Hybrid immune algorithms in the gene regulatory networks reconstruction

Fefelov A.O.[1][0000-0003-1140-0985], Lytvynenko V.I.[1][0000-0002-1536-5542],
Taif M.A.[1][0000-0002-3449-6791], Savina N.B.[2][0000-0001-8339-1219],
Voronenko M.A.[1][0000-0002-5392-5125], Lurie I.A.[1][0000-0001-8915-728X],
Boskin O.O.[1][0000-0001-7391-0986]

[1]Kherson National Technical University, 24, Beryslavskoe highway, Kherson, Ukraine,73008
fao1976@ukr.net, immun56@gmail.com, taifmohamedali@gmail.com,
mary_voronenko@i.ua, aandre.lenoge@gmail.com
[2] National University of Water Management and Environmental Engineering, 11, Soborna Street, Rivne, Ukraine, 33000
n.b.savina@nuwm.edu.ua

Abstract. The reconstruction is the process of identifying the structural and dynamic properties of the system on the basis of observations of its behavior and certain knowledge in the relevant subject area. Today, reconstruction plays a crucial role in biology as one of the main tools for modeling biological systems and their interactions, which is key to understanding the mechanisms of their functioning. One of the most widely used applications of reconstruction methodology is the gene regulatory networks identification. In this paper, we propose hybrid methods for the gene regulatory networks reconstruction, which allow us to increase the speed and accuracy of solving the S-system parametric identification problem. Experimental studies of the hybrid algorithm's parameters influence the quality of solving the S-system identification problem have been carried out on test examples. A comparative analysis of developments with other similar computational methods was carried out.

Keywords: gene regulatory networks, reverse engineering, gene expression, S-system, clonal selection algorithm, differential evolution.

1 Introduction

The modern world is characterized by the enormous complexity of the systems that fill it. The examples of such systems are biological organisms. Modeling of biological systems and their interactions is of key importance for understanding their functioning mechanisms, as evidenced by the numerous studies results that have been particularly intensive in the last few decades. During this time, a lot of tools and technologies have been developed that allow building and researching models of biological systems and processes. One of the fundamental modern science areas, the specialization of which is to decipher biological information at the genetic level, is molecular biology. Using gene expression profiles obtained using DNA microarrays [1], sets of genes involved in a particular biological process are identified. Interactions between different genes or their groups within the same process are modeled using the gene regulatory network (GRN).

Detection of the presence and nature of the interactions between the genes in the GRN is essential for the new drugs creation. The reconstruction helps researchers find answers to a number of questions, among which are the following: what body's processes are regulated by the gene under study; what genes influence the gene under study; how genes interact; which genes are responsible for a particular type of disease; which drugs will have an effect in case of a disease, etc. The recent interest in system biology, in the GRN's reconstruction, is primarily due to the intensive development of high technologies and the emergence of effective methods for measuring gene expression levels to produce impressive amounts of information available for processing. Expression data can be static with the ability to study the network structure. Expression data can be dynamic, which allows modeling of the complex behavior of the system over time. GRN reconstruction methods can be divided into several groups, the main of which are: relevant networks, Bayesian networks and differential equations. Relevant networks [2] are simplified models that use correlation or mutual information as a gene interactionmeasure. Due to the low computational complexity, these methods are suitable for the GRN structure reconstruction consisting of thousands of genes, however, they do not allow modeling the gene networks dynamics. Bayesian networks and dynamic Bayesian networks [3] are more complex models based on the fundamental concepts of probability theory and mathematical statistics. In these models, the relationship between genes is given in the form of conditional probability distributions. In addition, dynamic Bayesian networks allow simulating the behavior of a GRN in discrete time.

Models based on ordinary differential equations (SODE) systems have the ability to accurately reproduce the system evolution in continuous time [4]. Among the many linear and non-linear SODE models, the S-system is most widely accepted as a compromise between accuracy and mathematical flexibility. It is non-linear and therefore capable of reproducing different types of GRN behavior. On the other hand, the S-system structure features make its interpretation and analysis quite simple. The main difficulty of the GRN reconstruction, presented in the SODE form is the high computational complexity of the problem. It is necessary to take into account the fact that mathematical models, as a rule, have their own structure and a number of parameters that must be adjusted (identified). Moreover, the data used for the gene networks reconstruction are characterized by the highest dimensionality of the attribute space. Matrix gene expression profiles include tens of thousands of genes. In [5,6,7], the authors presented the studies results on the reduction and step-by-step clustering of gene expression profiles for the GRN reconstruction. However, it should be noted that this problem currently does not have a unique solution. Also unresolved is the problem of creating new, fast and accurate model identification methods that allow the use of expression data to reconstruct the architecture and behavior of GRN.

2 Formal problem statement

The GRN reconstruction problem involves two main aspects: the choice of a gene network model, and the development of a method for identifying the selected model. For a regulatory network consisting of N genes, the S-system is represented by the following SODE [8]:

$$\frac{dx_i}{dt} = \alpha_i \prod_{j=1}^{N} x_j^{g_{ij}} - \beta_i \prod_{j=1}^{N} x_j^{h_{ij}},\ i = 1 \ldots N \tag{1}$$

where $x_i(t)$ – state variable expressing the expression level of the i-th gene at time *t*; α_i, β_i – nonnegative numbers called rate constants; g_{ij}, h_{ij} – kinetic orders that determine the direction and regulatory action degree, i.e. stimulation or suppression. Minuend $\alpha_i \prod_{j=1}^{N} x_j^{g_{ij}}$ in the right part of equality (1) corresponds to the GRN elements cumulative effect, leading to an increase in the concentration x_i, and deducted $\beta_i \prod_{j=1}^{N} x_j^{h_{ij}}$ reflects all effects resulting in reduced concentrations x_i. With increasing concentration, positive values g_{ij} imply gene *j* activation on gene *i* (enhances gene production), while negative values g_{ij} indicate the gene *j* suppressive effect on gene *i* (weakens gene production). In the decreasing concentration phase, the amplification and weakening effects of degradation correspond to positive and negative values of the parameter h_{ij}. Values $g_{ij} = 0$ ($h_{ij} = 0$) indicate the relationship absence between genes *i* and *j*.

The S-system identification consists in finding the optimal parameters values from the set $\nu = \{\alpha, \beta, g, h\}$. The complexity of solving this problem is due to its high dimensionality. The number of parameters to be found is determined by the expression $2N(N+1)$. That is, for a GRN consisting of only four genes, the search space dimension is 40. For this reason, the reconstructing S-system task is almost impossible to solve by analytical methods, especially with a large number of genes participating in the GRN. It is known that in solving such problems, population methods, such as genetic algorithms or artificial immune systems, have proven themselves well. But on large dimensions, the convergence rate of population methods can be very low. Thus, the study purpose is to develop new, fast and accurate methods for optimizing the S-system parameters, taking advantage of the population-based approach and hybridization technology.

3 Review of the literature

Evolutionary algorithms (EA), as well as artificial immune systems (AIS), are powerful optimization tools often used to solve GRN reconstruction tasks. EA is a computational paradigm based on methods that simulate the natural evolution processes of living beings. AIS models are based on the theoretical immunology studies results. According to these studies, the natural immune mechanisms of higher beings possess features characteristic of pattern recognition systems. Immune mechanisms are adaptive and resistant to small changes in the recognizable antigens characteristics (viruses and bacteria). They are able to develop their skills in recognizing alien agents. Despite the differences in paradigms, EA and AIS have a lot in common. Both those and others use a population of individuals (solutions), which from generation to generation is exposed to evolutionary or immune operators. In general, these operators are mutation and selection. The population of solutions, initially generated randomly, under the mutation and selection influence, as well as other specific operators, in each new generation improves its characteristics. The population is quantified in quantitative terms by assigning to each individual values of suitability for EA or affinity for AIS. Based on the estimates, individuals can leave the population or move on to a new generation.

Currently, several varieties of EA and AIS have been proposed for the GRN reconstruction. A combined algorithm using genetic programming (GP) and the least squares method (LSM) was proposed in [9]. Here, the ODE system general form is chosen as the GRN model, and the individuals evaluation is performed by means of the rms discrepancy between the observed and experimentally obtained data of gene expression time series. In [10], a two-stage GRN reconstruction method was proposed, where, at the first stage, using GP, approximation of the expression data with obtaining differentiable functions is performed, and at the second stage, the derivative values calculated after differentiating the functions are used by the hybrid evolutionary algorithm to evaluate individuals and search for structure and parameters GRN model presented in the S-system form. This approach allows you to get rid of complex calculations of the integration stage and increase the calculations overall speed. In [11], an evolutionary algorithm called trigonometric differential evolution (TDE) was used to reconstruct the S-system [12]. The reconstruction process consists of three phases: structural identification, parametric identification and synchronization. In addition, each next phase uses the results obtained in the previous phase. In [11], by differentiating the expression data time series, the S system is translated into an untied form. In an isolated ODE system, every equation that is included in it can be identified independently of the others, which reduces the search space dimension. Then TDE method is applied to each equation to search for its structure and parameters. In [13], a method for reconstructing an S-system based on multi-purpose optimization was proposed. Individuals multipurpose EA consist of two parts: binary and real. The binary part encodes the GRN structure, namely the presence or absence of a link between the genes, and the real part contains the S-system parameters values. For each of the parts formulated its own objective function. The result of the algorithm is a set of Pareto-optimal alternatives from which the final decisions are formed using the automatic selection procedure. In work [14], AIS in the classical clonal algorithm form was used to optimize the S system parameters, which was later improved in [15] by hybridization with one of the quasi-Newton methods. In the resulting hybrid, AIS performs the global optimizer functions, while the quasi-Newtonian method deals with local search. In [16], the clonal algorithm is applied to the S-system linearized model, where the exponents are replaced by two weight matrices: the activation matrix and the suppression matrix. This allows you to reduce the optimized parameters number, and increase the algorithm convergence rate by eliminating significant dependencies. In [17], the authors used a clonal selection hybrid method and wavelet neural network for the GRN reconstruction, represented by the ODE system general form. In this case, the neural network, as a universal approximator, is chosen to describe the functional dependencies of the ODE system right part, and the clonal algorithm is used to identify the neural network structure and parameters.

4 Materials and methods

4.1 Clonal selection

In [2], the immune system is considered from the point of view of the clonal selection mechanism. On the basis of the clonal selection principle in [18], the optimization algorithm CLONALG was proposed, which is widely used today as one of the AIS varieties. In the clonal algorithm, affinity values express the individual's proximity measures to the optimal solution and are calculated based on the objective function of the problem. In CLONALG, depending on the problem type, you can use different ways of representing solutions. The most frequently used binary and real representations. Also, the conditions and goals of the task are decisive when choosing how to represent the immune operators, the affinity function type, the algorithm parameters values.

When calculating the main population affinity, conditions are created for the selection of those cells that most fully (at this stage) interact with the antigen, i.e. form the objective function minima. In the activation process, the selected antibodies increase their representation in the solution space due to cloning. Cells whose affinity is higher create more clones but are less susceptible to mutations. Mutation in CLONALG has a high intensity because is the main driving force of evolution. In the replacement process, cells with low affinity are removed from the main population, and new, randomly generated individuals take their place. Theoretically, this allows you to avoid local extremes, and explore the entire target surface.

In [19] and [20], using comparative experiments, it was shown that the clonal selection algorithm, in its classical version, is unable to solve the problem of optimizing the S-system parameters with the required accuracy. To improve the results, the authors apply hybridization technology.

4.2 Differential evolution

The differential evolution algorithm (DE) is one of the evolutionary algorithms types [21]. Possessing high efficiency, DE has found application in many subject areas, as a global optimization method. There are several versions of the DE, differing in the evolutionary operators' implementation details. In this paper, the variant presented in [21] is used. A brief description of the DE method is given below.

The problem of minimizing the objective function is considered:

$$f(\bar{x}) \to \min, \bar{x} = (x_1, \ldots, x_n) \tag{2}$$

where $\bar{x}$ – the parameters vector of the problem, on the basis of which individuals are built decision populations $\bar{x}_i^G, i = 1, \ldots, P$, P – decision population size; G – current generation.

The DE algorithm main difference from other evolutionary algorithms is the mutation operator implementation. DE mutation is as follows:

$$\bar{v}_i^{G+1} = \bar{x}_{r_3}^G + F\left(\bar{x}_{r_1}^G - \bar{x}_{r_2}^G\right) \tag{3}$$

где $\bar{v}_i^{G+1}, i = 1, \ldots, P$ – the individual that was received as a mutation result; $r_1, r_2, r_3 \in \{1, \ldots, P\}$ – indices of individuals that are randomly selected from a population of solutions in the current generation are such that $r_1 \neq r_2 \neq r_3 \neq i$; F – scale factor ($F \geq 0$).

Components of individuals $\bar{x}_i^G$ partially replaced by the corresponding components of the vectors $\bar{v}_i^{G+1}$ with the formation of a candidates population $\bar{u}_i^{G+1} = \left(u_{i1}^{G+1}, \ldots, u_{in}^{G+1}\right)$. Vector formation $\bar{u}_i^{G+1}$ occurs according to the following expression:

$$u_{ij}^{G+1} = \begin{cases} v_{ij}^{G+1}, if\ randEvent\left(p_{DE}\right) = 1 \vee j = k \\ x_{ij}^G, otherwise \end{cases}, j = 1, 2, \ldots, n \tag{4}$$

where $k \in \{1, \ldots, n\}$ – random parameter index, selected once for each individual, the meaning of which is to ensure the transition of at least one component of the vector $\bar{v}_i^{G+1}$ to vector $\bar{u}_i^{G+1}$; p_{DE} – transition probability of j-th component of the vector $\bar{v}_i^{G+1}$ to vector $\bar{u}_i^{G+1}$; *randEvent*() – binary function generating a random event with a given probability.

Since expression (4) is associated with the crossover operator in evolutionary algorithms, p_{DE} it is called the crossover probability.

The next generation population is formed from the current generation population and the candidate population through the DE-selection:

$$\overline{x}_i^{G+1} = \begin{cases} \overline{u}_i^{G+1}, if\ f\left(\overline{u}_i^{G+1}\right) \leq f\left(\overline{x}_i^{G}\right) \\ \overline{x}_i^{G}, otherwise \end{cases} \qquad (5)$$

that is, each individual in the candidate population is compared with the corresponding individual in the current population. If the candidate has a smaller value of the objective function, he moves to a new generation. Otherwise, the current individual enters a new generation.

In [12], the DE algorithm was supplemented by the introduction of a new operator, called the trigonometric operator of mutation (TOM). Studies show that TOM is a local search tool that provides the DE algorithm with additional stability and accelerated convergence. In [20], the CLONALG and DE hybrid proposed in [20] was supplemented with the TOM operator, for the description of which the following relation is used:

$$\begin{aligned} \overline{v}_i^{G+1} &= \left(\overline{x}_{r_1}^{G} + \overline{x}_{r_2}^{G} + \overline{x}_{r_3}^{G}\right)/3 + \left(p_2 - p_1\right)\left(\overline{x}_{r_2}^{G} - \overline{x}_{r_1}^{G}\right) + \left(p_3 - p_2\right)\left(\overline{x}_{r_2}^{G} - \overline{x}_{r_3}^{G}\right) + \\ &+ \left(p_1 - p_3\right)\left(\overline{x}_{r_3}^{G} - \overline{x}_{r_1}^{G}\right); \ p_1 = \left|f\left(\overline{x}_{r_1}^{G}\right)\right| / p'; \ p_2 = \left|f\left(\overline{x}_{r_2}^{G}\right)\right| / p'; \\ p_3 &= \left|f\left(\overline{x}_{r_3}^{G}\right)\right| / p'; \ p' = \left|f\left(\overline{x}_{r_1}^{G}\right)\right| + \left|f\left(\overline{x}_{r_2}^{G}\right)\right| + \left|f\left(\overline{x}_{r_3}^{G}\right)\right|, i = 1 \ldots P \qquad (6) \end{aligned}$$

where $f\left(\overline{x}_i^{G}\right)$ – the affinity function value of the i-th individual of the population in the current generation.

In each iteration, TOM is applied with probability m_{TDE}, and the operator DE-mutation, respectively, with probability $1 - m_{TDE}$. As in the case of the DE mutation, the parameter p_{TDE}, used in TOM is analogous p_{DE}, forming the final configuration of the solution vector by expression (4).

4.3 Hybrid solutions

The proposed hybrid optimization algorithms are based on the method of expanding the mutation phase of the clonal selection algorithm with operators taken from the differential evolution algorithm. The block diagram of the hybrid clonal selection and differential evolution algorithm is shown in the figure. 1.

In the second case, a TOM is introduced into the hybrid, working with the DE-mutation in the mutually exclusive mode. The block diagram of the hybrid clonal selection algorithm and trigonometric differential evolution is shown in the figure 2.

The description of the developed hybrids main blocks is presented below.

Randomly create an antibodies initial population Ab^0. An initial version of the possible size decisions basic population is created N_{Ab}. Population Individuals are encoded as strings of real numbers, which can acquire values in the range from 0.0 to 1.0. Each line contains the S-system parameters full set (Fig. 3). Each element of the initial population individual line is generated by generating a random number in the specified interval.

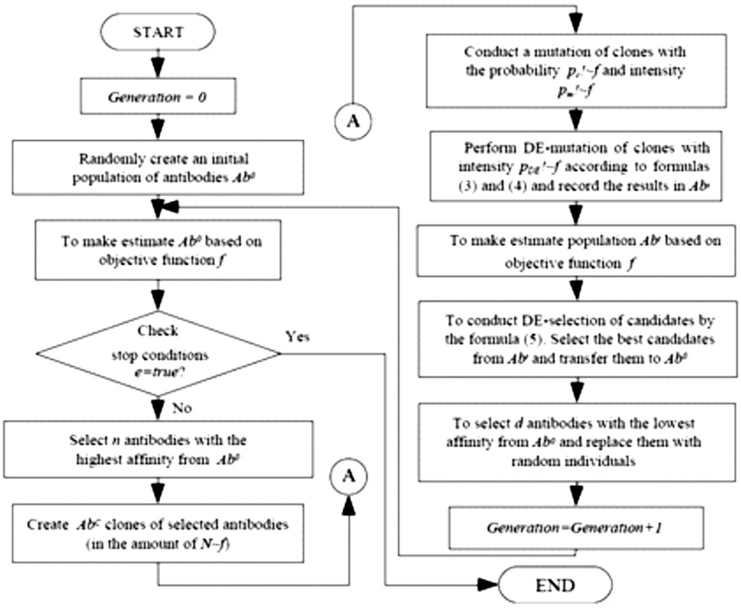
START
Generation = 0
Randomly create an initial population of antibodies Ab⁰
To make estimate Ab⁰ based on objective function f
Check stop conditions e=true?
Yes
No
Select n antibodies with the highest affinity from Ab⁰
Create Abᶜ clones of selected antibodies (in the amount of N-f)
A
A
Conduct a mutation of clones with the probability pc'-f and intensity pm'-f
Perform DE-mutation of clones with intensity pDE'-f according to formulas (3) and (4) and record the results in Abᵛ
To make estimate population Abᵛ based on objective function f
To conduct DE-selection of candidates by the formula (5). Select the best candidates from Abᵛ and transfer them to Ab⁰
To select d antibodies with the lowest affinity from Ab⁰ and replace them with random individuals
Generation=Generation+1
END

Fig. 1. Block diagram of the clonal selection hybrid algorithm and differential evolution

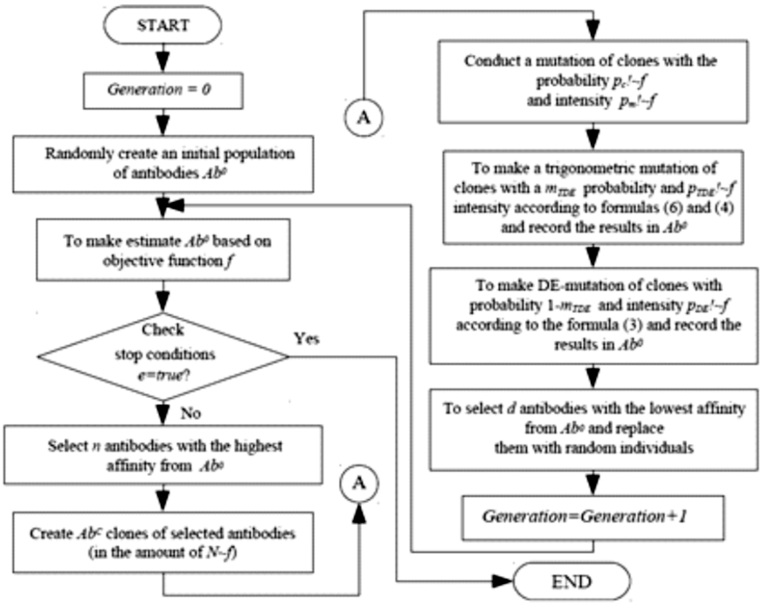
START
Generation = 0
Randomly create an initial population of antibodies Ab⁰
To make estimate Ab⁰ based on objective function f
Check stop conditions e=true?
Yes
No
Select n antibodies with the highest affinity from Ab⁰
Create Abᶜ clones of selected antibodies (in the amount of N-f)
A
A
Conduct a mutation of clones with the probability pc'-f and intensity pm'-f
To make a trigonometric mutation of clones with a mTDE probability and pTDE'-f intensity according to formulas (6) and (4) and record the results in Ab⁰
To make DE-mutation of clones with probability 1-mTDE and intensity pDE'-f according to the formula (3) and record the results in Ab⁰
To select d antibodies with the lowest affinity from Ab⁰ and replace them with random individuals
Generation=Generation+1
END

Fig. 2. Block diagram of the hybrid clonal selection algorithm and trigonometric differential evolution

Estimate Ab^0 *based on the objective function f.* When evaluating the next solution, the individual line values are recalculated into new ranges, in accordance with the change permissible intervals of one or another the S-system parameter, which are set before starting the algorithm.

α_1	β_1	g_{11}	...	g_{1N}	h_{11}	...	h_{1N}	...	α_i	β_i	g_{i1}	...	g_{iN}	h_{i1}	...	h_{iN}	...	α_N	β_N	g_{N1}	...	g_{NN}	h_{N1}	...	h_{NN}		

Fig. 3. Antibody coding for clonal hybrid algorithm

The problem objective function, used as an affinity measure of the antibodies to the antigen, is represented by the model error on the gene expression data time series. In this paper, the following expression is used to calculate the error [22]:

$$f = MIN \sum_{i=1}^{N} \sum_{j=1}^{T} \left(\frac{x_i^M (t_0 + j\Delta t) - x_i (t_0 + j\Delta t)}{x_i (t_0 + j\Delta t)} \right)^2 \tag{7}$$

where t_0 – time to start measuring gene expression levels; Δt – time step between successive measurements; T – number of measurements; $x_i^M (t_0 + j\Delta t)$ – the expression level values of the *i*-th gene, obtained using the model, i.e. SODE (1) solution (in this case, the fourth-order Runge-Kutta method); $x_i (t_0 + j\Delta t)$ – measured expression levels of the *i*-th gene.

Check break conditions. As a condition for stopping the algorithm, various indicators can be used. For example, the achievement of a given minimum value of the objective function, maximum operating time, cessation of the best antibody affinity growth over several generations, etc. In this paper, the stopping criterion is the established number of generations.

Select n antibodies with the highest affinity of Ab^0. In the CLONALG algorithm, only the antibodies with the highest affinity are selected for further cloning. Since affinity is represented by a scalar value, selection in a population is easy. You can simply sort the population and select the first n antibodies, you can use the roulette method, etc. In the present work, selection is made using a tournament, when *k* individuals are randomly selected from the population (*k*-size of the tournament) and a winner is selected by further comparison of their affinities, which proceeds to the next stage. The tournaments themselves are repeated *n* times..

Create clones Ab^C *of selected antibodies*. From the individuals selected in the previous step, a clones population of size N_{Ab^C} is created. When cloning, each individual is duplicated N times. The number of clones is directly proportional to the affinity of the antibody.

Conduct mutation of clones with probability $p_C ! \sim f$ *and intensity* $p_m ! \sim f$. The first phase of the mutation is represented by a simple scheme, which is used in the CLONALG algorithm. According to this scheme, the value of each element of the antibody string varies randomly (with a given probability p_m) by the following formula:

$$Ab_{ij} = \begin{cases} randInit(), if\ randEvent(p_m) = 1 \\ Ab_{ij}, otherwise \end{cases}, j = 1, 2, \ldots, L \qquad (8)$$

where Ab_{ij} – *j*-th element of the *i*-th individual in the population; *L* – individual line length; p_m – probability of changing the line element, called here the mutation intensity; *rndInit()* – random element initialization function; *rndEvent()* – binary function of generating a random event with a given probability.

In this case, the probability p_C determines the fact of applying the formula (8) to an individual. Also, according to the mechanism of clonal selection, mutation of antibodies with higher affinity is performed with less probability and with less intensity, i.e. there is an inverse relationship.

This is followed by blocks of diagrams representing operators taken from the differential and trigonometric differential evolution algorithm. Their detailed description is given above. Here Ab^t corresponds to the population of candidates, which is formed by the expression (4).

Select d antibodies with the lowest affinity of Ab^0 *and replace them with random individuals*. In this case, using the tournament selection, *d* individuals with the lowest affinity are selected from the population. Instead, new antibodies are placed in the population, which are created randomly. Thus, the algorithm provides the necessary population diversity level.

4.4 Search space transformation

Denote by *v* any of the parameters of the S-system *α*, *β*, *g* or *h*. To search for the optimal value of *v* by an evolutionary or immune algorithm, it is necessary that *v* be specified in the interval $v \in \Re : a \le v \le b$. The ends of the intervals *a* and *b* depend on the parameter type and are set on the basis of the subject area preliminary knowledge before the start of the optimization process. Since the search algorithms types considered here belong to combinatorial optimization tools, the values set $V, v \in V$ is finite. When using binary coding of individuals, the power *V* is defined as $|V| = 2^{m_v}$, where m_v – the length of the individual string segment (the bits number) allocated to encode the parameter *v*. The total number of solutions, among which the algorithm must find the optimal one, is equal to $2^{m_v^{2N(N+1)}}$. Obviously, the reduction m_v can significantly reduce the computational load and improve the performance of the search algorithm. However, here it is necessary to take into account the fact that all solutions are placed in the solution space not at random, but strictly in the nodes of the multidimensional grid formed by the values from the sets of $V_i, i = 1, \ldots, 2N(N+1)$ parameters of the S-system (Figure 4, *a*). If the global optimum P_{opt} does not coincide with any of this grid the nodes, then with this m_v search algorithm it is not able to find the optimal solution. Moreover, if it is small enough, then the best possible solutions, such as P_1, will be too far from the global optimum and thus will not meet the requirements for accuracy of the result.

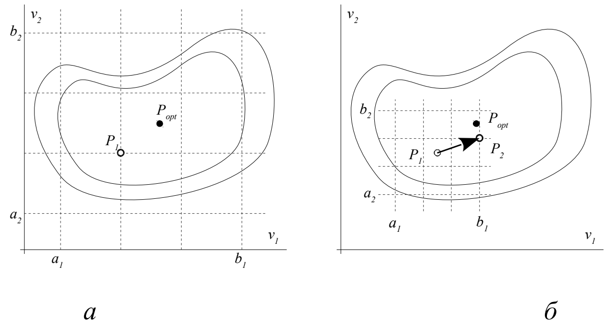

Fig. 4. *a* – initial placement of solutions in a two-dimensional search space; *б* – the transformation result of the search space relative to the solution P_1

The essence of this proposal is that, in the process of optimization, periodically, when certain conditions are reached, transform the search space by recalculating the possible values intervals of *v* with their simultaneous centering relative to the current best solution found (Figure 4, *б*). As can be seen in the figure, after receiving the solution P_1 the ends of the intervals of possible changes in parameters v_1 and v_2 were recalculated, which led to the transformation of the search space. As a result, without changing the values m_v, a more frequent grid is created, which allowed at the next iteration to get the best solution P_2, which is closer to P_{opt}, than the solution P_1. Then space is transformed relative to the solution P_2, and the process is repeated until the required accuracy of the result is achieved.

A positive feature of this approach is the ability to work with small values m_v, which significantly reduces the solution options number and speeds up the convergence of the search algorithm. The transformation involves not only compressing the space, as shown in Figure 1, *б*, but also a possible expansion if the best solution in the next iteration is at the ends of the gap of one or more model parameters. At the same time, the transformation can include simultaneous compression of space in one dimension and expansion in another, which makes the proposed approach more flexible.

Formally, the process of the solution space transformation can be represented as follows:

$$a^{j+1} = v_P^j - \frac{s^{j+1}\left(b^0 - a^0\right)}{2}; \; b^{j+1} = v_P^j + \frac{s^{j+1}\left(b^0 - a^0\right)}{2} \qquad (9)$$

where a^{j+1}, b^{j+1} – the ends of the parameter values interval v for the next transformation; a^0, b^0 – initial interval of parameter values v; v_P^j – best parameter value v, obtained using the optimization algorithm in the current (*j-th*) iteration; s^{j+1} – the scaling factor of the parameter values interval v for the next transformation, which is calculated by the formula:

$$s^{j+1} = \begin{cases} s^j k_g, \mathit{if} \left(v_P^j - a^j\right) \le \varepsilon \;\mathit{or}\; \left(b^j - v_P^j\right) \le \varepsilon \\ s^j k_s, \text{otherwise} \end{cases} \tag{10}$$

where s^j – parameter scaling factor v in the current (*j*-th) iteration; $k_g (k_g > 1)$ – span expansion factor; $k_s (0 < k_s < 1)$ – gap compression ratio; a^j, b^j – the ends of the parameter values interval v in the current (*j*-th) iteration; ε – threshold value that records the fact that the parameter v coincides with the left or right end of the gap.

It should be noted that expressions (9) and (10) are valid for both binary and real methods of encoding individuals of the search algorithm.

Figure 5 presents a flowchart showing the operation of the proposed method. In this scheme, v_P it implies the best value of the S-system parameter v obtained for the entire time of the algorithm.

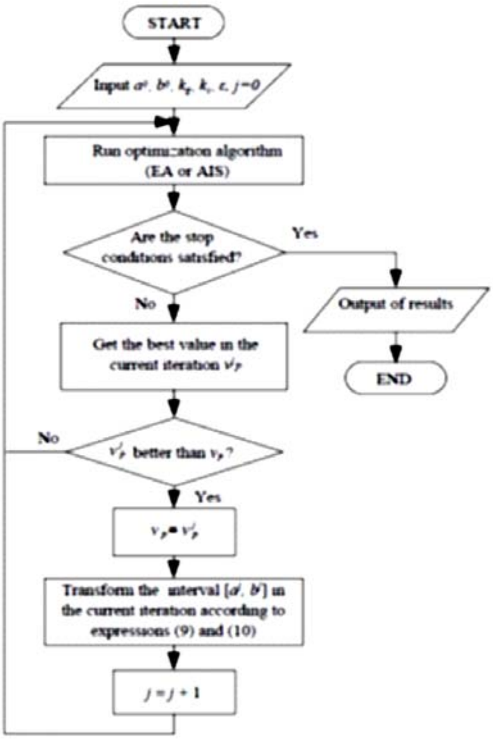
START
Input a^0, b^0, k_g, k_s, ε, $j=0$
Run optimization algorithm (EA or AIS)
Are the stop conditions satisfied?
Yes
Output of results
END
No
Get the best value in the current iteration v_P^j
No
v_P^j better than v_P?
Yes
$v_P = v_P^j$
Transform the interval $[a^j, b^j]$ in the current iteration according to expressions (9) and (10)
$j = j + 1$

Fig. 5. Block diagram of the search space transformation

The work of the proposed method and algorithm can be described as follows. After a single launch of an evolutionary or immune optimization algorithm with a fixed number of generations, the best solution obtained as a result of this start is compared with the best solution obtained for all previous launches. The comparison is made based on the approximation error values of the observed gene expression data. If the new solution turns out to be better than all previous ones, the solution space is transformed according to the formulas (9) and (10). Next, the optimization algorithm is restarted and the process is repeated until the stopping conditions are satisfied (the required number of iterations *j* or some minimum error value is reached). Thus, in each subsequent iteration *j* EA or AIS works with a new solution space, which is gradually compressed in the vicinity of the global optimum. As an optimization method, it is recommended to use either classic AIS or hybrid, discussed above in this paper.

5 Experiments

To test the proposed approach effectiveness, the authors chose two tasks to identify the S-system parameters. In both tasks, the same artificial GRN is used [10], consisting of four genes. Reconstruction is performed for a network with a fixed and non-fixed structure. The fixed network structure, in contrast to the non-fixed, implies the availability of information about the links between the genes (that is, the matrix of connections is considered constant), which simplifies the solution, since the algorithm searches only the weights of the existing links.

The parameters of the S-system for the regulatory network consisting of four genes (N=4), are presented in table 1. To build a table of observed expression data needed for the experiments, system (1) is solved with the parameters taken from table 1. Solution S -systems were carried out by the fourth-order Runge-Kutta method with the number of time samples T=26. The following initial conditions were selected in the experiments $(x_1(t_0), x_2(t_0), x_3(t_0), x_4(t_0)) = (1.4, 2.7, 1.2, 0.4)$.

Table 1. S-system parameters for GRN consisting of four genes

i	α_i	β_i	g_{i1}	g_{i2}	g_{i3}	g_{i4}	h_{i1}	h_{i2}	h_{i3}	h_{i4}
1	12.0	10.0	0.0	0.0	-0.8	0.0	0.5	0.0	0.0	0.0
2	8.0	3.0	0.5	0.0	0.0	0.0	0.0	0.75	0.0	0.0
3	3.0	5.0	0.0	0.75	0.0	0.0	0.0	0.0	0.5	0.2
4	2.0	6.0	0.5	0.0	0.0	0.0	0.0	0.0	0.0	0.8

A time series of changes in gene expression levels was obtained, its graph is shown in the figure. 6.

The developed algorithms parameters, selected for research, are presented in Table 2. The settings for simple mutation, DE and TDE mutations are set according to the results of studies conducted in [19,20,23], where at these values the maximum solution quality was obtained.

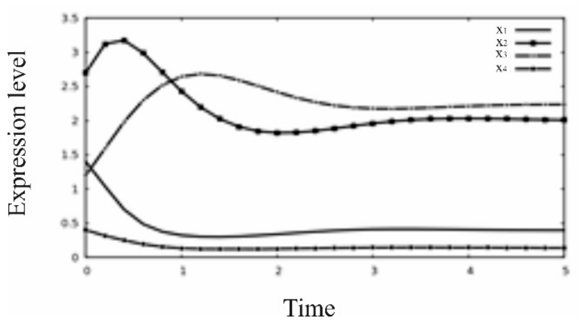

Fig. 6. The gene expression time series for experimental GRN

Table 2. Parameters of the proposed reconstruction algorithms

Parameter name	CLONALG + DE	CLONALG + TDE	CLONALG + transformation
Initial intervals of values *α* and *β*	[0.0, 15.0]	[0.0, 15.0]	[0.0, 15.0]
Initial intervals of values *g* and *h*	[-1.0, 1.0]	[-1.0, 1.0]	[-1.0, 1.0]
Coding method of individuals	real	real	binary
Parameter representation accuracy (m_v)	–	–	4 bit
Size of main population (N_{Ab})	300	300	300
Clone population size (N_{Ab^C})	900	900	900
Number of generations	3000	3000	50
Selection coefficient (n/N_{Ab})	1.0 (*n*=300)	1.0 (*n*=300)	0.7 (*n*=210)
Type of selection	tournament	tournament	tournament
Tournament size (*k*)	4	5	4
Probabilit of simple mutation (p_c)	0.1	0.1	0.01
Intensity of simple mutation (p_m)	0.05	0.05	–
Intensity of DE-mutation (p_{DE})	1.0	1.0	–
Probability TOM (m_{TDE})	–	0.6	–
Intensity TOM (p_{TDE})	–	0.7	–
Scale factor (*F*)	0.8	0.8	–
Span expansion factor (k_g)	–	–	1.5
Gap compression ratio (k_s)	–	–	0.8
Threshold value *ε*	0.0	0.0	0.0
Stop condition (number of space transformations)	–	–	50

For the experiments, the GRN reconstruction information system was used, the object-oriented architecture of which is presented in [24]. The results of each experiment were averaged on the basis of ten runs of the algorithm.

6 Results and Discussion

Comparison of the results of three AIS, based on clonal selection, as applied to the GRN reconstruction task, consisting of four genes with a fixed structure, is presented in the table 3.

Table 3. Comparative table of model errors in the study of three methods based on clonal selection for GRN with a fixed structure

Model error (*f*)	CLONALG + DE	CLONALG + transformation	CLONALG
Minimum	$1.634 \cdot 10^{-10}$	0.00112	0.03
Maximum	0.00068	0.0089	1.021
Average	0.000272	0.00466	0.6

It should be noted that in this test, out of ten starts, in 60% of cases, the CLONALG + DE algorithm converged to the minimum possible approximation error of $1.634 \cdot 10^{-10}$, and in 40% of cases, to error 0.00068, after which stagnation began. Stagnation is a differential evolution phenomenon characteristic that requires additional research that goes beyond the scope of this work.

Table 4. Comparative table of model errors in the study of three reconstruction methods for GRN with non-fixed structure

Model error (*f*)	CLONALG + TDE	CLONALG + DE	TDE
Minimum	0.00034	0.011	0.45
Maximum	0.0015	1.882	2.366
Average	0.00068	0.472	0.889

Table 4 presents comparative information on the GRN reconstruction results with a non-fixed structure. At the same time, the settings for the classical TDE algorithm are taken from the published research results and on the basis of recommendations found in the literature.

As can be seen from the table, the developed algorithms, thanks to the hybridization technology, provide a higher quality identification of the test SODE than the individual components that comprise them. In particular, the search space transformation procedure and the hybrid clonal selection and differential evolution algorithm give a smaller model error compared to the classical clonal algorithm. The clonal selection hybrid algorithm and trigonometric differential evolution shows better results than the classical differential evolution.

7 Conclusion

The paper proposes hybrid GRN reconstruction methods that allow increasing the rate of convergence and accuracy of the optimization algorithm in solving the identifying S-system problem. The methods are based on hybridization technology, combining an artificial immune system in the clonal selection algorithm form, differential evolution algorithms, and also a solution space transformation, in which space is compressed in the global optimum vicinity. To test the proposed approaches effectiveness, experiments on the artificial GRN reconstruction were carried out. During the experiments, the various parameters influence of hybrid algorithms on the solution to the S-system reconstruction problem quality was studied, and also a comparison was made with other computational methods. Experiments have shown the presence of a negative effect from the use of differential evolution operators, such as selection and crossing over. On the other hand, the operator DE-mutation showed a significant positive effect. The comparative experiments results confirmed the advantages of the developed hybrid methods and algorithms over similar classical computational methods.

Obviously, the proposed methods do not eliminate the search algorithm from the falling into local optima possibility, therefore the abilities development to choose from them is one of the priorities for further research. In addition, in the future we plan to test the development effectiveness on real biological data.

References

1. De Risi, J.: Exploring the metabolic and genetic control of gene expression on a genomic scale. Science. 1997, vol. 278, pp. 680-686 (1997)
2. Burnet, F.: A modification of jerne's theory of antibody production using the concept of clonal selection. CA: a cancer journal for clinicians. 1976, vol. 26(2), pp. 119-121 (1976)
3. Friedman, N.: Using Bayesian networks to analyze expression data. Journal of Computational Biology. 2000, vol. 7(3-4), pp. 601-620 (2000)
4. Chen, T.: Modeling gene expression with differential equations. 4th Pacific Symposium on Biocomputing, PSB '99 - Proceedings, Big Island of Hawaii, Hawaii, USA, January 1999, pp. 29-40 (1999)
5. Babichev, S., Lytvynenko V., Gozhyj, A., Korobchynskyi, M., Voronenko, M.: A fuzzy model for gene expression profiles reducing based on the complex use of statistical criteria and shannon entropy (2019) Advances in Intelligent Systems and Computing, 754, pp. 545-555 (2019). doi: 10.1007/978-3-319-91008-6_55
6. Babichev, S., Krejci, J., Bicanek, J., Lytvynenko, V.: Gene expression sequences clustering based on the internal and external clustering quality criteria (2017) Proceedings of the 12th International Scientific and Technical Conference on Computer Sciences and Information Technologies, CSIT 2017, 1, art. no. 8098744, pp. 91-94 (2017) doi: 10.1109/STC-CSIT.2017.8098744

7. Babichev, S., Lytvynenko, V., Skvor, J., Korobchynskyi, M., Voronenko, M.: Information Technology of Gene Expression Profiles Processing for Purpose of Gene Regulatory Networks Reconstruction (2018) Proceedings of the 2018 IEEE 2nd International Conference on Data Stream Mining and Processing, DSMP 2018, art. no. 8478452, pp. 336-341 (2018) doi: 10.1109/DSMP.2018.8478452
8. Savageau, M.: Introduction to S-systems and the underlying power-law formalism. Mathematical and Computer Modelling. 1988, vol. 11, pp. 546-551 (1988)
9. Sakamoto, E.: Inferring a system of differential equations for a gene regulatory network by using genetic programming. Congress on Evolutionary Computation, COEX 2001– Proceedings, Seoul, Korea, 27-30 May 2001, pp. 720-726 (2001)
10. Kim, K.: Multi-stage evolutionary algorithms for efficient identification of gene regulatory networks. Applications of evolutionary computing: EvoWorkshops– Proceedings, Budapest, Hungary, April 2006, pp. 45-56 (2006)
11. Noman, N.: Reverse engineering genetic networks using evolutionary computation. Genome Informatics. 2005, vol. 16(2), pp. 205-214 (2005)
12. Fan, H.: A trigonometric mutation operation to differential evolution. Journal of Global Optimization. 2003, vol. 27(1), pp. 105-129 (2003)
13. Chen, Y.: Inference of diochemical S-systems via mixed-variable multiobjective evolutionary optimization. Computational and mathematical methods in medicine. 2017, vol. 20. – 9p. (2017) id 3020326
14. Jereesh. A.: A clonal based algorithm for the reconstruction of genetic network using S-system. International Journal of Research in Engineering and Technology (IJRET). 2013, vol. 2, pp. 44-50 (2013)
15. Jereesh. A.: Clono-hybrid algorithm for the reconstruction of gene regulatory network using S-system. International Journal of Pure and Applied Bioscience. 2014, vol. 2(6), pp. 241-248 (2014)
16. Jereesh. A.: Two weight matrix model based GNET reconstruction using clonal algorithm. Journal of Advanced Bioinformatics Applications and Research. 2014, vol. 5(3), pp. 121-133 (2014)
17. Fefelov, A.: A hybrid approach in the reconstruction of gene regulatory networks. Control systems and machines. 2017, Kiev, № 3, pp. 63-72 (2017)
18. De Castro, L.: Learning and optimization using the clonal selection principle. IEEE Transactions on Evolutionary Computation. 2002, vol. 6(3), pp. 239-251 (2002)
19. Fefelov, A.: Parametric identification of the S-system using a modified clonal selection algorithm. Control systems and machines. 2017, Kiev, № 5, pp. 43-53 (2017)
20. Fefelov, A.: Reconstruction of the S-system by a hybrid algorithm of clonal selection and differential evolution. Control systems and machines. 2017, Kiev, № 6, pp. 41-51 (2017)
21. Storn, R.: Differential evolution — a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization. 1997, vol. 11(4). pp. 341-359 (1997)
22. Tominaga, D.: Efficient numerical optimization algorithm based on genetic algorithm for inverse problem. Genetic and Evolutionary Computation Conference, GECCO ’00 – Proceedings, Las Vegas, Nevada, USA, July 2000, vol. 251, pp. 251-258 (2000)
23. Fefelov, A.: Reconstruction of the Gene Regulatory Network by Hybrid Algorithm of Clonal Selection and Trigonometric Differential Evolution. IEEE 38th International Conference on Electronics and Nanotechnology, ELNANO 2018– Proceedings, Kyiv, Ukraine, April 24-26, pp. 305-309 (2018)

24.Fefelov, A.: Object-oriented architecture of the information system for the reconstruction of gene regulatory networks. Control systems and machines. 2017, Kiev, № 4, pp. 67-75, 82 (2017).