

Fuzzy-Based Self-Interactive Multiobjective Evolution Optimization for Reverse Engineering of Biological Networks

Shinq-Jen Wu, Cheng-Tao Wu, and Jyh-Yeong Chang, *Member, IEEE*

Abstract—S-system modeling from time series datasets can provide us with an interactive network. However, system identification is difficult since an S-system is described as highly nonlinear differential equations. Much research adopts various evolution computation technologies to identify system parameters, and some further achieve skeletal-network structure identification. However, the truncated redundant kinetic orders are not small enough as compared with the preserved terms. In this paper, we integrate quantitative genetics, bacterium movement, and fuzzy set theory into evolution computation to develop a new genetic algorithm to achieve convergence enhancement and diversity preservation. The proposed *exploration and exploitation* genetic algorithm (EEGA) can improve the best-so-far individual and ensure global optimal search at the same time. The EEGA enhances evolution convergence by golden section seed selection, normal-distribution reproduction, mixed inbreeding and backcrossing, competition elitism, and acceleration operations. Search-then-conquer evolution direction operations, eugenics-based screen-sifting mutation, eugenic self-mutation, and fuzzy-based tumble migration preserve population diversity to avoid premature convergence. Furthermore, to ensure that a reasonable gene regulation network is inferred, fuzzy composition is introduced to derive a reconstruction index. This performance index let EEGA possess self-interactive multiobjective learning. The proposed fuzzy-reconstruction-based multiobjective genetic algorithm is examined by three dry-lab biological systems. Simulation results show that a safety pruning action is guaranteed (the truncation threshold is set to be 10^{-15}), and only one- or two-step pruning action is taken.

Index Terms—Multiobjective, real-value coding, self-interactive, structure identification.

I. INTRODUCTION

THE inverse problem of identifying the topology of a biological network from their time-course response is a cornerstone challenge in systems biology [1]. Hill and Michaelis-Menten's [2] rate modeling is a forward approach and can provide local kinetic information of components. However, re-

peated modification and an undue amount of experiment data are necessary in parameter identification, especially for a system with many substances or reactions involved. S-system structure [3], [4] is another preferred nonlinear dynamic model. This model can uniquely map dynamic interaction onto its parameters and possesses good generalization characteristics. Some researchers used gradient-based computation technologies to identify the parameters of an S-system model. Marino and Voit [5] wrote an algorithm to gradually increase model complexity. Chou *et al.* [6] adopted an alternating regression (AR) method. Vilela *et al.* [1] further proposed an eigenvector optimization method to solve the convergence issues of an AR approach. Kotalik *et al.* [7] introduced the Newton flow analysis.

Recently, many researchers have tried to infer a gene regulatory network by stochastic search intelligent technologies such as genetic programming [8]–[11], evolutionary algorithms [12], evolution strategies [13], differential evolution [14]–[18], genetic algorithms (GAs) [19]–[21], simulated annealing [22], hybrid GA and simulated annealing [23], radial basis function networks [24], and a neural network with particle swarm optimization [25], [26]. However, as system states and parameters increase, parameter estimation becomes difficult and the solutions are problematic. Furthermore, pruning redundant kinetic orders to infer a suitable network structure is a big challenge. Based on the fact that a genetic network is connected sparsely [27], various penalty terms are introduced [12], [14], [19], [28]. However, no matter what kinds of penalty terms are chosen, the truncated redundant kinetic orders are not small enough to guarantee that a safely pruning operation is taken.

Technological contributions of this paper can be described as follows. A new GA with various advanced genetic operators is proposed in Section II-A. The proposed GA possesses a tradeoff between exploration and exploitation. In Section II-B, single-objective performance candidates for *parameter identification* are chosen from 28 indexes. These candidates are fuzzily integrated with a new kinetic order index for multiobjective *structure identification*. In Section III, the proposed fuzzy-reconstruction-based multiobjective genetic algorithm (FRMOGA) is examined by three genetic networks. Section IV is the conclusive remark.

II. FUZZY-BASED SELF-INTERACTIVE MULTIOBJECTIVE EVOLUTION COMPUTATION

S-system is a well-known canonical nonlinear model to capture genetic interactions/transcriptional regulation. Based on

Manuscript received March 21, 2011; revised August 12, 2011 and December 8, 2011; accepted December 19, 2011. Date of publication February 7, 2012; date of current version October 2, 2012. This work was supported by the National Science Council, Taiwan, under Grant NSC-99-2221-E-212-021.

S.-J. Wu is with the Department of Electrical Engineering, Da-Yeh University, Changhua 51591, Taiwan (e-mail: jen@mail.dyu.edu.tw).

C.-T. Wu and J.-Y. Chang are with the Department of Electrical and Control Engineering, National Chiao-Tung University, Hsinchu 300, Taiwan (e-mail: dau@cn.nctu.edu.tw; jychang@mail.nctu.edu.tw).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TFUZZ.2012.2187212

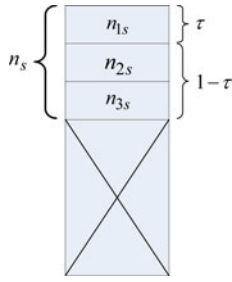


Fig. 1. Golden section.

biochemical system theory, the net influx (V_i^+) and efflux (V_i^-) of a system are approximated as power law functions. Each individual metabolite, protein, or gene is described as

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

where n and m are the numbers of dependent and independent variables, respectively; α_i and β_j are rate constants, g_{ij} and h_{ij} are kinetic orders to denote the interaction from X_j to X_i , where a positive value denotes excitatory effect and negative for inhibitory effect. Recently, various evolutionary optimization technologies were used to infer S-type gene regulatory networks (parameter estimation and structure identification of an S-system). However, as system states increase, the inference of a suitable network structure becomes a big challenge and the solutions are problematic. Local minimum solutions will generate an error structure. Avoiding from sticking into local minima is very important to infer such a high dimensional and nonlinear system by computation approach. Sometimes, in a biological system, an interaction order is small, but the interaction effect cannot be neglected. Besides, noise and uncertainty exist in a biological system, especially for dealing with microarray data. To achieve correct structure, the gap between true and redundant interactions should be increased. In other words, the pruning threshold should be small enough to ensure safety pruning. Furthermore, a wide searching space is necessary since no prior information is provided. Therefore, the used technologies should possess good search power and has the ability to escape from local minima. It is hard for a state-of-the-art GA with simplex crossover (SPXGA) to obtain a satisfactory solution in a limited computation time [21]. Ho *et al.* proposed a GA with intelligent crossover and Cauchy-Lorentz probability-distribution mutation to identify genetic networks [21]. We here propose an *exploration and exploitation* genetic algorithm (EEGA) to improve the best-so-far individual and to ensure global optimal search at the same time. EEGA enhances evolution convergence by golden section seeds selection, normal-distribution reproduction, mixed inbreeding and backcrossing, competition elitism, and acceleration operations. Search-then-conquer evolution direction (SCED) operation, eugenics-based screen-sifting mutation, eugenic self-mutation, and fuzzy-based tumble migra-

tion are to preserve population diversity to avoid premature convergence.

A. Exploration and Exploitation Genetic Algorithm

EEGA introduces seed group concept for selection (see Sections II-A2–II-A4) and mixes backcrossing into general inbreeding operations for crossover. Acceleration, eugenic self-mutation, and fuzzy-based tumble migration are for the best-so-far individual to further improve its local and global search power.

1) *Population Initialization*: We adopt real coding to exploit the gradualness of continuous variables. Each parameter is initialized to cover an entire search space. Therefore, an initial population with N_P individuals is chosen as

$$I_i^0 = I_{\min} + r(I_{\max} - I_{\min}), \quad i = 1, \dots, N_P \quad (2)$$

where r is a real number from a uniform distribution $[0,1]$, I_{\min} and I_{\max} are the lower and upper bounds of individual I_i .

2) *Golden Section Seeds Selection*: Individuals in a population are arranged according to their fitness values in a descending order. Then, a golden section method in Fig. 1 is used as follows to choose three Elitism-seeds groups, i.e., n_{1S} , n_{2S} , n_{3S} :

$$\begin{aligned} n_{1S} &= [\tau \cdot n_S] && \text{for the first seeds group} \\ n_{2S} &= [\tau \cdot (n_S - n_{1S})] && \text{for the second seeds group} \\ n_{3S} &= n_S - n_{1S} - n_{2S} && \text{for the third seeds group} \end{aligned} \quad (3)$$

where $[\bullet]$ is a Gauss mark, and τ is the golden section constant; the number of selected seeds is $n_S = [\tau \cdot N_P]$ for a small population ($N_P < 10$) and $n_S = [(1 - \tau) \cdot N_P]$ for $N_P \geq 10$.

3) *Search-Then-Conquer Evolution Direction*: We then choose three preferable individuals (I_b , I_s , and I_t) from these three seeds groups; one individual from each group. F_b , F_s , and F_t are the associated fitness. These three individuals are used in SCED operation to determine the evolutionary direction of a population. SCED proceeds a three-phase search. In the early-evolution phase, a random walk search is taken for wider searching

$$I_{\text{sced}} = I_b + r_1 * D_1 * (I_b - I_{r_1}) + r_2 * D_2 * (I_b - I_{r_2}). \quad (4)$$

In the middle-evolution phase, a directed-walk search is taken to provide a good searching direction

$$I_{\text{sced}} = I_b + r_1 * D_1 * (I_b - I_s) + r_2 * D_2 * (I_b - I_t). \quad (5)$$

Then, a conquer strategy is used to achieve fast convergence for the final phase

$$I_{\text{sced}} = I_b + r_1 * D_1 * (I_b - I_{b1}) + r_2 * D_2 * (I_b - I_{b2}) \quad (6)$$

where r_1 and r_2 are random factors; D_1 and D_2 denote evolutionary directions; I_{r_1} and I_{r_2} are two randomly selected individuals; and I_{b1} and I_{b2} are the perturbed values of I_b . The associated fitness value F_{sced} is estimated. If F_{sced} is better than one of the three preferable fitness values, the individual is replaced. Fig. 2 describes the SCED operation. SCED has the ability for local and global search synchronously. We can use this operator to achieve widely searching and

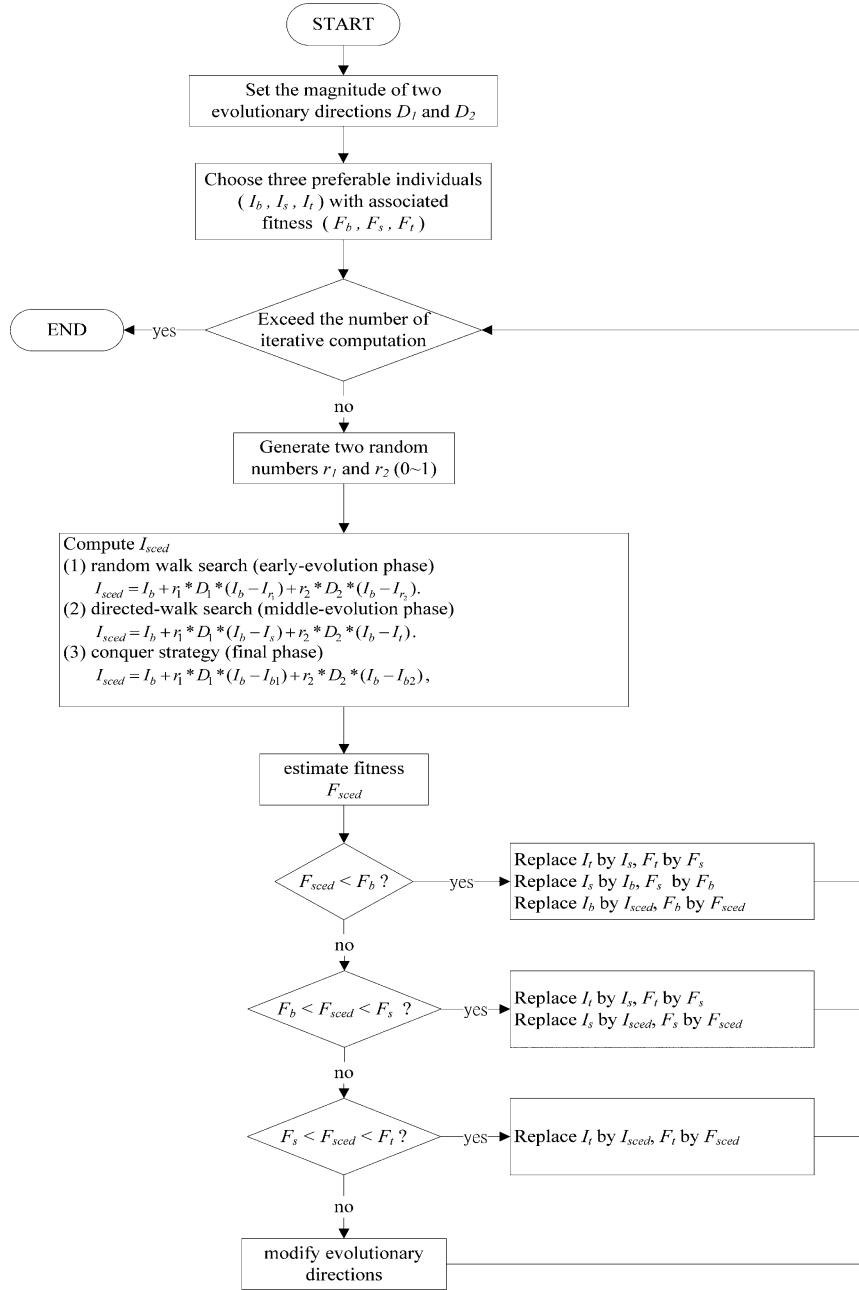


Fig. 2. SCED.

then quickly converging toward a global optimal solution. The operator helps us from bogging down into a local optimal solution.

4) *Normal-Distribution Reproduction*: Let the fitness of offspring be a Gaussian distribution. $[0.5\sigma N_p]$ individuals are related to the first seeds group only, $[\sigma N_p]$ individuals are related to the first two seeds groups only, and $[1.5\sigma N_p]$ individuals are related to these three seeds groups only. For diversity, the residue individuals are chosen from those eliminated individuals only. σ is the standard deviation of a normal distribution. Fig. 3 describes a parent distribution for crossover. During a crossover operation, if the same individual is chosen for a father and a mother, the individual is perturbed with a Gaussian/normal function $N(0, \sigma)$. Therefore, we denote the groups in Fig. 3 by

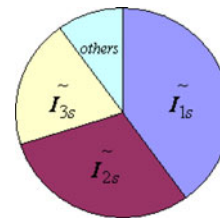


Fig. 3. Normal-distribution reproduction.

perturbed groups, where I_{is} denotes the individuals from the i th seeds group

$$\tilde{I}_{is} = \{I_{is} + N(0, \sigma)\}, \quad i = 1, 2, 3. \quad (7)$$

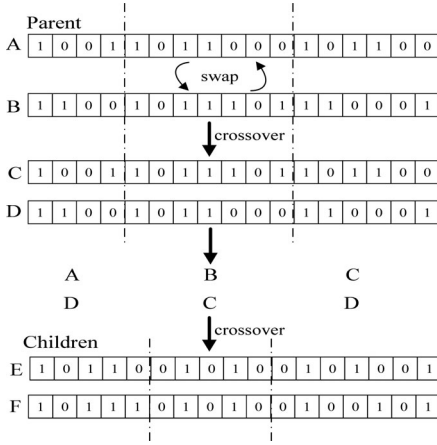


Fig. 4. Mixed inbreeding and backcrossing operation.

5) *Competition Elitism*: Competition carries on in each genetic operation. *Soft competition* is used for evolution direction operation and mutation. Three seeds groups, i.e., I_{1s} , I_{2s} , I_{3s} , take over the most responsibility; the more highly fit strings have a higher number of offspring. Only the mutated genes with better fitness values than their precursors can be accepted. *Winner-take-all* is adopted for acceleration, self-mutation, and migration. Only the best gene is chosen to participate in these operations. Competition also occurs between generations; only the winner can survive to the succeeding generation.

6) *Mixed Inbreeding and Backcrossing*: Randomly choose two points in parent strings (A and B) for two-point crossover to generate children C and D. These two children or just one is selected to replace the father or the mother for further crossover. In other words, the selected candidates are two children (C and D) for *close inbreeding*, or a parent and a child (A and D or B and C) for *backcrossing*. Then, a two-point crossover operation is used again to generate offspring E and F. Fig. 4 describes the proposed mixed inbreeding and backcrossing operation.

The inbreeding coefficient of offspring X_1 from two children (C and D) is denoted by f_{x_1} , and that of offspring X_2 from backcrossing between parent and children (A and D or B and C) is denoted by f_{x_2} [29]

$$\begin{aligned} f_{x_1} &= f_{CD} = \frac{1}{4}(f_{AA} + 2f_{AB} + f_{BB}) \\ &= \frac{1}{4} \left(1 + \frac{1}{2}(F_A + F_B) + 2f_{AB} \right) \\ f_{x_2} &= f_{AD} = \frac{1}{2}(f_{AA} + f_{AB}) = \frac{1}{2} \left(\frac{1}{2}(1 + F_A) + f_{AB} \right) \end{aligned} \quad (8)$$

where F_A and F_B are the inbreeding coefficient of individuals, A and B, respectively. The average relative coefficient f_{av} with backcrossing probability r_b is defined as

$$f_{av} = r_b \cdot \frac{1}{4}(f_{AD} + f_{AC} + f_{BC} + f_{BD}) + (1 - r_b) \cdot f_{CD}. \quad (9)$$

Therefore, $f_{av} = f_{av, \max}$ for parent coming from the same seeds group and $f_{av} = f_{av, \min}$ for no blood relationship parent. Based on the average relative coefficient, we set the selected candidates

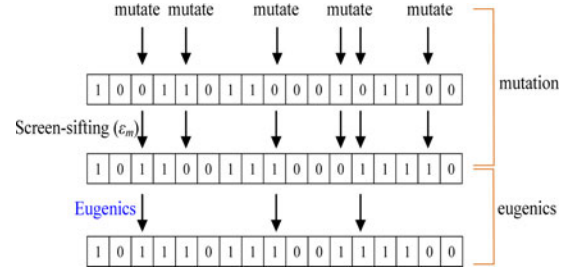


Fig. 5. Eugenics-based screen-sifting mutation.

for further crossover to be B and C in case of $r_b \leq f_{av, \min}$, to be A and D for $f_{av, \min} < r_b \leq f_{av, \max}$, and to be C and D for $r_b > f_{av, \max}$.

7) *Eugenics-Based Screen-Sifting Mutation*: In order to ensure global search and improve convergence, we here propose a mutation-then-eugenics operation. In stead of adopting only one gene (one-point mutation), two genes (two-point mutation), or some fixed-genes (mask mutation) for mutation, we let all genes or randomly choose some genes to mutate with mutation probability assigned by a designer. Those genes with qualified mutation rate will mutate (*screen sifting*). After this kind of mutation, a *variety* population is generated. We wish the mutated population distributes over a *wide but not an entire space*. A deviation threshold is then defined as

$$TA = \frac{|F_{\text{mut}} - F_b|}{F_b} \quad (10)$$

where F_{mut} is the fitness value of the mutated individual. Then, the mutated individuals compete with their source individuals. Those winners or losers with acceptable deviation ($TA < 10$) can pass down. In engineering, a ratio 3 or 5 means much far away or apart in space. Here, we use 10 to denote that mutated individuals distribute widely. The proposed eugenics-based mutation in Fig. 5 can not only increase population diversity but can guarantee fine offspring as well.

8) *Acceleration*: A steepest descent method or simplex method is used to enhance convergence. After acceleration, the best individual is

$$I_b = \begin{cases} \bar{I}_b, & \text{if } f(\bar{I}_b) < f(I_b^0) \\ \bar{I}_b - \alpha \nabla_f, & \text{otherwise} \end{cases} \quad (11)$$

where I_b^0 and \bar{I}_b are the best individuals before and after both mutation and crossover operations, α is the size of a step to determine the descent rate, and ∇_f is the gradient of an objective function. In other words, an acceleration operation starts when both crossover and mutation operations can no longer increase fitness.

9) *Eugenic Self-Mutation*: Inherent premature convergence and leak mountain climbing are two weak points of real-coding GA. Acceleration operations can compensate for the latter. We here propose eugenic self-mutation to avoid premature convergence. The mutation is applied to the best individual only. Each gene in the best individual is chosen and assigned a mutation probability. Competition carries on between the original

individual and the mutated individual. The winner will pass down to the next operation.

10) *Fuzzy-Based Tumble Migration*: We here integrate fuzzy concept and the movement of an *E.coli* bacterium into a migration operation. A population P is composed of N_p individuals $(I_1, I_2, \dots, I_{N_p})$. An individual I_i is composed of n chromosomes $(x_{i1}, x_{i2}, \dots, x_{in})$:

$$P = \{I_i\}_{i=1, \dots, N_p} = \begin{bmatrix} I_1 \\ I_2 \\ \vdots \\ I_{N_p} \end{bmatrix}$$

$$= \begin{bmatrix} x_{11} & x_{12} & \dots & x_{1n} \\ x_{21} & x_{22} & \dots & x_{2n} \\ \vdots & \vdots & \dots & \vdots \\ x_{N_p,1} & x_{N_p,2} & \dots & x_{N_p,n} \end{bmatrix}$$

$$I_i = \{x_{ij}\}_{j=1, \dots, n} = [x_{i1} \quad x_{i2} \quad \dots \quad x_{in}]. \quad (12)$$

Define fuzzy sets \tilde{I}_i , $i = 1, \dots, n$, to represent *individuals not close to I_b* . The relative cardinality of \tilde{I}_i describes the degree of the deviation of I_i from I_b :

$$|\tilde{I}_i|_{\text{rel}} = \frac{|\tilde{I}_i|}{|U_i|}$$

$$\tilde{I}_i = \{(x_{ij}, \mu_{\tilde{I}_i}(x_{ij})) \forall j = 1, \dots, n\}$$

$$= [\mu_{i1} \quad \mu_{i2} \quad \dots \quad \mu_{in}]$$

$$\mu_{ij} = \mu_{\tilde{I}_i}(x_{ij}) = \left| \frac{x_{ij} - x_{bj}}{x_{bj}} \right| \quad (13)$$

where U_i are the universal set of I_i . Therefore, the following fuzzy set \tilde{P} denotes *the deviation tendency of a population to the best individual I_b* , and its α level cut \tilde{P}_α contains the elements of a universal set U with membership grades greater than or equal to α (closeness tolerance). In other words, \tilde{P}_α contains the elements in \tilde{P} far enough to I_b . Therefore, we define the *population diversity* of P as η :

$$\eta \triangleq \frac{|\tilde{P}_\alpha|}{|U|}$$

$$\tilde{P} = \begin{bmatrix} \tilde{I}_1 \\ \tilde{I}_2 \\ \vdots \\ \tilde{I}_{N_p} \end{bmatrix} = \begin{bmatrix} \mu_{11} & \mu_{12} & \dots & \mu_{1n} \\ \mu_{21} & \mu_{22} & \dots & \mu_{2n} \\ \vdots & \vdots & \dots & \vdots \\ \mu_{N_p,1} & \mu_{N_p,2} & \dots & \mu_{N_p,n} \end{bmatrix}. \quad (14)$$

If the population diversity η is smaller than a diversity threshold ξ , then migration starts. The tumble movement of an *E.Coli* cell is adopted for migration operation: After a tumble migration, a new individual is generated as

$$I_{\text{new}} = I_b + c \cdot \left[\frac{\Delta}{\sqrt{\Delta^T \Delta}} \right] \quad (15)$$

where c is the size of the step taken in a random direction, and Δ indicates a vector in a random direction.

B. Preference-Based Self-Interactive Multiobjective Optimization

To infer a physically realizable skeletal-network structure, various pruning-penalty performance indexes are proposed. On the basis of that a genetic network is connected sparsely [27], Kikuchi *et al.* [19] introduced kinetic orders as the pruning (penalty) terms of a fitness function. Kimura *et al.* [12] replaced kinetic orders by their descending orders and introduced a maximum in degree l to determine the maximum number of genes that affect the i th gene. Norman and Iba [14] modified penalty terms by the sorted orders of both excitatory and inhibitory terms. Liu and Wang [28] introduced ε -constrain weighted sum. Ko *et al.* [30] considered a flux connectivity relationship and introduced a sensitivity term. However, no matter what kinds of penalty terms are used, pruning approaches face a challenge: Weighting coefficients need to be *carefully tuned*, and there are no clear guidelines for setting suitable penalty weights [31]. Furthermore, even if so much effort has been done, the pruned criteria in Table V, shown in the Appendix, are not small enough as compared with the preserved terms. In other words, a safely pruning action is not guaranteed. In the previous section, we integrate quantitative genetics, bacterium movement, and fuzzy set theory into evolution computation to develop a new GA to achieve convergence enhancement and diversity preservation. However, a performance index that is defined on various errors is a criterion to show how closing the estimated data to a real profile. This index determines both search directions and computation time. Therefore, the choice of a performance index is a key point for computation optimization. In this section, we shall get single-objective performance index candidates first. Those candidates are further integrated into a self-interactive multiobjective function.

1) *Single-Objective Performance Index Candidates*: The proposed EEGA algorithm with 28 performance indexes in Table VI, shown in the Appendix, is used to identify the parameters of S-systems for two dry-lab experiments in Figs. 6(a) and 6(b). Those well-performance indexes are chosen as candidates, and are further integrated into a multiobjective function. All computation is performed on an Intel core duo 3.16-GHz computer using Microsoft Windows XP. Parameter ranges are [0, 30] for rate constants and [-4, 4] for kinetic orders. 250 000 maximum iterations are performed to minimize those error criteria. Eight sets of experiment data x_{exp}^i and the corresponding slope information \dot{x}_{exp}^i are generated from the dynamic system in [20] for a genetic branch pathway, and in [15] for a cascade pathway. The simulation time of an experiment is from $t = 0$ s to $t = 8$ s, and sample time is set to be 0.02.

The genetic branch pathway in Fig. 6(a) has four dependent constituents, i.e., x_1, x_2, x_3 , and x_4 , and one constant source, i.e., x_0 . The results in Table VII, shown in the Appendix, give us some information. First, summation operations perform much better than the corresponding maximum operations except for J_5^l, J_7^l , and J_{14}^l . Second, J_1^l, J_2^l , and J_3^l are, respectively, better than J_4^l, J_5^l , and J_6^l ; J_7^l is better than J_{13}^l ; and J_{10}^l is better than J_{14}^l . Those results show that even if errors are always less than one and a error-squaring operation can scale

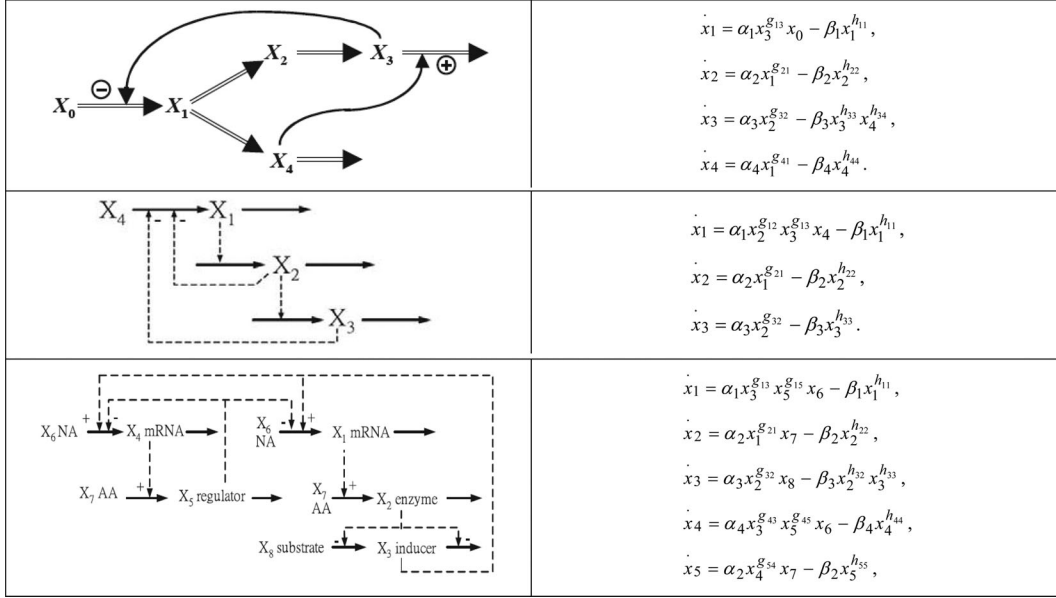


Fig. 6. Three gene regulation systems and their S-type Models. (a) Genetic branch pathway with two regulatory signals [20]. (b) Cascade pathway with three steps and two feedback signals [15]. (c) Small-scale genetic network [32], [19].

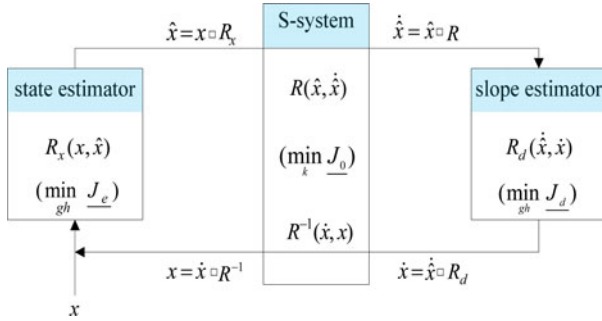


Fig. 7. Block diagram of an artificial system for structure identification.

down errors to generate a small-quantity criterion for updating, this operation still improves performance. Third, indexes with normalization have better performance (J_{10}^l, J_{11}^l , and J_{12}^l are, respectively, better than J_7^l, J_8^l , and J_9^l). Fourth, among summation operations, the performance of those weighting-related indexes is always $t_a > 1 > t_s$ (“>” denotes better). However, if slope normalization is taken ($J_{10}^l, J_{11}^l, J_{12}^l$), then it becomes $1 > t_a > t_s$. Based on these four phenomena, we suggest to choose $J_1^S, J_2^S, J_3^S, J_9^S, J_{10}^S, J_{12}^S, J_3^M$, and J_{10}^M as learning-criterion candidates.

We now consider another biological system. The cascade pathway in Fig. 6(b) has three dependent constituents x_1, x_2 , and x_3 , and one constant source x_4 . Table VIII in the Appendix has similar results as in Table VII except the third phenomenon. This is due to a dramatically sharp-slope point in x_2 profile. Therefore, slope normalization is not suitable. Based on Table VIII, shown in the Appendix, we know that $J_1^S, J_3^S, J_6^S, J_7^S, J_9^S, J_{13}^S, J_1^M, J_3^M$, and J_9^M are all suitable for learning indexes. In the following section, we shall use J_3^S and J_{10}^S as our error- and slope-performance candidates to develop a multiobjective optimization technology.

2) *Fuzzy-Reconstruction-Based Multiobjective Genetic Algorithm*: In the previous section, we examine parameter iden-

tification performance under 28 criteria. The results give us a very important concept. Except normalization operation, the dynamic behavior of a system is an important factor. We now extend the results to define a realizable learning index for the skeletal-structure identification of a biological system. Concentration error J_e and kinetic order J_o are two key penalties to examine fitness between measured data and estimated data, and to ensure that a genetic network is sparsely connected. Besides these two, we also introduce a slope error penalty J_d to achieve smooth time evolution. Furthermore, at any time instant, the *net interaction strength* (row sum of matrix K) of all species to some species is finite. Therefore, we define a kinetic order penalty as

$$J_o = \|K\|_\infty = \max_i \left\{ \sum_{j=1}^n |k_{ij}| \right\} \quad (\text{row sum})$$

$$K = \{k_{ij}, \quad i = 1, \dots, n, \quad j = 1, \dots, 2(n+m)\}$$

$$= \begin{bmatrix} g_{11} & \dots & g_{1,n+m} & \vdots & h_{11} & \dots & h_{1,n+m} \\ \vdots & & \vdots & & \vdots & & \vdots \\ g_{n1} & \dots & g_{n,n+m} & \vdots & h_{n1} & \dots & h_{n,n+m} \end{bmatrix}. \quad (16)$$

J_3^S and J_{10}^S are chosen as our error- and slope-penalty, respectively.

As we know, concentration error J_e , slope error J_d , and skeletal-structure penalty J_o are belong to different scales. We here normalize them as $J_e = J_e/J_{ee}$, $J_d = J_d/J_{de}$, and $J_o = J_o/J_{oe}$, where J_{ee}, J_{de} , and J_{oe} are the expected concentration error, slope error, and interaction measure, respectively. To infer a biological regulation network, our objective is to push both J_e and J_d to approach zero but to obtain a *nonzero* minimum value of J_o . The targets of J_e and J_d are the same, but they are totally different from that of J_o . Therefore, summing their scaled functions is not suitable. Moreover, the structure identification of a biological system is to get minimum value of J_o under

TABLE I
VARIABLES AND MODULUS OF SUBPROCESSES IN FIG. 7

		Step	input	output	Fuzzy relation equation	optimization
state estimator		1	x	\hat{x}	$\hat{x} = x \circ R_x$	$\min_{gh} \underline{J}_e$
S-system modeling	forward	2	\hat{x}	$\dot{\hat{x}}$	$\dot{\hat{x}} = \hat{x} \circ R$	$\min_k \underline{J}_0$
	backward	4	$\dot{\hat{x}}$	x	$x = \dot{\hat{x}} \circ R^{-1}$	
slope estimator		3	$\dot{\hat{x}}$	\dot{x}	$\dot{x} = \dot{\hat{x}} \circ R_d$	$\min_{gh} \underline{J}_d$

allowable J_d and J_e . It is different from fuzzy rule selection, where rules are extracted only if it is possible to either maintain or even improve the system's accuracy [33], [34] and general multiobjective optimization problems. Therefore, Pareto-optimal-based evolution computation approaches [35]–[41] are also unsuitable. In this paper, we use fuzzy composition “ \circ ” (min max) to integrate these three penalties into a reconstruction performance. Based on this performance, the proposed GA becomes a self-interactive multiobjective evolution optimization technology.

Biological systems are dynamic changeable systems. To guarantee well performance of computation approaches for gene network identification, except concentration error, we should consider slope error for data with close values but different slope. Additionally, sparse connection and the correct division of excitatory and inhibitory interactions are two important issues that are to be concerned. Therefore, the whole structure identification can be physically realized as three modules in series (a state estimator, S-system modeling, and a slope estimator). Since uncertainty and noise are two serious issues in a biological system, fuzzy relation and fuzzy functions are adopted to describe the dynamic behavior of subprocess modules. Fig. 7 shows a block diagram to describe the inference of structure identification.

R_x , R , and R_d are, respectively, fuzzy modules for a state estimator, an S-system, and a slope estimator. R^{-1} is the inverse fuzzy relation for S-system modeling. State estimator $R_x(x, \hat{x})$ describes the fuzzy relation between input x (true state) and output \hat{x} (estimated state). The corresponding fuzzy relation equation is $\hat{x} = x \circ R$, where “ \circ ” denotes min max composition. Here, our purpose is to minimize \underline{J}_e such that estimated state \hat{x} approaches to true state x . Similar operations are used for S-system modeling $R(\hat{x}, \dot{\hat{x}})$ and slope estimator $R_d(\dot{\hat{x}}, \dot{x})$. Table I lists the inputs, outputs, fuzzy equations, and optimization of these subprocesses. Steps 1–4 describe the inference flow for structure identification. Three normalized errors, i.e., \underline{J}_e , \underline{J}_d , and \underline{J}_0 , denote the distortion of these subprocess. The entire process is composed of a forward process ($R_x \circ R$) and a backward process ($R_d \circ R^{-1}$). Therefore, we have

$$\begin{aligned}
x &= \dot{\hat{x}} \circ R^{-1} \text{ (backward S-system modeling)} \\
&= (\dot{\hat{x}} \circ R_d) \circ R^{-1} \text{ (slope estimator)} \\
&= ((\hat{x} \circ R) \circ R_d) \circ R^{-1} \text{ (forward S-system modeling)} \\
&= (((x \circ R_x) \circ R) \circ R_d) \circ R^{-1} \text{ (state estimator)} \\
&= x \circ R_x \circ R \circ R_d \circ R^{-1}. \tag{17}
\end{aligned}$$

For *perfect* construction, the membership functions of composite fuzzy relation $R^* = R_x \circ R \circ R_d \circ R^{-1}$ always belong to 1. Therefore, the reconstruction ability of structure identification is dependent on the composition of the *distortion* of subsystems J_e , J_o , and J_d . Therefore, we define a reconstruction performance $J_{\text{rec-0}}$ as the *union* (U) of $\underline{J}_e \circ \underline{J}_0$ and $\underline{J}_d \circ \underline{J}_0$ (forward AND backward processing)

$$\begin{aligned}
\min_k J_{\text{rec-0}} &= \min_k [(\underline{J}_e \circ \underline{J}_0) U (\underline{J}_d \circ \underline{J}_0)] \\
&= \min_k [(\underline{J}_e U \underline{J}_d) \circ \underline{J}_0] \\
&= \min_k [\min_k (\underline{J}_e \vee \underline{J}_d) \vee \underline{J}_0] \\
&= \min_k [\underline{J}_e \vee \underline{J}_d \vee \underline{J}_0] \tag{18}
\end{aligned}$$

where maximum \vee is adopted as the union operation. In other words, $J_{\text{rec-0}} = \underline{J}_e \vee \underline{J}_d \vee \underline{J}_0$.

Based on this definition, an automatic self-interactive learning in Table IX, shown in the Appendix, is done. It is noticed that J_{ee} and J_{de} are values approaching zero, but J_{oe} is a value between 0.001 and 10. As \underline{J}_0 is comparable with \underline{J}_e or \underline{J}_d , J_e and J_d are values approaching zero. In the beginning of a learning process, concentration error \underline{J}_e and slope error \underline{J}_d are much larger than \underline{J}_0 . During the mid-learning stage, those three penalties compete with each other to get the opportunity to be reduced. At the final stage, J_o becomes the only winner, since both \underline{J}_e and \underline{J}_d are close to zero. Therefore, choosing $J_{\text{rec-0}}$ can divide our learning into two phases. In phase I, the learning purpose is to get an allowable space; concentration error \underline{J}_e is the key learning index and slope error \underline{J}_d is for smooth evolution. In phase II, \underline{J}_0 is the only kernel index to achieve spare network structure. Therefore, we modify $J_{\text{rec-0}}$ by adding phase I and phase II weighting factors w_I and w_{II} , i.e., $J_{\text{rec-1}} = \max\{w_{II} \underline{J}_e, w_I \underline{J}_d, \underline{J}_0\}$. This modification makes the performance index suitable for all kinds of biological systems ($w_I = w_{II} = 1$ for most systems). Besides, the dynamic behavior of a system is also an important factor for time-series-data-based learning. (We get this result from parameter identification in Section II-B.1.) Considering dynamic behavior helps us to simultaneously fine-tune the structure and parameters of a system in phase-II learning. Therefore, we further modify the performance as $J_{\text{rec}} = \max\{\eta_{II} \underline{J}_e, w_I \underline{J}_d, \underline{J}_0\}$ with an adaptive-learning ratio $\eta_{II} = w_{II} \times \mu$, where μ is an adaptive dynamic factor.

The reconstruction index is embedded into EEGA to infer biological regulatory networks from time series data. Fig. 8

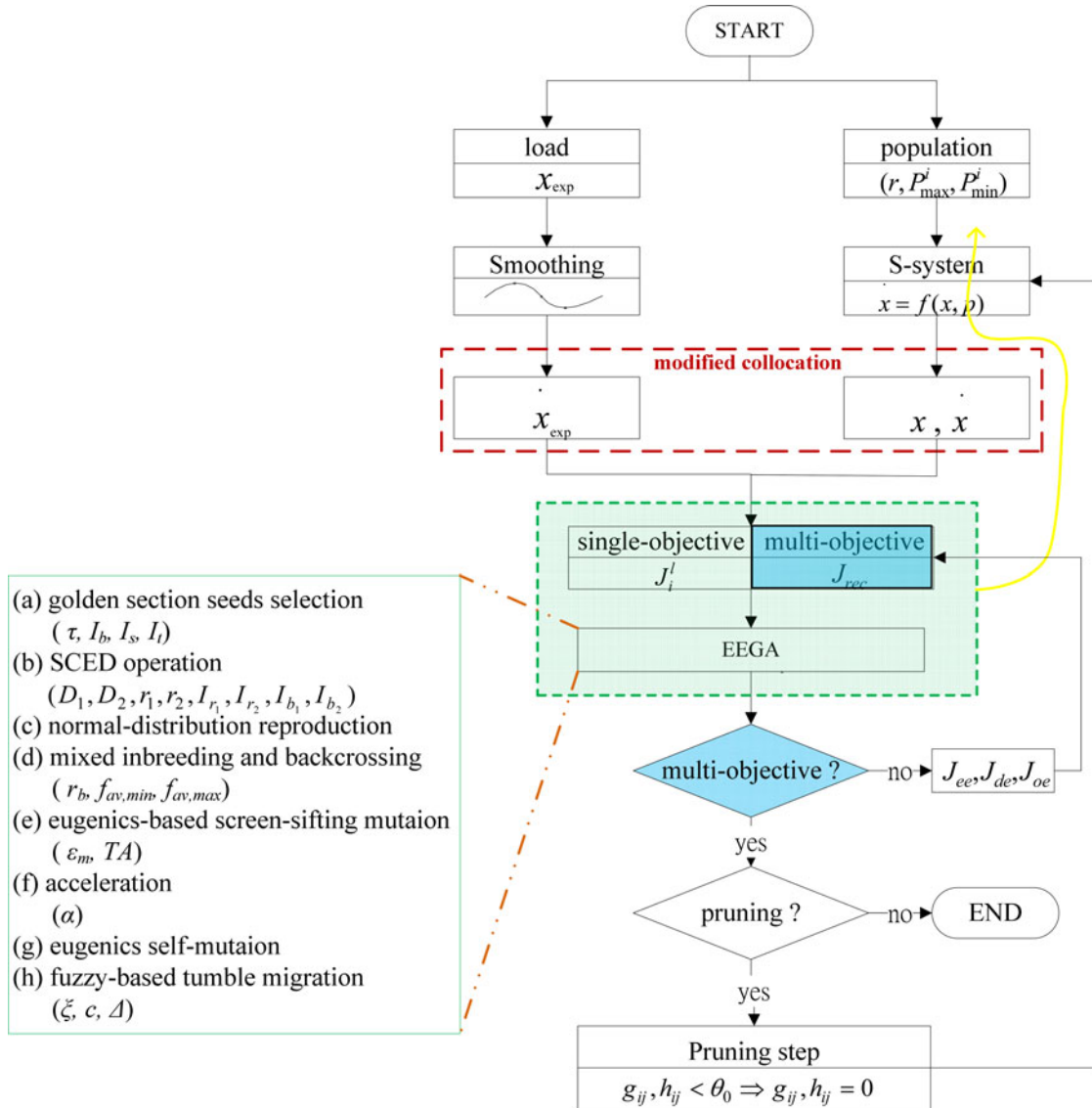


Fig. 8. Logical flow of FRMOGA for skeletal-network structure identification.

describes the proposed FRMOGA for skeletal-network structure identification.

III. S-SYSTEM MODELING OF GENE REGULATION NETWORKS

We now examine the proposed FRMOGA by identifying three biological systems (genetic branch pathway, cascade pathway, and small-scale genetic networks). Search space is set to be $[0, 30]$ for rate constants and $[-4, 4]$ for kinetic orders. In this paper, no further assumption for self-interaction (g_{ii} and h_{ii}) is made. All dependent and independent variables are included in an S-system. In other words, our learning is based on the super structure of an S-system model, where there are $2n(n + m + l)$ parameters to be identified. Table V in the Appendix lists the assumptions, dataset, and pruning thresholds used in the published papers. We use a cubic spline technology to get a smooth profile for time series data. Then, an integral-based-modified collocation method with piecewise linear Lagrange

polynomials as shape polynomials is adopted to approximate the generated dynamic profile [18]

$$x_i(t_l) = x_i(t_{l-1}) + 0.5\eta_l \{f_i(x_i(t_l), \theta) + f_i(x_i(t_{l-1}), \theta)\}$$

$$i = 1, \dots, n, \quad l = 1, \dots, N_S \quad (19)$$

where $x_i(t_l)$ and $f_i(x_i(t_l), \theta)$ are the expansion coefficients of the i th state and rate functions at the l th collocation point, and η_l is a time interval. Decoupled technology is adopted to get initial values for entire coupled S-type system learning.

A. Genetic Branch Pathway System

Eight sets of concentration data are generated from a true S-system; parameters are listed in Step 0 in Table II. Simulation time is set to be 8 s and sample time is 0.02. In this system, there are 48 parameters that are to be identified. Table II shows

TABLE II
TRUE AND ESTIMATED PARAMETERS OF AN S-TYPE SYSTEM FOR A BRANCH PATHWAY NETWORK IN FIG. 6(a)

Step	Variable	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	g_{i5}
0	x_1	20			-0.8		1
	x_2	8	0.5				
	x_3	3		0.75			
	x_4	2	0.5				
1	x_1	2.0213926E+01	2.9542619E-19	-2.7686232E-20	-7.7596018E-01	-3.6074108E-21	9.6914129E-01
	x_2	7.7341183E+00	5.0817954E-01	-1.2897465E-17	-8.1778006E-22	2.4361380E-20	-1.1246365E-21
	x_3	1.7384759E+00	-1.3891512E-17	9.5124850E-01	-1.2423557E-01	-6.6806669E-02	-1.1455635E-20
	x_4	2.1254852E+00	4.7163139E-01	6.8010305E-20	-7.8685546E-17	2.0875556E-19	4.3706924E-18
2	x_1	1.9997492E+01			-7.9985555E-01		9.9980586E-01
	x_2	7.9992661E+00	5.0004532E-01				
	x_3	3.2928899E+00		7.0991563E-01	5.3149755E-17	-7.2629749E-18	
	x_4	2.0000753E+00	4.9996471E-01				
3	x_1	1.9999710E+01			-8.0001586E-01		1.0000221E+00
	x_2	8.0001605E+00	4.9999047E-01				
	x_3	2.9999711E+00		7.5000521E-01			
	x_4	2.0000069E+00	4.9999807E-01				
Step	Variable	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	h_{i5}
0	x_1	10	0.5				
	x_2	3		0.75			
	x_3	5			0.5	0.2	
	x_4	6				0.8	
1	x_1	1.0319968E+01	4.8459428E-01	-1.6663995E-16	2.1738278E-16	-2.1797356E-16	-7.1926000E-17
	x_2	2.8551550E+00	-3.6875381E-17	7.6143539E-01	4.5844747E-23	-1.7584962E-18	-2.0146542E-20
	x_3	3.6484280E+00	-4.0181221E-21	-1.3548262E-01	6.0127279E-01	2.2377854E-01	-1.7665998E-19
	x_4	6.0478673E+00	-7.1963973E-19	3.3405014E-18	1.2849007E-16	7.5944616E-01	2.8543143E-18
2	x_1	1.0000123E+01	4.9991497E-01				
	x_2	2.9994523E+00		7.5006962E-01			
	x_3	5.3310011E+00		7.7174467E-21	4.7247779E-01	1.8833941E-01	
	x_4	5.9999186E+00				7.9995633E-01	
3	x_1	9.9997289E+00	5.0001243E-01				
	x_2	3.0001190E+00		7.4998464E-01			
	x_3	4.9999858E+00			5.0000240E-01	2.0000247E-01	
	x_4	6.0000076E+00				7.9999765E-01	

Step 0 lists the parameters of a true S-system. Steps 1--3 show estimated parameters and inferred structure.

TABLE III
TRUE AND ESTIMATED PARAMETERS OF AN S-TYPE SYSTEM FOR A CASCADE PATHWAY NETWORK IN FIG. 6(b)

Step	Variable	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}
0	x_1	10		-0.1	-0.05	1
	x_2	2	0.5			
	x_3	3		0.5		
1	x_1	1.3184830E+01	2.7902452E-17	-6.3869379E-02	-3.6599590E-02	6.5849933E-01
	x_2	1.9722535E+00	4.9636622E-01	6.1479326E-18	9.4662519E-19	-2.8361325E-18
	x_3	3.7347711E+00	7.4718549E-03	3.7992538E-01	3.2379347E-18	4.6108057E-18
2	x_1	9.9950744E+00		-1.0006935E-01	-5.0029445E-02	1.0006459E+00
	x_2	2.000010E+00	4.9998783E-01			
	x_3	2.9853968E+00	-3.2348485E-17	5.0175137E-01		
3	x_1	9.9999762E+00		-1.0000043E-01	-5.0000252E-02	1.0000041E+00
	x_2	2.0000040E+00	5.0002488E-01			
	x_3	2.9998517E+00		4.9999891E-01		
Step	Variable	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}
0	x_1	5	0.5			
	x_2	1.44		0.5		
	x_3	7.2			0.5	
1	x_1	8.4097688E+00	3.2571761E-01	1.0463913E-16	-3.0548374E-17	-5.5032187E-18
	x_2	1.4123528E+00	1.3525760E-18	5.0099319E-01	-2.2086675E-17	-1.6228509E-17
	x_3	7.3040329E+00	6.5479067E-18	3.5107504E-17	3.8270780E-01	-4.4658564E-17
2	x_1	4.9952221E+00	5.0033529E-01			
	x_2	1.4399719E+00		5.0001106E-01		
	x_3	7.1886307E+00			5.0196660E-01	
3	x_1	4.9999742E+00	4.9999861E-01			
	x_2	1.4400034E+00		5.0000203E-01		
	x_3	7.1999582E+00			5.0002460E-01	

Step 0 lists the parameters of a true S-system. Steps 1--3 show estimated parameters and inferred structure.

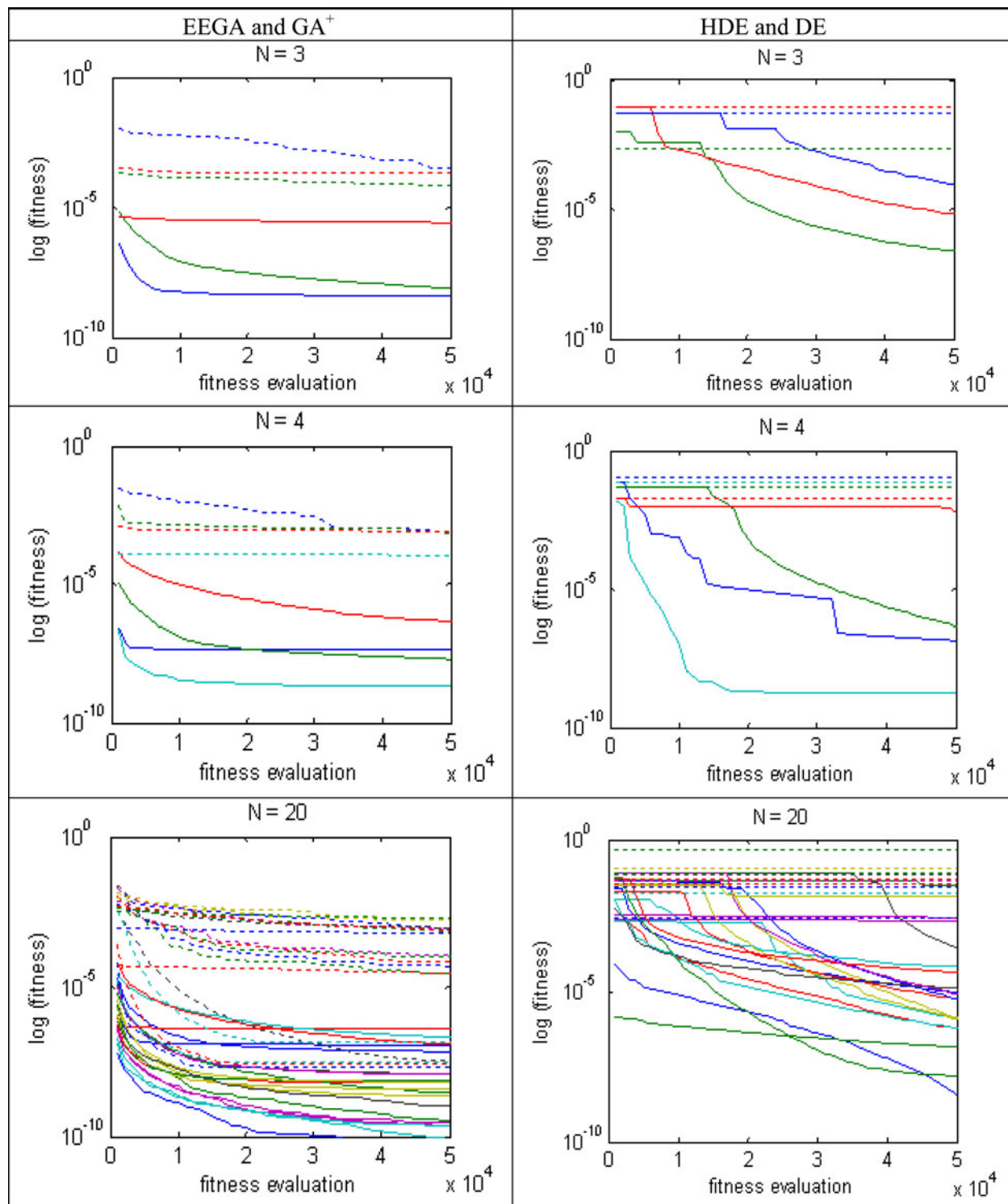


Fig. 9. Convergence results of EEGA, HDE [15], [17], [18], improved GA [42], and DE are examined by a wide search space ($[0, 100]$ for rate constants and $[-100, 100]$ for kinetic orders) with a bad initial start (80 for all parameters). Solid curves represent the estimated profiles of EEGA and HDE; dashed curves represent those of the improved GA (GA^+) and DE. Those curves are drawn for fitness evaluation from 1000 to 50 000.

that only two-step pruning action is taken. An obvious value gap exists between redundant interactions and possible interactions in Step 1. We subtract redundant interactions denoted by underlines with a threshold 10^{-15} . The pruned structure is further learned by FRMOGA to get the modified structure in Step 2. Step 3 shows the finalized structure. The inferred structure

is identical to true structure (Step 0), and parameter values are nearly the same as those in the true system.

B. Cascade Pathway System

Eight sets of concentration data are generated from *true* equations with parameters in Step 0 in Table III. Simulation is from

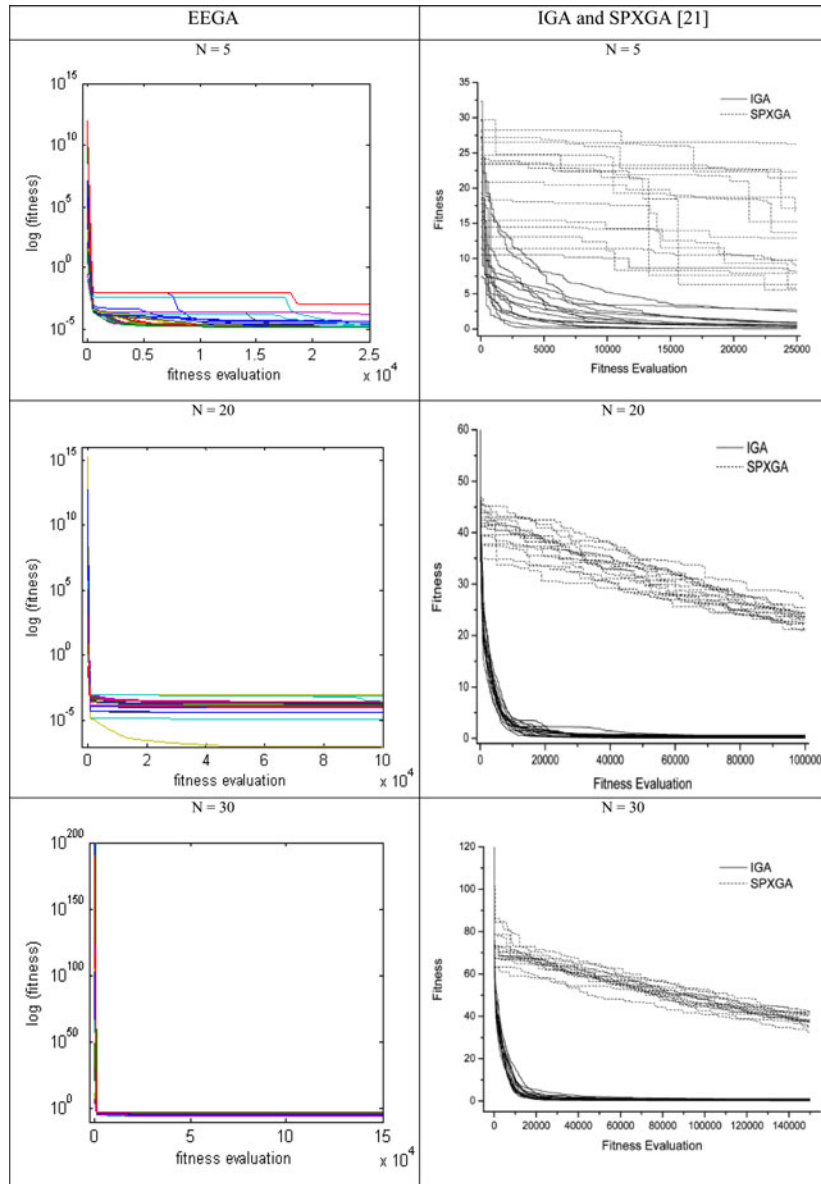


Fig. 10. Convergence comparison of EEGA with IGA and SPXGA cited from [21, Fig. 2] for a small-scale (five genes), a medium-scale (20 genes), and large-scale (30 genes) genetic networks. Six-set time series data with 11 sample points are used.

time $t = 0$ s to $t = 8$ s with a sample time of 0.02. The corresponding super structure of the S-system has 30 parameters that are to be identified. Step 1 in Table III shows the first-time learning result. Those interactions with kinetic order less than 10^{-15} are deleted. The pruned model is further modified into Step 2 in Table III by FRMOGA. After re truncating trivial terms and performing learning again, we have the final structure and associated parameters in Step 3 in Table III.

C. Small-Scale Genetic Network

Our third case is a small-scale genetic network with two regulatory signals. *True* rate constants and kinetic orders are listed in Step 0 of Table V in the Appendix. Each experiment is simulated from time $t = 0$ s to $t = 0.5$ s with a sample time of 0.0125. In this system, there are 90 parameters that are to

be identified. Inferred parameters and kinetic orders are shown in Steps 1 and 2. Pruning threshold is also set to be 10^{-15} . Those redundant connections to be truncated are denoted by underlines. In this system, only two-step learning is taken.

D. Convergence Comparison

Figs. 11–13, shown in the Appendix, describe the convergence of each step of EEGA for those three systems. Table V gives the comparison of the used dataset, assumptions, S-system models, and pruning thresholds for EEGA and the published computation algorithms. Pruning ratio ρ is defined as the ratio of the smallest preserved term to pruning threshold. A safety pruning is taken by EEGA with $\rho = 7.5 \times 10^{12}$. Except mutation and crossover, EEGA introduces fuzzy-based tumble migration

TABLE IV
TRUE AND ESTIMATED PARAMETERS OF AN S-TYPE SYSTEM FOR A SMALL-SCALE GENETIC NETWORK IN FIG. 6(c)

Step	Variable	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	g_{i5}	g_{i6}	g_{i7}	g_{i8}
0	x_1	5			1		-1	1		
	x_2	10	2						1	
	x_3	10		-1						1
	x_4	8			2		-1	1		
	x_5	10				2			1	
1	x_1	2.41256E+01	-3.89491E-18	-1.50184E-17	8.62468E-02	1.95320E-17	-1.60767E-01	1.39026E-01	-6.07649E-18	-6.69010E-18
	x_2	1.01404E+01	1.03706E+00	5.51269E-17	-1.53107E-17	3.84628E-17	-3.80445E-17	-2.10825E-16	4.68351E-01	5.57981E-17
	x_3	4.64020E+01	1.64492E-22	-3.73574E-01	-1.39241E-18	-4.98538E-24	-7.32593E-22	-9.73978E-22	-6.27446E-24	1.96541E-01
	x_4	2.83502E+01	-1.15628E-17	-5.07464E-17	2.02335E-01	-3.10366E-17	-2.25903E-01	8.61354E-02	7.41771E-19	-1.13540E-17
	x_5	1.44916E+01	-6.33232E-18	2.06808E-17	7.15398E-18	4.84820E-01	6.01047E-17	-9.45191E-17	3.14008E-01	2.92453E-17
2	x_1	5.00034E+00			9.99947E-01		-9.99924E-01	9.99948E-01		
	x_2	1.00008E+01	2.00022E+00						1.00002E+00	
	x_3	9.99864E+00		-1.00006E+00						1.00012E+00
	x_4	7.99995E+00			2.00004E+00		-1.00002E+00	1.00001E+00		
	x_5	9.99953E+00				2.00009E+00			1.00001E+00	
Step	Variable	β_i	h_{i1}	h_{i2}	h_{i3}	h_{i4}	h_{i5}	h_{i6}	h_{i7}	h_{i8}
0	x_1	10	2							
	x_2	10		2						
	x_3	10		-1	2					
	x_4	10				2				
	x_5	10					2			
1	x_1	2.59420E+01	1.83121E-01	2.96902E-17	1.74753E-17	-1.87971E-18	5.22547E-19	1.19782E-17	-5.24342E-17	1.38371E-19
	x_2	9.50466E+00	9.03112E-17	9.00464E-01	1.32261E-16	1.59152E-16	9.38945E-17	1.08191E-16	-6.48470E-17	-2.13658E-16
	x_3	4.68699E+01	5.33819E-21	-3.45276E-01	3.75987E-01	-1.98393E-20	1.62066E-22	-2.42022E-24	8.63917E-21	-3.44344E-17
	x_4	2.92271E+01	-4.41497E-19	4.47306E-17	1.91396E-18	1.39647E-01	4.00889E-17	-1.29410E-18	-1.78564E-19	3.42182E-17
	x_5	1.43446E+01	4.74966E-18	-5.89542E-17	3.20721E-16	2.53737E-17	5.82800E-01	-1.46697E-17	-3.90070E-16	8.08610E-17
2	x_1	1.00004E+01	1.99993E+00							
	x_2	1.00008E+01		2.00015E+00						
	x_3	9.99856E+00		-1.00006E+00	2.00026E+00					
	x_4	1.00000E+01				2.00003E+00				
	x_5	9.99957E+00					2.00011E+00			

Step 0 lists the parameters of a true S-system. Steps 1 and 2 show estimated parameters and inferred structure.

to escape from local minima. Some advanced operations on the best-so-far individual are to accelerate convergence. To show the global search power, a *bad* initial start in a *very wide* range is used in Fig. 9. Fig. 9 shows the convergence comparison of EEGA with DE, HDE [15], [17], [18] and our previously work improved GA (GA⁺) [42] for low- and medium-dimensional systems ($N = 3, 4$, and 20 genes) by a wide search space ([0, 100] for rate constants and [-100, 100] for kinetic orders) with a bad initial start (80 for all parameters). Fig. 10 shows convergence results of EEGA, intelligent two-stage evolutionary algorithm (IGA) [21], and SPXGA [19] for low-, medium-, and high-dimensional systems ($N = 5, 20$, and 30 genes) with the same search region as [21] ([0, 15] for rate constants and [-3, 3] for kinetic orders).

IV. CONCLUSION

The inverse problem of identifying a dynamic biological system from time series data is a central theme in systems biology. However, even if multiobjective computation approaches are used, structure identification is still a big challenge. How to avoid from sticking into local minima is very important to infer such a high dimensional and nonlinear system by computation approach. Additionally, a big gap between true and redundant interactions is necessary to ensure a correct struc-

ture, since noise and uncertainty exist in a biological system. In other words, the pruning threshold should be small enough. The used technology should possess good search power and has the ability to escape from local minima. In this paper, we propose an EEGA learning technology to achieve the improvement of the best-so-far individuals and to ensure global optimal search. A fuzzy-based multiobjective construction performance index is derived to infer a physically reasonable regulation network. Simulation results in Tables II–IV show that a very big gap exists in true and redundant interactions. The truncation threshold can be decreased down to 10^{-15} , and a correct structure is achieved after only one- or two-step pruning operations. Table V in the Appendix shows our pruning action is absolutely safe as compared with that in the published papers. Fig. 9 shows the good global search power of EEGA with a bad initial start (80 for all parameters) in a wide search space ([0, 100] for rate constants and [-100, 100] for kinetic orders) for three-, four-, and 20-dimensional systems. In this paper, we did not add noise in our time series data. Most researches use existing smoothing or filtering technique to deal with noise. However, those approaches depend too much on smoothing skills and lack clear guidelines, especially for structure identification. In the future, we shall propose a new estimator (filter) to solve those data seriously contaminated by white noise and color noise, or systems with bias and uncertainty.

APPENDIX

TABLE V
STRUCTURE IDENTIFICATION COMPARISON

paper	algorithm	system	data set	Noise (Gaussian)	assumption	super model	Pruning threshold (β)	Pruning ratio (ρ)
this paper		Fig. 7a, 7b, 7c	8 sets of time-series data are generated from S-system models.	no	no	Yes	10^{-15}	7.5×10^{12}
Voit and Almeida [20]		Fig. 7a	Artificial datasets are created from S-system model [20].	no	$g_i = 0, h_i > 0$	No	0.1	1.4
Tsai and Wang [15]	hybrid differential evolution (HDE)	Fig. 7b	10 sets 'true' time-course data are generated [15].	no	$g_i = 0, h_i > 0$	Yes	0.01	2.7
		Fig. 7c	10 sets 'true' time-course data are generated [15].	no	$g_i = 0, h_i > 0$	No	0.1	1.2
Kimura <i>et al.</i> [12]	cooperative coevolutionary algorithm	Fig. 7c	15 sets of noise-free time-series data are obtained by solving the set of differential equations [12].	no	-	No	0.001	-
		30 genes	20 sets of time-series data are obtained by solving the set of differential equations [12].	10%	<ul style="list-style-type: none"> * no simulation results * Weakly interactions were, especially, difficult to be correctly inferred, e.g. both methods often failed to infer interactions corresponding to $g_{1,14}$, $g_{24,18}$ and $g_{26,28}$ [12]. 			
Noman and Iba [14]	Differential Evolution (DE)	Fig. 7c	10 sets of time-series are obtained by solving the set of differential equations [14].	no	-	No	0.05	1
Gonzalez <i>et al.</i> [22]	Simulated Annealing	Fig. 7a	Artificial time courses datasets are created by using the S-system formulation Equation [22].	no	$g_i = 0, h_i > 0$	Yes	0.01	-
				10%	no simulation results			
Ho <i>et al.</i> [21]	intelligent two-stage evolutionary algorithm (IGA)	Fig. 7c	15 sets of noise-free time-series data are obtained by using differential equations [21].	no	-	No	0.03	-
		10 genes	-	no	-	No	0.1	3*
		5, 10, 15, 30 genes	-	3, 5 and 10%	no simulation results			
Liu and Wang [30]	hybrid differential evolution (HDE)	Fig. 7c	The eight sets of training data are generated by solving the set of differential equations [30].	no	$g_i = 0, h_i > 0$	Yes	0.01	6.9
			10% random noises are added into the eight sets of 'true' time-course data [30].	10 %	$g_i = 0, h_i > 0$	Yes	0.01	-
						g_{57} , g_{44} , h_{48} (second pruning), g_{42} (third pruning) and g_{35} (fifth pruning) were truncated beyond thresholds.		

- ♦ Super model is an augmented S-type model which includes independent terms.
- ♦ Pruning ratio ρ is the ratio of the smallest preserved term to pruning threshold β .
- ♦ 3* denotes the ratio β is from the final pruning step (Authors do not provide entire pruning steps.).
- ♦ "-" denotes no information is provided in the published papers.

TABLE VI
SINGLE-OBJECTIVE PERFORMANCE INDEX

entire performance		ith-individual performance	entire performance		ith-individual performance
Summation	Maximum		Summation	Maximum	
$J_i^S = \frac{1}{N} \sum_{j=1}^N J_{ji}$	$J_i^M = \max\{J_{i,1}, \dots, J_{i,N}\}$		$J_i^S = \frac{1}{N} \sum_{j=1}^N J_{ji}$	$J_i^M = \max\{J_{i,1}, \dots, J_{i,N}\}$	
J_1^S	J_1^M	$J_{1i} = \left(\frac{x^i - x_{\text{exp}}^i}{\max(x_{\text{exp}}^i)} \right)^2$	J_9^S	J_9^M	$J_{9i} = t_a \left(\frac{x^i - x_{\text{exp}}^i}{\max(x_{\text{exp}}^i)} \right)^2$
J_2^S	J_2^M	$J_{2i} = t_s \left(\frac{x^i - x_{\text{exp}}^i}{\max(x_{\text{exp}}^i)} \right)^2$	J_{10}^S	J_{10}^M	$J_{10i} = \left(\frac{x^i - x_{\text{exp}}^i}{\max(x_{\text{exp}}^i)} \right)^2$
J_3^S	J_3^M	$J_{3i} = t_a \left(\frac{x^i - x_{\text{exp}}^i}{\max(x_{\text{exp}}^i)} \right)^2$	J_{11}^S	J_{11}^M	$J_{11i} = t_s \left(\frac{x^i - x_{\text{exp}}^i}{\max(x_{\text{exp}}^i)} \right)^2$
J_4^S	J_4^M	$J_{4i} = \sqrt{\left(\frac{x^i - x_{\text{exp}}^i}{\max(x_{\text{exp}}^i)} \right)^2}$	J_{12}^S	J_{12}^M	$J_{12i} = t_a \left(\frac{x^i - x_{\text{exp}}^i}{\max(x_{\text{exp}}^i)} \right)^2$
J_5^S	J_5^M	$J_{5i} = t_s \sqrt{\left(\frac{x^i - x_{\text{exp}}^i}{\max(x_{\text{exp}}^i)} \right)^2}$	J_{13}^S	J_{13}^M	$J_{13i} = \sqrt{\left(\frac{x^i - x_{\text{exp}}^i}{\max(x_{\text{exp}}^i)} \right)^2}$
J_6^S	J_6^M	$J_{6i} = t_a \sqrt{\left(\frac{x^i - x_{\text{exp}}^i}{\max(x_{\text{exp}}^i)} \right)^2}$	J_{14}^S	J_{14}^M	$J_{14i} = \sqrt{\left(\frac{x^i - x_{\text{exp}}^i}{\max(x_{\text{exp}}^i)} \right)^2}$
J_7^S	J_7^M	$J_{7i} = \left(\frac{\dot{x}^i - \dot{x}_{\text{exp}}^i}{\max(\dot{x}_{\text{exp}}^i)} \right)^2$			
J_8^S	J_8^M	$J_{8i} = t_s \left(\frac{\dot{x}^i - \dot{x}_{\text{exp}}^i}{\max(\dot{x}_{\text{exp}}^i)} \right)^2$			

- ♦ J_1^i, \dots, J_6^i are defined on concentration error and J_7^i, \dots, J_{14}^i are on slope error ($i = M$ or S).
- ♦ Normalization is used to ensure comparable competition in different-scale species.
- ♦ $x^i (i = 1, \dots, n)$ is the i th estimated concentration, x_{exp}^i is the i th measured concentration, and $\max(x_{\text{exp}}^i)$ is the maximum of measured concentrations; \dot{x}^i is the i th estimated slope, \dot{x}_{exp}^i is the i th measured slope, and $\max(\dot{x}_{\text{exp}}^i)$ is the maximum of measured slopes.
- ♦ t_s and t_a are time-weighting factors increasing and decreasing with time, respectively.
- ♦ N is the number of data and n is the number of states.

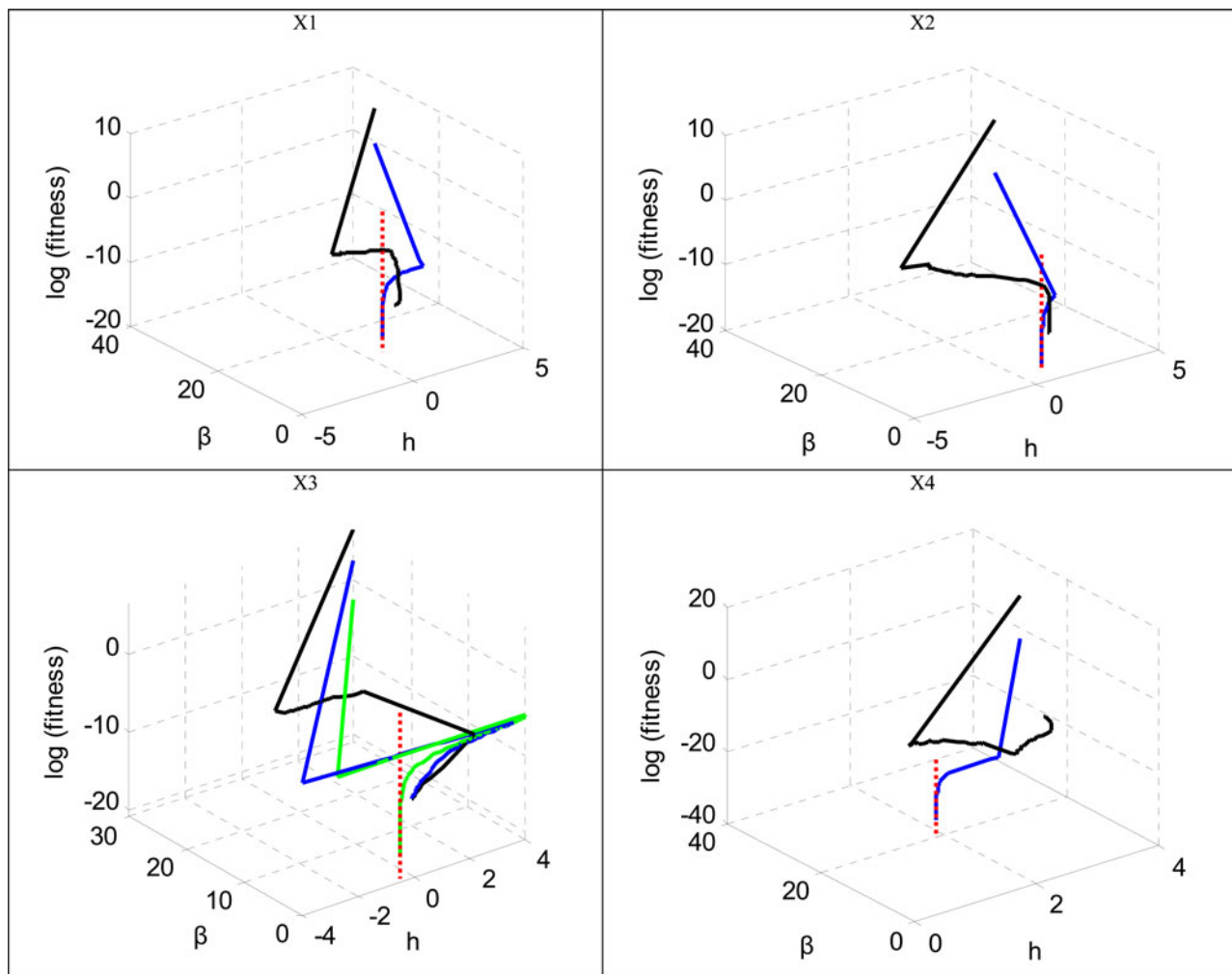


Fig. 11. Convergence of EEGA for each step in the structure identification of a branch network in Fig. 6(a) starting from $(\beta_i, h_{ii}) = (30, 4)$. Dash line denotes true value. Black solid curve shows the convergence of Step 1, blue for Step 2, and Green for Step 3.

TABLE VII
PERFORMANCE COMPARISON FOR A GENETIC BRANCH PATHWAY IN FIG. 6(a)

Error criterion , $E = \frac{1}{N} \sum_{i=1}^N \left(\frac{x^i - x_{exp}^i}{\max(x^i)} \right)^2$			
Sum	Optimal fitness	Max	Optimal fitness
J_1^S	4.5835935E-10	J_1^M	5.0488280E-08
J_2^S	1.1989171E-09	J_2^M	5.7832483E-08
J_3^S	4.5960822E-11	J_3^M	1.0699997E-10
J_4^S	3.4262501E-07	J_4^M	3.0536607E-06
J_5^S	1.1229400E-04	J_5^M	7.7593655E-05
J_6^S	1.8479043E-07	J_6^M	2.2431690E-05
J_7^S	4.7749491E-07	J_7^M	5.5962543E-08
J_8^S	1.0961241E-08	J_8^M	3.9076341E-05
J_9^S	2.9665960E-10	J_9^M	7.8697814E-08
J_{10}^S	4.8324012E-11	J_{10}^M	1.7753159E-09
J_{11}^S	1.2970259E-08	J_{11}^M	4.7993926E-07
J_{12}^S	1.2807886E-10	J_{12}^M	3.5800163E-07
J_{13}^S	1.8329152E-06	J_{13}^M	5.1474803E-06
J_{14}^S	3.9287059E-05	J_{14}^M	6.0951341E-06

TABLE VIII
PERFORMANCE COMPARISON FOR A CASCADE PATHWAY SYSTEM IN FIG. 6(b)

Error criterion , $E = \frac{1}{N} \sum_{i=1}^N \left(\frac{x^i - x_{exp}^i}{\max(x^i)} \right)^2$			
Sum	Optimal fitness	Max	Optimal fitness
J_1^S	5.7719659E-09	J_1^M	7.1645410E-09
J_2^S	1.1456407E-05	J_2^M	7.7100757E-05
J_3^S	5.7623443E-12	J_3^M	5.1791243E-10
J_4^S	1.0183205E-05	J_4^M	1.5735404E-05
J_5^S	1.4208229E-04	J_5^M	7.1913330E-05
J_6^S	1.7395375E-09	J_6^M	6.5643269E-08
J_7^S	5.1764368E-11	J_7^M	1.9206372E-08
J_8^S	1.1109550E-05	J_8^M	5.7691971E-06
J_9^S	2.3851452E-09	J_9^M	9.9108335E-09
J_{10}^S	1.4667368E-07	J_{10}^M	1.5248148E-06
J_{11}^S	6.7262942E-05	J_{11}^M	5.6203930E-05
J_{12}^S	1.4635411E-06	J_{12}^M	3.1027507E-06
J_{13}^S	5.1961041E-09	J_{13}^M	4.9981500E-07
J_{14}^S	1.5947509E-05	J_{14}^M	3.9171596E-05

TABLE IX
UPDATED TERMS IN SELF-INTERACTIVE LEARNING

	iteration		$\text{Max}(\underline{J}_e, \underline{J}_0)$	$\text{Max}(\underline{J}_d, \underline{J}_0)$	Updated term $\text{Max}(\underline{J}_e, \underline{J}_0, \underline{J}_d)$
	phase I	early stage	k=0	\underline{J}_e	\underline{J}_d
⋮			⋮	⋮	⋮
k=m-1		\underline{J}_e	\underline{J}_d	\underline{J}_e or \underline{J}_d	
middle stage		k=m	\underline{J}_e or \underline{J}_0	\underline{J}_d or \underline{J}_0	\underline{J}_e or \underline{J}_d or \underline{J}_0
	⋮	⋮	⋮	⋮	
		k=l-1	\underline{J}_e or \underline{J}_0	\underline{J}_d or \underline{J}_0	\underline{J}_e or \underline{J}_d or \underline{J}_0
phase II	final stage	k=l	\underline{J}_0	\underline{J}_0	\underline{J}_0
		⋮	⋮	⋮	⋮
		⋮	⋮	⋮	⋮
		k=K _{ic}	\underline{J}_0	\underline{J}_0	\underline{J}_0

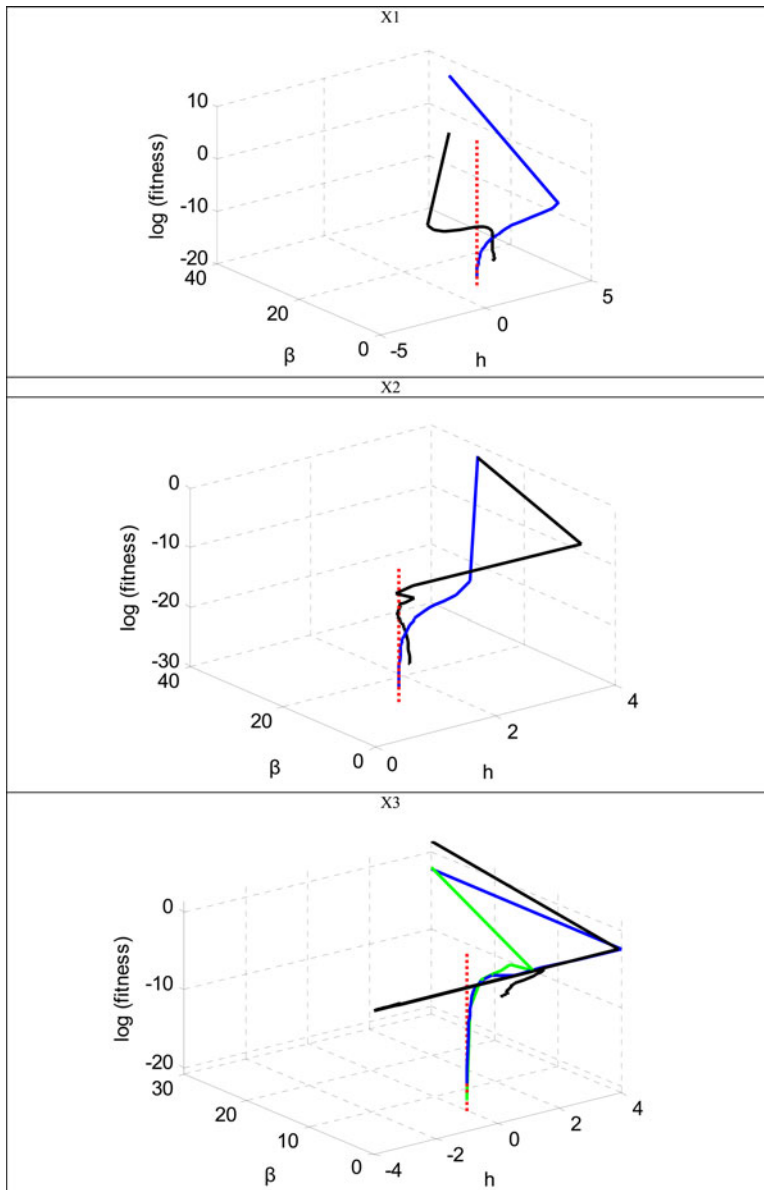


Fig. 12. Convergence of EEGA for each step in the structure identification of a cascade network in Fig. 6(b) starting from $(\beta_i, h_{ii}) = (30, 4)$. Dash line denotes true value. Black solid curve shows the convergence of Step 1, blue for Step 2, and Green for Step 3.

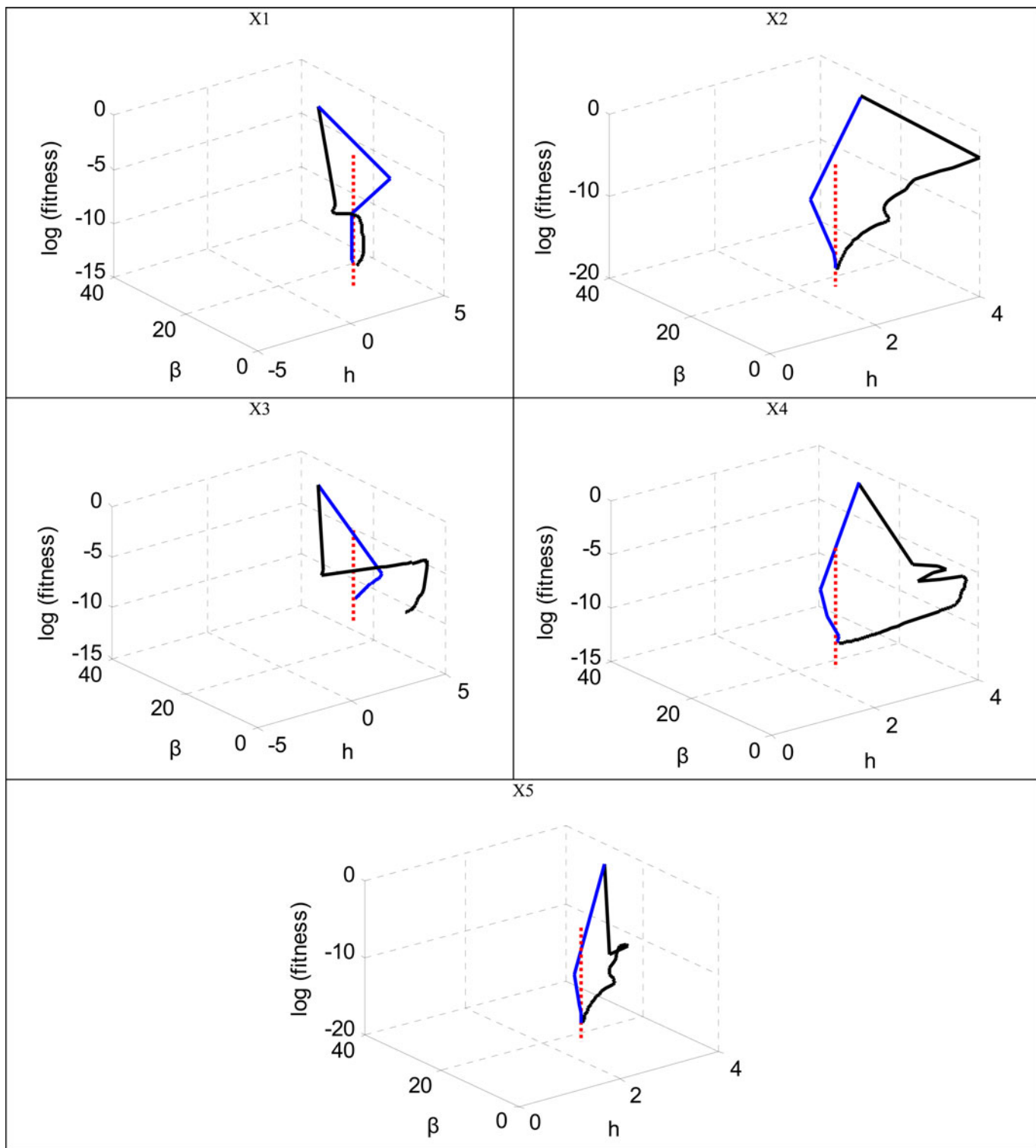


Fig. 13. Convergence of EEGA for each step in the structure identification of a small-scale genetic network in Fig. 6(c) starting from $(\beta_i, h_{ii}) = (30, 4)$. Dash line denotes true value. Black solid curve shows the convergence of Step 1 and blue for Step 2.

ACKNOWLEDGMENT

The authors would like to thank Prof. F.-S. Wang of the Chemical Engineering Department, National Chung-Chen University, for his help with reverse engineering technologies.

REFERENCES

- [1] M. Vilela, I. C. Chou, S. Vinga, A. T. R. Vasconcelos, E. O. Voit, and J. S. Almeida, "Parameter optimization in S-system models," *BMC Syst. Biol.*, vol. 2, no. 35, 2008.
- [2] L. Michaelis and M. I. Menton, "Die kinetik der invertinwirkung," *Biochem. Z.*, vol. 49, pp. 333–369, 1913.
- [3] M. A. Savageau, *Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology*. Reading, MA: Addison-Wesley, 1976.
- [4] E. O. Voit, *Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists*. Cambridge, U.K.: Cambridge Univ. Press, 2000.
- [5] S. Marino and E. O. Voit, "An automated procedure for the extraction of metabolic network information from time series data," *Bioinform. Comput. Biol.*, vol. 4, no. 3, pp. 665–691, 2006.
- [6] I. C. Chou, H. Martens, and E. O. Voit, "Parameter estimation in biochemical systems models with alternating regression," *Theor. Biol. Med. Model.*, vol. 3, no. 25, 2006.
- [7] Z. Kutalik, W. Tucker, and V. Moulton, "S-system parameter estimation for noisy metabolic profiles using Newton flow analysis," *IET Syst. Biol.*, vol. 1, pp. 174–180, 2007.
- [8] E. Sakamoto and H. Iba, "Inferring a system of differential equations for a gene regulatory network by using genetic programming," in *Proc. Congr. Evol. Comput.*, 2001, vol. 1, pp. 720–726.
- [9] S. Ando, E. Sakamoto, and H. Iba, "Evolutionary modeling inference of gene network," *Inf. Sci.*, vol. 145, pp. 237–259, 2002.
- [10] D. Y. Cho, K. H. Cho, and B. T. Zhang, "Identification of biochemical networks by S-tree based genetic programming," *Bioinform.*, vol. 22, pp. 1631–1640, 2006.
- [11] H. Wang, L. Qian, and E. Dougherty, "Inference of gene regulatory networks using S-system: A unified approach," *IET Syst. Biol.*, vol. 4, pp. 145–156, 2010.
- [12] S. Kimura, K. Ide, A. Kashihara, M. Kano, H. Mariko, R. Masui, N. Nakagawa, S. Yokoyama, S. Kuramitsu, and A. Konagaya, "Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm," *Bioinform.*, vol. 21, pp. 1154–1163, 2005.
- [13] C. G. Moles, P. Mendes, and J. R. Banga, "Parameter estimation in biochemical pathways: A comparison of global optimization methods," *Genome Res.*, vol. 13, pp. 2467–2474, 2003.
- [14] N. Noman and H. Iba, "Inference of gene regulatory networks using S-system and differential evolution," in *Proc. Conf. Genetic Evol. Comput.*, 2005, vol. 1, pp. 439–446.
- [15] K. Y. Tsai and F. S. Wang, "Evolutionary optimization with data collocation for reverse engineering of biological networks," *Bioinform.*, vol. 21, pp. 1180–1188, 2005.
- [16] S. J. Wu, C. T. Wu, and T. T. Lee, "Computation intelligent for eukaryotic cell-cycle gene network," in *Proc. IEEE Eng. Med. Biol. Soc. Conf.*, 2006, pp. 2017–2020.
- [17] P. K. Liu and F. S. Wang, "Hybrid differential evolution with geometric mean mutation in parameter estimation of bioreaction systems with large parameter search space," *Comput. Chem. Eng.*, vol. 33, pp. 1851–1860, 2009.
- [18] F. S. Wang and P. K. Liu, "Inverse problems of biochemical systems using hybrid differential evolution and data collocation," *Int. J. Syst. Synthetic Biol.*, vol. 1, pp. 21–38, 2010.
- [19] S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, and M. Tomita, "Dynamic modeling of genetic networks using genetic algorithm and S-system," *Bioinform.*, vol. 19, pp. 643–650, 2003.
- [20] E. O. Voit and J. Almeida, "Decoupling dynamical systems for pathway identification from metabolic profiles," *Bioinformatics*, vol. 20, pp. 1670–1681, 2004.
- [21] S. Y. Ho, C. H. Hsieh, F. C. Yu, and H. L. Huang, "An intelligent two-stage evolutionary algorithm for dynamic pathway identification from gene expression profiles," *IEEE/ACM Trans. Comput. Biol. Bioinform.*, vol. 4, no. 4, pp. 648–660, Oct./Dec. 2007.
- [22] O. R. Gonzalez, C. Küper, K. Jung, P. C. Naval, E. Mendoza, Jr., and E. Mendoza, "Parameter estimation using simulated annealing for S-system models of biochemical networks," *Bioinform.*, vol. 23, pp. 480–486, 2007.
- [23] C. M. Chen, C. Lee, C. L. Chuang, C. C. Wang, and G. S. Shieh, "Inferring genetic interactions via a nonlinear model and an optimization algorithm," *BMC Syst. Biol.*, vol. 4, no. 16, 2010.
- [24] Y. Matsubara, S. Kikuchi, M. Sugimoto, and M. Tomita, "Parameter estimation for stiff equations of biosystems using radial basis function networks," *BMC Bioinform.*, vol. 7, no. 230, 2006.
- [25] H. Murata, M. Koshino, M. Mitamura, and H. Kimura, "Inference of S-system models of genetic networks using product unit neural networks," in *Proc. IEEE Conf. Syst. Man Cybern.*, 2008, pp. 1390–1395.
- [26] R. Xu, D. C. Wunsch II, and R. L. Frank, "Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization," *IEEE Trans. Comput. Biol. Bioinform.*, vol. 4, no. 4, pp. 1545–5963, Oct./Dec. 2007.
- [27] D. Thieffry, A. M. Huerta, E. Perez-Rueda, and J. Collado-Vides, "From specific gene regulation to genomic networks: A global analysis of transcriptional regulation in *Escherichia coli*," *BioEssays*, vol. 20, pp. 433–440, 1998.
- [28] P. K. Liu and F. S. Wang, "Inference of biochemical network models in S-system using multiobjective optimization approach," *Bioinform.*, vol. 24, pp. 1085–1092, 2008.
- [29] D. S. Falconer and T. F. C. MacKay, *Introduction to Quantitative Genetics*, 4th ed. White Plains, NY: Longman, 1996.
- [30] C. L. Ko, E. O. Voit, and F. S. Wang, "Estimating parameters for generalized mass action models with connectivity information," *BMC Bioinform.*, vol. 10, no. 140, 2009.
- [31] I. C. Chou and E. O. Voit, "Recent developments in parameter estimation and structure identification of biochemical and genomic systems," *Math. Biosci.*, vol. 219, pp. 57–83, 2009.
- [32] W. S. Hlavacek and M. A. Savageau, "Rules for coupled expression of regulator and effector genes in inducible circuits," *J. Mol. Biol.*, vol. 255, pp. 121–139, 1996.
- [33] M. J. Gacto, R. Alcalá, and F. Herrera, "Integration of an index to preserve the semantic interpretability in the multiobjective evolutionary rule selection and tuning of linguistic fuzzy systems," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 3, pp. 515–531, Jun. 2010.
- [34] C. J. Carmona, P. Gonzalez, M. J. del Jesus, and F. Herrera, "NMEEFSD: Non-dominated multiobjective evolutionary algorithm for extracting fuzzy rules in subgroup discovery," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 5, pp. 958–970, Oct. 2010.
- [35] G. P. Rangaiah, *Multi-Objective Optimization: Techniques and Applications in Chemical Engineering*. Hackensack, NJ: World Scientific, 2009.
- [36] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: NSGA-II," *IEEE Trans. Evol. Comput.*, vol. 6, no. 2, pp. 182–197, Apr. 2002.
- [37] K. Deb, "Current trends in evolutionary multi-objective optimization," *Int. J. Simul. Multidisci. Design Optim.*, vol. 1, pp. 1–8, 2007.
- [38] M. Sakawa and K. Kato, "An interactive fuzzy satisficing method for multiobjective nonlinear integer programming problems with block-angular structures through genetic algorithms with decomposition procedures," *Adv. Oper. Res.*, vol. 2009, 2009.
- [39] K. Deb, A. Sinha, P. J. Korhonen, and J. Wallenius, "An interactive evolutionary multiobjective optimization method based on progressively approximated value functions," *IEEE Trans. Evol. Comput.*, vol. 14, no. 5, pp. 723–739, Oct. 2010.
- [40] A. Sukstienwong, "Solving multiobjective optimization under bounds by genetic algorithms," *Int. J. Comput.*, vol. 5, pp. 18–25, 2011.
- [41] P. Pulkkinen and H. Koivisto, "A dynamically constrained multiobjective genetic fuzzy system for regression problems," *IEEE Trans. Fuzzy Syst.*, vol. 18, no. 1, pp. 161–177, Feb. 2010.
- [42] S. J. Wu, C. H. Chou, C. T. Wu, and T. T. Lee, "Inference of genetic network of xenopus frog egg: Improved genetic algorithm," in *Proc. IEEE Eng. Med. Biol. Soc. Conf.*, 2006, pp. 4147–4150.



Shinq-Jen Wu received the B.S. degree in chemical engineering from National Taiwan University, Taipei, Taiwan, in 1986, the M.S. degree in chemical engineering from National Tsing-Hua University, Hsinchu, Taiwan, in 1989, the M.S. degree in electrical engineering from the University of California, Los Angeles, in 1994, and the Ph.D. degree in electrical and control engineering from National Chiao-Tung University, Hsinchu, in 2000.

From September 1989 to July 1990, she was with the Laboratory for Simulation and Control

Technology, Chemical Engineering Division, Industrial Technology Research Institute, Hsinchu. She then joined the Chemical Engineering Department, Kao-Yuan Junior College of Technology and Commerce, Kaohsiung, Taiwan. From 1995 to 1996, she was an Engineer with the Integration Engineering Department, Macronix International Co., Ltd., Hsinchu. She is currently with the Department of Electrical Engineering, Da-Yeh University, Changhua, Taiwan. Her research interests include ergonomics-based smart cars, advanced vehicle control and safety systems, Petri-net modeling for cancer mechanisms, robust identification of genetic networks, soft sensor for online tuning, soft-computation-based protein structure prediction, very-large-scale integration process technology, optimal fuzzy control/tracking, optimal fuzzy estimation, and artificial-and-swarm intelligent modeling techniques.

Dr. Wu is a member of the Phi Tau Phi Scholastic Honor Society. She is the Editor of *Advances in Fuzzy Sets and Systems* (Pushpa, Allahabad, India). Her name is included in *Asian Admirable Achievers*, *Asia/American Who's Who*, *Asia/Pacifica Who's Who*, and in *Marquis Who's Who in Science and Engineering in the World/in America/in Asia*. She is a Life Fellow of the International Biographical Association. She is a Scientific Adviser to the International Biographical Centre (IBC) Director General. She received the 21st Century Award for Achievement from the IBC and *The Albert Einstein Award of Excellent* from the American Biographical Institute, Inc.



Cheng-Tao Wu received the M.S. degree in electrical engineering from Da-Yeh University, Changhua, Taiwan, in 2006.

He is currently with the Department of Electrical and Control Engineering National Chiao-Tung University, Hsinchu, Taiwan. He is involved in projects concerning optimization and detection in control systems.



Jyh-Yeong Chang (S'84–M'86) received the B.S. degree in control engineering in 1976 and the M.S. degree in electronic engineering in 1980, both from National Chiao-Tung University (NCTU), Hsinchu, Taiwan, and the Ph.D. degree in electrical engineering from North Carolina State University, Raleigh, in 1987.

During 1976–1978 and 1980–1982, he was a Research Fellow with the Chung Shan Institute of Science and Technology, Lung-Tan, Taiwan. In 1987, he was an Associate Professor with the Department of Electrical and Control Engineering, NCTU, where he is currently a Professor. His current research interests include neural fuzzy systems, video processing and surveillance, and bioinformatics.