Mathematical Biosciences 245 (2013) 299–313

Contents lists available at ScienceDirect

Mathematical Biosciences

journal homepage: www.elsevier.com/locate/mbs

Computational optimization for S-type biological systems: Cockroach genetic algorithm

Shinq-Jen Wu [a],*, Cheng-Tao Wu [b]

[a] Department of Electrical Engineering, Da-Yeh University, Chang-Hwa, Taiwan, ROC
[b] Department of Electrical and Control Engineering, National Chiao-Tung University, Hsin-Chu, Taiwan, ROC

ARTICLE INFO

Article history:
Received 22 February 2013
Received in revised form 24 July 2013
Accepted 26 July 2013
Available online 6 August 2013

Keywords:
Inverse problem
S-system
Memetic algorithm
Cockroach swarm evolution
Structure identification

ABSTRACT

S-type biological systems (S-systems) are demonstrated to be universal approximations of continuous biological systems. S-systems are easy to be generalized to large systems. The systems are identified through data-driven identification techniques (cluster-based algorithms or computational methods). However, S-systems' identification is challenging because multiple attractors exist in such highly nonlinear systems. Moreover, in some biological systems the interactive effect cannot be neglected even the interaction order is small. Therefore, learning should be focused on increasing the gap between the true and redundant interaction. In addition, a wide searching space is necessary because no prior information is provided. The used technologies should have the ability to achieve convergence enhancement and diversity preservation. Cockroaches live in nearly all habitats and survive for more than 300 million years. In this paper, we mimic cockroaches' *competitive* swarm behavior and integrated it with advanced evolutionary operations. The proposed cockroach genetic algorithm (CGA) possesses strong snatching-food ability to rush forward to a target and high migration ability to escape from local minimum. CGA was tested with three *small-scale* systems, a twenty-state *medium-scale* system and a thirty-state large-scale system. A wide search space ([0,100] for rate constants and [−100,100] for kinetic orders) with random or bad initial starts are used to show the high exploration performance.

1. 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]. Parameter estimation is the limiting step for biological modeling. Hill and Michaelis-Menten [2] rate modeling is a forward approach. These models use local kinetic information. Chou and Voit [3] estimated the parameter and the functional forms through dynamic flux. S-system [4,5] is another popular nonlinear dynamic model to show direct state-interactive information. The model is composed of highly nonlinear differential equations. Parameter estimation of both models becomes increasingly challenge when the number of state variables increases. Traditional gradient-based approaches have the possibility to get trapped at local optima. Population-based approaches have problems in finding the global optima in a limited time. Wang et al. used a two-step approach to determine the ranges and the mean values of the parameters [6]. Voit and collaborators proposed algorithms to gradually increase the model complexity [7]. They also introduced alternating regression [1] and solved the convergence issues [8]. Kutalik et al. used Newton-flow analysis [9]. Iba and collaborators used genetic programming (GP) [10,11]. Wang and collaborators integrated migration and acceleration into differential evolutionary algorithms (hybrid differential evolution (HDE)) [12–14]. Ho et al. proposed genetic algorithms with intelligent crossover (IGA) [15]. Gonzalez et al. used simulated annealing [16]. Chen et al. hybridized genetic algorithm and simulated annealing [17]. Matsubara et al. introduced radial basis function [18]. Some researchers introduced neural networks and particle swarm optimization (PSO) [19,20]. Various penalty terms were introduced to infer sparsely connected networks [21–26]. Chou et al. [27] and Sun et al. [28] reviewed various approaches that have been developed for the S-system identification. Some important issues and possible research directions were proposed in these two papers. Voit took a comprehensive review in the models and identification technologies of biochemical systems [29].

Memetic algorithms (MAs) use various methodological hybridization methods to integrate efficient local-improvement operations into a population-based algorithm. MAs have advantages in exploitation (local-search) and exploration (global-search). MAs have successfully solved various optimization problems *in other fields*. Harman et al. discussed local and global optimization through theoretical and empirical studies [30]. Soh et al. identified low-energy pure water isomers [31]. Ahn et al. proposed a GA-based MA for electromagnetic systems [32]. Meuth et al. intro-

* Corresponding author. Tel.: +886 4 8511888; fax: +886 4 8511245.
E-mail address: jen@mail.dyu.edu.tw (S.-J. Wu).

0025-5564/$ - see front matter © 2013 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.mbs.2013.07.019

duced meta-memes for high-order learning [33]. Kramer integrated iterated local search with Powell gradient method [34]. Caponio et al. introduced Hooke–Jeeves and Nelder–Mead memes to control the synchronous magnet drive [35]. Netri et al. introduced adaptive multi-meme-MA for HIV therapies [36]. Wang et al. introduced dual-mapping and random-immigrants MA [37]. Shen et al. proposed a hybrid algorithm of PSO and Tabu Search [38]. Song et al. proposed a hybrid algorithm of PSO, SA (simulated annealing) and the Simplex method [39]. Keedwell and Khu proposed a heuristic MA for water distribution network [40]. Tsoulos and Lagris hybridized these two in series through the continuous repetition of a GA and a subsequent local search [41]. Yang and Jat proposed a guided-search GA in which local-search methods were used to improve each individual of a population in each generation [42]. Some researchers combined GA with other global-search methods to achieve a balance between exploration and exploitation; for example, the hybrid GA–SA algorithm [43], the K-means-cluster-based GA [44], the hybrid GA-FNN (fuzzy neural network) [45], and the hybrid GA-PSO [46–48].

The proposed CGA is a MA in nature. In this paper, we focus on developing a competitive swarm operation with the ability to intensify and diversify the search at the same time. Technological contributions of this paper are described as follows. In Section 2, we mimicked cockroaches′ competitive behavior for food during shortages as multi-meme operations and simulated their migration. These two operations were organically integrated with advanced genetic operations. The new optimization algorithm CGA largely improves the explorative (global-search) and exploitative (local-search) of genetic algorithms. The proposed CGA is used to estimate the parameters of S-system models of five biological systems in Section 3. Robustness, exploration and convergence performances are examined in this section. Section 4 is the conclusion.

2. Cockroach genetic algorithm (CGA)

S-systems, which are rooted on biochemical system theory, express gene interaction, protein regulation and metabolic reactions as power functions. At any time instant the net influx (v_i^+) and efflux (v_i^-) of the constitute (metabolite, protein or gene) x_i are approximated as power-law functions: For a system with n dependent constitutes and m independent constitutes, the change of gene expression level x_i is

$$\frac{dx_i}{dt} = v_i^+ - v_i^- = \alpha_i \Pi_{j=1}^{n+m} x_j^{g_{ij}} - \beta_i \Pi_{j=1}^{n+m} x_j^{h_{ij}}, \quad i = 1, 2, \ldots, n, \tag{1}$$

where α_i and β_i are the rate constants, and g_{ij} and h_{ij} are the kinetic orders. Various evolutionary optimization technologies were used to identify the S-system models of gene regulatory networks or protein metabolic systems. How to avoid getting stuck in local minima is critical for inferring such a high dimensional and nonlinear system by computational approaches. It is hard for a state-of-the-art GA with simplex crossover (SPXGA) to obtain a satisfactory solution in a limited computation time [15].

We adopt real-value coding for exploiting the gradualness of continuous variables. The unknown parameters of an S-system are encoded as a cockroach individual (a chromosome in evolution operations):

α_1	…	α_n	β_1	…	β_n	g_{11}	…	$g_{n,(n+m)}$	h_{11}	…	$h_{n,(n+m)}$

Cockroaches live in a wide range of environments around the world. We randomly disperse cockroach individuals over the entire search space. The initial population is composed of $I_i, i = 1, \ldots, N_P$:

$$I_i = I_{\min} + r(I_{\max} - I_{\min}), \tag{2}$$

where r is a random number, N_P is the total number of individuals in the population, and $I_{\min}$ and $I_{\max}$ are the lower and upper bounds of individual I_i. The parameter vector I_i denotes the *position* of the ith cockroach. For simplification, we call I_i the ith cockroach individual. The strongest cockroach implies that the cockroach occupies the place with the most food resources. (In the entire space the place with most food resources denotes the global optimum.) *Parameter learning* is to minimize the residual error J_e (denoting unfitness),

$$J_e = \frac{1}{nQ} \sum_{k=1}^{Q} \sum_{i=1}^{n} t_a \left(\frac{x_i^k - x_{i,\exp}^k}{\max\left(x_{i,\exp}^k\right)} \right)^2, \tag{3}$$

where $x_i^k, x_{i,\exp}^k$ are, respectively, the kth estimated and artificially experimental data of the constitute x_i, t_a is a time-weighting factor, and Q is the number of the sampled data. Normalization is to ensure comparable competition in different-scales species. Parameters in CGA are set to be random such that the algorithm will not move back and forth between two local attractors.

Ant colony optimization (ACO) is inspired by the swarm intelligence of real ants that are capable of finding the shortest path from a food source to their nest. Each ant constructs a solution that is expressed in terms of the feasible paths on the graph, which is composed of vertices and edges. While walking, ants deposit pheromone on the edge and follow pheromone previously deposited by other ants. At each construction step, an ant chooses the edge to follow in probability associated with the amount of pheromone and the heuristic information. After all ants complete their tour, the pheromone level is updated through pheromone evaporation and according to the performance of a set of good solutions. Particle swarm optimization (PSO) mimics the intelligence of fish schooling and bird flocking for foraging efficiency and defensing over against predators. Each particle (agent, individual) has a position and keeps moving through the search space. A particle successively adjusts the velocity according to its personal experience (local-best position) as well as the experience of the particles in the neighborhood (global-best position). PSO lies in accelerating each particle toward the local-best and the global-best locations. The algorithm uses an inertia weighting factor to balance the global and local search. Both swarm intelligence are all related to their cooperative behavior. This becomes a reason that both algorithms have a lack of diversity and are easily trapped in a local optimum. Instead of considering cockroaches′ cooperative behavior [49,50], we here focus on their competitive behaviors. Cockroaches prefer dark, warm and humid environment. Their behavioral tactics for coping with extreme conditions is migration. They like to share information (food location) via secreting pheromone. However, they always compete with each other for food during shortages. We integrate the artificial-cockroach behaviors with biological evolution to promote the exploration and exploitation of algorithms. Cockroaches are able to jump away to numbers-of-meters distance. Driving-out-induced jumping and harsh-environment-induced escaping will significantly enhance the exploration ability. Therefore, the algorithm is able to prevent premature convergence.

2.1. Leader generation (golden section)

Individuals in a population are arranged according to their fitness values in descending order. To increase the diversity of search we divide the ordered individuals into four teams by the golden section method, as shown in Fig. 1. The best individual in each team is chosen as the leader. Leaders will be responsible for exploitation and others for exploration. The number of members in these four teams are n_i, $i = 1,\ldots,4$ and their leaders are I_l^i, $i = 1,\ldots,4$. For

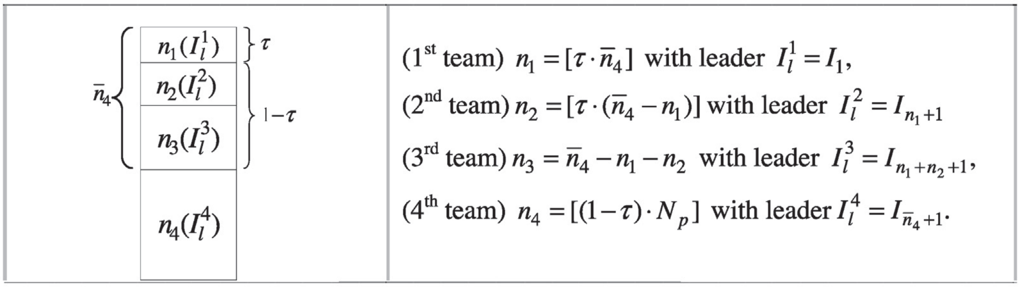
$\bar{n}_4$
$n_1(I_l^1)$
$n_2(I_l^2)$
$n_3(I_l^3)$
$n_4(I_l^4)$
τ
$1-\tau$
(1st team) $n_1 = [\tau \cdot \bar{n}_4]$ with leader $I_l^1 = I_1$,
(2nd team) $n_2 = [\tau \cdot (\bar{n}_4 - n_1)]$ with leader $I_l^2 = I_{n_1+1}$
(3rd team) $n_3 = \bar{n}_4 - n_1 - n_2$ with leader $I_l^3 = I_{n_1+n_2+1}$,
(4th team) $n_4 = [(1-\tau) \cdot N_p]$ with leader $I_l^4 = I_{\bar{n}_4+1}$.

Fig. 1. Generation of leaders and teams by golden-section method.

more huge and complex biological systems more teams are needed. We use $\bar{n}_4$ to denote the number of individuals in the first three teams; i.e., the total number of individuals in the population is $Np = \bar{n}_4 + n_4$. $[\bullet]$ is a Gauss mark and τ is the golden-section constant.

$$\begin{aligned}
&(\text{1st team})n_1 = [\tau \cdot \bar{n}_4] \quad \text{with leader} \quad I_l^1 = I_1,\\
&(\text{2nd team})n_2 = [\tau \cdot (\bar{n}_4 - n_1)] \quad \text{with leader} \quad I_l^2 = I_{n_1+1}\\
&(\text{3rd team})n_3 = \bar{n}_4 - n_1 - n_2 \quad \text{with leader} \quad I_l^3 = I_{n_1+n_2+1},\\
&(\text{4th team})n_4 = [(1-\tau) \cdot N_p] \quad \text{with leader} \quad I_l^4 = I_{\bar{n}_4+1.}
\end{aligned} \tag{4}$$

2.2. Competition for food during shortages

2.2.1. Snatching food (leader-take-all exploitation)

Competition occurs during food shortages. Only the leader in a team has the chance to snatch food: The strongest cockroach rushes to food as soon as possible. Other leaders run at a lower speed. In order to let snatching behavior more flexible we assume the strongest cockroach run in a derivative-based (straight forward) way, but other leaders crawl in a derivative-free (circuitous) way. Hungry cockroaches search for food. At the sight of food the strongest cockroach (the best individual) $I_b(=I_l^1)$ dashes forward to food at the speed of the steepest descent,

$$I_l^1 = I_b - \lambda \nabla_f, \tag{5}$$

where λ is the size of a step to determine the descent rate and ∇_f is the gradient of an objective function. Other leaders I_l^i, $i = 2, 3, 4$ rush for food at the speed of the downhill Simplex: Replace I_l^i with the better point generated from the following reflection, expansion and contraction operations. If this operation fails, shrink towards the best vertex I_b.

$$\begin{aligned}
&(\text{reflection})x_r = c + \alpha(c - I_l^i),\\
&(\text{expansion})x_e = c + \beta(x_r - c),\\
&(\text{contraction})x_c \text{ is the better one of } c + \gamma(x_r - c) \text{ and } c + \gamma(I_l^i - c).\\
&(\text{shrink})I_l^i \text{ is replaced by } (I_l^i + I_b)/2,
\end{aligned} \tag{6}$$

where α, β, γ are reflection, expansion and contraction coefficients, respectively, and c is the centroid of all vertices better than I_l^i (centroid of the better side). Parameters α, β, γ, λ are set to be random such that the algorithm is stochastic and is able to get out of local minima. When the exploitation successes the leader goes to a better position; otherwise the leader stays at the same place.

2.2.2. Driving out (weaker-migration exploration)

Hungry leaders are selfish in not only rushing to food but also driving out the weaker. At this time the cockroach I_j^i jumps away from their leader I_l^i:

$$I_j^i = \begin{cases} I_l^i + r_2 * \left(I_{\min} - I_l^i\right), & r_1 \leqslant \lambda_l^i,\\ I_l^i + r_2 * \left(I_{\max} - I_l^i\right), & \textit{otherwise}, \end{cases} \tag{7}$$

for $j = 2,\ldots,n^i$, $i = 1,\ldots,4$ where $\lambda_l^i = \frac{I_l^i - I_{\min}}{I_{\max} - I_{\min}}$, and r_1 and r_2 are random factors. After the driving-out operation the weaker individuals are updated. If the dispelled cockroaches luck into food-rich places, then the winner (best-so-far individual I_b) exchange with those better individuals one by one: If I_3^1 (the third member in the first team) and I_l^2 (the leader in the second team) are better than I_b, then I_b exchange with I_3^1 first. A new winner I_b' is generated ($I_b' = I_3^1$ and $I_3^1 = I_b$). Once I_l^2 is better than I_b' then exchange again. In this way we can improve the best-so-far individual and preserve the population diversity at the same time.

2.2.3. Winner walkabout (neighbor exploitation)

To further improve the best-so-far individual we let the *current best* cockroach I_b take walkabout for neighbor search,

$$\tilde{I}_{bm} = I_b + \eta * rand(-1, 1), \tag{8}$$

where η denotes walkabout radius ($\eta = 1$ for our systems). Once exploitation successful I_b is updated again; otherwise I_b stays at the same place.

2.3. Advanced genetic operations (biological evolution)

2.3.1. Replacement

The weaker cockroach is one of the food sources (the stronger eats the weaker). To fasten the evolution we mimic this behavior as follows. The first-three strongest cockroaches are assumed to be the invaders, and others to be the victims. The victim I_m will be replaced by the invader I_k. Let the new generated individual I_m' is a normal random variable with expected value $E[I_m'] = I_k$ and variance $E\left[(I_m' - E(I_m'))^2\right] = \sigma\delta_{km}$. Therefore, we express

$$I_m' = N(I_k, \sigma\delta_{km}), \tag{9}$$

where $\delta_{km} = |I_k - I_m|$ is a perturbed value to denote the power gap of the couple (I_k, I_m), and σ is the standard deviation. The parameter δ_{km} shows the variation of the degree of slaughter for different invader-and-victim couples. We further introduce new immigrating cockroach for population diversity. The ratio of occupation (replacement) percentage is 1:2:3:4 for new immigrating cockroaches to the victims of the third-strongest cockroach, to the victims of the second-strongest cockroach and to the victims of the strongest cockroach.

2.3.2. Mixed inbreeding and backcrossing

Fig. 2 describes the used mixed inbreeding and backcrossing operation [26]. Two parents (A and B) are randomly chosen from the population. The two-point crossover is used to generate chil-

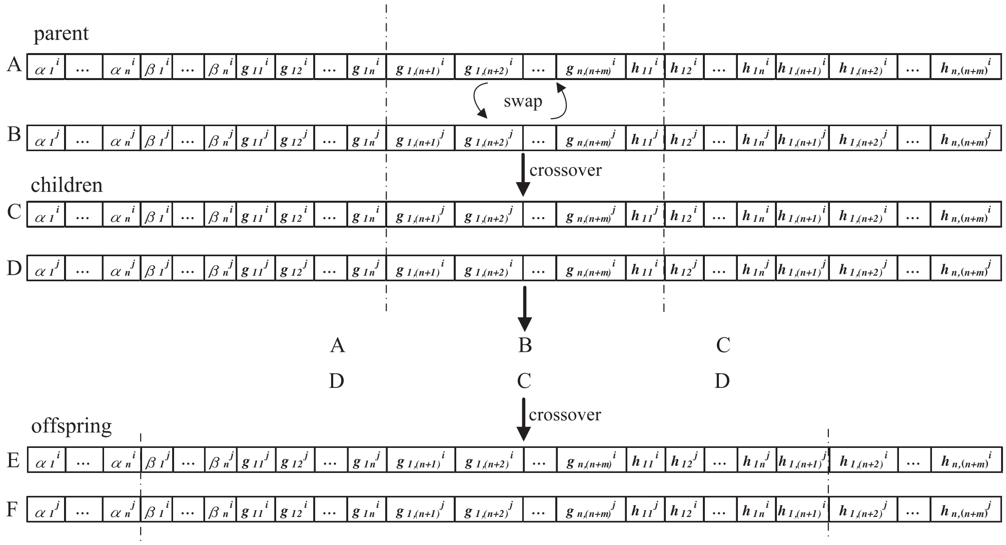
parent
A
α_1^i … α_n^i β_1^i … β_n^i g_{11}^i g_{12}^i … g_{1n}^i $g_{1,(n+1)}^i$ $g_{1,(n+2)}^i$ … $g_{n,(n+m)}^i$ h_{11}^i h_{12}^i … h_{1n}^i $h_{1,(n+1)}^i$ $h_{1,(n+2)}^i$ … $h_{n,(n+m)}^i$
swap
B
α_1^j … α_n^j β_1^j … β_n^j g_{11}^j g_{12}^j … g_{1n}^j $g_{1,(n+1)}^j$ $g_{1,(n+2)}^j$ … $g_{n,(n+m)}^j$ h_{11}^j h_{12}^j … h_{1n}^j $h_{1,(n+1)}^j$ $h_{1,(n+2)}^j$ … $h_{n,(n+m)}^j$
crossover
children
C
D
A
B
C
D
C
D
crossover
offspring
E
F

Fig. 2. Mixed inbreeding and backcrossing (the upper index i denotes parent A and j for parent B).

Table 1
(*parameter estimation*) Dataset and search range for a branch system (4-gene), a cascade system (3-gene), a small scale (5-gene), a medium scale (20-gene) and a large scale (30-gene) genetic networks.

Systems	4 Genes (Fig. 3)	3 Genes (Fig. 4)	5 Genes (Fig. 5)	20 Genes (Fig. 6)	30 Genes (Fig. 7)
Data set	8 Sets	8 Sets	8 Sets	8 Sets	8 Sets
time Period (s)	[0,8]	[0,8]	[0,0.5]	[0,1.8]	[0,10]
Sample time	0.02	0.02	0.0125	0.01	0.05
Search space	[0,100] for rate constants and [−100,100] for kinetic orders				

dren C and D. We then choose two candidates from the parents and the children for further crossover to generate offspring E and F. *Close inbreeding* denotes that the candidates are two children (C and D). *Backcrossing* denotes that the candidates are a parent and a child (A and D or B and C). For the parents A and B with the inbreeding coefficient F_A and F_B, the inbreeding coefficient f_{x_1} and the backcrossing coefficient f_{x_2} are [51]

$$\begin{aligned} f_{x_1} &= f_{CD} = \frac{1}{4}(f_{AA} + 2f_{AB} + f_{BB}) \\ &= \frac{1}{4}\left(1 + \frac{1}{2}(F_A + F_B) + 2f_{AB}\right), \\ f_{x_2} &= f_{AD} = \frac{1}{2}(f_{AA} + f_{AB}) \\ &= \frac{1}{2}\left(\frac{1}{2}(1 + F_A) + f_{AB}\right). \end{aligned} \tag{10}$$

The average relative coefficient f_{av} with backcrossing probability r_b is defined as

$$f_{av} = r_b \cdot \frac{1}{4}(f_{AD} + f_{AC} + f_{BC} + f_{BD}) + (1 - r_b) \cdot f_{CD}. \tag{11}$$

The selected candidates for further crossover are B and C in case of $r_b \leqslant f_{av,\min}$, are A and D for $f_{av,\min} < r_b \leqslant f_{av,\max}$, and are C and D for $r_b > f_{av,\max}$. The extreme case, $f_{av} = f_{av,\max}$, is for parents coming from the same group and the case $f_{av} = f_{av,\min}$ is for no blood relationship parents.

2.3.3. Competition-based screen-sifting mutation

In order to ensure global search and improve convergence we here introduce the mutation-then-competition operation [26]. Each gene is assigned a random mutation probability. Those genes with qualified mutation rate will mutate (*screen-sifting*). The mutated individuals compete with their source individuals. Those winners or losers with acceptable deviation (TA < 10) can pass down. The ratio 10 denotes that the mutated individual is far from the best individual. With this deviation threshold the mutated population will distribute *over a wide but not the entire space.*

$$TA = \frac{|F_{mut} - F_b|}{F_b}, \tag{12}$$

where F_{mut} and F_b are the residual errors of the mutated individual and the best individual, respectively.

2.4. Habitat migration (periodic)

After one-epoch iterations environment becomes severe and population emigration occurs. The emigration covers the entire searching place. Except the strongest cockroach I_b, all other cockroaches are forced to move away from him,

$$I_i = \begin{cases} I_b + r_2 * (I_{\min} - I_b), & r_1 \leqslant \lambda_b \\ I_b + r_2 * (I_{\max} - I_b), & \textit{otherwise}, \end{cases} \tag{13}$$

for $i = 2, \ldots, N_P$, where $\lambda_b = \frac{I_b - I_{\min}}{I_{\max} - I_{\min}}$. The strongest cockroach I_b has the right to determine if he will stay in his original place for leftover food or takes several chemotaxis movements (*tumble-then-run*) to a better place: (*tumble*)

$$\tilde{I}_b^i = I_b + r \cdot \left[\frac{\Delta(i)}{\sqrt{\Delta^T(i)\Delta(i)}}\right], \quad i = 1, 2, \ldots, l, \tag{14}$$

where r is a random value and $\Delta(i)$ denotes a vector in a random direction. If this operation is successful (tumble to a rich-food place), then I_b *runs* in that direction or along a local-search way; otherwise I_b stay at the original place.

Table 2

(*parameter estimation*) Estimated parameters of the S-type **large-scale network (30 genes)** in Fig. 7. Column "true" lists the parameters of the true S-system. Column "simulation" lists the estimated parameters for a ***wide*** *search space* ([0, 100] for rate constants and [−100, 100] for kinetic orders).

	α_i		β_j		g_{ij}									h_{ij}		
	True	Simulation	True	Simulation		True	Simulation		True	Simulation		True	Simulation		True	Simulation
x_1	1	1.0000126E+00	1	1.0000130E+00	$g_{1,4}$	−0.1	−9.9996528E−02							$h_{1,1}$	1	9.9998254E−01
x_2	1	9.9998940E−01	1	9.9998908E−01										$h_{2,2}$	1	1.0000019E+00
x_3	1	9.9997693E−01	1	9.9997639E−01										$h_{3,3}$	1	1.0000159E+00
x_4	1	9.9998236E−01	1	9.9998237E−01										$h_{4,4}$	1	1.0000091E+00
x_5	1	9.9998221E−01	1	9.9998207E−01	$g_{5,1}$	1	1.0000047E+00							$h_{5,5}$	1	1.0000075E+00
x_6	1	9.9989652E−01	1	9.9989455E−01	$g_{6,1}$	1	1.0000668E+00							$h_{6,6}$	1	1.0000816E+00
x_7	1	1.0002472E+00	1	1.0002458E+00	$g_{7,2}$	0.5	4.9992588E−01	$g_{7,3}$	0.4	3.9988473E−01				$h_{7,7}$	1	9.9977602E−01
x_8	1	1.0001088E+00	1	1.0001104E+00	$g_{8,4}$	0.2	1.9997917E−01	$g_{8,17}$	−0.2	−1.9998185E−01				$h_{8,8}$	1	9.9987040E−01
x_9	1	1.0006263E+00	1	1.0006269E+00	$g_{9,5}$	1	9.9948913E−01	$g_{9,6}$	−0.1	−9.9960598E−02				$h_{9,9}$	1	9.9945890E−01
x_{10}	1	1.0001168E+00	1	1.0001171E+00	$g_{10,7}$	0.3	2.9995245E−01							$h_{10,10}$	1	9.9988003E−01
x_{11}	1	1.0004151E+00	1	1.0004233E+00	$g_{11,4}$	0.4	3.9973748E−01	$g_{11,7}$	−0.2	−1.9977102E−01	$g_{11,22}$	0.4	3.9987111E−01	$h_{11,11}$	1	9.9961584E−01
x_{12}	1	1.0000422E+00	1	1.0000423E+00	$g_{12,23}$	0.1	9.9991685E−02							$h_{12,12}$	1	9.9994338E−01
x_{13}	1	9.9939412E−01	1	9.9937961E−01	$g_{13,8}$	0.6	6.0031656E−01							$h_{13,13}$	1	1.0006332E+00
x_{14}	1	9.9996771E−01	1	9.9996641E−01	$g_{14,9}$	1	1.0000195E+00							$h_{14,14}$	1	1.0000244E+00
x_{15}	1	1.0004812E+00	1	1.0004819E+00	$g_{15,10}$	0.2	1.9993068E−01							$h_{15,15}$	1	9.9962277E−01
x_{16}	1	1.0007666E+00	1	1.0007740E+00	$g_{16,11}$	0.5	4.9964286E−01	$g_{16,12}$	−0.2	−1.9987503E−01				$h_{16,16}$	1	9.9923183E−01
x_{17}	1	9.9996402E−01	1	9.9996361E−01	$g_{17,13}$	0.5	5.0001123E−01							$h_{17,17}$	1	1.0000267E+00
x_{18}	1	9.9998647E−01	1	9.9998605E−01										$h_{18,18}$	1	1.0000052E+00
x_{19}	1	1.0003037E+00	1	1.0003069E+00	$g_{19,14}$	0.1	9.9975697E−02							$h_{19,19}$	1	9.9971925E−01
x_{20}	1	1.0000949E+00	1	1.0000957E+00	$g_{20,15}$	0.7	6.9992255E−01	$g_{20,26}$	0.3	2.9996594E−01				$h_{20,20}$	1	9.9990257E−01
x_{21}	1	1.0002323E+00	1	1.0002328E+00	$g_{21,16}$	0.6	5.9985588E−01							$h_{21,21}$	1	9.9978813E−01
x_{22}	1	1.0000823E+00	1	1.0000828E+00	$g_{22,16}$	0.5	4.9995417E−01							$h_{22,22}$	1	9.9990040E−01
x_{23}	1	1.0003912E+00	1	1.0003945E+00	$g_{23,17}$	0.2	1.9992442E−01							$h_{23,23}$	1	9.9966772E−01
x_{24}	1	1.0012928E+00	1	1.0012845E+00	$g_{24,15}$	−0.2	−1.9991302E−01	$g_{24,18}$	−0.1	−9.9592950E−02	$g_{24,19}$	0.3	2.9934183E−01	$h_{24,24}$	1	9.9885165E−01
x_{25}	1	1.0000588E+00	1	1.0000581E+00	$g_{25,20}$	0.4	3.9996678E−01							$h_{25,25}$	1	9.9994277E−01
x_{26}	1	1.0002307E+00	1	1.0002322E+00	$g_{26,21}$	−0.2	−1.9993470E−01	$g_{26,28}$	0.1	9.9947068E−02				$h_{26,26}$	1	9.9980655E−01
x_{27}	1	1.0016699E+00	1	1.0016669E+00	$g_{27,24}$	0.6	5.9926628E−01	$g_{27,25}$	0.3	2.9961896E−01	$g_{27,30}$	−0.2	−1.9977462E−01	$h_{27,27}$	1	9.9872062E−01
x_{28}	1	1.0000797E+00	1	1.0000796E+00	$g_{28,25}$	0.5	4.9994686E−01							$h_{28,28}$	1	9.9992051E−01
x_{29}	1	1.0001221E+00	1	1.0001229E+00	$g_{29,26}$	0.4	3.9994688E−01							$h_{29,29}$	1	9.9989082E−01
x_{30}	1	1.0003259E+00	1	1.0003253E+00	$g_{30,27}$	0.6	5.9981936E−01							$h_{30,30}$	1	9.9974045E−01

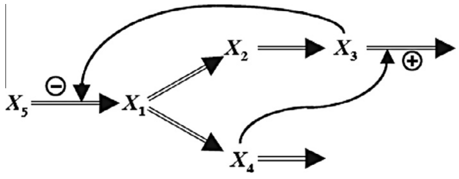

Fig. 3. A branch system with two regulatory signals [16].

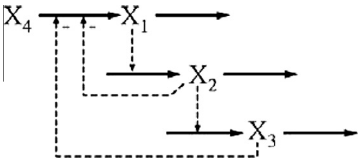

Fig. 4. A cascade system with three steps and two feedback signals [12].

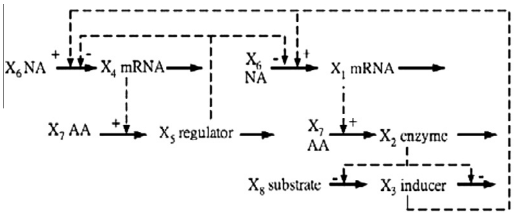

Fig. 5. A Small-scale network [52].

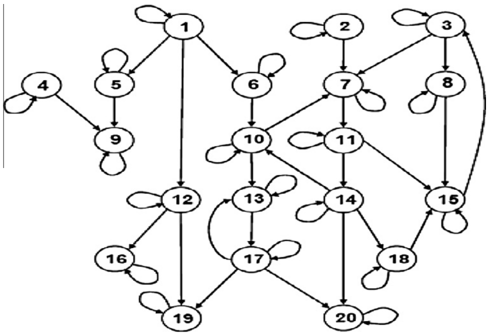

Fig. 6. A medium-scale network [53].

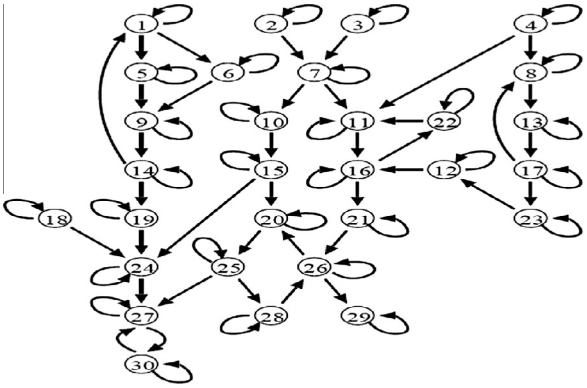

Fig. 7. A large-scale network with 30 genes [21].

Table 3
The probability of achieving global optimum for *50 independent runs*. Case 1 is a general search range with a random initial start ([0, 30] for rate constants and [−4, 4] for kinetic orders). Case 3 starts at a random point and Case 2 starts at a very bad point (80 for all parameters). Both cases are searching for a wide search range ([0, 100] for rate constants and [−100, 100] for kinetic orders).

Algorithm	Initial parameter value	Search range	Variable	Global optimal probability
CGA (static population)	Random	General (*Case 1*)	x_1	100%
			x_2	100%
			x_3	100%
		Wide (*Case 3*)	x_1	100%
			x_2	100%
			x_3	100%
	Bad	Wide (*Case 2*)	x_1	98%
			x_2	98%
			x_3	100%
CGA$^+$ (dynamic population)	Bad	Wide (*Case 2*)	x_1	100%
			x_2	100%
			x_3	100%

3. Dry-lab experiments

We now test CGA with five biological systems: a four-gene branch system [26], a three-gene cascade system [26], a five-gene small-scale network [26,33], a twenty-gene medium-scale network [34] and a thirty-gene large-scale network [21]. Table 1 lists the used dataset and search range. Cubic splines were first used for smoothing and to generate the artificial slope information for structure identification. The estimated gene-expression-levels data were generated through collocation methods. The modified collocation method is cited from Tsai and Wang [12,13]. Through collocation methods the state vector $X(t)$, $[x_1(t),\ldots,x_n(t)]^{\mathrm{T}}$, in an S-system is approximated by a set of polynomial X_m, which is further expressed as a linear combination of m bases functions, $X_m(t) = \sum_{j=0}^{m} X_c(j)\phi_j(t)$, where $X_c(j)$ is the expansion coefficients at the jth collocation points, $j = 1,\ldots,m$ and $\phi_j(t)$ is the bases functions. Let Eq. (1) be satisfied by $X_m(t)$ at the finite numbers of collocation points. By using piecewise linear Lagrange polynomial as bases functions, the dynamic behavior of the differential equations in Eq. (1) are approximated by the algebraic equations [12,13],

$$X_c(j) \cong X_c(j-1) + 0.5\eta_j\{f[X_c(j),\theta] + f[X_c(j-1),\theta]\}, \quad j = 1,\ldots,m, \qquad (15)$$

where the expansion-coefficient vector $X_c(j)$ equals to the solution $X(t)$ at the time instant $t = t_j$, $f[X_c(j),\theta]$ is the rate-function vector of the S-system in Eq. (1), θ is the parameters of the S system and η_l is a time interval.

System identification is to infer the structure (structure inference or structure identification) and to estimate the unknown parameters (parameter estimation). This study develops a new algorithm to perform grey-box system identification for estimating the unknown parameters of user-defined S-system prototypes which are constructed from the genetic networks. In addition, black-box system identification for the system with four dependent variables and one independent variable is done to further exhibit the performance of the algorithm. (No prior interactive information of the branched system is available.)

3.1. Parameter estimation (single-objective optimization)

We first do the *parameter* estimation of five biological systems. Eight-set artificial data for gene expression levels were generated from the true systems. The parameters of these true systems are

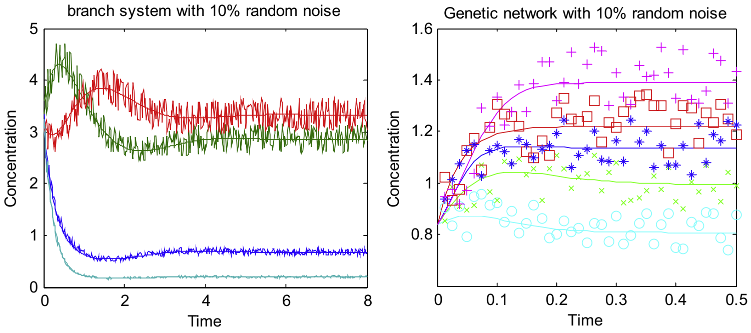

Fig. 8. Robustness examination of CGA for the branch system and the small-scale genetic network. Dot points and square-, star-, circle-, cross- and plus-signs denote data with 10% random-noise contaminate. Solid curves are the estimated profiles. A wide search space ([0, 100] for rate constants and [−100, 100] for kinetic orders) and a bad initial start (80 for all parameters) are used. Initial conditions are 20% beyond the training range.

Table 4
(*Structure identification*) comparison of this study to current research in the pruning threshold, pruning ratio, assumption and search region. Pruning ratio (ρ) is the ratio of the smallest preserved term to pruning threshold δ_s. "–" Denotes no information was provided.

Algorithm	System	Data set	Assumption	Search space	Pruning threshold (δ_s)	Pruning ratio (ρ)
CGA	4-Gene	8 Sets	no	$\alpha,\ \beta \in [0,30]$ $g,\ h \in [-4,4]$	10^{-14}	1.89×10^{13}
SPXGA (Kikuchi et al. [23])	5-Gene	10 Sets	-	$\alpha,\ \beta \in [0,15]$ $g,\ h \in [-3,3]$	–	–
GA (Voit and Almeida [55])	4-Gene	–	$g_{ii} = 0,\ h_{ii} > 0$	- $g,\ h \in [-1,1]$	0.1	1.4
HDE (Tsai and Wang [12])	3-Gene	10 Sets	$g_{ii} = 0,\ h_{ii} > 0$	$\alpha,\ \beta \in [0,20]$ $g,\ h \in [-4,4]$	0.01	2.7
	5-Gene	10 Sets	$g_{ii} = 0,\ h_{ii} > 0$	$\alpha,\ \beta \in [0,20]$ $g,\ h \in [-4,4]$	0.1	1.2
CCA (Kimura et al. [21])	5-Gene	15 Sets	–	$\alpha,\ \beta \in [0,20]$ $g,\ h \in [-3,3]$	0.001	–
DE (Noman and Ib [22])	5-Gene	10 Sets	–	$\alpha,\ \beta \in [0,15]$ $g,\ h \in [-3,3]$	0.05	–
SA (Gonzalez et al. [16])	4-Gene	–	$g_{ii} = 0,\ h_{ii} > 0$	$-g,\ h \in [-2,2]$	0.01	–
IGA (Ho et al. [15])	5-Gene	15 Sets	–	$\alpha,\ \beta \in [0,15]$ $g,\ h \in [-3,3]$	0.03	–
HDE (Liu and Wang [25])	5-Gene	8 Sets	$g_{ii} = 0,\ h_{ii} > 0$	$\alpha,\ \beta \in [0,20]$ $g,\ h \in [-4,4]$	0.01	6.9

Table 5
Pruning conditions in each step for the ten independent runs.

Variable	No. of the true redundant connections	No. of the deleted connections		
		Step 1	Step 2	Run
x_1	7	6 (88%)	1	Runs 9 and 10
		7 (100%)	–	Others
x_2	8	8 (100%)	–	All
x_3	7	7 (100%)	–	All
x_4	8	8 (100%)	–	All

listed in Row "true" in Tables 7–9 and Column "true" in Tables 2 and 10–12. Tables 10 and 12 and Row "Case 1" in Tables 7–9 list the learning results for the ***general*** *search space* ([0, 30] for rate constants and [−4, 4] for kinetic orders) with random *initial starts*. Tables 2 and 11 and Row "Case 2" in Tables 7–9 list the results for the ***wide*** *search space* ([0, 100] for rate constants and [−100, 100] for kinetic orders) with a ***bad*** *initial start* (the initial values of all parameter are set to be 80). Simulation results show that estimated parameters for these five systems are nearly the same as those of the true S-systems.

3.1.1. A branch system (4 genes)

We first consider the branch system [16], as shown in Fig. 3 with parameters listed in Table 7. This is a four-dimension system. x_1, x_2, x_3 and x_4 are the dependent constituents and x_5 is the source input (the independent constituent). Based on the network, we have the respective S-system model prototype:

$$\begin{aligned}
\dot{x}_1 &= \alpha_1 x_3^{g_{13}} x_5 - \beta_1 x_1^{h_{11}}, \\
\dot{x}_2 &= \alpha_2 x_1^{g_{21}} - \beta_2 x_2^{h_{22}}, \\
\dot{x}_3 &= \alpha_3 x_2^{g_{32}} - \beta_3 x_3^{h_{33}} x_4^{h_{34}}, \\
\dot{x}_4 &= \alpha_4 x_1^{g_{41}} - \beta_4 x_4^{h_{44}}.
\end{aligned} \tag{16}$$

The unknown parameters are encoded as a 17-gene cockroach individual:

α_1	α_2	α_3	α_4	β_1	β_2	β_3	β_4	g_{13}	g_{21}	g_{32}	g_{41}	h_{11}	h_{22}	h_{33}	h_{34}	h_{44}

We generated eight sets of artificial experimental data from the true S-system with parameters listed in Row "true" of Table 7. The simulation time of an experiment is 8 s and the sample time is 0.02. Row "Case 1" in Table 7 lists the estimated parameters for a general search space ([0, 30] for rate constants and [−4, 4] for kinetic orders) with a *random* initial start. Row "Case 2" in Table 7 lists the estimated parameters for a wide search space ([0, 100] for rate constants and [−100, 100] for kinetic orders) with a *bad* initial start (the initial values of all parameter are set to be 80). Simulation results in both cases show that the inferred parameters are nearly the same as those in the true S-system.

3.1.2. A cascade system (3 genes)

The second system is the cascade network in Fig. 4 [12]. This is a three-step system with two feedback signals. x_1,x_2 andx_3 are the

Table 6
(***structure identification***) Structure identification for **a branch system** in Fig. 3. Step 0 lists the parameters of the true S-system. Steps 1 to 2 show the estimated parameters and inferred structures.

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	1.9869487E+01	2.0350824E−15	−1.0121408E−16	−7.5888690E−01	−2.0782613E−17	9.2950239E−01
	x_2	8.3421078E+00	4.4928402E−01	−1.2346232E−15	−5.2359489E−16	−5.5289283E−16	−5.5075463E−17
	x_3	3.1105792E+00	−1.8536181E−16	7.2585091E−01	−1.6276487E−16	−3.2226031E−16	3.5381885E−16
	x_4	2.2156428E+00	4.5628239E−01	8.1023598E−16	−7.8526363E−16	5.7838675E−16	1.1071731E−16
2	x_1	1.9989784E+01			−8.0017394E−01		1.0002165E+00
	x_2	8.0134341E+00	4.9921193E−01				
	x_3	3.0050430E+00		7.4929863E−01			
	x_4	2.0001301E+00	4.9995519E−01				
Step	Variable	β_j	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	1.0311030E+01	4.6583387E−01	−3.7362789E−15	−1.4502508E−15	5.3440492E−16	−4.9975398E−17
	x_2	3.5238444E+00	−1.5478921E−16	6.4537002E−01	−1.6607947E−16	1.1598826E−15	4.2922820E−17
	x_3	5.0633937E+00	−3.7595700E−16	−2.7055843E−16	4.8018180E−01	1.8858591E−01	2.0750533E−17
	x_4	6.1116142E+00	3.9237470E−18	−9.4828264E−16	−9.6898885E−16	7.3635861E−01	−1.9775796E−16
2	x_1	9.9937089E+00	5.0014031E−01				
	x_2	3.0098595E+00		7.4870596E−01			
	x_3	5.0059692E+00			4.9949949E−01	1.9979963E−01	
	x_4	5.9999738E+00				7.9994288E−01	

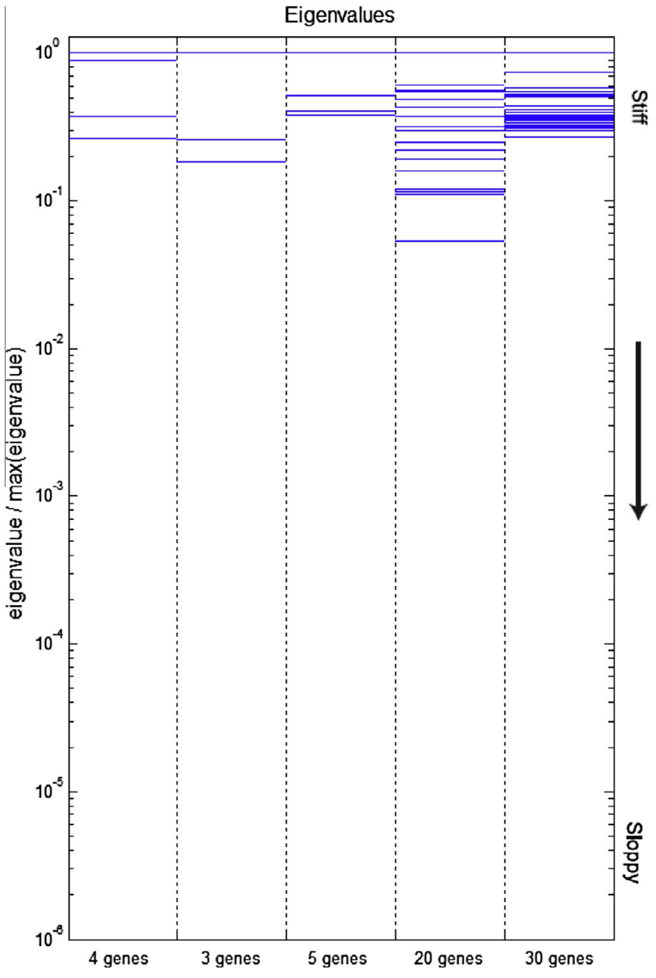

Fig. 9. Eigenvalues of the sensitivity-related Hessian matrix of the five S-systems.

three dependent constituents and x_4 is the independent constituent. According to the network, we have the S-system prototype:

$$\begin{aligned}\dot{x}_1 &= \alpha_1 x_2^{g_{12}} x_3^{g_{13}} x_4 - \beta_1 x_1^{h_{11}},\\ \dot{x}_2 &= \alpha_2 x_1^{g_{21}} - \beta_2 x_2^{h_{22}},\\ \dot{x}_3 &= \alpha_3 x_2^{g_{32}} - \beta_3 x_3^{h_{33}}\end{aligned} \tag{17}$$

and the respective cockroach individual (13-gene):

α_1	α_2	α_3	β_1	β_2	β_3	g_{12}	g_{13}	g_{21}	g_{32}	h_{11}	h_{22}	h_{33}

The true parameters are shown in Row “true” of Table 8. Simulation was from time $t = 0$ to $t = 8$ s with sample time 0.02. Rows “Case 1” and “Case 2” in Table 8 show, respectively, the estimated parameters for a *general* search space with a *random* initial start and a *wide* search space with a *bad* initial start. Both simulation results are nearly the same as the true values.

3.1.3. A small-scale network (5 genes)

Our third case is a small-scale network with two regulatory signals in Fig. 5 [52]. We write the respective S-system prototype for the network as

$$\begin{aligned}\dot{x}_1 &= \alpha_1 x_3^{g_{13}} x_5^{g_{15}} x_6 - \beta_1 x_1^{h_{11}},\\ \dot{x}_2 &= \alpha_2 x_1^{g_{21}} x_7 - \beta_2 x_2^{h_{22}},\\ \dot{x}_3 &= \alpha_3 x_2^{g_{32}} x_8 - \beta_3 x_2^{h_{32}} x_3^{h_{33}},\\ \dot{x}_4 &= \alpha_4 x_3^{g_{43}} x_5^{g_{45}} x_6 - \beta_4 x_4^{h_{44}},\\ \dot{x}_5 &= \alpha_2 x_4^{g_{54}} x_7 - \beta_2 x_5^{h_{55}},\end{aligned} \tag{18}$$

where x_i, $i = 1,\ldots,5$ are the dependent constitutes, and x_i, $i = 6,7,8$ are the constant sources (independent constituents). The unknown parameters were encoded as a 23-gene cockroach individual:

α_1	α_2	α_3	α_4	α_5	β_1	β_2	β_3	β_4	β_5	g_{13}	g_{15}	g_{21}	g_{32}	g_{43}	g_{45}	g_{54}	h_{11}	h_{22}	h_{32}	h_{33}	h_{44}	h_{55}

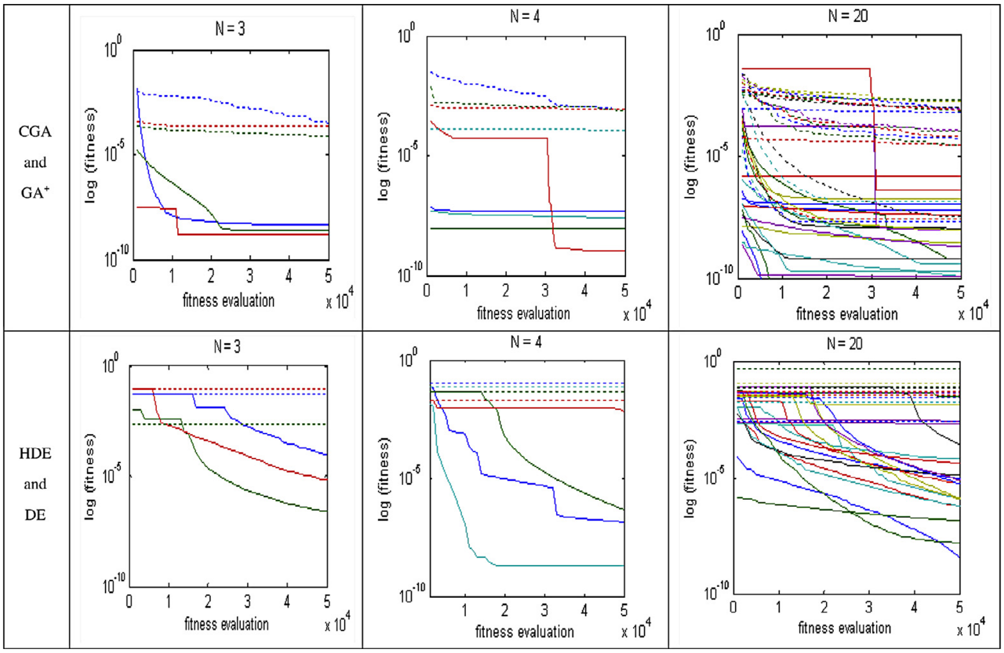

Fig. 10. Convergence comparison of CGA to HDE [12–14], improve GA (GA$^+$) [54] and DE. The wide search space ([0,100] for rate constants and [−100,100] for kinetic orders) starting at a bad point (80 for all parameters) is used. **Solid curves** denote CGA and HDE and **dash curves** denote GA$^+$ and DE. Curves are drawn for fitness evaluation from 1000 to 50,000. N denotes system dimension.

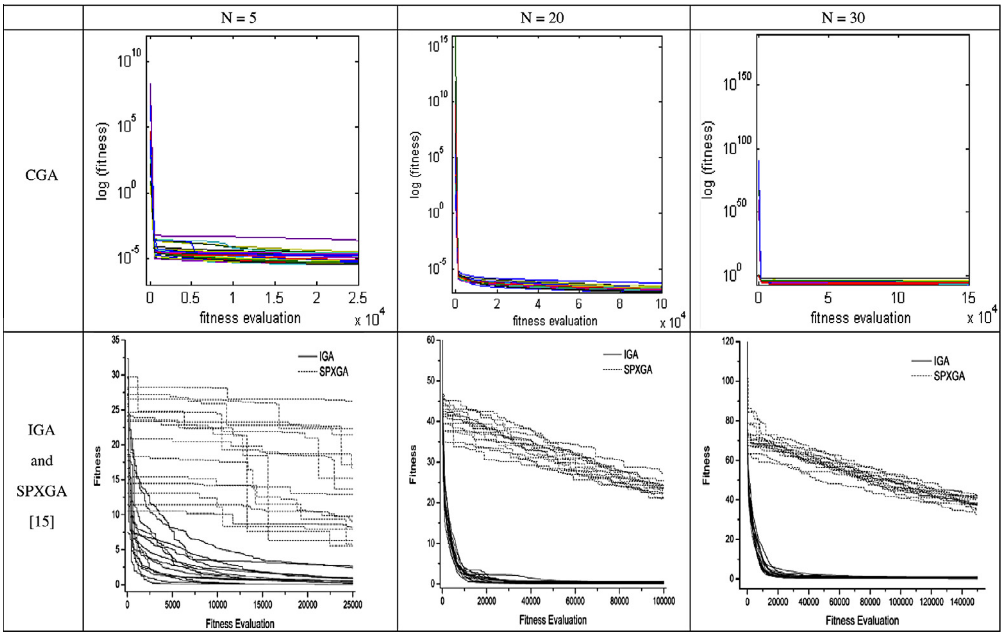

Fig. 11. Convergence comparison of CGA to IGA and SPXGA for a small-scale ($N = 5$), a medium-scale ($N = 20$) and large-scale ($N = 30$) networks. Six-set time series data with 11 sample points are used as in [15]. The general search space ([0,15] for rate constants and [−3,3] for kinetic orders) starting at nearby midpoint (zero for kinetic order and 7.5 for rate constant) is used.

The *true* rate constants and kinetic orders are listed in Row “true” of Table 9. The same cubic spline technology is adopted for smoothing the eight-set data. Each experiment was simulated from time $t = 0$ to $t = 0.5$ s with a sample time 0.0125. Rows “Case 1” and “Case 2” in Table 9 show, respectively, the estimated parameters for a *general space* with a *random* initial start and a *wide* search space with a *bad* initial start.

3.1.4. A medium-scale network (20 genes)

Our fourth system is the medium-scale network in Fig. 6 [53]. Twenty dependent constituents (x_i, $i = 1,\ldots,20$) are considered.

The respective S-system formulation with respect to twenty genes, x_i, $i = 1,\ldots,20$, is as follows.

$$
\begin{aligned}
\dot{x}_1 &= \alpha_1 - \beta_1 x_1^{h_{1,1}}, & \dot{x}_8 &= \alpha_8 x_3^{g_{8,3}} - \beta_8 x_8^{h_{8,8}}, & \dot{x}_{15} &= \alpha_{15} x_8^{g_{15,8}} x_{11}^{g_{15,11}} x_{18}^{g_{15,18}} - \beta_{15} x_{15}^{h_{15,15}},\\
\dot{x}_2 &= \alpha_2 - \beta_2 x_2^{h_{2,2}}, & \dot{x}_9 &= \alpha_9 x_4^{g_{9,4}} x_5^{g_{9,5}} - \beta_9 x_9^{h_{9,9}}, & \dot{x}_{16} &= \alpha_{16} x_{12}^{g_{16,12}} - \beta_{16} x_{16}^{h_{16,16}},\\
\dot{x}_3 &= \alpha_3 x_{15}^{g_{3,15}} - \beta_3 x_3^{h_{3,3}}, & \dot{x}_{10} &= \alpha_{10} x_6^{g_{10,6}} x_{14}^{g_{10,14}} - \beta_{10} x_{10}^{h_{10,10}}, & \dot{x}_{17} &= \alpha_{17} x_{13}^{g_{17,13}} - \beta_{17} x_{17}^{h_{17,17}},\\
\dot{x}_4 &= \alpha_4 - \beta_4 x_4^{h_{4,4}}, & \dot{x}_{11} &= \alpha_{11} x_7^{g_{11,7}} - \beta_{11} x_{11}^{h_{11,11}}, & \dot{x}_{18} &= \alpha_{18} x_{14}^{g_{18,14}} - \beta_{18} x_{18}^{h_{18,18}},\\
\dot{x}_5 &= \alpha_5 x_1^{g_{5,1}} - \beta_5 x_5^{h_{5,5}}, & \dot{x}_{12} &= \alpha_{12} x_1^{g_{12,1}} - \beta_{12} x_{12}^{h_{12,12}}, & \dot{x}_{19} &= \alpha_{19} x_{12}^{g_{19,12}} x_{17}^{g_{19,17}} - \beta_{19} x_{19}^{h_{19,19}},\\
\dot{x}_6 &= \alpha_6 x_1^{g_{6,1}} - \beta_6 x_6^{h_{6,6}}, & \dot{x}_{13} &= \alpha_{13} x_{10}^{g_{13,10}} x_{17}^{g_{13,17}} - \beta_{13} x_{13}^{h_{13,13}}, & \dot{x}_{20} &= \alpha_{20} x_{14}^{g_{20,14}} x_{17}^{g_{20,17}} - \beta_{20} x_{20}^{h_{20,20}}.\\
\dot{x}_7 &= \alpha_7 x_2^{g_{7,2}} x_3^{g_{7,3}} x_{10}^{g_{7,10}} - \beta_7 x_7^{h_{7,7}}, & \dot{x}_{14} &= \alpha_{14} x_{11}^{g_{14,11}} - \beta_{14} x_{14}^{h_{14,14}}, & &
\end{aligned}
\tag{19}
$$

The unknown parameters include forty rate constants and forty six kinetic orders. These unknown parameters are encoded as an 86-gene cockroach individual. The true rate constants and kinetic orders are listed in Column "true" of Tables 10 and 11. Eight sets of artificial experiment data were generated from the *true* S-type system. Each experiment was simulated from time $t = 0$ to $t = 1.8$ with a sample time 0.01. The estimated parameters are listed in Column "Simulation" of Tables 10 and 11 for a *general* search space with a *random* initial start and for a *wide* search space with a *bad* initial start, respectively. Simulation results in both tables demonstrate that even system dimension is high and searching covers such a wide range the proposed CGA still has good performance.

3.1.5. A large-scale network (30 genes)

We now discuss a large system with thirty dimensions, which is shown in Fig. 7 [21]. This system has thirty dependent constituents (x_i, $i = 1,\ldots,30$). The respective S-system is described as

$$
\begin{aligned}
\dot{x}_1 &= \alpha_1 x_{14}^{g_{1,14}} - \beta_1 x_1^{h_{1,1}} & \dot{x}_{16} &= \alpha_{16} x_{11}^{g_{16,11}} x_{12}^{g_{16,12}} - \beta_{16} x_{16}^{h_{16,16}}\\
\dot{x}_2 &= \alpha_2 - \beta_2 x_2^{h_{2,2}} & \dot{x}_{17} &= \alpha_{17} x_{13}^{g_{11,13}} - \beta_{17} x_{17}^{h_{17,17}}\\
\dot{x}_3 &= \alpha_3 - \beta_3 x_3^{h_{3,3}} & \dot{x}_{18} &= \alpha_{18} - \beta_{18} x_{18}^{h_{18,18}}\\
\dot{x}_4 &= \alpha_4 - \beta_4 x_4^{h_{4,4}} & \dot{x}_{19} &= \alpha_{19} x_{14}^{g_{19,14}} - \beta_{19} x_{19}^{h_{19,19}}\\
\dot{x}_5 &= \alpha_5 x_1^{g_{5,1}} - \beta_5 x_5^{h_{5,5}} & \dot{x}_{20} &= \alpha_{20} x_{15}^{g_{20,15}} x_{26}^{g_{20,26}} - \beta_{20} x_{20}^{h_{20,20}}\\
\dot{x}_6 &= \alpha_6 x_1^{g_{6,1}} - \beta_6 x_6^{h_{6,6}} & \dot{x}_{21} &= \alpha_{21} x_{16}^{g_{21,16}} - \beta_{21} x_{21}^{h_{21,21}}\\
\dot{x}_7 &= \alpha_7 x_2^{g_{7,2}} x_3^{g_{7,3}} - \beta_7 x_7^{h_{7,7}} & \dot{x}_{22} &= \alpha_{22} x_{16}^{g_{22,16}} - \beta_{22} x_{22}^{h_{22,22}}\\
\dot{x}_8 &= \alpha_8 x_4^{g_{8,4}} x_{17}^{g_{8,17}} - \beta_8 x_8^{h_{8,8}} & \dot{x}_{23} &= \alpha_{23} x_{17}^{g_{23,17}} - \beta_{23} x_{23}^{h_{23,23}}\\
\dot{x}_9 &= \alpha_9 x_5^{g_{9,5}} x_6^{g_{9,6}} - \beta_9 x_9^{h_{9,9}} & \dot{x}_{24} &= \alpha_{24} x_{15}^{g_{24,15}} x_{18}^{g_{24,18}} x_{19}^{g_{24,19}} - \beta_{24} x_{24}^{h_{24,24}}\\
\dot{x}_{10} &= \alpha_{10} x_7^{g_{10,7}} - \beta_{10} x_{10}^{h_{10,10}} & \dot{x}_{25} &= \alpha_{25} x_{20}^{g_{25,20}} - \beta_{25} x_{25}^{h_{25,25}}\\
\dot{x}_{11} &= \alpha_{11} x_4^{g_{11,4}} x_7^{g_{11,7}} x_{22}^{g_{11,22}} - \beta_{11} x_{11}^{h_{11,11}} & \dot{x}_{26} &= \alpha_{26} x_{21}^{g_{26,21}} x_{28}^{g_{26,28}} - \beta_{26} x_{26}^{h_{26,26}}\\
\dot{x}_{12} &= \alpha_{12} x_{23}^{g_{12,23}} - \beta_{12} x_{12}^{h_{12,12}} & \dot{x}_{27} &= \alpha_{27} x_{24}^{g_{27,24}} x_{25}^{g_{27,25}} x_{30}^{g_{27,30}} - \beta_{27} x_{27}^{h_{27,27}}\\
\dot{x}_{13} &= \alpha_{13} x_8^{g_{13,8}} - \beta_{13} x_{13}^{h_{13,13}} & \dot{x}_{28} &= \alpha_{28} x_{25}^{g_{28,25}} - \beta_{28} x_{28}^{h_{28,28}}\\
\dot{x}_{14} &= \alpha_{14} x_9^{g_{14,9}} - \beta_{14} x_{14}^{h_{14,14}} & \dot{x}_{29} &= \alpha_{29} x_{26}^{g_{29,26}} - \beta_{29} x_{29}^{h_{29,29}}\\
\dot{x}_{15} &= \alpha_{15} x_{10}^{g_{15,10}} - \beta_{15} x_{15}^{h_{15,15}} & \dot{x}_{30} &= \alpha_{30} x_{27}^{g_{30,27}} - \beta_{30} x_{30}^{h_{30,30}}
\end{aligned}
\tag{20}
$$

The unknown parameters include sixty rate constants and sixty eight kinetic orders. These unknown parameters are encoded as a 128-gene cockroach individual. The rate constants and kinetic orders are listed in Column "true" of Tables 2 and 12. Eight sets of training data were generated from the *true* S-system. Each dry-lab experiment was simulated from time $t = 0$ to $t = 10$ with a sample time 0.05. The estimated parameters are shown in Column "Simulation" of Table 12 for the *general* search space with *random* initial start, and Table 2 for the *wide* search space with a *bad* initial start. Simulation results in both tables demonstrate that CGA achieves perfect results even the dimension of the system is as high as thirty and searching is in a wide range.

3.2. Further discussion

3.2.1. Exploration and convergence

We so far have demonstrated that CGA possesses the ability to achieve perfect searching even in a wide search space with a bad initial start; most variables achieve satisfied values after a few evaluations calls. *Exploration ability* is shown in Table 2 (30 genes), in Row "Case 2" of Tables 7–9 (3-, 4- and 5-gene), and in Column "Simulation" of Table 11 (20 genes) for the case where rate range is set to be [0,100], order range is [−100,100] and all parameters are initialized as 80. We now further consider an extra case: a wide search space with random initial start. In other words, three different cases are considered: Case 1 is a general search space with random initial start. Case 2 is a wide search space with a bad initial start. Case 3 is a wide search space with random initial start. 50 independent runs were done to compare the effectiveness of CGA. The probability of achieving global minimum is shown in Table 3. Simulation results show that CGA has 100% probability to find the global minimum when the learning starts from random points no matter the searching space is general or wide (*Cases 1 and 3*). The performance falls down to 98% when the search space is wide and learning starts at *very bad* values, as shown in *Case 2* of Table 3. To solve this problem we further introduced *dynamic-population* strategy. The results for the modified CGA (CGA$^+$) are shown in the last row of Table 3. We observe that 100% searching ability is guaranteed for CGA starting at a very bad point in a wide searching space when the dynamic population is used.

Snatching food, winner walkout and stronger-eat-weaker replacement are all for exploitation-ability enhancement. Figs. 10 and 11 show the *convergence comparison* of CGA to DE, HDE [12–14], SPXGA [23], intelligent two-stage evolution algorithm (IGA, intelligent GA) [15], and GA with migration and acceleration operations (GA$^+$, improved GA) [54]. Fig. 10 shows the results of Low- and medium-dimensional systems ($N = 3$, 4, 20 genes) in the wide search space ([0,100] for rate constants and [−100,100] for kinetic orders) and a bad initial start (80 for all parameters). Low-, medium- and large-dimensional systems ($N = 5$, 20, 30 genes) with the same search space and initial start as [15] are used in Fig. 11, where the nearby-midpoint initial start (zero for kinetic order and 7.5 for rate constant) is used and search range is [0,15] for rate constants and [−3,3] for kinetic orders. Simulation results show CGA converges faster than those algorithms.

3.2.2. Robustness

To examine the *robustness* of CGA we consider systems that are contaminated with 10% random noise (10% random noise is added to the true datum for training). Fig. 8 is the results for the initial

Table 7
True and estimated parameters of the S-type **branch system (4 genes)** in Fig. 3. Row "true" lists the parameters of the true S-system. Row "Case 1" shows the estimated parameters for a ***general*** *search space* ([0,30] for rate constants and [−4,4] for kinetic orders) with a ***random*** *initial start*. Row "Case 2" shows the estimated parameters for a ***wide*** *search range* ([0,100] for rate constants and [−100,100] for kinetic orders) with a ***bad*** *initial start* (80 for all parameters).

	Variable	α_i	β_j	g_{i1}	g_{i2}	g_{i3}	h_{i1}	h_{i2}	h_{i3}	h_{i4}
True	x_1	20	10			−0.8	0.5			
	x_2	8	3	0.5				0.75		
	x_3	3	5		0.75				0.5	0.2
	x_4	2	6	0.5						0.8
Case1	x_1	1.9996496E+01	9.9986163E+00			−7.9994627E−01	4.9999159E−01			
	x_2	8.0002030E+00	3.0001482E+00	4.9998922E−01				7.4997682E−01		
	x_3	3.0048335E+00	5.0056201E+00		7.4932029E−01				4.9952091E−01	1.9980073E−01
	x_4	2.0000161E+00	5.9998200E+00	4.9996746E−01						7.9996349E−01
Case2	x_1	1.9998961E+01	1.0000011E+01			−7.9989604E−01	4.9996658E−01			
	x_2	8.0039345E+00	3.0028797E+00	4.9977047E−01				7.4961995E−01		
	x_3	3.0017547E+00	5.0025795E+00		7.4979131E−01				4.9981801E−01	1.9996480E−01
	x_4	2.0032071E+00	6.0041227E+00	4.9962411E−01						7.9938683E−01

Table 8
True and estimated parameters of the S-type **cascade system (3 genes**) in Fig. 4. Row "true" is the parameters of the true S-system. Row "Case 1" shows the estimated parameters for a ***general*** *search space* ([0,30] for rate constants and [−4,4] for kinetic orders) with a ***random*** *initial start*. Row "Case 2" shows the estimated parameters for a ***wide*** *search range* ([0,100] for rate constants and [−100,100] for kinetic orders) with a ***bad*** *initial start* (80 for all parameters).

	Variable	α_i	β_j	g_{i1}	g_{i2}	g_{i3}	h_{i1}	h_{i2}	h_{i3}
True	x_1	10	5		−0.1	−0.05	0.5		
	x_2	2	1.44	0.5				0.5	
	x_3	3	7.2		0.5				0.5
Case1	x_1	9.9983000E+00	4.9991321E+00		−1.0001159E−01	−5.0009436E−02	5.0000103E−01		
	x_2	2.0036028E+00	1.4436798E+00	4.9929160E−01				4.9901351E−01	
	x_3	2.9993602E+00	7.1998806E+00		5.0011201E−01				5.0011092E−01
Case2	x_1	9.9980369E+00	4.9984476E+00		−1.0007160E−01	−4.9959651E−02	5.0001419E−01		
	x_2	2.0064165E+00	1.4466246E+00	4.9875874E−01				4.9821637E−01	
	x_3	3.0019340E+00	7.2004725E+00		4.9966966E−01				4.9967402E−01

Table 9
True and estimated parameters of the S-type**small-scale network (5 genes)** in Fig. 5. Row "true" lists the parameters of the true S-system. Row "Case 1" shows the estimated parameters for a ***general*** *search space* ([0,30] for rate constants and [−4,4] for kinetic orders) with a ***random*** *initial start*. Row "Case 2" shows the estimated parameters for a ***wide*** *search range* ([0,100] for rate constants and [−100,100] for kinetic orders) with a ***bad*** *initial start* (80 for all parameters).

	Variable	α_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	g_{i5}
True	x_1	5			1		−1
	x_2	10	2				
	x_3	10		−1			
	x_4	8			2		−1
	x_5	10				2	
Case1	x_1	5.0025244E+00			1.0004463E+00		−1.0005928E+00
	x_2	1.0000364E+01	2.0002799E+00				
	x_3	1.0008258E+01		−1.0003655E+00			
	x_4	8.0002327E+00			2.0010040E+00		−1.0005881E+00
	x_5	1.0000137E+01				2.0004907E+00	
Case2	x_1	5.0016401E+00			1.0004428E+00		−1.0007198E+00
	x_2	9.9991671E+00	2.0001203E+00				
	x_3	1.0018684E+01		−9.9862400E−01			
	x_4	7.9989991E+00			2.0012278E+00		−1.0006360E+00
	x_5	9.9986509E+00				2.0003787E+00	
	Variable	β_j	h_{i1}	h_{i2}	h_{i3}	h_{i4}	h_{i5}
True	x_1	10	2				
	x_2	10		2			
	x_3	10		−1	2		
	x_4	10				2	
	x_5	10					2
Case1	x_1	1.0002901E+01	1.9990869E+00				
	x_2	9.9998865E+00		2.0001225E+00			
	x_3	1.0007753E+01		−1.0005392E+00	1.9998620E+00		
	x_4	1.0000772E+01				1.9998910E+00	
	x_5	9.9999571E+00					2.0002320E+00
Case2	x_1	1.0001007E+01	1.9990220E+00				
	x_2	9.9986681E+00		1.9999833E+00			
	x_3	1.0018041E+01		−9.9883469E−01	1.9998284E+00		
	x_4	9.9991415E+00				1.9998054E+00	
	x_5	9.9984596E+00					2.0001887E+00

Table 10
True and estimated parameters of the S-type **medium-scale network (20 genes)** in Fig. 6. Column "true" lists the parameters of the true S-system. Column "simulation" lists the estimated parameters for a ***genera**l search range* ([0,30] for rate constants and [−4,4] for kinetic orders). Parameters are initialized to be ***random***.

	α_i		β_j		g_{ij}									h_{ij}		
	True	Simulation	True	Simulation		True	Simulation		True	Simulation		True	Simulation		True	Simulation
x_1	10	9.9999702E+00	10	9.9999711E+00										$h_{1,1}$	1	9.9999482E−01
x_2	10	9.9994099E+00	10	9.9994048E+00										$h_{2,2}$	1	1.0000487E+00
x_3	10	9.9887145E+00	10	9.9871483E+00	$g_{3,15}$	−0.7	−6.9933524E−01							$h_{3,3}$	1	1.0019234E+00
x_4	10	9.9997768E+00	10	9.9997738E+00										$h_{4,4}$	1	1.0000132E+00
x_5	10	9.9987314E+00	10	9.9987248E+00	$g_{5,1}$	1	1.0000857E+00							$h_{5,5}$	1	1.0001086E+00
x_6	10	9.9988066E+00	10	9.9987938E+00	$g_{6,1}$	2	2.0000382E+00							$h_{6,6}$	1	1.0000612E+00
x_7	10	1.0002369E+01	10	1.0002337E+01	$g_{7,2}$	1.2	1.1997564E+00	$g_{7,3}$	−0.8	−7.9969190E−01	$g_{7,10}$	1.6	1.5996903E+00	$h_{7,7}$	1	9.9980710E−01
x_8	10	1.0006045E+01	10	1.0006077E+01	$g_{8,3}$	−0.6	−5.9938707E−01							$h_{8,8}$	1	9.9913529E−01
x_9	10	1.0004408E+01	10	1.0004395E+01	$g_{9,4}$	0.5	4.9976112E−01	$g_{9,5}$	0.7	6.9977223E−01				$h_{9,9}$	1	9.9961772E−01
x_{10}	10	9.9927401E+00	10	9.9930444E+00	$g_{10,6}$	−0.3	−3.0014597E−01	$g_{10,14}$	0.9	9.0059441E−01				$h_{10,10}$	1	1.0008388E+00
x_{11}	10	1.0001934E+01	10	1.0001944E+01	$g_{11,7}$	0.5	4.9991849E−01							$h_{11,11}$	1	9.9982349E−01
x_{12}	10	1.0001310E+01	10	1.0001306E+01	$g_{12,1}$	1	9.9985573E−01							$h_{12,12}$	1	9.9988233E−01
x_{13}	10	1.0009689E+01	10	1.0009665E+01	$g_{13,10}$	−0.4	−3.9892298E−01	$g_{13,17}$	1.3	1.2978355E+00				$h_{13,13}$	1	9.9889870E−01
x_{14}	10	9.9971482E+00	10	9.9971440E+00	$g_{14,11}$	−0.4	−4.0006827E−01							$h_{14,14}$	1	1.0002839E+00
x_{15}	10	1.0012384E+01	10	1.0012618E+01	$g_{15,8}$	0.5	4.9870465E−01	$g_{15,11}$	−1	−9.9817511E−01	$g_{15,18}$	−0.9	−8.9834477E−01	$h_{15,15}$	1	9.9860871E−01
x_{16}	10	9.9994897E+00	10	9.9995480E+00	$g_{16,12}$	2	1.9999732E+00							$h_{16,16}$	1	9.9997887E−01
x_{17}	10	9.9948458E+00	10	9.9948611E+00	$g_{17,13}$	−0.5	−5.0018209E−01							$h_{17,17}$	1	1.0006012E+00
x_{18}	10	9.9992990E+00	10	9.9992924E+00	$g_{18,14}$	1.2	1.2000460E+00							$h_{18,18}$	1	1.0000609E+00
x_{19}	10	9.9997656E+00	10	9.9997242E+00	$g_{19,12}$	1.4	1.4000532E+00	$g_{19,17}$	0.6	5.9988064E−01				$h_{19,19}$	1	1.0000268E+00
x_{20}	10	9.9939167E+00	10	9.9938882E+00	$g_{20,14}$	1	1.0004408E+00	$g_{20,17}$	1.5	1.5008207E+00				$h_{20,20}$	1	1.0006102E+00

Table 11
True and estimated parameters of the S-type **medium-scale network (20 genes)** in Fig. 6. Column "true" lists the parameters of the true S-system. Column "simulation" lists the estimated parameters for a ***wide** search space* ([0,100] for rate constants and [−100,100] for kinetic orders). Parameters are initialized to be ***80***.

	α_i		β_j		g_{ij}									h_{ij}		
	True	Simulation	True	Simulation		True	Simulation		True	Simulation		True	Simulation		True	Simulation
x_1	10	9.9999077E+00	10	9.9999075E+00										$h_{1,1}$	1	1.0000001E+00
x_2	10	9.9999329E+00	10	9.9999332E+00										$h_{2,2}$	1	9.9999757E−01
x_3	10	9.9887189E+00	10	9.9871538E+00	$g_{3,15}$	−0.7	−6.9933480E−01							$h_{3,3}$	1	1.0019228E+00
x_4	10	9.9981307E+00	10	9.9981060E+00										$h_{4,4}$	1	1.0001721E+00
x_5	10	9.9998440E+00	10	9.9998422E+00	$g_{5,1}$	1	9.9998184E−01							$h_{5,5}$	1	1.0000035E+00
x_6	10	9.9959505E+00	10	9.9958949E+00	$g_{6,1}$	2	2.0003236E+00							$h_{6,6}$	1	1.0002390E+00
x_7	10	1.0001018E+01	10	1.0001006E+01	$i_{7,2}$	1.2	1.1998864E+00	$g_{7,3}$	−0.8	−7.9978166E−01	$g_{7,10}$	1.6	1.5998481E+00	$h_{7,7}$	1	9.9991099E−01
x_8	10	1.0001452E+01	10	1.0001483E+01	$g_{8,3}$	−0.6	−5.9960246E−01							$h_{8,8}$	1	9.9961698E−01
x_9	10	1.0001662E+01	10	1.0001652E+01	$g_{9,4}$	0.5	4.9988616E−01	$g_{9,5}$	0.7	6.9991284E−01				$h_{9,9}$	1	9.9984072E−01
x_{10}	10	1.0005413E+01	10	1.0005729E+01	$g_{10,6}$	−0.3	−2.9982576E−01	$g_{10,14}$	0.9	8.9974094E−01				$h_{10,10}$	1	9.9981510E−01
x_{11}	10	9.9993773E+00	10	9.9993848E+00	$g_{11,7}$	0.5	5.0002529E−01							$h_{11,11}$	1	1.0000310E+00
x_{12}	10	9.9902724E+00	10	9.9902942E+00	$g_{12,1}$	1	1.0008127E+00							$h_{12,12}$	1	1.0007973E+00
x_{13}	10	1.0007261E+01	10	1.0007218E+01	$g_{13,10}$	−0.4	−3.9900039E−01	$g_{13,17}$	1.3	1.2981217E+00				$h_{13,13}$	1	9.9915384E−01
x_{14}	10	9.9987190E+00	10	9.9987245E+00	$g_{14,11}$	−0.4	−4.0001924E−01							$h_{14,14}$	1	1.0001411E+00
x_{15}	10	1.0012200E+01	10	1.0012434E+01	$g_{15,8}$	0.5	4.9871156E−01	$g_{15,11}$	−1	−9.9818480E−01	$g_{15,18}$	−0.9	−8.9835616E−01	$h_{15,15}$	1	9.9862287E−01
x_{16}	10	9.9928299E+00	10	9.9928554E+00	$g_{16,12}$	2	2.0007276E+00							$h_{16,16}$	1	1.0004011E+00
x_{17}	10	9.9252898E+00	10	9.9252332E+00	$g_{17,13}$	−0.5	−5.0303316E−01							$h_{17,17}$	1	1.0070983E+00
x_{18}	10	9.9998227E+00	10	9.9998274E+00	$g_{18,14}$	1.2	1.1999980E+00							$h_{18,18}$	1	1.0000109E+00
x_{19}	10	9.9983984E+00	10	9.9983659E+00	$g_{19,12}$	1.4	1.4001749E+00	$g_{19,17}$	0.6	5.9990701E−01				$h_{19,19}$	1	1.0001175E+00
x_{20}	10	9.9936784E+00	10	9.9936483E+00	$g_{20,14}$	1	1.0004558E+00	$g_{20,17}$	1.5	1.5008471E+00				$h_{20,20}$	1	1.0006296E+00

Table 12
True and estimated parameters of the S-type **large-scale network (30 genes)** in Fig. 7. Column "true" lists the parameters of the true S-system. Column "simulation" lists the estimated parameters for a ***general*** *search range* ([0,30] for rate constants and [−4,4] for kinetic orders). Parameters are initialized to be ***random***.

	α_i		β_j		g_{ij}									h_{ij}		
	True	Simulation	True	Simulation		True	Simulation		True	Simulation		True	Simulation		True	Simulation
x_1	1	9.9996143E−01	1	9.9996097E−01	$g_{1,4}$	−0.1	−1.0000476E−01							$h_{1,1}$	1	1.0000303E+00
x_2	1	9.9998739E−01	1	9.9998702E−01										$h_{2,2}$	1	1.0000072E+00
x_3	1	9.9998978E−01	1	9.9998956E−01										$h_{3,3}$	1	1.0000067E+00
x_4	1	9.9998988E−01	1	9.9998998E−01										$h_{4,4}$	1	1.0000053E+00
x_5	1	9.9998556E−01	1	9.9998542E−01	$g_{5,1}$	1	1.0000043E+00							$h_{5,5}$	1	1.0000072E+00
x_6	1	9.9999396E−01	1	9.9999385E−01	$g_{6,1}$	1	1.0000008E+00							$h_{6,6}$	1	1.0000032E+00
x_7	1	1.0000402E+00	1	1.0000399E+00	$g_{7,2}$	0.5	4.9994609E−01	$g_{7,3}$	0.4	4.0001770E−01				$h_{7,7}$	1	9.9996973E−01
x_8	1	9.9999024E−01	1	9.9998991E−01	$g_{8,4}$	0.2	1.9999715E−01	$g_{8,17}$	−0.2	−2.0000167E−01				$h_{8,8}$	1	1.0000026E+00
x_9	1	1.0000656E+00	1	1.0000655E+00	$g_{9,5}$	1	9.9993786E−01	$g_{9,6}$	−0.1	−9.9990357E−02				$h_{9,9}$	1	9.9993853E−01
x_{10}	1	1.0000155E+00	1	1.0000154E+00	$g_{10,7}$	0.3	2.9999026E−01							$h_{10,10}$	1	9.9997191E−01
x_{11}	1	1.0001693E+00	1	1.0001788E+00	$g_{11,4}$	0.4	3.9984830E−01	$g_{11,7}$	−0.2	−1.9982997E−01	$g_{11,22}$	0.4	3.9995295E−01	$h_{11,11}$	1	9.9981619E−01
x_{12}	1	9.9995657E−01	1	9.9995637E−01	$g_{12,23}$	0.1	1.0001119E−01							$h_{12,12}$	1	1.0000336E+00
x_{13}	1	1.0000043E+00	1	1.0000042E+00	$g_{13,8}$	0.6	5.9999029E−01							$h_{13,13}$	1	9.9999063E−01
x_{14}	1	9.9999249E−01	1	9.9999176E−01	$g_{14,9}$	1	1.0000021E+00							$h_{14,14}$	1	1.0000042E+00
x_{15}	1	1.0000179E+00	1	1.0000174E+00	$g_{15,10}$	0.2	1.9999232E−01							$h_{15,15}$	1	9.9998452E−01
x_{16}	1	1.0006722E+00	1	1.0006788E+00	$g_{16,11}$	0.5	4.9968772E−01	$g_{16,12}$	−0.2	−1.9989018E−01				$h_{16,16}$	1	9.9932747E−01
x_{17}	1	1.0000211E+00	1	1.0000215E+00	$g_{17,13}$	0.5	4.9998547E−01							$h_{17,17}$	1	9.9997448E−01
x_{18}	1	9.9998799E−01	1	9.9998761E−01										$h_{18,18}$	1	1.0000070E+00
x_{19}	1	9.9999691E−01	1	9.9999679E−01	$g_{19,14}$	0.1	1.0000139E−01							$h_{19,19}$	1	1.0000083E+00
x_{20}	1	9.9990322E−01	1	9.9990243E−01	$g_{20,15}$	0.7	7.0006999E−01	$g_{20,26}$	0.3	3.0002558E−01				$h_{20,20}$	1	1.0000867E+00
x_{21}	1	9.9999388E−01	1	9.9999285E−01	$g_{21,16}$	0.6	5.9999223E−01							$h_{21,21}$	1	1.0000026E+00
x_{22}	1	9.9998611E−01	1	9.9998523E−01	$g_{22,16}$	0.5	4.9999168E−01							$h_{22,22}$	1	9.9999922E−01
x_{23}	1	1.0000134E+00	1	1.0000139E+00	$g_{23,17}$	0.2	1.9999672E−01							$h_{23,23}$	1	9.9998451E−01
x_{24}	1	9.9992788E−01	1	9.9992826E−01	$g_{24,15}$	−0.2	−2.0003160E−01	$g_{24,18}$	−0.1	−9.9988814E−02	$g_{24,19}$	0.3	3.0000607E−01	$h_{24,24}$	1	1.0000445E+00
x_{25}	1	1.0001049E+00	1	1.0001044E+00	$g_{25,20}$	0.4	3.9994772E−01							$h_{25,25}$	1	9.9990513E−01
x_{26}	1	9.9996435E−01	1	9.9996495E−01	$g_{26,21}$	−0.2	−2.0001342E−01	$g_{26,28}$	0.1	1.0001311E−01				$h_{26,26}$	1	1.0000247E+00
x_{27}	1	9.9996723E−01	1	9.9996845E−01	$g_{27,24}$	0.6	6.0003899E−01	$g_{27,25}$	0.3	2.9997533E−01	$g_{27,30}$	−0.2	−2.0000408E−01	$h_{27,27}$	1	1.0000142E+00
x_{28}	1	9.9993430E−01	1	9.9993352E−01	$g_{28,25}$	0.5	5.0001672E−01							$h_{28,28}$	1	1.0000448E+00
x_{29}	1	9.9999578E−01	1	9.9999553E−01	$g_{29,26}$	0.4	3.9999878E−01							$h_{29,29}$	1	1.0000025E+00
x_{30}	1	9.9998567E−01	1	9.9998559E−01	$g_{30,27}$	0.6	6.0000487E−01							$h_{30,30}$	1	1.0000090E+00

condition of testing at 20% beyond the training range. For instance, in the branch system the range for training is [0.4, 2.7] and 2.7(1 + 20%) = 3.24. So the initial condition for testing is set at 3.25 (beyond 3.24). In the branch system, the average accuracy rate for rate constants is 91.526105%, and that for kinetic orders is 95.192224%. In the small-scale genetic network, the average accuracy rate for rate constants is 97.919166%, and that for kinetic orders is 96.455424%.

3.2.3. Multi-objective optimization (structure identification)

We further examine CGA by identifying the *structure* of the branch system. Structure identification is a multi-objective optimization problem. To infer a sparsely connected network we have to minimize the normalized error for gene expression levels, $\underline{J_e}$, and the normalized kinetic order $\underline{J_0}$. The former is to examine the fitness between the measured data and the estimated data. The latter is to get a sparsely connected network. In our previous paper [26], we further introduced the normalized slope-error penalty $\underline{J_d}$ for smooth evolution profiles. The objective of S-system modeling is to push the gene-expression-levels error and the slope error to approach zero but to obtain a *nonzero* minimum value of the kinetic-order penalty. In other words, our objective is to get minimum value of the kinetic-order penalty under *allowable* gene-expression-levels error and slope error. Therefore, it is not suitable to summate these three because their targets are different. In our previous paper [26], we proposed a reconstruction performance index J_{rec} = -max $\{\eta_{II}\underline{J_e}, w_I\underline{J_d}, \underline{J_0}\}$ where $\eta_{II} = w_{II}\times\mu$, μ is an adaptive dynamic factor and $w_{I,}$ w_{II} are two weighting factors ($w_I = w_{II} = 1$ for most systems) [26].

Structure identification was based on the *super structure* of S-systems. For a S-system with n independent variables and m independent variables, there are $2n(n + m + 1)$ connections to be identified. For the branch system ($n = 4$ and $m = 1$), the parameters to be estimated were encoded as 48-gene cockroach individual:

α_1	α_2	α_3	α_4	β_1	β_2	β_3	β_4	g_{11}	...	g_{15}	g_{21}	...	g_{25}	...	g_{45}	h_{11}	...	h_{15}	h_{21}	...	h_{25}	...	h_{45}			

Table 4 compares the threshold, pruning ratio, assumption and search region in this study and the published research. We took ten independent runs to show the repeatability of the proposed algorithm. Table 5 is the pruning condition in each step for the ten runs. Only two of the ten runs (Runs 9 and 10) need two pruning steps to infer the correct structure. These two runs fail to truncate the redundant connection of x_4–x_1 (denoted by g_{14}) in the first step. The interaction is successfully inferred after two pruning operations. Table 6 shows the results of the first run. Due to space limitation, the results of other nine runs are shown in the supplement file. The simulation results in Table 6 show that the value gap between the redundant (value below 10^{-14}) and possible connections is obvious, as shown in Step 1 of Table 6. The pruned structure is relearned to get a modified structure. The inferred structure shown in Step 2 is identical to the true structure shown in Step 0, and the estimated parameters are all nearly the same as those of the true system.

3.2.4. Parameter sloppiness

We now further discuss if S-systems possess the sloppy features: a few stiff parameters, many sloppy parameters and the distribution of the eigenvalues of the sensitivity-related Hessian matrix H, which is expressed in Eq. (22), over many decades. For the parameter vector changing from θ to θ^*, we define the average squared change of the constitute x_i as $\aleph(\theta)$ [56],

$$\aleph(\theta) = \frac{1}{2nK}\sum_{k=1}^{k=K}\sum_{i=1}^{i=n}\frac{1}{T_k}\int_0^{T_k}\left[\frac{x_i^k(\theta,t)-x_i^k(\theta^*,t)}{x_{i,\max}}\right]^2 dt \tag{21}$$

where n is the number of constitutes, $x_{i,\max}$ is the maximum of $x_i^k, i = 1,\ldots,n,\ k = 1,\ldots,K$ and T_k is the time period of the kth conditions (accounting periods). System sensitivity to parameter variation is estimated through the Hessian matrix,

$$H^x_{l,m} = \frac{\partial^2\aleph}{\partial\log\theta_l\cdot\partial\log\theta_m} \tag{22}$$

Fig. 9 shows the respective normalized eigenvalue spectra of the Hessian matrixes for the five S-systems. We observe that nearly all eigenvalues span less than one decade (except one eigenvalue of the 20-gene S-system, which is around 0.05). Models are sloppy when the respective eigenvalues span over more than six decades (10^{-6}). Therefore, the sloppy phenomenon observed in Hill and Michaelis–Menten models [56] does not exist in these five S-systems.

4. Conclusion

Identifying a dynamic biological system from time-series data is a central theme in systems biology. S-system model is good in showing the net interactive effect. In this paper, we mimic cockroaches′ competition behavior which is then embedded into advanced genetic algorithms to increase exploration and exploitation abilities. Simulation results show that the global-search ability is ensured even in a wide search space with a bad initial start. CGA is demonstrated to learn with a rather fast speed as compared to the state-of-the-art GA and DE (SPXGA [23], intelligent GA [15], improved GA [35], DE, HDE [12–14]). We also examine the robustness of CGA with systems under noise contaminate. Training data is generated from the true data with 10% random noise. 20% deviation from the training range is used for testing. Simulation results show that CGA is robust to system noise.

Appendix A

See Figs. 10 and 11, Tables 7-12.

Appendix B. Supplementary data

Supplementary data associated with this article can be found, in the online version, at http://dx.doi.org/10.1016/j.mbs.2013.07.019.

References

[1] M. Vilela, I.C. Chou, S. Vinga, A.T.R. Vasconcelos, E.O. Voit, J.S. Almeida, Parameter optimization in S-system models, BMC Syst. Biol. 2 (35) (2008).

[2] L. Michaelis, M.I. Menten, Die kinetik der invertinwirkung, Biochem. Z. 49 (1913) 333.

[3] I.C. Chou, E.O. Voit, Estimation of dynamic flux profiles from metabolic time series data, BMC Syst. Biol. (2012).

[4] M.A. Savageau, Biochemical Systems Analysis: A Study of Function and Design in Molecular Biology, Addison-Wesley, Reading, Massachusetts, 1976.

[5] E.O. Voit, Computational Analysis of Biochemical Systems: A Practical Guide for Biochemists and Molecular Biologists, Cambridge University Press, Cambridge, UK, 2000.

[6] H. Wang, L. Qian, E. Dougherty, Inference of Gene Regulatory Networks using S-system: A Unified Approach, IET Syst. Biol. 4 (2010) 145.

[7] S. Marino, E.O. Voit, An automated procedure for the extraction of metabolic network information from time series data, Bioinf.. Comput. Biol. 4 (3) (2006) 665.

[8] I.C. Chou, H. Martens, E.O. Voit, Parameter estimation in biochemical systems models with alternating regression, Theory Biol. Med. Model. 3 (25) (2006).

[9] Z. Kutalik, W. Tucker, V. Moulton, S-system parameter estimation for noisy metabolic profiles using Newton-flow analysis, IET Syst. Biol. 1 (2007) 174.

[10] E. Sakamoto, H. Iba, Inferring a system of differential equations for a gene regulatory network by using genetic programming, in: 2001 CEC: Proceedings of Congress Evolutionary Computation, vol. 1, pp. 720–726.

[11] S. Ando, E. Sakamoto, H. Iba, Evolutionary modeling inference of gene network, Inf. Sci. 145 (2002) 237.

[12] K.Y. Tsai, F.S. Wang, Evolutionary optimization with data collocation for reverse engineering of biological networks, Bioinformatics 21 (2005) 1180.

[13] P.K. Liu, F.S. Wang, Hybrid differential evolution with geometric mean mutation in parameter estimation of bioreaction systems with large parameter search space, Comput. Chem. Eng. 33 (2009) 1851.

[14] F.S. Wang, P.K. Liu, Inverse problems of biochemical systems using hybrid differential evolution and data collocation, Int. J. Syst. Synth. Biol. 1 (2010) 21.

[15] S.Y. Ho, C.H. Hsieh, F.C. Yu, H.L. Huang, An intelligent two-stage evolutionary algorithm for dynamic pathway identification from gene expression profiles, IEEE/ACM Trans. Comput. Biol. Bioinf. 4 (2007) 648.

[16] O.R. Gonzalez, C. Küper, K. Jung, P.C. Naval, E. Mendoza Jr, E. Mendoza, Parameter estimation using simulated annealing for S-system models of biochemical networks, Bioinformatics 23 (2007) 480.

[17] C.M. Chen, C. Lee, C.L. Chuang, C.C. Wang, G.S. Shieh, Inferring genetic interactions via a nonlinear model and an optimization algorithm, BMC Syst. Biol. 4 (16) (2010).

[18] Y. Matsubara, S. Kikuchi, M. Sugimoto, M. Tomita, Parameter estimation for stiff equations of biosystems using radial basis function networks, BMC Bioinf. 7 (230) (2006).

[19] H. Murata, M. Koshino, M. Mitamura, H. Kimura, Inference of S-system models of genetic networks using product unit neural networks, in 2008 SMC: IEEE Conference on Systems, Man, and Cybernetics, pp. 1390–1395.

[20] R. Xu, D.C. Wunsch II, R.L. Frank, Inference of genetic regulatory networks with recurrent neural network models using particle swarm optimization, IEEE Trans. Comput. Biol. Bioinf. 4 (2007) 1545.

[21] S. Kimura, K. Ide, A. Kashihara, M. Kano, H. Mariko, R. Masui, N. Nakagawa, S. Yokoyama, S. Kuramitsu, A. Konagaya, Inference of S-system models of genetic networks using a cooperative coevolutionary algorithm, Bioinformatics 21 (2005) 1154.

[22] N. Noman, H. Iba, Inference of gene regulatory networks using S-system and differential evolution, in: 2005 GECCO: Proceedings of Conference Genetic Evolutionary Computation, vol. 1, pp. 439–446.

[23] S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, M. Tomita, Dynamic modeling of genetic networks using genetic algorithm and S-system, Bioinformatics 19 (2003) 643.

[24] D. Thieffry, A.M. Huerta, E. Perez-Rueda, J. Collado-Vides, From specific gene regulation to genomic networks: a global analysis of transcriptional regulation in *Escherichia coli*, BioEssays 20 (1998) 433.

[25] P.K. Liu, F.S. Wang, Inference of biochemical network models in S-system using multiobjective optimization approach, Bioinformatics 24 (2008) 1085.

[26] S.J. Wu, C.T. Wu, J.Y. Chang, Fuzzy-based self-interactive multi-objective evolution optimization for reverse engineering of biological networks, IEEE Trans. Fuzzy Syst. 20 (5) (2012) 865.

[27] I.C. Chou, E.O. Voit, Recent developments in parameter estimation and structure identification of biochemical and genomic systems, Math. Biosci. 219 (2009) 57.

[28] J. Sun, J.M. Garibaldi, C. Hodgman, Parameter estimation using metaheuristics in systems biology: a comprehensive review, IEEE/ACM Trans. Comput. Biol. Bioinf. 9 (1) (2012) 185–202.

[29] E.O. Voit, Biochemical systems theory: a review, ISRN Biomath. vol. 2013 (2013), Article ID 897658, 53 pages.

[30] M. Harman, P. McMinn, A theoretical and empirical study of search based testing: local, global and hybrid search, IEEE Trans. Software Eng. 36 (2) (2010) 226.

[31] H. Soh, Y.S. Ong, Q.C. Nguyen, Q.H. Nguyen, M.S. Habibullah, T. Hung, J.L. Kuo, Discovering unique, low-energy pure water isomers: memetic exploration, optimization, and landscape analysis, IEEE Trans. Evol. Comput. 14 (3) (2010) pp. 419–437.

[32] Y. Ahn, J. Park, C.G. Lee, J.W. Kim, S.Y. Jung, Novel memetic algorithm implemented with GA (genetic algorithm) and MADS (mesh adaptive direct search) for optimal design of electromagnetic system, IEEE Trans. Mag. 46 (6) (2010) 1982.

[33] R. Meuth, M.H. Lim, Y.S. Ong, D.C. Wunsch, A proposition on memes and meta-memes in computing for higher-order learning, Memetic Comput. 1 (2) (2009) 85.

[34] O. Kramer, Iterated local search with Powell's method: a memetic algorithm for continuous global optimization, Memetic Comput. 2 (1) (2010) 69.

[35] A. Caponio, G.L. Cascella, F. Neri, N. Salvatore, M. Sumner, A fast adaptive memetic algorithm for online and offline control design of PMSM drives, IEEE Trans. Syst. Man Cybern.-Part B: Cybern. 37 (1) (2007) 28.

[36] F. Neri, J. Toivanen, G. Cascella, Y.S. Ong, An adaptive multimeme algorithm for designing HIV multidrug therapies, IEEE/ACM Trans. Comput. Biol. Bioinf. 4 (2) (2007) 264.

[37] H. Wang, D. Wang, S. Yang, A memetic algorithm with adaptive hill climbing strategy for dynamic optimization problems, Soft Comput. 13 (8–9) (2008) 763.

[38] Q. Shen, W.M. Shi, W. Kong, Hybrid particle swarm optimization and tabu search approach for selecting genes for tumor classification using gene expression data, Comput. Biol. Chem. 32 (2007) 53.

[39] S. Song, L. Kong, Y. Gan, R. Su, Hybrid particle swarm cooperative optimization algorithm and its application to MBC in alumina production, Prog. Nat. Sci. 18 (2008) 1423.

[40] E. Keedwell, S.T. Khu, A hybrid genetic algorithm for the design of water distribution networks, Eng. Appl. Artif. Intel. 18 (4) (2005) 461.

[41] I.G. Tsoulos, I.E. Lagaris, GenMin: an enhanced genetic algorithm for global optimization, Comput. Phys. Commun. 178 (2008) 843–851.

[42] S.X. Yang, S.N. Jat, Genetic algorithms with guided and local search strategies for university course timetabling, IEEE Trans. Syst. Man Cybern. Part C – Appl. Rev. 41 (Jan. 2011) 93.

[43] K. Ganesh, M. Punniyamoorthy, Optimization of continuous-time production planning using hybrid genetic algorithms-simulated annealing, Int. J. Adv. Manuf. Technol. 26 (2004) 148.

[44] M.Y. Cheng, K.Y. Huang, Genetic algorithm-based chaos clustering approach for nonlinear optimization, J. Mar. Sci. Technol. 18 (Jun. 2010) 435.

[45] X.M. Jiang, H. Adel, Neuro-genetic algorithm for non-linear active control of structures, Int. J. Numer. Methods Eng. 75 (2008) 770–786.

[46] Y.T. Kao, E. Zahara, A hybrid genetic algorithm and particle swarm optimization for multimodal functions, Appl. Soft Comput. 8 (2008) 849.

[47] A. Shunmugalatha, S.M.R. Slochanal, Optimum cost of generation for maximum loadability limit of power system using hybrid particle swarm optimization, Int. J. Electr. Power Energy Syst. 30 (2008) 486.

[48] S. Jeong, S. Hasegawa, K. Shimoyama, S. Obayashi, Development and investigation of efficient GA/PSO-hybrid algorithm applicable to real-world design optimization, IEEE Comput. Intell. Mag. 1 (2009) 36–44.

[49] T.C. Havens, C.J. Spain, N.G. Salmon, J.M. Keller, Roach infestation optimization, in: 2008 Conference on IEEE Swarm Intelligence, Symposium, pp. 21–23.

[50] Z. Chen, A Modified Cockroach Swarm Optimization, in: 2011 Conferene, Energy Procedia, pp. 4–9.

[51] D.S. Falconer, T.F.C. MacKay, Introduction to Quantitative Genetics, fourth ed., Longman Inc, New York, 1996.

[52] W.S. Hlavacek, M.A. Savageau, Rules for coupled expression of regulator and effector genes in inducible circuits, J. Mol. Biol. 255 (1996) 121.

[53] N. Noman, H. Iba, Inference of genetic networks using S-system: Information criteria for model selection, in: 2006 GECCO: Proceedings of Conference Genetic Evolutionary Computation, pp. 263–270.

[54] S.J. Wu, C.H. Chou, C.T. Wu, T.T. Lee, Inference of genetic network of xenopus frog egg: improved genetic algorithm, in: 2006 Conference Proceedings on IEEE Engineering in Medicine and Biology Society, pp. 4147–4150.

[55] E.O. Voit, J. Almeida, Decoupling dynamical systems for pathway identification from metabolic profiles, Bioinformatics 20 (2004) 1670.

[56] R.N. Gutenkunst, J.J. Waterfall, F.P. Casey, K.S. Brown, C.R. Myers, J.P. Sethna, Universally sloppy parameter sensitivities in systems biology models, PLoS Comput. Biol. 3 (10) (2007) e189.