

# Gene Regulatory Network Model Identification Using Artificial Bee Colony and Swarm Intelligence

Zary Forghany\*, Mohsen Davarynejad<sup>†</sup> and B.Ewa Snaar-Jagalska\*

\*Gorlaeus Laboratory, Institute of Biology, Leiden University, The Netherlands  
Email: Z.Forghany@umail.leidenuniv.nl, B.E.Snaar-Jagalska@biology.leidenuniv.nl

<sup>†</sup>Systems Engineering Section, Faculty of Technology, Policy and Management,  
Delft University of Technology, The Netherlands  
Email: M.Davarynejad@tudelft.nl

**Abstract**—Gene association/interaction networks have complex structures that provide a better understanding of mechanisms at the molecular level that govern essential processes inside the cell. The interaction mechanisms are conventionally modeled by nonlinear dynamic systems of coupled differential equations (S-systems) adhering to the power-law formalism. Our implementation adopts an S-system that is rich enough in structure to capture the dynamics of the gene regulatory networks (GRN) of interest. A comparison of three widely used population-based techniques, namely evolutionary algorithms (EAs), local best particle swarm optimization (PSO) with random topology, and artificial bee colony (ABC) are performed in this study to rapidly identify a solution to inverse problem of GRN reconstruction for understanding the dynamics of the underlying system. A simple yet effective modification of the ABC algorithm, shortly ABC\* is proposed as well and tested on the GRN problem. Simulation results on two small-size and a medium size hypothetical gene regulatory networks confirms that the proposed ABC\* is superior to all other search schemes studied here.

## I. INTRODUCTION

A genome-wide interaction analysis would open new horizons for biologists, providing a unique ubiquitous structural component of the genetic architecture of human diseases. Since many diseases are the result of polygenic and pleiotropic effects controlled by multiple genes, genome-wide interaction analysis are preferable over single locus study.

The activation and inhibition of genes are governed by factors within a cellular environment and outside of the cell. This level of activation and inhibition of genes are integrated by gene regulatory networks (GRNs), forming an organizational level in the cell with complex dynamics [7].

Mathematical modeling of gene is becoming popular in the post-genome era [18], [21], providing a powerful tool not only for the better understanding of such complex systems but also for developing new hypotheses on underlying mechanisms. The availability of high-throughput technologies provide time course expression data; and GRN model built by reverse engineering, may explain the data [24]. Model parameter estimation is a challenging task and is normally formulated as an optimization problem [27]. Based on gene expression data over time, these optimization techniques enable genetic network architectures to be reconstructed. In this studies, S-systems, a set of non-linear differential equations of a special

form belonging to the power-law formalism are adopted as model.

S-system based GRN inference was formulated by Tomimaga et al. [28] as an optimization problem to minimize the difference between the model and the system. To optimize the network parameters and to capture the dynamics in gene expression data, they used standard evolutionary algorithms (EAs) to estimate the model parameters. Evolutionary computation is becoming a popular approach for solving S-system parameter optimization [23], [3], [17], mainly due to the multimodality and strong non-linear parameter-dependencies in the problem. Extensive simulations were reported in [26] where the performance of different genetic and evolutionary strategies are compared. [20] presents a hybrid genetic algorithm-particle swarm optimization method to infer appropriate network parameters. The results are compared to that of standard PSO.

To the best of our knowledge, this is the first attempt to evaluate the ABC algorithm for S-system parameter optimization for GRN inference by reverse engineering of three gene networks. We have also proposed a simple modification to the ABC algorithm. Moreover, when comparing heuristic optimization methods, a PSO with random topology is adopted here. Numerical experiments show that the proposed enhancement on standard ABC algorithm attain higher accuracy and computational efficiency compared to that of EAs, PSO with random topology and the standard ABC.

The remainder of this paper is organized as follows. A short introduction to GRNs and to S-systems is provided in Section II. The population based model for S-system parameter identification is presented in Section III with a short introduction to EA and PSO with random topology. ABC and ABC\* (a simple yet effective modification on standard ABC algorithm) are presented in Section III-C and III-D respectively. Experimental setup adopted to check the suitability of ABC and ABC\* for gene network estimation as well as a comparison with EA and PSO are presented in Section IV. The final section draws conclusions and considers implications for future research.

## II. GENE REGULATORY NETWORKS

GRNs in the cell are a complex dynamic network of interactions between the products of genes (mRNAs) and the proteins they produce, some of which in return act as regulators of the expression of other genes (or even their own gene) in the production of mRNA. While low cost methods to monitor gene expression through microarrays exist, we still know little about the complex interactions of these cellular components. Usually, sets of ordinary differential equations (ODEs) are used as mathematical models for these systems [29], however they suffer from many assumptions critical to the equations themselves. S-system approaches, on the other hand, use time-independent variables to model these processes. Assuming the concentration of  $N$  proteins, mRNAs, or small molecules at time  $t$  is given by  $y_1(t), y_2(t), \dots, y_i(t), \dots, y_N(t)$ , S-systems models the temporal evolution of the  $j$ th component at time  $t$  by power-law functions of the form (1).

$$\frac{dy_i(t)}{d(t)} = \alpha_i \left( \prod_{j=1}^N y_j^{g_{ij}(t)} \right) - \beta_i \left( \prod_{j=1}^N y_j^{h_{ij}(t)} \right), \quad (1)$$

where  $N$  represents the number of genes involved in the GRN. The first term represents all factors that promote the expression of component  $i$ ,  $y_i$ , whereas the second term represents all factors that inhibit its expression. In a biochemical engineering context, the non-negative parameters  $\alpha_i$ ,  $\beta_i$  are called rate constants, and real-valued exponents  $g_{ij}$  ( $G$  matrix,  $[G]$ ) and  $h_{ij}$  ( $H$  matrix,  $[H]$ ) are, respectively, referred to as kinetic order for synthesis and kinetic order for degradation.

$\{\alpha, \beta, [G], [H]\}$  are the parameters that define the S-system. The total number of parameters in the S-system is  $2N(N+1)$ , meaning the number of parameters increases quadratically and can quickly become very large. The parameter estimation is to determine model parameters so that the dynamic profiles fit the observation. Attaining appropriate model parameter values is a challenging task since there is no general solution guaranteed to solve the problem in nonlinear parameter estimation.

## III. POPULATION BASED S-SYSTEMS MODEL PARAMETER IDENTIFICATION

Let us define the search space as the following:

$$E = \bigotimes_{d=1}^D [\min_d, \max_d], \quad (2)$$

To guide the population in the search space, some measure of discrimination is needed. The most commonly used quality assessment criterion is the mean quadratic discrepancy between the observed expression pattern  $y_i^t$  and the model output  $\hat{y}_i^t$  [25].

$$f = \sum_{i=1}^N \sum_{t=1}^T \left( \frac{\hat{y}_i^t - y_i^t}{y_i^t} \right)^2, \quad (3)$$

where  $N$  represents the number of genes of the network of interest and  $T$  is the number of time points.

## A. Evolutionary algorithms

Evolutionary algorithms (EAs) are stochastic population-based search method inspired from the evolutions scheme of natural selection [12]. EA includes methods of genetic algorithms (GAs) [2], [13], evolution strategies (ESs) and some other derivatives including genetic programming [19]. Although they are different in some details, they are broadly similar. They all maintain a population of solutions of which are randomly generated in initialization phase. To guide the evolution by discriminating good solutions from bad ones, a fitness is assigned to every individual. Selection and genetic operators are iteratively evolving the population [9]. Recombination and mutation are often used genetic operators. The Simulated Binary Crossover (SBX) [11], a real-parameter recombination operator that emulates the single-point crossover, is a widely used operator to combine pieces of parental solutions to form offsprings. Mutation, that restores genetic diversity lost, is an operator whereby random changes are introduced into a solution, thereby reducing the risk of the search algorithm becoming trapped in local. Throughout this study SBX crossover and polynomial mutation are adopted.

## B. Particle swarm optimization with random topology

In classical particle swarm optimization (PSO), every particle is a potential solution moving in a  $D$ -dimensional search space. A collection of particles are known as *swarm*. Each particle  $i$  has a position,  $x_i \in \mathbb{R}^D$ , a velocity,  $v_i \in \mathbb{R}^D$  and the best position found so far,  $p_i \in \mathbb{R}^D$ . In contrast to *global best PSO*, in *local best PSO* each particle informs only  $k$  other randomly chosen particles, known as *information links* or *neighbors*. Neighbors are, often, defined once at the beginning of the search process [1]. Derived from the fact that no topology is proven to be globally the best [6], we used a randomly chosen neighbors both at the beginning of search as well as once the previous global best position found by swarm,  $g_i \in \mathbb{R}^D$  has not improved. Random topology [16], a popular *local best PSO* is a particular case of a stochastic star [22].

The movement equations are given by expressions (4) and (5).

$$v_i = wv_i + x'_i - x_i, \quad (4)$$

$$x_i = x_i + v_i, \quad (5)$$

where  $w$  is a predefined constant representing the confidence of particle on its own movements.  $x'_i \in \mathbb{R}^D$  denoted a point drawn uniformly at random from a hypersphere  $\mathcal{H}$  with center  $G_i \in \mathbb{R}^D$  and radius  $\|G_i - x_i\|$ .  $G_i$ , the *center of gravity* [5], also known as *queen* [4] is defined as:

$$G_i = x_i + c \frac{p_i + l_i - 2x_i}{3}, \quad (6)$$

$l_i \in \mathbb{R}^D$  is the best previous position found in the neighbor particles and  $c \in \mathcal{U}(0, c_{\max})$  represents the confidence of particle on its best performance and that of its best neighbor respectively.

### C. Artificial Bee Colony (ABC)

ABC algorithm [14] is one of the most recently introduced population-based algorithms inspired from the intelligent foraging behavior of a honeybee swarm. Honeybees have many well-defined behaviors. Information exchange among bees, contributing in system's collective knowledge formation, happens in 'dance language' reflecting the quality of the food source visited by employed forager, its distance from the hive and the direction. Food source exploitation is a process carried out by both employed and onlooker bees. Onlookers try to find trace information on food source provided by employed bees. The probability an onlooker chooses a food source is proportional to the profitability of the food source. The exploration process is controlled by scouts, moving randomly in the solution space seeking for new sources of food.

In the ABC algorithm, the food source position,  $x_i \in \mathbb{R}^D$ ,  $i = 1, 2, \dots, SN$  represents a  $D$ -dimensional candidate solution. Food source profitability is the fitness,  $f_i$ , of candidate solution. An employed bee explores its surroundings according to (7).

$$v_{ij} = x_{ij} + \phi_{ij} (x_{ij} - x_{kj}) \quad (7)$$

where  $k \in \{1, 2, \dots, SN\}$  and  $k \neq i$ .  $\phi_{ij}$  is a random number generated from  $\mathcal{U}(-1, 1)$  and  $j \in \{1, 2, \dots, D\}$  respectively.  $v_{ij}$ , a new food position, is perturbation of  $x_{ij}$  in dimension  $j \in \{1, 2, \dots, D\}$ . If the fitness of the new position is superior to the old one, the new position,  $v_{ij}$  is memorized by the bee. Otherwise the previous position is kept unchanged.

The information-sharing process begins when all employed bees complete the search process. The probability of a food source to be chosen by an onlooker is calculated by expression (8).

$$p_i = \frac{\text{fit}_i}{\sum_{n=1}^{SN} \text{fit}_n} \quad (8)$$

### D. Modified Artificial Bee Colony (ABC\*)

Our modification of the ABC algorithm is as following. The employed bees explore their neighbor according to (7), but the onlookers exploration takes place according to (9)

$$v_{i\bar{j}} = x_{i\bar{j}} + \phi_{i\bar{j}} (x_{i\bar{j}} - x_{k\bar{j}}) \quad (9)$$

where again  $k \in \{1, 2, \dots, SN\}$  and  $k \neq i$ .  $\phi_{i\bar{j}}$  is a random number generated from  $\mathcal{U}(-1, 1)$  for all  $\bar{j}$ . Without loss of generality that  $j_l \leq j_u$ ,  $\bar{j}$  is an interval with  $j_l$  and  $j_u$  corresponding to its end points where  $j_l, j_u \in \{1, 2, \dots, D\}$ . In this study the performance of both ABC and ABC\* are compared with a set of well-established inference methods, evolutionary algorithms and swarm intelligence, effective and efficient population-based optimization method capable of handling nonlinear and multimodal objective functions.

## IV. EXPERIMENTAL SETUP AND NUMERICAL RESULTS

As the evolutionary optimization techniques are stochastic in nature, 30 independent runs are performed for each experimental setup with average and std of results being reported along with the results of Mann-Whitney U-test for statistical test of significance.

### A. Settings

In all experiments in this section, the common parameters used in each algorithm such as population size and total number of fitness evaluation were chosen to be the same. Unless otherwise mentioned, the population size was set at 50. The maximum number of fitness evaluation was set at 200,000 for the first two networks (NET1 and NET2) and 500,000 for the third network (NET3).

1) *EA settings*: A real-coded evolutionary algorithm, for the following GRN model identification was run using a crossover rate of 1.0 (using SBX), a binary tournament for mating selection with replacement and a polynomial mutation (PM) as described by Deb [11]. Mutation rate was set at 0.1 and the distribution indices  $\eta_c = 20$  and  $\eta_m = 20$  are applied.

2) *PSO settings*: [5] notes that the appropriate swarm size  $s$  is usually smaller than EAs population size and suggests a swarm size of  $s = 10 + 2 * \sqrt{D}$ . In an attempt to achieve a fair comparison of results, for NET1 and NET2, a swarm size of 20 running for 10000 iterations (PSO2) in addition to a swarm size of 50 running for 4000 iterations (PSO1) was adopted. In the case of NET3, a swarm size of 40 was adopted.

The PSO parameters across the experiments are  $w = \frac{1}{2 \ln(2)}$ ,  $c_{\max} = \frac{1}{2} + \ln(2)$ , [6].  $k$ , the number of neighbors was set at 3.

3) *ABC and ABC\* settings*: Apart from the common parameters, namely the number of population and maximum number of fitness evaluation, ABC and ABC\* use only one control parameter, which that is called *limit*. If a food source cannot be further improved within *limit* 'trial numbers', it is assumed to be abandoned and the corresponding employed bee becomes a scout. This means that these solutions will not be exploited anymore. In [15], the *limit* value was determined using the dimension of the problem and the colony size according to (10).

$$\text{limit} = (SN * D) \quad (10)$$

where  $D$  is the dimension of the problem and  $SN$  is the population size or the number of food sources.

To assess the performance of the methodologies, two small-size hypothetical gene regulatory networks, NET1 and NET2, each consist of a network of two genes generated by the parameters given in Table I and II as well as a medium-size GRN, NET3, with parameters given in Table III are adopted.

The gene expression levels of the networks are plotted in Figures 1, 2 and 3 each consist of 50 time course of expression

$i$	$\alpha_i$	$\beta_i$	$g_{i1}$	$g_{i2}$	$h_{i1}$	$g_{i2}$
1	3	3	0	2.5	-1	0
2	3	3	-2.5	0	0	2

TABLE I: S-System Parameters for Network Model NET1 adopted for model validation [28].

$i$	$\alpha_i$	$\beta_i$	$g_{i1}$	$g_{i2}$	$h_{i1}$	$g_{i2}$
1	3	3	0	-2.5	.1	0
2	3	3	2.5	0	0	.1

TABLE II: S-System Parameters for Network Model NET2 adopted for model validation.

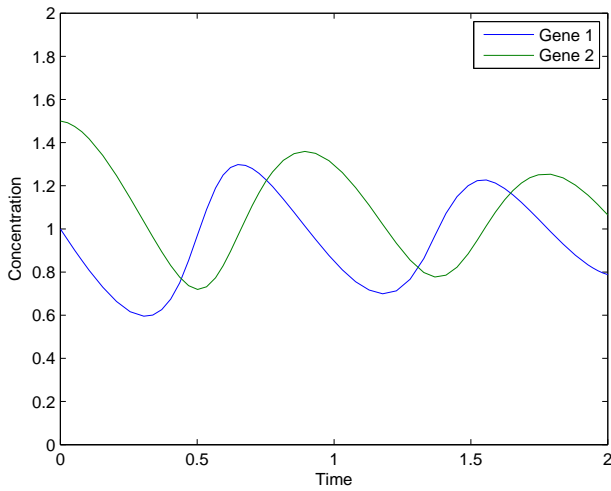


Fig. 1: Target time dynamics of first gene network, NET1.

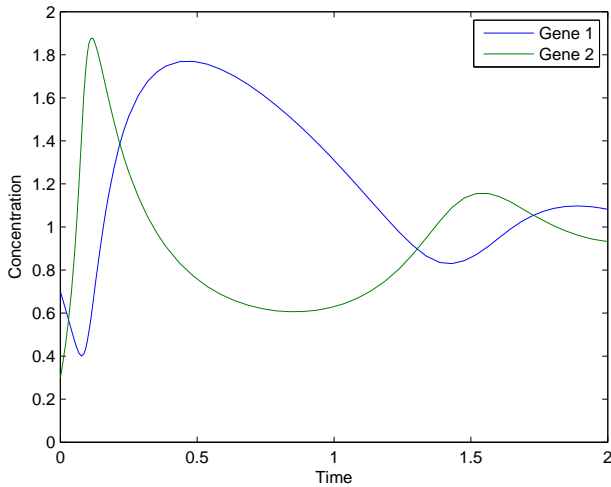


Fig. 2: Target time dynamics of second gene network, NET2.

$i$	$\alpha_i$	$g_{i1}$									
1	5	0	0	0	1	0	-2	0	0	0	0
2	10	0	0	1	0	0	0	0	-1	0	0
3	8	-1	0	0	-1	0	0	0	0	0	0
4	10	0	0	0	0	2	0	0	0	1	0
5	10	0	2	0	0	0	-1	0	0	0	0
6	5	0	0	0	0	0	0	0	0	2	-2
7	10	0	0	0	0	0	1	0	0	0	-1
8	5	1	-2	0	0	0	0	1	0	0	0
9	10	0	0	1	0	0	0	0	-2	0	0
10	8	2	0	0	0	0	0	-1	0	0	0

$i$	$\beta_i$	$h_{i1}$										
1	10	2	0	0	0	0	0	0	0	0	0	0
2	10	0	2	0	0	0	0	0	0	0	0	0
3	10	0	0	2	0	0	0	0	0	0	0	0
4	10	0	0	0	2	0	0	0	0	0	0	0
5	10	0	0	0	0	2	0	0	0	0	0	0
6	10	0	0	0	0	0	2	0	0	0	0	0
7	10	0	0	0	0	0	0	2	0	0	0	0
8	10	0	0	0	0	0	0	0	2	0	0	0
9	10	0	0	0	0	0	0	0	0	2	0	0
10	10	0	0	0	0	0	0	0	0	0	0	2

TABLE III: S-System Parameters for Network Model NET3 adopted for model validation [20].

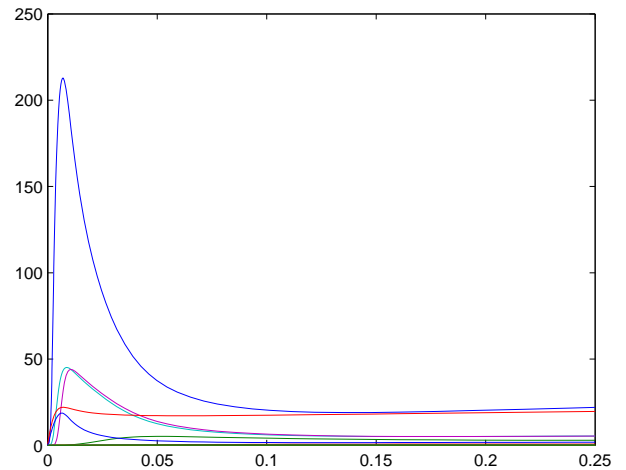


Fig. 3: Target time dynamics of third gene network, NET3.

level per gene. The search space for  $\alpha_i$  and  $\beta_i$  is limited to  $[20, 0.0]$  and for  $g_{ij}$  and  $h_{ij}$  to  $[-4.0, 4.0]$ .

The fitness transitions for different methodologies for NET1-3 are plotted in a logarithmic scale in Figures 4, 5 and Figure 6 respectively. Though all algorithms discussed here start with randomly generated population of solutions, PSO2 starts with worse fitness value due to the smaller population size. The Figures are averaged over 30 independent runs.

In NET1 and NET2, PSO2 starts with a sharp fitness decrease in the beginning, but started to become almost stagnate after approximately 20,000 fitness evaluations in case of NET1 and 50,000 fitness evaluations in case of NET2 on average. The EA in both cases had a slow progression compared to all other studied schemes whereas PSO1 and ABC\* had much

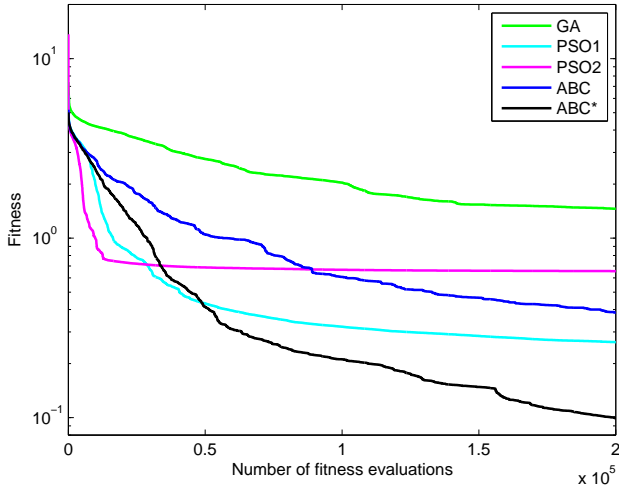


Fig. 4: Performance comparison of the EA, PSO1, PSO2, ABC and ABC\* on NET1.

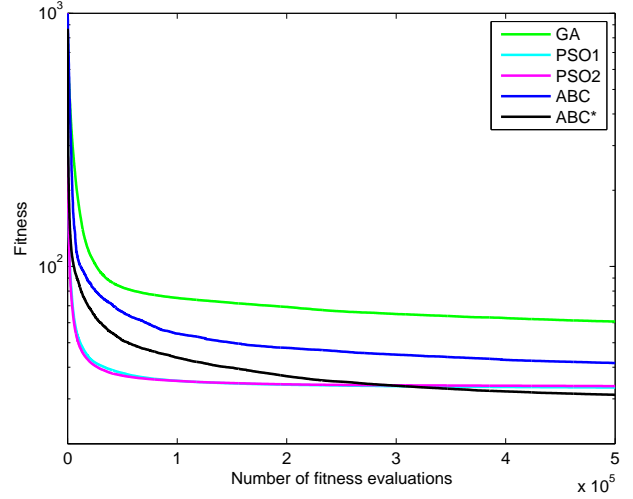


Fig. 6: Performance comparison of the EA, PSO1, PSO2, ABC and ABC\* on NET3.

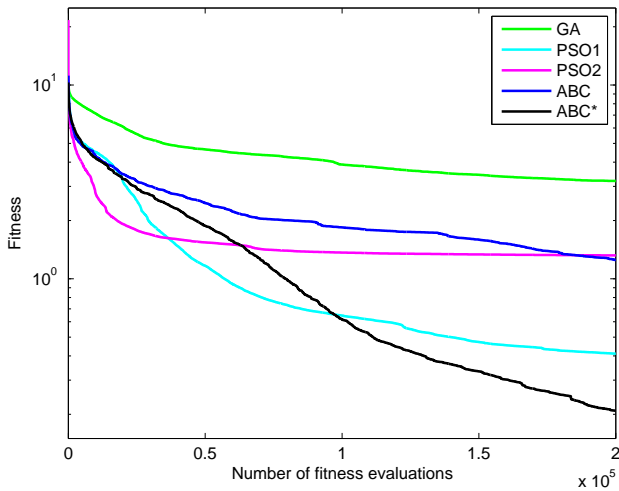


Fig. 5: Performance comparison of the EA, PSO1, PSO2, ABC and ABC\* on NET2.

better progression compared to the rest. ABC presents the slowest convergence rate in the beginning after EA, but unlike EA, shows steady improvement in fitness value. In the case of NET3, PSO1 and PSO2 are presenting almost a same fitness evolution.

According to the Mann-Whitney U-test (Table IV, V and VI), the results of proposed ABC\* approach, are better than that of ABC with a significance level of 5%. For the case of NET2 and NET3, ABC\* is significantly superior to all other schemes treated here, whereas in case of NET1, while ABC\* is not significantly superior to that of PSO1, it has a much lower mean error and lower variance.

NET1	Simulation results		
	Mean	std.	<i>p</i> -Value
EA	1.46	1.37	$1.1737 \times 10^{-09}$
SPSO1	0.26	0.29	0.2772
SPSO2	0.65	0.75	$9.06 \times 10^{-8}$
ABC	0.38	0.18	$1.0702 \times 10^{-9}$
ABC*	0.09	0.08	-

TABLE IV: A Mann-Whitney U test of the fitness of last generation for NET1 (30 runs).

NET2	Simulation results		
	Mean	std.	<i>p</i> -Value
EA	3.20	2.51	$5.0922 \times 10^{-08}$
SPSO1	0.41	0.44	0.0461
SPSO2	1.32	1.59	0.0067
ABC	1.25	0.53	$3.1589 \times 10^{-10}$
ABC*	0.21	0.23	-

TABLE V: A Mann-Whitney U test of the fitness of last generation for NET2 (30 runs).

NET2	Simulation results		
	Mean	std.	<i>p</i> -Value
EA	60.6	22.5	$3.0199 \times 10^{-11}$
SPSO1	33.2	0.9172	$5.9242 \times 10^{-09}$
SPSO2	33.7	1.21	$2.6547 \times 10^{-09}$
ABC	41.5	2.65	$3.0199 \times 10^{-11}$
ABC*	31.1	1.11	-

TABLE VI: A Mann-Whitney U test of the fitness of last generation for NET3 (30 runs).

## V. CONCLUSIONS AND FUTURE WORK

In this paper, the standard artificial bee colony (ABC) and its variant proposed in this paper, ABC\*, are employed for estimating genetic networks using S-system formalism. The performance of the ABC\* method, which enhance the global searching capability of ABC, was verified using two small-scale and one medium-scale hypothetical networks and the experiments shows that the proposed method is capable to identify model parameters. From the simulation results it is concluded that the proposed ABC\* is superior to all other search schemes studied here, namely evolutionary algorithms and *local best* particle swarm optimization with random topology, in terms of smaller mean of error and smaller variance.

Much of the computational complexity involved in the use of population-based optimization tools is due to the fitness function evaluation that may either be very difficult to define or be computationally very expensive. One of the popular solutions to this challenge is to replace the expensive fitness evaluation step with an approximate model. As part of our future work, we are interested in studying the performance of some of this type of solutions, like those proposed in [8], [10] on reverse engineering gene regulatory networks.

Finally, we wish to apply the ABC and ABC\* algorithms to actual biological gene network to further verify its effectiveness more comprehensively.

## ACKNOWLEDGMENT

The authors would like to acknowledge Dr. Gary Fogel from Natural Selection Inc. for his constructive comments and suggestions on several earlier versions of this work.

## REFERENCES

- [1] D. Bratton and J. Kennedy. Defining a standard for particle swarm optimization. In *Swarm Intelligence Symposium*, pages 120–127, 2007.
- [2] H.J. Bremermann. Optimization through evolution and recombination. *Self-organizing systems*, pages 93–106, 1962.
- [3] A.R. Chowdhury and M. Chetty. An improved method to infer gene regulatory network using s-system. In *Congress of Evolutionary Computation (CEC'11)*, pages 1012–1019, 2011.
- [4] M. Clerc. The swarm and the queen: towards a deterministic and adaptive particle swarm optimization. In *Congress on Evolutionary Computation (CEC'99)*, volume 3, pages 1951–1957.
- [5] M. Clerc. *Particle swarm optimization*. Wiley-ISTE, 2006.
- [6] M. Clerc. Stagnation analysis in particle swarm optimisation or what happens when nothing happens. 2006.
- [7] A. Crombach and P. Hogeweg. Evolution of evolvability in gene regulatory networks. *PLoS computational biology*, 4(7):e1000112, 2008.
- [8] M. Davarynejad, C.W. Ahn, J.L.M. Vrancken, J. van den Berg, and C.A. Coello Coello. Evolutionary hidden information detection by granulation-based fitness approximation. *Applied Soft Computing*, 10(3):719–729, 2010.
- [9] M. Davarynejad, M.-R. Akbarzadeh-T, and N. Pariz. A novel general framework for evolutionary optimization: Adaptive fuzzy fitness granulation. In *Congress on Evolutionary Computation (CEC'07)*, pages 951–956, 2007.
- [10] M. Davarynejad, J. Vrancken, J. van den Berg, and C.A. Coello Coello. A Fitness Granulation Approach for Large-Scale Structural Design Optimization. In Raymond Chiong, Thomas Weise, and Zbigniew Michalewicz, editors, *Variants of Evolutionary Algorithms for Real-World Applications*, pages 245–280. Springer-Verlag, 2012.
- [11] K. Deb and R.B. Agrawal. Simulated binary crossover for continuous search space. *Complex systems*, 9(2):115–148, 1995.
- [12] D.E. Goldberg. *The Design of Innovation: Lessons from and for Competent Genetic Algorithms*. Kluwer Academic Publishers, 2002.
- [13] J.H. Holland. Adaptation in natural and artificial systems, 1975.
- [14] D. Karaboga. An idea based on honey bee swarm for numerical optimization. *Techn. Rep. TR06, Erciyes Univ. Press, Erciyes*, 2005.
- [15] D. Karaboga and B. Akay. A comparative study of artificial bee colony algorithm. *Applied Mathematics and Computation*, 214(1):108–132, 2009.
- [16] J. Kennedy and R. Mendes. Population structure and particle swarm performance. In *Congress on Evolutionary Computation (CEC'02)*, volume 2, pages 1671–1676, 2002.
- [17] S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, and M. Tomita. Dynamic modeling of genetic networks using genetic algorithm and s-system. *Bioinformatics*, 19(5):643–650, 2003.
- [18] S. Kimura, K. Ide, A. Kashiwara, M. Kano, M. Hatakeyama, R. Masui, N. Nakagawa, S. Yokoyama, S. Kuramitsu, and A. Konagaya. Inference of s-system models of genetic networks using a cooperative coevolutionary algorithm. *Bioinformatics*, 21(7):1154, 2005.
- [19] J.R. Koza. Genetic programming: On the programming of computers by means of natural selection. 1992.
- [20] W.P. Lee and Y.T. Hsiao. An adaptive gapso approach with gene clustering to infer s-system models of gene regulatory networks. *The Computer Journal*, 54(9):1449–1464, 2011.
- [21] W.P. Lee and W.S. Tzou. Computational methods for discovering gene networks from expression data. *Briefings in bioinformatics*, 10(4):408–423, 2009.
- [22] V. Miranda, H. Keko, and A.J. Duque. Stochastic star communication topology in evolutionary particle swarms (epso). *International Journal of Computational Intelligent Research*, 4(2):105116, 2008.
- [23] T. Nakayama, S. Seno, Y. Takenaka, and H. Matsuda. Inference of s-system models of gene regulatory networks using immune algorithm. *Journal of Bioinformatics and Computational Biology*, 9:75–86, 2011.
- [24] S. Navlakha and Z. Bar-Joseph. Algorithms in nature: the convergence of systems biology and computational thinking. *Molecular Systems Biology*, 7(1), 2011.
- [25] N. Noman and H. Iba. Inference of gene regulatory networks using s-system and differential evolution. In *Genetic and Evolutionary Computation Conference, Washington, DC*, pages 439–446, 2005.
- [26] A. Sirbu, H. Ruskin, and M. Crane. Comparison of evolutionary algorithms in gene regulatory network model inference. *BMC Bioinformatics*, 11(1):59, 2010.
- [27] J. Sun, J.M. Garibaldi, and C. Hodgman. Parameter estimation using metaheuristics in systems biology: A comprehensive review. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 9(1):185–202, 2012.
- [28] D. Tominaga, M. Okamoto, Y. Maki, S. Watanabe, and Y. Eguchi. Non-linear numerical optimization technique based on a genetic algorithm for inverse problems: Towards the inference of genetic networks. In *Computer Science and Biology (Proc. German Conf. on Bioinformatics)*, pages 127–140, 1999.
- [29] K.Y. Tsai and F.S. Wang. Evolutionary optimization with data collocation for reverse engineering of biological networks. *Bioinformatics*, 21(7):1180, 2005.