5163379

Yang, X., Dent, J. E., & Nardini, C. (2012). An S-system parameter estimation method (SPEM) for biological networks. *Journal of Computational Biology, 19*(2), 175–187. doi:10.1089/cmb.2011.0269