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

Shinq-Jen Wu, Cheng-Tao Wu

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

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.

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

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-
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 $n$ independent constitutes and $m$ independent constitutes, the change of gene expression level $x_i$ is

$$
\frac{dx_i}{dt} = v_i^* - v_i^* = g_i \Pi_{j=1}^{n-m} x_j^{\beta_i} - h_i \Pi_{j=1}^{n-m} x_j^{\gamma_i}, \quad i = 1, 2, \ldots, n,
$$

where $g_i$ and $h_i$ are the rate constants, and $g_i$ and $h_i$ 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 exploring the gradualness of continuous variables. The unknown parameters of an S-system are encoded as a cockroach individual (a chromosome in evolution operations):

$$
\begin{array}{cccccccc}
\alpha_0 & \ldots & \alpha_x & \beta_0 & \ldots & \beta_x & g_1 & \ldots & g_{x(n-m)} & h_1 & \ldots & h_{x(n-m)}
\end{array}
$$

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 $I_t$, $i = 1, \ldots, N_P$:

$$
I_t = I_{\text{min}} + r(I_{\text{max}} - I_{\text{min}}),
$$

where $r$ is a random number, $N_P$ is the total number of individuals in the population, and $I_{\text{min}}$ and $I_{\text{max}}$ are the lower and upper bounds of individual $I_i$. The parameter vector $I_i$ denotes the position of the $i$th cockroach. For simplification, we call $I_t$ the $i$th 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),

$$
J_e = \frac{1}{nQ} \sum_{k=1}^{Q} \sum_{t=1}^{n} \left( \frac{x_i^{\text{exp}} - x_i^*}{\max(x_i^{\text{exp}})} \right)^2,
$$

where $x_i^{\text{exp}}$ and $x_i^*$ are, respectively, the $k$th estimated and artificially experimental data of the constitute $x_i$, $t_s$ 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 defending 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 the neighborhood (global-best position). PSO lies in accelerating the movement of 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
more huge and complex biological systems more teams are needed. We use $n_0$ to denote the number of individuals in the first three teams; i.e., the total number of individuals in the population is $N_p = n_0 + n_4$. $[\star]$ is a Gauss mark and $\tau$ is the golden-section constant.

\[ (1^{st} \text{team}) n_1 = [\tau \cdot n_4] \text{ with leader } l_1^1 = I_1, \]
\[ (2^{nd} \text{team}) n_2 = [\tau \cdot (n_4 - n_1)] \text{ with leader } l_2^2 = I_{n_1+1} \]
\[ (3^{rd} \text{team}) n_3 = n_4 - n_1 - n_2 \text{ with leader } l_3^3 = I_{n_1+n_2+1}, \]
\[ (4^{th} \text{team}) n_4 = [(1 - \tau) \cdot N_p] \text{ with leader } l_4^4 = I_{n_4+1}. \]

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 snatch behavior more flexible we assume the strongest cockroach run in a derivative-based (straight forward) way, but other leaders crawl in a derivative-free (circuital) way. Hungry cockroaches search for food. At the sight of food the strongest cockroach (the best individual $l_4 = l^4_i$) dashes forward to food at the speed of the steepest descent,

\[ l_1^1 = I_b - \lambda \nabla l_1, \]

where $\lambda$ is the size of a step to determine the descent rate and $\nabla l_1$ is the gradient of an objective function. Other leaders $l_i, i = 2, 3, 4$ rush for food at the speed of the downhill Simplex: Replace $l_i$ with the better point generated from the following reflection, expansion and contraction operations. If this operation fails, shrink towards the best vertex $I_b$.

\[ \text{(reflection) } \tilde{x}_i = c + \alpha (c - l_i), \]
\[ \text{(expansion) } x_i = c + \beta (x_i - c), \]
\[ \text{(contraction) } x_i = c + \gamma (x_i - c) \text{ and } c + \gamma (l_i - c). \]
\[ \text{(shrink) } l_i^j \text{ is replaced by } (l_i + I_b)/2, \]

where $\alpha, \beta, \gamma$ are reflection, expansion and contraction coefficients, respectively, and $c$ is the centroid of all vertives better than $l_i$ (centroid of the better side). Parameters $\alpha, \beta, \gamma, \lambda$ are set to be random such that the algorithm is stochastic and is able to get out of local minima. When the exploitation success 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 cockroaches $l_i$ jumps away from their leader $l_i^j$;
and \( I_b \) stay at the original place.

\[ I_{b+} = \frac{1}{C_3} \text{ with backcrossing probability} \]

These are \[ 51 \] if no blood relationship

\[ 6 \]

\[ 1 \]

\[ = \frac{1}{C_0} \]

\[ = 1 \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]

\[ = \frac{1}{C_0} \]
Table 2  
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).

<table>
<thead>
<tr>
<th>$x_i$</th>
<th>$f_i$</th>
<th>$g_{ij}$</th>
<th>$h_{ij}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>Simulation</td>
<td>True</td>
<td>Simulation</td>
</tr>
<tr>
<td>1</td>
<td>1.0000126E+00</td>
<td>1</td>
<td>1.0000130E+00</td>
</tr>
<tr>
<td>2</td>
<td>9.9998940E-01</td>
<td>1</td>
<td>9.9998908E-01</td>
</tr>
<tr>
<td>3</td>
<td>9.9997693E-01</td>
<td>1</td>
<td>9.9997639E-01</td>
</tr>
<tr>
<td>4</td>
<td>9.9998236E-01</td>
<td>1</td>
<td>9.9998237E-01</td>
</tr>
<tr>
<td>5</td>
<td>9.9998652E-01</td>
<td>1</td>
<td>9.9998545E-01</td>
</tr>
<tr>
<td>6</td>
<td>1.0002472E+00</td>
<td>1</td>
<td>1.0002458E+00</td>
</tr>
<tr>
<td>7</td>
<td>1.00001088E+00</td>
<td>1</td>
<td>1.00001104E+00</td>
</tr>
<tr>
<td>8</td>
<td>1.0006263E+00</td>
<td>1</td>
<td>1.0006260E+00</td>
</tr>
<tr>
<td>9</td>
<td>1.0001683E+00</td>
<td>1</td>
<td>1.0001717E+00</td>
</tr>
<tr>
<td>10</td>
<td>1.0000415E+00</td>
<td>1</td>
<td>1.0000423E+00</td>
</tr>
<tr>
<td>11</td>
<td>1.0000037E+00</td>
<td>1</td>
<td>1.0000036E+00</td>
</tr>
<tr>
<td>12</td>
<td>1.0000049E+00</td>
<td>1</td>
<td>1.0000050E+00</td>
</tr>
<tr>
<td>13</td>
<td>9.9993941E-01</td>
<td>1</td>
<td>9.9997396E-01</td>
</tr>
<tr>
<td>14</td>
<td>9.9996771E-01</td>
<td>1</td>
<td>9.9996641E-01</td>
</tr>
<tr>
<td>15</td>
<td>9.9996812E-01</td>
<td>1</td>
<td>9.9996812E-01</td>
</tr>
<tr>
<td>16</td>
<td>1.0000766E+00</td>
<td>1</td>
<td>1.0000774E+00</td>
</tr>
<tr>
<td>17</td>
<td>1.0000047E+00</td>
<td>1</td>
<td>1.0000046E+00</td>
</tr>
<tr>
<td>18</td>
<td>9.9996642E-01</td>
<td>1</td>
<td>9.9996650E-01</td>
</tr>
<tr>
<td>19</td>
<td>9.9996642E-01</td>
<td>1</td>
<td>9.9996650E-01</td>
</tr>
<tr>
<td>20</td>
<td>1.0000049E+00</td>
<td>1</td>
<td>1.0000050E+00</td>
</tr>
<tr>
<td>21</td>
<td>1.0000323E+00</td>
<td>1</td>
<td>1.0000323E+00</td>
</tr>
<tr>
<td>22</td>
<td>1.0000983E+00</td>
<td>1</td>
<td>1.0000983E+00</td>
</tr>
<tr>
<td>23</td>
<td>1.0000394E+00</td>
<td>1</td>
<td>1.0000394E+00</td>
</tr>
<tr>
<td>24</td>
<td>1.0002889E+00</td>
<td>1</td>
<td>1.0002889E+00</td>
</tr>
<tr>
<td>25</td>
<td>1.0000588E+00</td>
<td>1</td>
<td>1.0000588E+00</td>
</tr>
<tr>
<td>26</td>
<td>1.0000237E+00</td>
<td>1</td>
<td>1.0000237E+00</td>
</tr>
<tr>
<td>27</td>
<td>9.9916699E-01</td>
<td>1</td>
<td>9.9916699E-01</td>
</tr>
<tr>
<td>28</td>
<td>1.0000797E+00</td>
<td>1</td>
<td>1.0000797E+00</td>
</tr>
<tr>
<td>29</td>
<td>1.0001221E+00</td>
<td>1</td>
<td>1.0001221E+00</td>
</tr>
</tbody>
</table>
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 \(\mathbf{X}(t)\) is approximated by a set of polynomial \(X_m\), which is further expressed as a linear combination of \(m\) bases functions, 

\[
X_m(t) = \sum_{j=1}^{m} \phi_j(t)X_c(j),
\]

where \(X_c(j)\) is the expansion coefficients at the \(j\)th 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],

\[
X_c(j) \cong X_c(j-1) + 0.5\eta_t (f[X_c(j), \theta] + f[X_c(j-1), \theta]), \quad j = 1, \ldots, m. \quad (15)
\]

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), \(\theta\) is the parameters of the S system and \(\eta_t\) 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

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Initial parameter value</th>
<th>Search range</th>
<th>Variable</th>
<th>Global optimal probability</th>
</tr>
</thead>
<tbody>
<tr>
<td>CGA (static population)</td>
<td>Random</td>
<td>General</td>
<td>(x_1)</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(x_2)</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(x_3)</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Wide</td>
<td>(x_1)</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(x_2)</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(x_3)</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Bad</td>
<td>(x_1)</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(x_2)</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(x_3)</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td>CGA+ (dynamic population)</td>
<td>Bad</td>
<td>Wide</td>
<td>(x_1)</td>
<td>100%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(x_2)</td>
<td>100%</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>(x_3)</td>
<td>100%</td>
<td></td>
</tr>
</tbody>
</table>
and 

\[
\begin{aligned}
\frac{v}{C_0} & = 0.3 \\
\frac{w}{C_0} & = 0.4 \\
\frac{x}{C_0} & = 0.2 \\
\frac{y}{C_0} & = 0.1 \\
\end{aligned}
\]

are the dependent constituents and \( z \) is the source input (the independent constituent). Based on the network, we have the respective S-system model prototype:

\[
\begin{aligned}
d_{0} & = 2 \\
d_{1} & = 8 \\
d_{2} & = 5 \\
d_{3} & = 2 \\
d_{4} & = 1 \\
\end{aligned}
\]

are the unknown parameters. The simulation time of an experiment is 8 s and the sample time is 100,100 s. 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 s. Row “Case 1” in Table 7 lists the estimated parameters for a general search space [0, 30] for rate constants and [-100, 100] for kinetic orders with a random 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. The unknown parameters are encoded as a 17-gene cockroach individual:

\[
\begin{aligned}
X_1 & = 2X_3^2X_5 - \beta_1X_1^{11} \\
X_2 & = 2X_1^2 - \beta_2X_2^2 \\
X_3 & = 3X_3^2X_5 - \beta_3X_3^{13}X_4^{11} \\
X_4 & = 3X_1X_4 - \beta_4X_4^{12} \\
\end{aligned}
\]

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

\[
\begin{aligned}
\alpha, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8, \beta_9, \beta_{10}, \beta_{11}, \beta_{12}, \beta_{13}, \beta_{14}, \beta_{15}, \beta_{16}, \beta_{17} \\
\end{aligned}
\]

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

\[
\begin{aligned}
\alpha, \beta_1, \beta_2, \beta_3, \beta_4, \beta_5, \beta_6, \beta_7, \beta_8, \beta_9, \beta_{10}, \beta_{11}, \beta_{12}, \beta_{13}, \beta_{14}, \beta_{15}, \beta_{16}, \beta_{17} \\
\end{aligned}
\]

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 s. Row “Case 1” in Table 7 lists the estimated parameters for a general search space [0, 30] for rate constants and [-100, 100] 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.

### 3.1.1. 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 \( x_0 \) is the source input. Based on the network, we have the respective S-system model prototype:

\[
\begin{aligned}
x_1 & = 2X_3^2X_5 - \beta_1X_1^{11} \\
x_2 & = 2X_1^2 - \beta_2X_2^2 \\
x_3 & = 3X_3^2X_5 - \beta_3X_3^{13}X_4^{11} \\
x_4 & = 3X_1X_4 - \beta_4X_4^{12} \\
\end{aligned}
\]
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.

<table>
<thead>
<tr>
<th>Step</th>
<th>Variable</th>
<th>$x_i$</th>
<th>$g_{i1}$</th>
<th>$g_{i2}$</th>
<th>$g_{i3}$</th>
<th>$g_{i4}$</th>
<th>$g_{i5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$x_1$</td>
<td>20</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td>-0.8</td>
</tr>
<tr>
<td></td>
<td>$x_2$</td>
<td>8</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>$x_3$</td>
<td>3</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>0.75</td>
</tr>
<tr>
<td></td>
<td>$x_4$</td>
<td>2</td>
<td>0.5</td>
<td></td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>$x_1$</td>
<td>1.9869487E+01</td>
<td>2.0350824E-15</td>
<td>-1.0121408E-16</td>
<td>-7.5888690E-01</td>
<td>-2.0782613E-17</td>
<td>9.2950239E-01</td>
</tr>
<tr>
<td></td>
<td>$x_2$</td>
<td>8.3421078E+00</td>
<td>4.6928402E-01</td>
<td>-1.2346223E-15</td>
<td>-5.259489E-16</td>
<td>-5.5289283E-16</td>
<td>-5.5075463E-17</td>
</tr>
<tr>
<td></td>
<td>$x_3$</td>
<td>3.1105792E+00</td>
<td>-1.8536116E-16</td>
<td>7.2585091E-01</td>
<td>-1.6276487E-16</td>
<td>-3.2226031E-16</td>
<td>3.5381885E-16</td>
</tr>
<tr>
<td></td>
<td>$x_4$</td>
<td>2.2156428E+00</td>
<td>4.5628239E-01</td>
<td>8.1032598E-16</td>
<td>-7.8526363E-16</td>
<td>5.7838675E-16</td>
<td>1.1071731E-16</td>
</tr>
<tr>
<td>2</td>
<td>$x_1$</td>
<td>1.9989784E+01</td>
<td>4.9921193E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$x_2$</td>
<td>8.0134341E+00</td>
<td>4.9995519E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$x_3$</td>
<td>3.0050430E+00</td>
<td>4.9928636E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>$x_4$</td>
<td>2.0001301E+00</td>
<td>4.9995519E-01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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

$$
\dot{x}_1 = \alpha_1 x_2^2 x_3 x_4 - \beta_1 x_1^{13},
\dot{x}_2 = \alpha_2 x_1^2 x_1 x_2 - \beta_2 x_2^{12},
\dot{x}_3 = \alpha_3 x_2^2 x_3 - \beta_3 x_2^2 x_3^{32},
\dot{x}_4 = \alpha_4 x_4^3 x_5 x_6 - \beta_4 x_4^{33},
\dot{x}_5 = \alpha_5 x_5^3 x_7 - \beta_5 x_5^{35},
$$

where $x_i, i = 1, \ldots, 5$ are the dependent constituents, and $x_6, i = 6, 7, 8$ are the constant sources (independent constituents). The unknown parameters were encoded as a 23-gene cockroach individual.
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.

\[
\begin{align*}
\dot{x}_1 &= -\beta_1 x_1^{a_1}, \\
\dot{x}_2 &= -\beta_2 x_2^{a_2}, \\
\dot{x}_3 &= -\beta_3 x_3^{a_3}, \\
\dot{x}_4 &= -\beta_4 x_4^{a_4}, \\
\dot{x}_5 &= -\beta_5 x_5^{a_5}, \\
\dot{x}_6 &= -\beta_6 x_6^{a_6}, \\
\dot{x}_7 &= -\beta_7 x_7^{a_7}, \\
\dot{x}_8 &= -\beta_8 x_8^{a_8}, \\
\dot{x}_9 &= -\beta_9 x_9^{a_9}, \\
\dot{x}_{10} &= -\beta_{10} x_{10}^{a_{10}}, \\
\dot{x}_{11} &= -\beta_{11} x_{11}^{a_{11}}, \\
\dot{x}_{12} &= -\beta_{12} x_{12}^{a_{12}}, \\
\dot{x}_{13} &= -\beta_{13} x_{13}^{a_{13}}, \\
\dot{x}_{14} &= -\beta_{14} x_{14}^{a_{14}}, \quad \text{(19)}
\end{align*}
\]

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

\[
\begin{align*}
\dot{x}_1 &= -\beta_1 x_1^{a_1}, \\
\dot{x}_2 &= -\beta_2 x_2^{a_2}, \\
\dot{x}_3 &= -\beta_3 x_3^{a_3}, \\
\dot{x}_4 &= -\beta_4 x_4^{a_4}, \\
\dot{x}_5 &= -\beta_5 x_5^{a_5}, \\
\dot{x}_6 &= -\beta_6 x_6^{a_6}, \\
\dot{x}_7 &= -\beta_7 x_7^{a_7}, \\
\dot{x}_8 &= -\beta_8 x_8^{a_8}, \\
\dot{x}_9 &= -\beta_9 x_9^{a_9}, \\
\dot{x}_{10} &= -\beta_{10} x_{10}^{a_{10}}, \\
\dot{x}_{11} &= -\beta_{11} x_{11}^{a_{11}}, \\
\dot{x}_{12} &= -\beta_{12} x_{12}^{a_{12}}, \\
\dot{x}_{13} &= -\beta_{13} x_{13}^{a_{13}}, \\
\dot{x}_{14} &= -\beta_{14} x_{14}^{a_{14}}, \\
\dot{x}_{15} &= -\beta_{15} x_{15}^{a_{15}}, \\
\dot{x}_{16} &= -\beta_{16} x_{16}^{a_{16}}, \\
\dot{x}_{17} &= -\beta_{17} x_{17}^{a_{17}}, \\
\dot{x}_{18} &= -\beta_{18} x_{18}^{a_{18}}, \\
\dot{x}_{19} &= -\beta_{19} x_{19}^{a_{19}}, \\
\dot{x}_{20} &= -\beta_{20} x_{20}^{a_{20}}, \quad \text{(20)}
\end{align*}
\]

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 \((N = 3, 4, 20 \text{ genes})\) in the wide search space \([0,100]\) for rate constants and \([-100,100]\) for kinetic orders and a bad initial start \((80 \text{ for all parameters})\). Low-, medium- and large-dimensional systems \((N = 5, 20, 30 \text{ genes})\) with the same search space and initial start as [15] are used in Fig. 11, where the nearby-midpoint initial start \((\text{zero for kinetic order and 7.5 for rate constant})\) 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 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 I 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.
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).

<table>
<thead>
<tr>
<th>Variable</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$g_{i1}$</th>
<th>$g_{i2}$</th>
<th>$g_{i3}$</th>
<th>$h_{i1}$</th>
<th>$h_{i2}$</th>
<th>$h_{i3}$</th>
<th>$h_{i4}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>20</td>
<td>10</td>
<td>8</td>
<td>3</td>
<td>0.5</td>
<td>0.75</td>
<td>0.5</td>
<td>0.5</td>
<td>0.5</td>
<td>0.2</td>
<td>0.8</td>
</tr>
<tr>
<td></td>
<td>1.9996496E+01</td>
<td>9.9986163E+00</td>
<td></td>
<td>–7.9994627E–01</td>
<td>4.9999159E–01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case1 $x_1$</td>
<td>8.0020030E+00</td>
<td>3.0001482E+00</td>
<td>4.9999822E–01</td>
<td></td>
<td>7.4997682E–01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.0043335E+00</td>
<td>5.0056201E+00</td>
<td>7.4932029E–01</td>
<td></td>
<td>4.9952091E–01</td>
<td>1.9980073E–01</td>
<td>7.9996349E–01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.9998961E+01</td>
<td>1.0000011E+01</td>
<td></td>
<td>–7.9989604E–01</td>
<td>4.9996658E–01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case2 $x_1$</td>
<td>8.0039345E+00</td>
<td>3.0028797E+00</td>
<td>4.9977047E–01</td>
<td></td>
<td>7.4961995E–01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.0017547E+00</td>
<td>5.0025795E+00</td>
<td>7.4979131E–01</td>
<td></td>
<td>4.9981801E–01</td>
<td>1.9996480E–01</td>
<td>7.9938683E–01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th>Variable</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$g_{i1}$</th>
<th>$g_{i2}$</th>
<th>$g_{i3}$</th>
<th>$h_{i1}$</th>
<th>$h_{i2}$</th>
<th>$h_{i3}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>10</td>
<td>5</td>
<td>2</td>
<td>0.5</td>
<td>–0.1</td>
<td>–0.05</td>
<td>0.5</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.9983000E+00</td>
<td>4.9991321E+00</td>
<td></td>
<td>–1.0001159E–01</td>
<td>–5.0009436E–02</td>
<td>5.0000103E–01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case1 $x_2$</td>
<td>2.0036028E+00</td>
<td>1.4436798E+00</td>
<td>4.9929160E–01</td>
<td></td>
<td>4.9901331E–01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>2.9993602E+00</td>
<td>7.1998806E+00</td>
<td>5.0011201E–01</td>
<td></td>
<td>5.0011992E–01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.9980346E+00</td>
<td>4.9984476E+00</td>
<td></td>
<td>–1.0001760E–01</td>
<td>–4.9995651E–02</td>
<td>5.0014191E–01</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case2 $x_3$</td>
<td>2.00664165E+00</td>
<td>1.4466246E+00</td>
<td>4.9875874E–01</td>
<td></td>
<td>4.9821637E–01</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.00193140E+00</td>
<td>7.2004725E+00</td>
<td></td>
<td>4.9966966E–01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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).

<table>
<thead>
<tr>
<th>Variable</th>
<th>$x_1$</th>
<th>$x_2$</th>
<th>$x_3$</th>
<th>$x_4$</th>
<th>$x_5$</th>
<th>$g_{i1}$</th>
<th>$g_{i2}$</th>
<th>$g_{i3}$</th>
<th>$g_{i4}$</th>
<th>$g_{i5}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>5</td>
<td>10</td>
<td>8</td>
<td>10</td>
<td>1</td>
<td>–1</td>
<td>–1</td>
<td>–1</td>
<td>–1</td>
<td>–1</td>
</tr>
<tr>
<td></td>
<td>1.00025244E+00</td>
<td>2.0002799E+00</td>
<td></td>
<td>1.0004463E+00</td>
<td>–1.0005928E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case1 $x_1$</td>
<td>1.0000364E+01</td>
<td>1.0003655E+00</td>
<td></td>
<td>2.0010040E+00</td>
<td>2.0004907E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0001371E+01</td>
<td>1.00016401E+00</td>
<td></td>
<td>1.0004428E+00</td>
<td>–1.0007198E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.9996171E+00</td>
<td>2.0001203E+00</td>
<td></td>
<td>2.00012278E+00</td>
<td>–1.0006360E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case2 $x_2$</td>
<td>1.00018854E+01</td>
<td>9.9989991E+00</td>
<td></td>
<td>2.0003787E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.9986509E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Variable</td>
<td>$h_{i1}$</td>
<td>$h_{i2}$</td>
<td>$h_{i3}$</td>
<td>$h_{i4}$</td>
<td>$h_{i5}$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td>----------</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>True</td>
<td>10</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.0002901E+01</td>
<td>1.9990869E+00</td>
<td></td>
<td>2.0001225E+00</td>
<td>1.9998620E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Case1 $x_1$</td>
<td>9.9998865E+00</td>
<td>1.0005392E+00</td>
<td></td>
<td>1.9998620E+00</td>
<td>1.9998910E+00</td>
<td>2.0002320E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.9999571E+00</td>
<td>1.9990220E+00</td>
<td></td>
<td>1.9998824E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.9986818E+00</td>
<td></td>
<td></td>
<td>1.9998054E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>9.9984596E+00</td>
<td></td>
<td></td>
<td>2.0001887E+00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
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 general search range ([0,30] for rate constants and [-4,4] for kinetic orders). Parameters are initialized to be random.

<table>
<thead>
<tr>
<th>$i$</th>
<th>True Simulation</th>
<th>$j$</th>
<th>True Simulation</th>
<th>$h_{ij}$</th>
<th>True Simulation</th>
<th>$b_{ij}$</th>
<th>True Simulation</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x_1$</td>
<td>9.9999702E+00</td>
<td>10</td>
<td>9.9999711E+00</td>
<td>0.75</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_2$</td>
<td>9.9994099E+00</td>
<td>10</td>
<td>9.9994048E+00</td>
<td>6.3</td>
<td>9.9981438E+00</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_3$</td>
<td>9.9987145E+00</td>
<td>10</td>
<td>9.9987143E+00</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_4$</td>
<td>9.9997768E+00</td>
<td>10</td>
<td>9.9997738E+00</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_5$</td>
<td>9.9987314E+00</td>
<td>10</td>
<td>9.9987248E+00</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_6$</td>
<td>9.9988066E+00</td>
<td>10</td>
<td>9.9987936E+00</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_7$</td>
<td>1.00004395E+01</td>
<td>10</td>
<td>1.00004395E+01</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_8$</td>
<td>9.9997420E+00</td>
<td>10</td>
<td>9.9993444E+00</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_9$</td>
<td>1.0000009E+01</td>
<td>10</td>
<td>1.0000009E+01</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_{10}$</td>
<td>1.0001310E+01</td>
<td>10</td>
<td>1.0001310E+01</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_{11}$</td>
<td>1.0001786E+01</td>
<td>10</td>
<td>1.0001786E+01</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_{12}$</td>
<td>1.0002384E+01</td>
<td>10</td>
<td>1.0002384E+01</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_{13}$</td>
<td>1.0003161E+01</td>
<td>10</td>
<td>1.0003161E+01</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_{14}$</td>
<td>1.0004985E+01</td>
<td>10</td>
<td>1.0004985E+01</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_{15}$</td>
<td>1.0007613E+01</td>
<td>10</td>
<td>1.0007613E+01</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_{16}$</td>
<td>1.0009364E+01</td>
<td>10</td>
<td>1.0009364E+01</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_{17}$</td>
<td>1.0010822E+01</td>
<td>10</td>
<td>1.0010822E+01</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_{18}$</td>
<td>1.0012305E+01</td>
<td>10</td>
<td>1.0012305E+01</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_{19}$</td>
<td>1.0013892E+01</td>
<td>10</td>
<td>1.0013892E+01</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
<tr>
<td>$x_{20}$</td>
<td>1.0015484E+01</td>
<td>10</td>
<td>1.0015484E+01</td>
<td>0.1</td>
<td>-6.9933524E-01</td>
<td>0.9999842E-01</td>
<td></td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>( x_i )</th>
<th>( \beta_j )</th>
<th>( \gamma_{ij} )</th>
<th>( \beta_{ij} )</th>
<th>( h_{ij} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>True</td>
<td>Simulation</td>
<td>True</td>
<td>Simulation</td>
<td>True</td>
</tr>
<tr>
<td>9.9996143E-01</td>
<td>1</td>
<td>9.9996097E-01</td>
<td>1</td>
<td>9.9999670E-00</td>
</tr>
<tr>
<td>9.9998739E-01</td>
<td>1</td>
<td>9.9998702E-01</td>
<td>1</td>
<td>9.9999817E-00</td>
</tr>
<tr>
<td>9.9998978E-01</td>
<td>1</td>
<td>9.9998956E-01</td>
<td>1</td>
<td>9.9999935E-00</td>
</tr>
<tr>
<td>9.9998880E-01</td>
<td>1</td>
<td>9.9998890E-01</td>
<td>1</td>
<td>9.9999935E-00</td>
</tr>
<tr>
<td>9.9998556E-01</td>
<td>1</td>
<td>9.9998542E-01</td>
<td>1</td>
<td>9.9999935E-00</td>
</tr>
<tr>
<td>9.9999396E-01</td>
<td>1</td>
<td>9.9999385E-01</td>
<td>1</td>
<td>9.9999935E-00</td>
</tr>
<tr>
<td>9.9999880E-01</td>
<td>1</td>
<td>9.9999890E-01</td>
<td>1</td>
<td>9.9999935E-00</td>
</tr>
<tr>
<td>9.9999856E-01</td>
<td>1</td>
<td>9.9999890E-01</td>
<td>1</td>
<td>9.9999935E-00</td>
</tr>
<tr>
<td>9.9999873E-01</td>
<td>1</td>
<td>9.9999890E-01</td>
<td>1</td>
<td>9.9999935E-00</td>
</tr>
</tbody>
</table>
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 95.192244%. 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, \( J_1 \) and the normalized kinetic order \( J_0 \). 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 \( J_0 \) 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 \( J_{rec} = -\max \{ J_{II}, J_{wI}, J_{wII} \} \) where \( J_{II} = w_{II} J_{II} \) \( J_{wI} = w_I J_{wI} \) and \( J_{wII} = w_{II} J_{wII} \) where \( w_I = w_{II} = \mu \), \( \mu \) 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 crocodile individual:

\[
\begin{align*}
&\alpha_1, \beta_1, \gamma_1, \delta_4, \beta_5, \gamma_5, \delta_6, g_{11}, \ldots, g_{21}, g_{22}, \ldots, g_{33}, g_{34}, h_{11}, \ldots, h_{21}, h_{22}, \ldots, h_{32}
\end{align*}
\]

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 \( \theta \) to \( \theta' \), we define the average squared change of the constitue \( x_i \) as \( N(\theta) \) [56].

\[
N(\theta) = \frac{1}{2nk} \sum_{i=1}^{n} \sum_{t=1}^{n} \frac{1}{T_k} \int_0^{T_k} \frac{x_i'(0, t) - x_i'(\theta', t)}{x_{i,\text{max}}}^2 dt
\]  

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

\[
H_{ij} = \frac{\partial^2 N}{\partial \log \theta_i \partial \log \theta_j}
\]

Fig. 9 shows the respective normalized eigenvalue spectra of the Hessian matrices 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 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.

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 \( x_{4} \) on \( x_{3} \) (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.

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
