Evolutionary optimization with data collocation for reverse engineering of biological networks
Kuan-Yao Tsai and Feng-Sheng Wang*
Department of Chemical Engineering, National Chung Cheng University, Chia-yi 621-02, Taiwan
Received on July 19, 2004; revised on September 30, 2004; accepted on October 15, 2004
Advance Access publication October 28, 2004

ABSTRACT
Motivation: Modern experimental biology is moving away from analyses of single elements to whole-organism measurements. Such measured time-course data contain a wealth of information about the structure and dynamic of the pathway or network. The dynamic modeling of the whole systems is formulated as a reverse problem that requires a well-suited mathematical model and a very efficient computational method to identify the model structure and parameters.
Numerical integration for differential equations and finding global parameter values are still two major challenges in this field of the parameter estimation of nonlinear dynamic biological systems.
Results: We compare three techniques of parameter estimation for nonlinear dynamic biological systems. In the proposed scheme, the modified collocation method is applied to convert the differential equations to the system of algebraic equations. The observed time-course data are then substituted into the algebraic system equations to decouple system interactions in order to obtain the approximate model profiles. Hybrid differential evolution (HDE) with population size of five is able to find a global solution. The method is not only suited for parameter estimation but also can be applied for structure identification. The solution obtained by HDE is then used as the starting point for a local search method to yield the refined estimates.
Contact: chmfsw@ccu.edu.tw

INTRODUCTION
Mathematical modeling and dynamic simulation of genetic networks, metabolic networks and signal transduction cascades is a central theme in systems biology and is increasingly attracting attention in post-genomic era (Voit, 2002). The current interest is not due to particular biological breakthroughs facilitated by mathematical models but is largely based on the promise that signal- and systems-oriented approaches have. The ultimate goal of mathematical modeling is to obtain expressions that quantitatively understand every detail and principle of biological systems. Building a mathematical model comprises mainly two aspects: (1) deciding on the model structure and (2) estimating the involved parameter values. Given a model structure, parameter estimation remains the limiting step in the modeling and simulation of biological systems. There have been in-depth studies regarding parameter estimation for nonlinear dynamic systems (Kuzmic, 1996; Mendes and Kell, 1998) and for steady-state systems (Jacquez and Perry, 1990; Hernandez and Ruiz, 1998). However, there still remain difficulties due to the measurement noise and according to the increasing number of parameters to be estimated (Kimura et al., 2004).

Various in silico models have been proposed to quantitatively describe the dynamic behaviors of biological systems. A serious bottleneck of in silico modeling in the past has often been the lack of sufficient data for parameter estimation. The advent of comprehensive gene expression data and metabolic profiles provides new means of parameter estimation (Moles et al., 2003; Voit, 2002). The estimation of parameter values from time-course data is a part of regression problems. There are two major challenges in this field of the parameter estimation of nonlinear dynamic biological systems. The first challenge is to develop an efficient optimization method to find the global estimates. Several conventional methods, such as the graphical method and the gradient-based nonlinear optimization methods (Mendes and Kell, 1998), have been employed as estimation techniques to obtain the parameter values. The graphical method can be applied only to those problems that can be converted to linear regression problems. Basically, a gradient-based nonlinear optimization method does not have such a restriction, but it requires gradient information about the error function with respect to the parameter estimates and often converges to local minima. Evolutionary algorithms which include genetic algorithms, evolutionary programming, evolution strategies, genetic programming and their variants (Back et al., 1997; McKay et al., 1997; Edwards et al., 1998) can be applied to overcome such drawbacks.

The second challenge is to develop a powerful numerical integration method to solve differential equations. In conventional parameter estimation, it requires numerical integration to yield dynamic profiles for the model. Numerical integration failure is the major problem in the optimization search. In addition, numerical integration is time consuming. Almeida and Voit (2003) have applied an artificial neural network to smooth the measured profile data for obtaining slopes of the response curves. Such slopes are then employed in the reverse problem as the measured data to compare with the slopes of the model so that such an approach avoids the numerical integration of differential equations.

In this study, hybrid differential evolution (HDE) is applied as a global search to determine a satisfied solution (Chiu and Wang, 1999). Such satisfied estimates are then provided as the starting point for a gradient-based optimization method to obtain the refined
solution. To avoid numerical integration, we will introduce a modified collocation method (Wang, 2000) to directly use the measured dynamic data in order to yield the approximate model profiles for evaluating the error criterion of parameter estimation problems.

**METHOD**

**Estimation technique**

We are interested in dynamics of biological systems at the level of pools of molecular species. Such dynamics can be described mathematically by the S-system as follows (Voit, 2000):

\[
\dot{X} = f(X, p)
\]

where \(X\) is \(n\)-dimensional components or pools, the parameter vector \(p\) consists of rate constants, \(\alpha_j\) and \(\beta_j\), and kinetic orders, \(g_j\) and \(h_j\). \(f\) is the set of net rate equations, which consists of the influxes and the effluxes. The \(m\)-dimensional independent variables in the S-system equations are expressed as \(X_{e1}, \ldots, X_{em}\). The parameter estimation is to determine model parameters so that the dynamic profiles satisfactorily fit the measured observation. Figure 1 shows the concepts of conventional technique of parameter estimation. As shown in the dash box of Figure 1, we provide some adjustable inputs to perform the experiment in order to obtain measurable output data. Such inputs are also provided for the proposed model to compute the system output data. Meanwhile, an optimization method numerically generates model parameters in order to provide for the model. Differential equations including these model parameters are numerically solved to yield dynamic profiles. Difference between the measured data and the computed dynamic profiles becomes a least squared error and is expressed in the form

\[
J_k = \frac{1}{N_t} \sum_{i=1}^{N_t} \sum_{j=1}^{N_s} \frac{(X_{ei(t_j)} - X_{i(t_j)})^2}{X_{i,max}^2}, \quad k = 1, \ldots, N_{exp}
\]

where \(X_{ei(t_j)}\) is the measured data for the \(i\)-th component at \(t = t_j\); \(X_{i(t_j)}\) is the computed concentration for the \(i\)-th component at \(t = t_j\) and \(X_{i,max}\) are the maximum measured concentration of the \(i\)-th component. \(N_t\) is the number of the sampling data, and \(N_{exp}\) is the number of experiments. A least squared regression sums up every observed error in (2) to yield a criterion in order to inspect how the experimental data are close to the computed profiles. If the error criterion is not satisfied, then new model parameters are generated by the optimization algorithm, and repeat the procedures until a satisfied result is obtained.

There is no general solution guaranteed to solve the problem in nonlinear parameter estimation. Voit and Almeida (2004) have proposed a strategy for improving efficiency to avoid the need for solving the differential equations. As shown in Figure 2, an artificial neural network is applied to smooth the dynamic data in order to yield the approximate model profiles for evaluating the error criterion of parameter estimation problems. V oit and Almeida (2004) have proposed a strategy for improving efficiency to avoid the need for solving the differential equations. Such smooth data provide for the slope evaluation and approximation procedure. The modified collocation method is applied to convert differential equations into algebraic equations. Such algebraic equations directly adopt the measured data to approximately yield dynamic profiles to evaluate the error criterion.

In collocation methods, dynamic profiles \(X\) in equation (1) are spanned by a set of shape functions as

\[
X(t) = \sum_{j=0}^{N_s} x(j) \phi_j(t)
\]

where \(x(j)\) is an expansion coefficient of \(X(t)\) and \(\phi_j(t)\) is a set of polynomial shape function. Conventional collocation methods on the whole-time domain...
and is equal to the solution $X$ where $x_j$ is the $j$-th component of $X$.

Collocation equations are therefore expressed in the form

$$ f(x(j), p) = 0 $$

where $x(j)$ is a vector of expansion coefficients at the $j$-th collocation point and $f(x(j), p)$ is a vector function of expansion coefficients at the $j$-th collocation point. $p$ is the vector of decision parameters.

The solution for the dynamic system (1) yields the expansion coefficients by solving the implicit algebraic equations. However, such recursively solving procedures may spend most of computational burden.

Approximated profiles can be applied to circumvent these recursively solving procedures in order to reduce computational time. If we substitute the measured data at each collocation point into the right-hand side of equation (5), the approximated data can be expressed as

$$ X(t_j) = X_{e}(t_j) + f_1(X(t_j), p) $$

where $X(t_j)$ is the measured output at time $t_j$, and $X_{e}(t_j)$ is the estimated output at time $t_j$.

The modified collocation method redefined that the weighted residual criterion to yield expansion coefficients. For the whole-time domain approach, the accuracy and efficiency largely depend on the degree of shape polynomials. In addition, more computational efforts are required to solve the residual equation using the higher degree polynomials. Collocation on finite elements can be applied to overcome such drawbacks (Zienkiewicz and Morgan, 1984). By using conventional collocation on finite elements, additional equality constraints are required to include in the residual equations for enforcing continuity of the solution profiles at each element boundary. Wang (2000) has proposed the modified collocation method to leave out such a requirement.

The modified collocation method redefined that the weighted residual criterion is based on the integral expression of the dynamic system. As a result, the equality constraints are automatically satisfied for the modified collocation method so that we can derive the unified formulas for computing the expansion coefficients element by element. Here, we use the simplest shape function, linear Lagrange polynomial, to illustrate how to apply the modified collocation equation for parameter estimation problems. The modified collocation equations are therefore expressed in the form

$$ x(j) = x(j-1) + 0.5\eta_j \left[ f(x(j), p) + f(x(j-1), p) \right], \quad j = 1, \ldots, N $$

where $x(j)$ is a vector of expansion coefficients at the $j$-th collocation point and is equal to the solution $X(t)$ at time $t = t_j$. $f(x(j), p)$ is a vector function of expansion coefficients at the $j$-th collocation point. $\eta_j$ is the time interval between the $j$-th collocation point and the $(j-1)$-th point and $N$ is number of collocation points.

The reverse problem is thus reformulated from one involving $n$ differential equations to a larger system of $n \times N$ decoupled algebraic equations. The reformulation of the system of differential equations as a system of algebraic equations has the obvious advantage of making numerical integration unnecessary, but it has other very beneficial consequences as similar to Voi and Almeida (2004).

**Optimization technique**

We apply HDE to the estimation problem to globally determine a satisfied result, and then the estimates are used as the starting point for a gradient-based optimization method to obtain a finer solution. The HDE algorithm is a simple population-based, stochastic function method and has extended from the original algorithm of differential evolution (DE) as introduced by Storn and Price (1996, 1997). The basic operations of DE are similar to the conventional evolutionary algorithms as shown in Table 1. The basic operations of HDE are also shown in Table 1 for comparison.

The DE and HDE structure is a parallel direct search algorithm which utilizes $N_p$ vectors of the decision parameters in the estimation problem as a population for each generation. The initial population is randomly selected and should attempt to cover the entire search space uniformly. The mutation of DE and HDE is the essential ingredient, compared with the other evolutionary algorithms. The mutation in DE and HDE uses the difference between two randomly chosen individuals as a search direction. Such a mutation is different from the conventional evolutionary algorithms. A mutant individual is yielded from a parent individual and the perturbed mutation. In DE, a fixed mutation factor multiplies the difference vector to yield the perturbed mutation. In HDE, a random mutation factor is applied to obtain more diversified individuals. The crossover operation in DE and HDE is employed to increase the local population diversity, which is similar to the conventional evolutionary algorithms.

In DE and HDE, the fitness of an offspring competes one to one with that of its parent. This competition, which is also different from the conventional evolutionary algorithms, gives rise to a faster convergence rate. However, this faster convergence also leads to a higher probability of obtaining a local optimum since the population diversity descends faster during the solution progress. This drawback can be overcome by using a larger population size, although much computation time is expended to evaluate the fitness function. This fact is particularly serious when using DE to solve optimal control problems (Wang and Chiou, 1997). The HDE can avoid using a larger population size and has been applied to solve bioreaction process optimization problems (Wang et al., 1998; Wang and Sheu, 2000).

When using an evolutionary algorithm to optimize a function, an acceptable trade-off between convergence and population diversity must be generally determined. Convergence implies a fast convergence even though it may be
to a local optimum. On the other hand, the population diversity guarantees a high probability of obtaining the global optimum. When the population diversity is small, the candidate individuals are closely clustered. Therefore, the mutation and crossover operations can no longer generate the next better individual because a premature solution is obtained. In the HDE, an accelerated operation and a migration are embedded in the original DE algorithm, and these two operations act as a trade-off operator. The accelerated operation is used to speed up convergence. According to our experience, by using DE to solve optimization problems, the best fitness does not descend continuously from generation to generation. It usually descends toward a better fitness after several generations. Under this situation, the acceleration can be used to speed up convergence. When the best fitness in the present generation is not improved any longer by the mutation and crossover operations, a descent method is then applied to push the best individual for obtaining a better solution.

The rate of convergence can be improved by the acceleration. However, faster descent usually results in finding a local minimum. In addition, performing this operation frequently can make the candidate individuals gradually cluster around the best individual so that the population diversity decreases. Furthermore, the closely clustered individuals cannot reproduce better individuals by mutation and crossover operations. As a result, a migration operation is switched on to escape this local cluster. The new candidate individuals are regenerated on the basis of the best individual in the current generation. Correspondingly, the diversity of the candidates can be retained by using such a regeneration procedure. The migration in HDE is performed only if a measure of the population diversity fails to satisfy the desired tolerance. Chiou and Wang (1999) proposed a measure called the degree of population diversity to check whether the migration should be switched on.

The degree of population diversity is between 0 and 1. A zero value implies that all of the genes are clustered around the best individual. On the other hand, the value of 1 indicates that the current candidate individuals are a completely diversified population. The desired tolerance for population diversity is accordingly assigned within this region. A zero tolerance implies that the migration in HDE is switched off. A tolerance of 1 implies that the migration is performed in every generation.

RESULTS

In order to examine the effectiveness of the proposed techniques, we have applied the method to determine model parameters for a cascaded pathway and a small-scale genetic network.

Cascaded pathway

Cascaded mechanisms are found in diverse areas of biochemistry and physiology, including hormonal control, gene regulation, immunology, blood clotting and visual excitation. Purely biochemical examples are the phosphorylase activation cascade in muscle and the bicyclic glutamine synthase cascade. In this example, we consider the "true" cascade system shown in Figure 4. The system is described in the S-system equations as follows:

\[
\begin{align*}
\dot{X}_1 &= \alpha_1 X_4^{\beta_1} X_3^{\beta_3} - \beta_1 X_1^{\beta_1} \\
\dot{X}_2 &= \alpha_2 X_1^{\beta_1} - \beta_2 X_2^{\beta_2} \\
\dot{X}_3 &= \alpha_3 X_2^{\beta_2} - \beta_3 X_3^{\beta_3}
\end{align*}
\]

(7)

where \(X_4\) is the precursor, which is considered as the independent variable in the experiment. We first generated 10 sets of random initial conditions for dependent variables within the range \([0.1, 0.6]\) and the precursor within \([0.6, 0.9]\) to generate 40 "true" time-course data. The "true" kinetic orders and rate constants are shown in Case I of Table 2.

All computations were performed on a Pentium IV computer using Microsoft Windows 2000. The HDE algorithm is implemented in Compaq Visual Fortran. The user has to provide four setting factors for HDE. The setting factors used for all runs in the case studies are listed as follows. The crossover factor is set to 0.5. Two tolerances used in the migration are set to 0.05. The population size of 5 is used in the computation. In HDE, the mutation factor is taken as a random number in \([0, 1]\). The range used for the parameter estimation is \(\alpha_i\) and \(\beta_i \in [0.0, 20.0]\) and \(g_i, f_i, \beta_i \in [−4.0, 4.0]\).

The HDE with 50,000 generations is used to solve the estimation problem. This computation uses the total function calls of 314,022 including the migration operations of 189 and acceleration operations of 2,663. The computational time required for the global search about 2.3 min on a single-CPU Pentium IV 2.4 GHz. The best objective function value, denoted as sum of squared errors for all state variables and experiments, is 1.373E−3. In order to yield more accurate estimates, the solution obtained by HDE is employed as the starting point for a gradient-based method, a subroutine BCONF in IMSL Math/Library, to solve the parameter estimation problem. The local search procedure, as shown in Figure 1, employs Runge–Kutta pairs of various orders, a subroutine IVMRK in IMSL Math/Library, to solve differential equations for obtaining time-course profiles of the system. The refined objective function value of 2.978E−5 is obtained. The estimates are listed in Case II of Table 2. The computational time required for the local search about 4.67 min on a single-CPU Pentium IV 2.4 GHz. The local search is able to obtain more accurate solution, but it spends much computational time on numerical integration. The HDE with 100,000 generations was also applied to solve the estimation problem. The computational time required for the global search was about 6.6 min to obtain the best objective function value of 8.707E−5. After that, the local search spent about 2.02 min to yield the refined objective function value of 4.103E−6.

We applied DE with population size of 5 to solve the estimation problem for comparison. In the original DE algorithm, the mutation factor is fixed and provided by a user. We tried various pairs of mutation and crossover factors for the original DE to solve the problem. However, we always obtained a premature solution using the fixed mutation factor. We then used the random mutation factor at every generation. We still obtained the premature objective function value of 1.811. This solution was unsatisfactory by using the five-population size because of mutation and crossover operations in DE being vanished. Moreover, we were unable to find a converged result when the solution was applied as the starting point for the local search because of a numerical integration failure. A satisfied solution can be obtained by using DE with the population size of 30. The computational time...
required for the global and local search is ~15.83 min to achieve the best objective function value of 5.522E−4.

A simple genetic algorithm was also applied to solve the estimation problem. We used the binary coding for the variables with the population size of 100 to solve the estimation problem. However, we were unable to find a satisfied solution. In this work, we then applied the improved genetic algorithm (IGA), which embedded the acceleration and migration of the HDE algorithm into the GA algorithm (Chiang et al., 2002), to inspect the computation effect. The best objective function value of 5.495E−4 was obtained by IGA and local search by spending a total CPU processing time of 10.62 min. This effect indicates that the acceleration and migration operations could be indeed applied to improve the solution quality.

We applied the HDE global search using slope approximation, as shown in Figure 2, to evaluate error criterion. The computational time required for the global search was ~2 min. The best slope error criterion for state variables and experiments was 8.806E−2. The solution was then assigned as the starting point for the local search method including Runge–Kutta integration to refine the estimates. The refined objective function value of 7.041E−4 was obtained by the local search with a CPU processing time of 8.03 min. The estimates are listed in Case III of Table 2. The estimates are clearly different from the true values. For instance, $g_{13}$ is positive, which implies that $X_3$ is activating the synthesis of $X_1$ rather than inhibiting. This situation is similar to the result in Voit (2000). Nevertheless, the fitness value of 7.041E−4 is larger than the proposed method but acceptable.

Realistic experimental data are, in general, subjective natural variations and uncertainties. Such data require to be smoothed by a filter, such as artificial neural network, in order to yield noise-free data for providing parameter estimation. Accordingly, such an estimation quality should depend on the employed filter. In order to evaluate robustness of the proposed method, we suppose that very rough filtered data, which the previous experimental data are included with random noise, are directly applied for the modified collocation block in Figure 3 to approximate the dynamic profiles. The global HDE search with 50,000 generations is then used to solve the estimation problem for obtaining the best objective function value of 8.50E−2.

Two test experiments were used to validate fitness for these estimates. The initial conditions and independent variable for the first test experiment are $[0.5, 0.2, 0.5, 0.9]$ and $[0.72, 0.08, 0.72, 1.05]$, respectively. The solid, dash and dotted curves are the computed profiles using the estimates for Cases II, III and IV in Table 2, respectively. The dash-dot and dash-dot-dot curves are the computed profiles using the estimates for Cases III and IV in Table 3, respectively.

The refined objective function value of 1.637E−3 is obtained by the local search technique. The estimates are listed in Case IV of Table 2. The parameters, $g_{12}$ and $g_{13}$, are identical inhibition effect to the true system.

Fig. 5. Model validation using various estimates. The triangle and circle data points are the “true” dynamic profiles using the initial conditions for $[0.5, 0.2, 0.5, 0.9]$ and $[0.72, 0.08, 0.72, 1.05]$, respectively. The solid, dash and dotted curves are the computed profiles using the estimates for Cases II, III and IV in Table 2, respectively. The dash-dot and dash-dot-dot curves are the computed profiles using the estimates for Cases III and IV in Table 3, respectively.

The parameters, $g_{12}$ and $g_{13}$, are identical inhibition effect to the true system.
Evolutionary optimization with data collocation

Table 3. Sequential identification for parameter values and regulation structure of the cascaded pathway

<table>
<thead>
<tr>
<th>Case</th>
<th>Variable</th>
<th>(a_i)</th>
<th>(g_{i1})</th>
<th>(g_{i2})</th>
<th>(g_{i3})</th>
<th>(g_{i4})</th>
<th>(b_{i1})</th>
<th>(b_{i2})</th>
<th>(b_{i3})</th>
<th>(b_{i4})</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>(X_1)</td>
<td>13.671</td>
<td>−0.02731</td>
<td>−0.02743</td>
<td>0.4732</td>
<td>9.7694</td>
<td>0.2344</td>
<td>0.02974</td>
<td>−1.569E−5</td>
<td>−6.339E−3</td>
</tr>
<tr>
<td></td>
<td>(X_2)</td>
<td>15.673</td>
<td>0.06289</td>
<td>−7.953E−7</td>
<td>3.873E−5</td>
<td>15.067</td>
<td>−1.450E−4</td>
<td>0.05841</td>
<td>2.676E−5</td>
<td>6.460E−6</td>
</tr>
<tr>
<td></td>
<td>(X_3)</td>
<td>0.9672</td>
<td>1.308E−3</td>
<td>0.4917</td>
<td>1.309E−5</td>
<td>4.8591</td>
<td>−0.05101</td>
<td>−0.3890</td>
<td>0.9562</td>
<td>−5.308E−5</td>
</tr>
<tr>
<td>II</td>
<td>(X_1)</td>
<td>11.827</td>
<td>−0.05829</td>
<td>−0.03894</td>
<td>0.5555</td>
<td>8.1013</td>
<td>0.2745</td>
<td>9.317E−4</td>
<td>1.187E+0</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(X_2)</td>
<td>0.8072</td>
<td>0.9895</td>
<td>1.1206</td>
<td>0.3374</td>
<td>6.8355</td>
<td>−1.232E−5</td>
<td>−1.435E−8</td>
<td>1.1302</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(X_3)</td>
<td>0.9524</td>
<td>1.1206</td>
<td>0.3374</td>
<td>6.8355</td>
<td>7.3485</td>
<td>0.123</td>
<td>8.6439</td>
<td>0.8649</td>
<td></td>
</tr>
<tr>
<td>III</td>
<td>(X_1)</td>
<td>10.751</td>
<td>−0.05962</td>
<td>−0.04931</td>
<td>0.5961</td>
<td>7.2252</td>
<td>0.297</td>
<td>7.3485</td>
<td>0.123</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(X_2)</td>
<td>7.9426</td>
<td>0.1371</td>
<td>0.8569</td>
<td>6.4389</td>
<td>6.2996</td>
<td>0.4282</td>
<td>6.2996</td>
<td>0.4282</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(X_3)</td>
<td>1.4269</td>
<td>0.8569</td>
<td>6.4389</td>
<td>7.1858</td>
<td>0.1304</td>
<td>0.1304</td>
<td>7.1858</td>
<td>0.1304</td>
<td></td>
</tr>
<tr>
<td>IV</td>
<td>(X_1)</td>
<td>11.591</td>
<td>−0.09788</td>
<td>−0.03566</td>
<td>0.8506</td>
<td>0.04931</td>
<td>0.5961</td>
<td>6.2996</td>
<td>0.4282</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(X_2)</td>
<td>7.9426</td>
<td>0.1371</td>
<td>0.8569</td>
<td>7.1858</td>
<td>0.1304</td>
<td>0.1304</td>
<td>7.1858</td>
<td>0.1304</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(X_3)</td>
<td>1.4454</td>
<td>0.8867</td>
<td>6.7823</td>
<td>0.8745</td>
<td>0.8745</td>
<td>0.8745</td>
<td>0.8745</td>
<td>0.8745</td>
<td></td>
</tr>
</tbody>
</table>

The value with underline indicates that the parameter is deleted at the next identification procedure. Parameter values of Case IV are the minimum solution obtained by the local search, which uses the starting point yielded from Case III.

The proposed global/local search method is not only suited for parameter estimation but also applies for structure identification. A priori knowledge about the biological systems is introduced into the reverse problem in order to determine a correct solution. The pruning term introduced by Kikuchi et al. (2003) is used as the penalty into the objective function to detect a suitable skeletal structure. In addition, if it is a priori known that \(X_1\) does not directly affect \(X_2\), then the corresponding kinetic orders are zero and this reduces the parameter space that needs to be searched by a full dimension (Almeida and Voit, 2003; Voit and Almeida, 2004). In this example, we set the kinetic orders \(g_{i1} = 0\), which precluded a direct effect of a variable on its own production and required the kinetic orders \(h_{ij}\) to be >0, indicating that the degradation of compounds almost always depends on the concentration. Under the assumption, the search range used for the regression is \(a_i\) and \(b_i\) \(\in [0.0, 20.0\), \(g_{i1}\) and \(h_{ij}\) \(\in [-4.0, 4.0\), \(i \neq j\) and \(h_{ij}\) \(\in [0.0, 4.0]\). The HDE algorithm with 100,000 generations was used to solve the estimation problem to yield the best objective function value of 1.382E−2. The optimal parameter values are shown in Case I of Table 3. We delete the smaller kinetic orders, (magnitude <0.01), as shown in Case II (underlined to zero). The pruned model is then refitted by the global search to obtain the optimal estimates as given in Case II of Table 3. Following similar procedures, we obtained the optimal solution in Case III of Table 3. Such a solution is essentially identical to the ‘true’ structure. The parameter values were then applied as the starting point for the local search method to yield the optimal objective function value of 4.358E−4. The optimal estimates are given in Case IV of Table 3. Total computational time for the reverse problem required for the global and local search was \(\sim 1.03\) h on a single-CPU Pentium IV 2.4 GHz. In order to validate the estimates for Case III and Case IV, both parameter values were applied to evaluate two test experiments. The computational results are shown in Figure 5 by dash-dot and dash-dot-dot curves for Cases III and IV, respectively. Both sets of model parameters can satisfactorily fit the test experiments.

Small-scale genetic network

The second case study is a small-scale genetic network shown in Figure 6 (Hlavacek and Savageau, 1996). This is a typical gene interaction system consisting of two genes (genes 1 and 4). The ‘true’ system is described in the S-system equations as follows:

\[
\begin{align*}
\dot{X}_1 &= a_1 X_1^{g_{11}} X_5^{g_{15}} X_6 - b_1 X_1^{h_{11}} \\
\dot{X}_2 &= a_2 X_2^{g_{21}} X_7 - b_2 X_2^{h_{22}} \\
\dot{X}_3 &= a_3 X_3^{g_{31}} X_8 - b_3 X_3^{h_{33}} X_3 \\
\dot{X}_4 &= a_4 X_4^{g_{41}} X_5^{g_{45}} X_6 - b_4 X_4^{h_{44}} \\
\dot{X}_5 &= a_5 X_5^{g_{52}} X_7 - b_5 X_5^{h_{55}}
\end{align*}
\]

where \(X_1\) is an mRNA produced from gene 1, \(X_2\) is an enzyme protein it produces and \(X_3\) is an inducer protein catalyzed by \(X_2\). \(X_4\) is an mRNA produced from gene 4 and \(X_5\) is a regulator protein it produces. Positive feedback from the inducer protein \(X_1\) and negative feedback from the regulator protein \(X_4\) are assumed in the mRNA production processes of genes 1 and 4. \(X_6\), \(X_7\) and \(X_8\) denote a pool of nucleic acid, amino acid and substrate, respectively, and are considered as independent variables in the system.

Following similar procedures as discussed in the previous case study, we applied several methods to determine the kinetic parameters of the genetic network. Those results are summarized in the supplements. The optimal solution obtained by HDE using the modified collocation approximation is more accurate and less time consuming than the other methods. Moreover, the proposed method also suits for the small noisy time-course data. Here, we show the estimated results for the reverse problem only. Under the same search

![Genetic network used in the second computational experiments.](https://academic.oup.com/bioinformatics/article-abstract/21/7/1180/268883)
Table 4. Parameter value of the ‘true’ system (Case I) and sequential identification for parameter values and regulation structure (Cases II to V)

<table>
<thead>
<tr>
<th>Case</th>
<th>Variables</th>
<th>$\alpha_i$</th>
<th>$g_{i1}$</th>
<th>$g_{i2}$</th>
<th>$g_{i3}$</th>
<th>$g_{i4}$</th>
<th>$g_{i5}$</th>
<th>$\beta_i$</th>
<th>$h_{i1}$</th>
<th>$h_{i2}$</th>
<th>$h_{i3}$</th>
<th>$h_{i4}$</th>
<th>$h_{i5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>I</td>
<td>$X_1$</td>
<td>5.0</td>
<td>1.0</td>
<td>-1.0</td>
<td>10.0</td>
<td>2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_2$</td>
<td>10.0</td>
<td>2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_3$</td>
<td>10.0</td>
<td>-1.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_4$</td>
<td>8.0</td>
<td>2.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_5$</td>
<td>10.0</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>II</td>
<td>$X_1$</td>
<td>7.3764</td>
<td>-2.32E-4</td>
<td>0.5711</td>
<td>-8.75E-7</td>
<td>-0.4776</td>
<td>13.35</td>
<td>1.7019</td>
<td>1.48E-4</td>
<td>-1.00E-4</td>
<td>9.67E-8</td>
<td>0.2955</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_2$</td>
<td>7.8208</td>
<td>1.1384</td>
<td>1.18E-9</td>
<td>3.58E-7</td>
<td>-2.62E-4</td>
<td>8.67</td>
<td>-6.32E-2</td>
<td>1.6918</td>
<td>2.08E-8</td>
<td>-6.08E-7</td>
<td>4.21E-7</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_3$</td>
<td>12.449</td>
<td>-7.82E-9</td>
<td>-0.7112</td>
<td>-6.49E-2</td>
<td>3.02E-5</td>
<td>12.542</td>
<td>1.31E-7</td>
<td>-0.7049</td>
<td>2.0056</td>
<td>-5.97E-2</td>
<td>3.21E-6</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_4$</td>
<td>10.046</td>
<td>4.63E-7</td>
<td>2.35E-7</td>
<td>1.5244</td>
<td>-0.7407</td>
<td>12.236</td>
<td>1.13E-6</td>
<td>2.35E-2</td>
<td>-9.21E-7</td>
<td>1.8087</td>
<td>7.13E-5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_5$</td>
<td>7.5119</td>
<td>2.11E-5</td>
<td>-7.74E-2</td>
<td>6.17E-7</td>
<td>1.1965</td>
<td>7.4972</td>
<td>-6.35E-2</td>
<td>1.64E-7</td>
<td>-1.38E-7</td>
<td>0.1239</td>
<td>1.7002</td>
<td></td>
</tr>
<tr>
<td>III</td>
<td>$X_1$</td>
<td>6.0793</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.6624</td>
<td></td>
<td></td>
<td>11.349</td>
<td>1.8348</td>
<td></td>
<td>8.68E-2</td>
</tr>
<tr>
<td></td>
<td>$X_2$</td>
<td>10.028</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.0015</td>
<td></td>
<td></td>
<td>10.024</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_3$</td>
<td>11.153</td>
<td></td>
<td>0.8419</td>
<td></td>
<td></td>
<td>11.81</td>
<td></td>
<td></td>
<td></td>
<td>0.841</td>
<td>2.0131</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_4$</td>
<td>9.8109</td>
<td></td>
<td>1.5523</td>
<td></td>
<td></td>
<td>0.7722</td>
<td></td>
<td></td>
<td>11.876</td>
<td></td>
<td>1.8061</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_5$</td>
<td>9.932</td>
<td></td>
<td>2.0141</td>
<td></td>
<td></td>
<td>9.9354</td>
<td></td>
<td></td>
<td></td>
<td>3.52E-6</td>
<td>2.0114</td>
<td></td>
</tr>
<tr>
<td>IV</td>
<td>$X_1$</td>
<td>5.9438</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.6943</td>
<td></td>
<td></td>
<td>11.042</td>
<td>1.8467</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_2$</td>
<td>10.074</td>
<td></td>
<td>2.0032</td>
<td></td>
<td></td>
<td>0.7591</td>
<td></td>
<td></td>
<td>10.092</td>
<td>2.005</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_3$</td>
<td>10.883</td>
<td></td>
<td>0.8539</td>
<td></td>
<td></td>
<td>10.903</td>
<td></td>
<td></td>
<td>10.903</td>
<td>-0.8534</td>
<td>2.0166</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_4$</td>
<td>9.5202</td>
<td></td>
<td>1.5685</td>
<td></td>
<td></td>
<td>-0.7947</td>
<td></td>
<td></td>
<td>11.551</td>
<td></td>
<td>1.8088</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_5$</td>
<td>10.042</td>
<td></td>
<td>2.0032</td>
<td></td>
<td></td>
<td>10.046</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2.0015</td>
<td></td>
</tr>
<tr>
<td>V</td>
<td>$X_1$</td>
<td>5.0145</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.0128</td>
<td></td>
<td></td>
<td>-1.0031</td>
<td>10.01</td>
<td>1.9936</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_2$</td>
<td>9.9001</td>
<td>1.9935</td>
<td></td>
<td></td>
<td></td>
<td>9.8709</td>
<td></td>
<td></td>
<td>9.8709</td>
<td></td>
<td>1.9932</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_3$</td>
<td>10.321</td>
<td></td>
<td>-0.9631</td>
<td></td>
<td></td>
<td>10.344</td>
<td></td>
<td></td>
<td>10.344</td>
<td>-0.9594</td>
<td>1.9987</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_4$</td>
<td>7.9934</td>
<td></td>
<td>2.0157</td>
<td></td>
<td></td>
<td>9.9809</td>
<td></td>
<td></td>
<td>9.9809</td>
<td></td>
<td>2.0018</td>
<td></td>
</tr>
<tr>
<td></td>
<td>$X_5$</td>
<td>9.9661</td>
<td></td>
<td>1.9854</td>
<td></td>
<td></td>
<td>9.9668</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1.9974</td>
<td></td>
</tr>
</tbody>
</table>

The value with underline indicates that the parameter is deleted at the next identification procedure. Parameter values of Case V are the minimum solution obtained by the local search, which uses the starting point yielded from Case IV.
Evolutionary optimization with data collocation

Table 5. Comparison between the proposed method and two reported methods for the reverse problem of the second case study

<table>
<thead>
<tr>
<th>Method</th>
<th>ODE solving method</th>
<th>CPU time</th>
<th>Estimated result</th>
</tr>
</thead>
<tbody>
<tr>
<td>This work</td>
<td>Converted into algebraic equations</td>
<td>2.84 h required from Cases II through V on a single-CPU Pentium IV 2.4 GHz</td>
<td>The parameter values are nearly identical to the true value</td>
</tr>
<tr>
<td>PEACE 1</td>
<td>Numerical integration</td>
<td>10 h for one loop on PC cluster with 1040 CPUs and Pentium III 933 MHz</td>
<td>The structure is not completely identical to the true system</td>
</tr>
<tr>
<td>GLSDC</td>
<td>Decomposed numerical integration</td>
<td>58.8 min for minimizing each subsystem on a single-CPU Pentium III 1 GHz</td>
<td>This method was unable to estimate the parameter values with perfect precision</td>
</tr>
</tbody>
</table>

PEACE 1 was reported by Kikuchi et al. (2003). GLSDC was reported by Kimura et al. (2004).

range as discussed in the previous case study, the HDE algorithm with 100,000 generations was used to solve the estimation problem to yield the best objective function value of 8.287E−3. The optimal parameter values are shown in Case II of Table 4. We subtracted the smaller kinetic orders, which the magnitude is <0.01, as shown in Case II with underline to zero. The pruned model was then refit by the global search to obtain the optimal estimates given in Case III of Table 4. Using the same type of simplification again, we obtained the results in Case IV of Table 4. This structure is essentially identical to the ‘true’ system. Such parameter values were then applied as the starting point for the local search method to yield the optimal objective function value of 5.7576E−6. The optimal estimates are given in Case V of Table 4. In Table 4, we used the skeletal threshold of 0.1 to prune small parameters. When a smaller threshold is used, for example 1.0E−3, more sequential identification stages are required, six stages for this case study, to yield the identical result.

The estimated parameter values were applied to evaluate two extra test experiments in order to validate the model. Input conditions for the first test experiment are greater than the upper bounds for the training experiments. However, conditions for the second test experiment are less than the lower bounds for the training experiments. Both ‘true’ dynamic profiles are shown as the circle and triangle data points in Figure 7a and b, respectively. The parameter values for Case V in Table 4 are much close to the ‘true’ process so that the computed profiles (solid curves) are nearly identical to both test experiments shown in Figure 7. Cases II, III and IV were applied to evaluate the fitness. Because the model for Case II uses too many parameters to describe the dynamic behavior, some profiles are less fit to the test experiments, as observed from Figure 7.

Table 5 summarizes the comparison between the proposed method and two reported methods for the reverse problem. The proposed method applied the modified collocation to convert differential equations into a set of algebraic equations. This approach is not only suitable for parallel computation, but also saves a lot of computation time. The total CPU time for this parameter and structure identification from Case II to Case V was ~2.84 h on a single-CPU computer. The PEACE 1 used numerical integration to solve differential equations. Such a method using a modified genetic algorithm and running on a PC cluster with 1040 CPUs took ~10 h for one loop (Kikuchi et al., 2003). In addition, the structure is not completely identical to the true system. Kimura et al. (2004) applied a decomposed method to integrate differential equations. Such an algorithm applied a real-coded genetic algorithm with parallel computation to solve the problem (referred to as GLSDC). It required ~58.8 min on a single-CPU Pentium III 1 GHz to optimize each subproblem. However, such a method was unable to estimate the parameter values with precision.
CONCLUSION
Parameter estimation is a well-established technique and is routinely applied to estimate biochemical kinetics and enzyme kinetic parameters in many laboratories. However, there are still many challenges in this field for the parameter estimation of nonlinear dynamic biological systems. Numerical integration for differential equations and finding global parameter values are two major problems. In this study, we applied the modified collocation method to convert differential equations into a set of algebraic equations. Such an approximation could yield accurate model profiles and save a lot of computational time. HDE was applied as a global search to determine a satisfied solution, and then such estimates were used as the starting point for a gradient-based optimization method to obtain refined estimates. Two case studies were used to illustrate the effectiveness of the global/local search method and to compare with several methods.

ACKNOWLEDGEMENTS
The financial support from the National Science Council, ROC (Grant No. NSC92-2214-E194-004) is highly appreciated.

REFERENCES