Investigation of coevolutionary approach in gene regulatory network inference

Danko Komlen*, Domagoj Jakobović†
University of Zagreb, Faculty of Electrical Engineering and Computing
Unska 3, 10000 Zagreb, Republic of Croatia
* Email: dkomlen@gmail.com
† Email: domagoj.jakobovic@fer.hr

***Abstract*—Inference of gene regulatory networks is currently an active field of research in system biology. Evolutionary computation algorithms are lately applied for finding the optimal parameters of models. This paper presents a comparison of four evolutionary algorithms (DE, GA, PSO and the hybrid Hooke-Jeeves GA) used with a linear time-variant gene network model. The paper also investigates the efficiency of cooperative coevolution approach to cope with the increased complexity of networks with large number of genes. Experiments were performed on two artificially generated and one real microarray data set. The results are twofold: the efficiency comparison may serve as a guideline for future research, and the application of coevolution proved to be successful for most algorithms.**

I. Introduction

Each cell in a living organism has the same genome[1]. Complex mechanisms at molecular level that interpret that information are responsible for their differentiation and functionality.

A key role in such mechanisms are transcription factors that bind themselves to parts of DNA [1]. They can activate or suppress activity of a particular gene. Complex systems emerge because transcription factors are products of genes and can be regulated by other transcription factors (Fig. 1). Feedback loops are also possible.

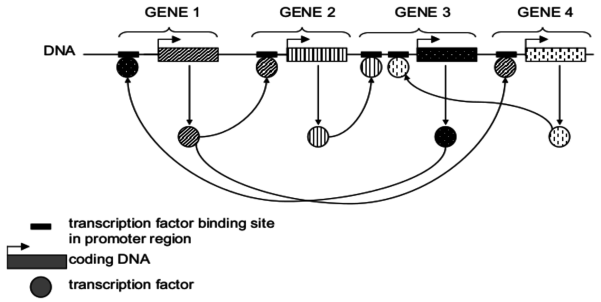

Figure 1. A simplified transcription factor network [2]

Research of these complex systems requires a way to measure the activity of individual genes within the cell. Such a procedure is called gene expression profiling and there are several different technologies available. One of the most common is DNA microarray technology [3], where levels of mRNA[2] for multiple genes are monitored in parallel.

Gene regulatory network (GRN) is an abstraction that allows understanding of these complex dynamic systems and provides an explanation for the expression of genes that are observed. Constructing a gene regulatory network involves finding a solution to a complex set of conditions, which is usually approximated by an appropriate optimization model. Descriptions of models for inferring GRN networks that are used in the paper are given in Section II.

Evolutionary computation (EC) algorithms have been successfully applied for finding the optimal or near-optimal parameters of models. This paper presents a comparison of various evolutionary algorithms used to infer the parameters of a GRN model. The comparison is based on two artificially generated data sets that were introduced in paper [4] and a real microarray data set that was also used in paper [5].

The efficiency of evolutionary algorithms may be further increased with the concept of coevolution, which is applied to all the algorithms in this work. This approach could be particularly useful for networks with a large number of genes, and consequently a large number of parameters. The results reflect the efficiency of each algorithm and the effect of applied coevolution, which may provide useful guidelines for future research.

The remainder of this paper is organized as follows: Section II outlines the GRN models, and Section III briefly covers the coevolutionary approach. Section IV details the optimization of GRN model parameters and Section V presents the results. Conclusions and future work are given in Section VI.

II. Gene Regulatory Network Models

A. S-systems

GRN networks describe biomolecular interactions that are non-linear and can be expressed by the general system of differential equations:

$$\frac{dx_i(t)}{dt} = f_i[x_1(t), ..., x_N(t)], \tag{1}$$

for i = 1,...,N, where N is number of genes, x_i gene expression level and f_i a function that describes the influence

[1] inherited information encoded in DNA

[2] messenger RNA, conveys genetic information from DNA to the ribosome

of all genes on gene i. For example, if a gene j activates gene i, then f_i increases with x_j and the other way around if a gene j inhibits gene i.

Determination of functions f_i that would define a successful model is an ill-placed problem. For this reason, we use different approximations of these functions.

S-systems are a special type of systems of differential equations where the function f_i is approximated by the following expression [6]:

$$\frac{dx_i(t)}{dt} = \alpha_i \prod_{j=1}^{N} {x_j}^{g_{i,j}} - \beta_i \prod_{j=1}^{N} {x_j}^{h_{i,j}} \tag{2}$$

Such a model is defined by a total of $2N^2+2N$ parameters.

B. Linear time-variant model

The paper [7] presents a linear time-varying model (LTV) where the following approximation is used for f_i:

$$\frac{dx_i(t)}{dt} = \sum_{j=1}^{N} W_{i,j}(t)x_j(t), \tag{3}$$

where $W_{i,j}(t)$ is a matrix of gene interactions (the control matrix). Its values depend on the current time t, which allows describing the nonlinearities in the system. Matrix values are calculated as the sum of the first two members of the Fourier series:

$$W_{i,j}(t) = \alpha_{i,j} sin(\omega_i t + \phi_{i,j}) + \beta_{i,j}. \tag{4}$$

The coefficient $W_{i,j}$ determines the strength of the influence of gene j on the regulation of gene i. A positive value indicates that the gene j activates gene i, the negative that it inhibits the gene, and zero indicates that the gene j does not affect the transcription of the gene i.

This model was also used in [8] with a slightly different approach. The authors used a discrete form of the model so that the values of genes expressions (x_i) were calculated directly by applying the sigmoid function to regulation input:

$$x_i(t+1) = \frac{1}{1+e^{(-Z_i(t))}}. \tag{5}$$

The regulation input $Z_i(t)$ denotes the influence of all genes on the gene i, and is defined as:

$$Z_i(t) = \sum_{j=1}^{N} W_{i,j}(t)x_j(t). \tag{6}$$

III. Cooperative coevolutionary algorithms

The term coevolution is most commonly used in context of biology and scientific fields devoted to the study of complex ecosystems. It is a biological change of an object that is caused by changing of another object that is interacting with it. Coevolutionary algorithms (CA) apply the concept of coevolution on metaheuristic optimization methods. Algorithms are divided into cooperative CA and competitive CA, each applicable to a specific kind of problems.

Basic classification of coevolutionary algorithms according to [9] is:

- one population competitive coevolution (1PC),
- two population competitive coevolution (2PC),
- N-population cooperative coevolution (NPC).

NPC coevolution, applied in this work, is used for problems with large solution space and where it is possible to make a decomposition into smaller subproblems. Each of the N subproblems is solved by a separate population. The fitness of the individual is calculated by the success of the entire solution, which includes itself and the best individuals from each of the other N-1 populations.

Coevolutionary algorithms can be realized sequentially and in parallel. For algorithms that operate over each population, various optimization algorithms (metaheuristics) could be used.

IV. GRN inference problem

Each GRN network model contains a set of parameters that define it. After selecting the type of model, the next step is to determine the values of parameters. This is a continuous optimization problem where the objective is to minimize the error between the data obtained by model simulation and experimental data (MAD measurements).

A. Problem description

For the modeling of GRN network, a linear time-varying model was used described in the [8], where the gene expressions were calculated by expression 5. Parameter set for the network of N genes consists of the following $3N^2 + N$ parameters:

$$\{\alpha_{i,j}, \beta_{i,j}, \phi_{i,j}, \omega_i | i,j \in \{1,...,N\}\}. \tag{7}$$

The fitness function is defined as

$$f = -\sum_{k=1}^{M}\sum_{t=1}^{T_k}\sum_{i=1}^{N}\left(\frac{X_{k,i,mod}(t) - X_{k,i,measured}(t)}{X_{k,i,measured}(t)}\right)^2, \tag{8}$$

where M is the number of data sets, t_k the number of measurements in the k-th set, and $X_{k,i,mod}(t)$ and $X_{k,i,measured}(t)$ are expression levels of the gene in time t as a result from the model and experimental results, respectively.

Each individual in EC algorithm contains a specific set of parameter values. Also, for each parameter type boundary values are predefined.

B. Application of coevolutionary algorithm

The number of parameters of the linear time-varying model grows quadratically depending on the number of genes in the data set. For this reason, evolutionary algorithms are searching a large solution space and the execution time increases. This kind of problem is appropriate for cooperative coevolution with the condition that division of the problem into smaller subproblems can be defined. Two different ways of divisions used will be described next.

The first problem division approach is the use of N populations, where individual from population i contains $3N + 1$ parameters: $\{_{i,j}, \beta_{i,j}, \phi_{i,j}, \omega_i | j \in \{1, ..., N\}\}$. Thus, each population optimizes parameters for a particular gene. During the evaluation, the best individuals from other populations create the total solution with all the parameters that are evaluated over the given data set (Fig. 2).

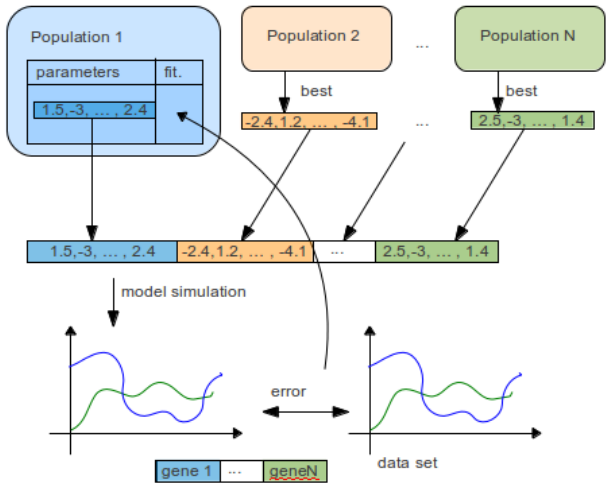

Figure 2. The first approach for problem division in NPC coevolution

The second approach, used in [10], shares model parameters in the same way by use of N populations. However, when evaluating the parameters for a particular gene, the following modified expression for the regulation input is used:

$$Z_i(t) = \sum_{j=1}^{N} W_{i,j}(t) Y_j(t)$$
$$Y_j(t) = \begin{cases} x_j(t), & j = i \\ \hat{x}_j(t), & \text{otherwise} \end{cases} \quad (9)$$

where $\hat{x}_j(t)$ is the expression of j-th gene at time t obtained by simulation of the model whose parameters are taken from the best individuals from the previous generation. So, at the end of each iteration the model is simulated using the best individuals and thus generates $\hat{x}_i(t)$ for each gene, which is then used in calculating the regulation inputs for the next generation (Fig. 3).

C. Test sets

A test set refers to a set of one or more data sets, which are used for testing the algorithms. Their overview is given in table I.

Data sets Tominaga2 and Tominaga5 are artificially obtained using S-systems that were presented in the paper [4] and are often used for measurement of algorithms performance. Data set Tominaga2 is shown in figure 4. The x axis indicates time; the time *intervals* between the two measurements can vary, but for LTV model only their ordinal number (T) is relevant.

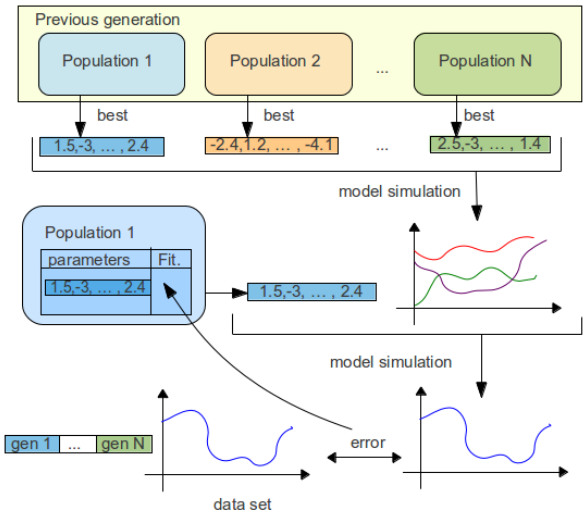

Figure 3. The second approach for problem division in NPC coevolution

Table I
TEST SETS OVERVIEW

Name	Gene count (N)	Measurement count (T)	Data sets count (K)	Data sets type
IS-param	5	20	1	Tominaga5
IS-t2	2	11	1	Tominaga2
IS-t5	5	11	10	Tominaga5
IS-yeast6	6	18	1	Spellman d.s.

The y axis indicates the gene expression levels and contains values between 0 and 1.

Test set IS-yeast6 contains experimental measurements of two cell cycles of the *Saccharomyces cerevisiae* organism, and the data was taken from Spellman data set [3].

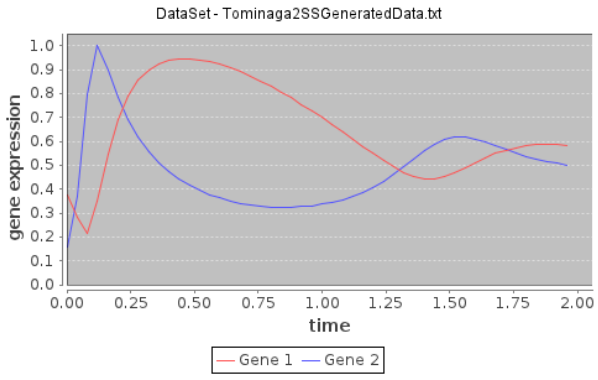

Figure 4. Tominaga2 data set

D. Overview of related work

The problem of modeling GRN network is one of the actual problems in systems biology field of research. Application

[3]available in the Kegg database (http://www.genome.jp/kegg/)

of evolutionary computation to optimize the parameters of the model is only one possible approach. Table II gives the representation of used algorithms, models, fitness functions and data sets from several papers related to application of evolutionary algorithms for GRN model inference.

The paper [5] gives comparison of the different evolutionary algorithms in modeling GRN network with an S-system. Seven algorithms were compared: GA, MOGA (multiobjective GA), GA + ES, GA + ANN (GA with artificial neural networks), PEACE1 [6] GLSDC [11] and DE. GA + ANN and DE proved to be the best on real world GRN networks. Tests were carried out on artificially generated networks and real DNA microarray data (Spellman data set).

Table II
OVERVIEW OF THE RELATED WORK

Paper title	**Evolutionary algorithm**	**Model type**	**Data sets (size)**
[8]	SA-DE	LTV	Generated (5), E. Coli (6)
[10]	coev-GLSDC	SS	Generated (5,30), Theromophilus (25)
[5]	GA, DE, GA+ANN, MOGA, PACE1, GLSDC, GA+ES	SS	Generated (10-50), Spellman d.s.
[12]	GA-simplex	diff. equ.	Rice (1)

The paper [8] used a linear time-varying model with differential evolution algorithm and self-tuning parameters. Tests were performed on Tominaga5, E.coli SOS and cAMP data sets with additional 5% and 10% noise. The algorithm has achieved satisfactory results after a relatively short running time.

In [10] a cooperative coevolution was used in combination with GLSDC algorithm on Tominaga5, S-system with 30 genes with 10% noise and *Thermus thermophilus* HB8 MAD data sets. The approach was successful in overcoming the problem of dimensionality with a larger number of genes. The results obtained for a set Tominaga5 are roughly $2 \cdot 10^{-3}$, and approach proved more successful than the usual problem decompositions.

V. ALGORITHM EVALUATION

The first series of experiments served to determine the efficiency of each of the applied algorithms: genetic algorithm (GA), differential evolution (DE), particle swarm optimization (PSO) and a hybrid algorithm based on GA and Hooke-Jeeves local search (HIB). For every evaluated algorithm we need to specify:

1) algorithm parameters,
2) model parameter intervals,
3) test set,
4) stop condition.

Algorithm parameters used for evaluation were determined with param-IS test set, while the other sets were used for comparison of performance. Stopping condition and the model parameter intervals have been specified separately for each of the tests.

A. Algorithm parameters search

Each of the four tested algorithms has its own set of parameters that determine its behavior for a given test set. Optimal parameters were determined over the IS-param test set with best solution stagnation and a time limit as a stopping condition, and 10 repetitions for each parameter value.

The algorithm would stop if the fitness of the best individual in 100 iterations has not increased by at least 0.01 or if the time limit of 3 minutes has been reached. The value of a parameter that would give the best average value of fitness of best individuals from each of the trials, would then be taken as optimal. For testing, independence of the parameters has been assumed and the parameters were determined one after another. At each test, the optimal values of the previously determined parameters were used. Initial and final optimal parameter values are given in tables (III - VI). Parameters are determined from left to right as they appear in the tables.

For population size (n) used values were 50, 100, 150, 200 and 250. Crossover probability (p_c) in the genetic algorithm and hybrid Hooke-Jeeves GA assumed values 0.5, 0.6, 0.7, 0.8, 0.9, and mutation probability (p_m) was 0.02, 0.04, 0.06, 0.08, 0.1 and 0.12. For differential evolution three types of differentiation were tested (introduced in paper [13]) with uniform and exponential crossover. For values of the parameter F the following values were used: 0.5, 1, 1.5, 2 and 2.5. In particle swarm optimization the size of the neighborhood was 2, 4, 8, 16, 32, and parameters C_1 and C_2 assumed values 0.5, 1, 1.5, 2, 2.5 and 3.

After the estimation of each algorithm's parameters, two coevolutionary problem division approaches (see Sec. IV-B) were investigated, as well as the number of iterations for the base algorithm that is executed over each subpopulation in a cooperative environment. The second approach proved to be superior (Table VII shows the average error), and results for the number of iterations are shown in Table VIII.

Table III
GA ALGORITHM PARAMETERS

	n	p_c	p_m
initial	200	0.8	0.05
optimal	250	0.8	0.06

Table IV
DE ALGORITHM PARAMETERS

	n	differentiation	crossover	F	p_c
initial	200	DE/best/1	uniform	0.5	0.9
optimal	200	DE/best/1	uniform	0.5	0.7

B. Evaluation results

The aim of the second series of experiments is the comparison of all the algorithms and their coevolutionary versions

Table V
PSO ALGORITHM PARAMETERS

	n	k	C_1	C_2
initial	200	2	2	2
optimal	250	8	0.5	1

Table VI
HIB ALGORITHM PARAMETERS

	n	p_c	p_m	Δx_0	ε
initial	50	0.8	0.05	1	10^{-6}
optimal	50	0.7	0.1	-	-

Table VII
COMPARISON OF COEVOLUTION PROBLEM DIVISION APPROACHES

	DE	GA	HIB	PSO
first approach	-1.1254	-0.076895	-4.1002	-6.0824
second approach	-0.099778	-0.025158	-0.21125	-0.023291

Table VIII
NUMBER OF ITERATIONS FOR COEVOLUTIONARY ALGORITHM

	DE	GA	HIB	PSO
initial	5	5	5	5
optimal	2	10	2	5

on different data sets. In all the following tests a stopping condition of maximum $3 \cdot 10^6$ evaluations and a time limit of 10 minutes was used. Each test has been repeated 50 times.

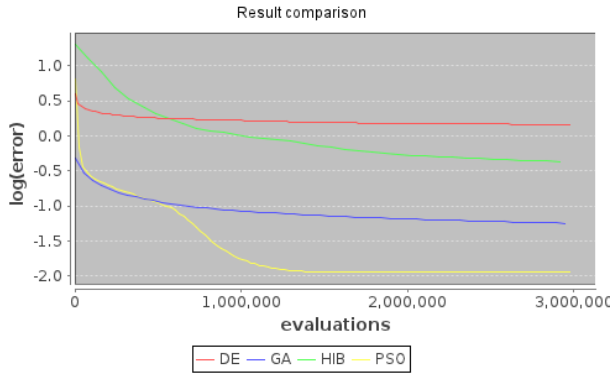

Figure 5. Average fitness for IS-param test set

1) Test set IS-param: Figure 5 shows the average value of the best solutions of all 50 repetitions, depending on the number of evaluations for the IS-param test set. PSO algorithm gives best solutions on average, and hybrid algorithm found the best overall solution with an error of 0.0022.

For further comparison of the results a boxplot graph type was used [14]. The graph shows the median, upper and lower quartile and a minimum and maximum value that is within the interval size of 1.5 IQR (interquartile range, the difference between the upper and lower quartiles). Outliers are marked with circles.

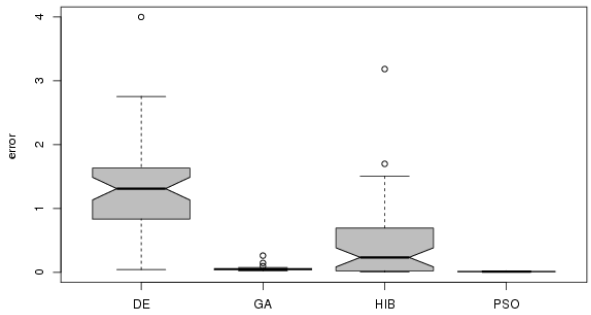

Figure 6. Result comparison for IS-param test set without coevolution

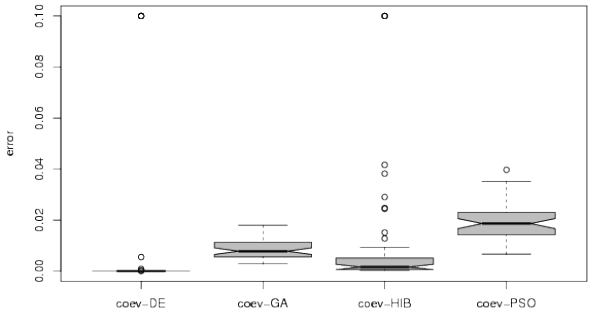

Figure 7. Result comparison for IS-param test set with coevolution

Fig. 6 shows that GA and PSO have the smallest and DE the greatest discrepancies in the results.

Coevolution results show improvements in all algorithms except the PSO [4]. The best solution is found with the DE algorithm with error of $2,122 \cdot 10^{-7}$, which is a significant improvement. Also from the Fig. 7 it is noticeable that the solution quality of DE is rather uniform, as opposed to tests without coevolution.

2) Test set IS-t2: The best average error and the best solution for the IS-t2 test set was achieved by the hybrid algorithm (Fig. 8). As in the previous case improvement with the use of coevolution is visible in all algorithms except for PSO (Fig. 9). Improvements are somewhat lower than in the previous set, which can be attributed to a small number of model parameters, where algorithms without coevolution give good results in the first place.

3) Test set IS-t5: IS-t5 test set has the most data sets and from this point of view is more difficult compared to other sets. On average, the best solution was obtained by PSO algorithm, and the best overall solution was found by the DE algorithm. Fig. 10 shows that error for the hybrid algorithm has

[4] when comparing algorithms with and without coevolution the average error was observed

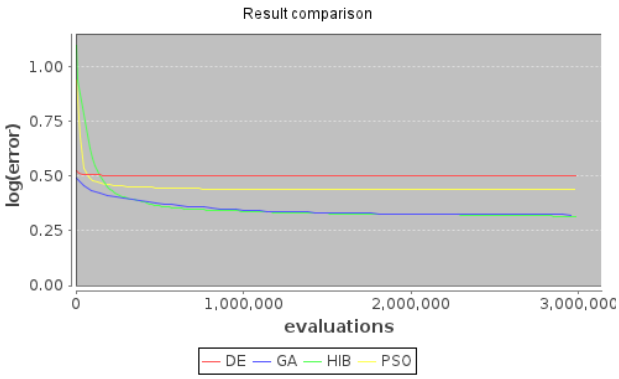

Figure 8. Average fitness for IS-t2 test set

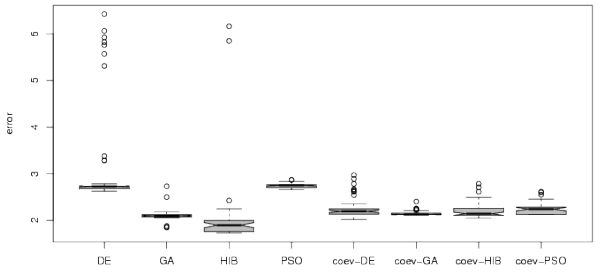

Figure 9. Result comparison for IS-t2 test set

the slowest decline, from which it can be assumed that with increasing number of evaluations it could have better results. With the application of coevolution all algorithms except PSO were improved (Fig. 11).

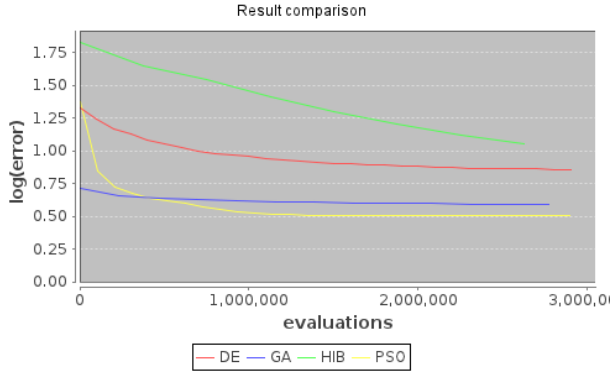

Figure 10. Average fitness for IS-t5 test set

4) Test set IS-yeast6: The best results in the last test set were obtained by PSO, and the evolution rate is shown in Fig. 12). It is interesting that the application of coevolution in this test set also improves all algorithms except the PSO. One possible explanation of this behavior is that the PSO does not adapt to constant changes in the solution space. In coevolution the fitness of each individual is calculated in

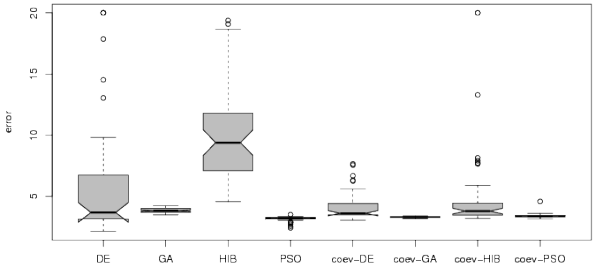

Figure 11. Result comparison for IS-t5 test set

relation to the other populations, which are also changing. The PSO algorithm contains additional information about the particle momentum, which could have a negative impact at the next iteration after the other populations change.

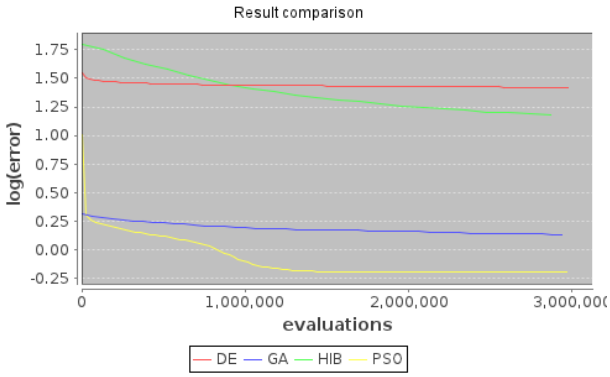

Figure 12. Average fitness for IS-yeast6 test set

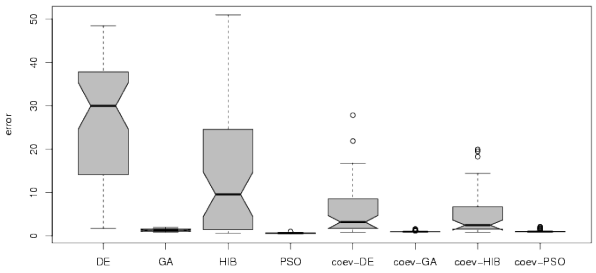

Figure 13. Result comparison for IS-yeast6 test set

C. Statistical analysis on applied coevolution

The impact of the application of the coevolution is further evaluated using T-test for the solutions obtained with and without coevolution. This type of test gives the probability that two samples originated from the same population with

the same mean. For each algorithm 50 solutions obtained without and 50 obtained with the use of coevolution were compared for each of the test sets, and the results are shown in Table IX. It is evident that most of the values are very small, indicating that the application of coevolution produces statistically significant differences. In PSO algorithm, however, the difference comes at the expense of coevolution. The values for the hybrid algorithm for two test sets are slightly larger and it is generally difficult to conclude whether in this case coevolution is a benefit.

Table IX
T-TEST RESULTS ON APPLIED COEVOLUTION

	DE	GA	HIB	PSO
IS-param	$< 10^{-6}$	$< 10^{-6}$	$2.342 \cdot 10^{-5}$	$< 10^{-6}$
IS-t2	$< 10^{-6}$	0.011	0.132	$< 10^{-6}$
IS-t5	0.053	$2.902 \cdot 10^{-4}$	0.106	$< 10^{-6}$
IS-yeast6	$< 10^{-6}$	$< 10^{-6}$	$2.87 \cdot 10^{-5}$	$< 10^{-6}$

VI. CONCLUSION

Modeling of gene regulatory networks is currently an active area of research in the field of systems biology. Creating successful models provides greater insight into cellular processes and improves the possibility of their predictions. Evolutionary computation algorithms allow search for optimal parameters of the model and, according to recent contributions, give promising results.

It is difficult to pinpoint the algorithm that would obtain the best results for every data set, so a viable approach suggests the use of multiple algorithms. The hybrid algorithm provided a more stable convergence, although requiring a larger number of evaluations. Coevolution was also proved beneficial, as it has caused improvements for all algorithms (except PSO) in each test set. We believe that the additional data structures the PSO maintains with every individual may be the main cause of its deterioration with the use of coevolution.

Possible further research includes assessing additional types of algorithms and model types, testing larger data sets and conducting experiments in the presence of noise.

REFERENCES

[1] S. K. Pal, S. Bandyopadhyay, and S. S. Ray, "Evolutionary computation in bioinformatics: a review," vol. 36, no. 5, pp. 601–615, 2006.

[2] T. Schlitt and A. Brazma, "Current approaches to gene regulatory network modelling," *BMC Bioinformatics*, vol. 8, no. Suppl 6, pp. S9+, 2007.

[3] P. Baldi, G. W. Hatfield, and W. G. Hatfield, *DNA Microarrays and Gene Expression: From Experiments to Data Analysis and Modeling*, 1st ed. Cambridge University Press, 2002.

[4] D. Tominaga, M. Okamoto, Y. Maki, S. Watanabe, and Y. Eguchi, "Nonlinear Numerical Optimization Technique Based on a Genetic Algorithm for Inverse Problems: Towards the Inference of Genetic Networks," in *German Conference on Bioinformatics (GCB)*, 1999.

[5] A. Sirbu, H. Ruskin, and M. Crane, "Comparison of evolutionary algorithms in gene regulatory network model inference," *BMC Bioinformatics*, vol. 11, no. 1, p. 59, 2010.

[6] S. Kikuchi, D. Tominaga, M. Arita, K. Takahashi, and M. Tomita, "Dynamic modeling of genetic networks using genetic algorithm and s-system," *Bioinformatics*, vol. 19, no. 5, pp. 643 – 650, 2003.

[7] J. Kim, D. G. Bates, I. Postlethwaite, H. P. Harrison, and K. H. Cho, "Linear time-varying models can reveal non-linear interactions of biomolecular regulatory networks using multiple time-series data," *Bioinformatics*, vol. 24, no. 10, pp. 1286–1292, 2008.

[8] M. Kabir, N. Noman, and H. Iba, "Reverse engineering gene regulatory network from microarray data using linear time-variant model," *BMC Bioinformatics*, vol. 11, no. Suppl 1, p. S56, 2010.

[9] S. Luke, "Essentials of metaheuristics," pp. 103–125, 2009.

[10] S. Kimura, K. Ide, A. Kashihara, 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*, vol. 21, no. 7, pp. 1154–1163, 2005.

[11] S. Kimura and A. Konagaya, "High dimensional function optimization using a new genetic local search suitable for parallel computers," *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, vol. 1, pp. 335 – 342, 2003.

[12] P. Koduru, S. Das, S. Welch, and J. L. Roe, "Fuzzy dominance based multi-objective ga-simplex hybrid algorithms applied to gene network models," in *GECCO (1)*, 2004, pp. 356–367.

[13] K. V. Price, "An introduction to differential evolution," pp. 79–108, 1999.

[14] T. Schlitt and A. Brazma, "Current approaches to gene regulatory network modelling," *BMC Bioinformatics*, vol. 8, no. Suppl 6, pp. S9+, 2007.