

# Evolving biochemical systems

Silvia Rausanu<sup>1</sup>, Crina Grosan<sup>2,4</sup>, Zujian Wu<sup>3</sup>, Ovidiu Parvu<sup>4</sup> and David Gilbert<sup>4</sup>

<sup>1</sup>ISDC Cluj Napoca, Romania

<sup>1</sup>Department of Computer Science, Babes-Bolyai University  
Cluj-Napoca, Romania

<sup>3</sup>School of Natural Computing Sciences  
University of Aberdeen, UK

<sup>2</sup>Department of Information Systems and Computing  
Brunel University  
London, UK

*silvia.rausanu@gmail.com, zujian.wu@abdn.ac.uk, {crina.grosan, ovidiu.parvu, david.gilbert}@brunel.ac.uk,*

**Abstract**—The interaction of biological compounds in cells has been enforced to a proper understanding by the numerous bioinformatics projects which contributed with a vast amount of biological information. The construction of biochemical systems (systems of chemical reactions) which include both topology and kinetic rates of the chemical reactions is an NP-hard problem. In this paper we propose a hybrid architecture which combines genetic programming and simulated annealing in order to generate and optimize both the topology (the network) and the reaction rates of a biochemical systems. Simulations and analysis of two real models show promising results for the proposed method.

**Keywords**—*biochemical systems, genetic programming, simulated annealing, optimization*

## I. INTRODUCTION

In the context of theoretical chemistry as well as systems and synthetic biology, a problem of interest is the collection detailed information on time-dependent chemical concentration data for large networks of biochemical reactions. This is done with the purpose to identifying the exact structure of a network of chemical reactions for which the identity of the chemical species present in the network is known but a priori no information is available on the species interactions.

The most convenient way of visualizing biochemical systems is by using a graphical representation. The graphical representation of a pathway in terms of chemical structures offers a great flexibility in visualising a biochemical system. The system contains a list of components and interactions between these components which are then transcribed in terms of mathematical equations.

We consider three types of entities that contribute to the composition of a system:

- Metabolites, whose concentrations change during an experiment
- Enzymes, which do not change appreciably during an experiment
- Parameters (or kinetic rates) which have constant value during the experiment.

Modelling of a biochemical system involves inferring from observed data the complex interlinked chains of biochemical reactions that lead to a biochemical product of interest. The observed data are time-course measurements of the concentrations of a variety of biological entities over time. The kinetic model involves a set of substances interacting through a network of reactions.

In this paper we focus on the automatic identification of networks (pathways) structures and their corresponding kinetic rates from observed time-domain concentrations alone (without assuming a given basic structure or any given reaction kinetics).

## II. BIOCHEMICAL SYSTEMS-REPRESENTATION AND MODELLING

Mathematical formulations of metabolic represent a biochemical system as a series of differential equations by providing a kinetic equation for each reaction of the pathway. Petri net theory is an alternative formulation based on discrete-event systems.

### A. Reaction kinetics

The basic units of a biochemical pathway are reactions between pairs of biomolecules. In the field of biochemistry, a reaction is defined as the process of transforming the molecules of the reactants in a different product within a time period. There exist two main types of reaction: spontaneous reactions, and enzymatic reactions.

A spontaneous reaction is a decaying reaction which involves the conversion of the components of a reactant into another product. Due to the forward and reverse reaction rates existing in a biochemical system, any spontaneous reaction can become reversible between the reactant and the product.

The other type of reaction is one that is mediated (or catalyzed) by an enzyme. The enzyme is simply a protein that facilitates a chemical reaction). In catalyzed reactions, the enzyme enters and exits the reaction unchanged, but is critical to yielding the product of the reaction from the reactant's constituent parts.

An enzymatic reaction is in fact a catalysed biochemical reaction which encourages the transformation of a set of

reactants into a set of products. The catalysis of the reaction is enforced by the enzyme reducing the amount of energy which is required to reach a higher energy transitional state [heiner]. An enzymatic reaction assumes the presence of at least a substrate, as reactant, a product and of an enzyme for the process of molecule conversion.

Mass action kinetics are a set of rules which are used in chemistry and chemical engineering to describe the dynamics of a reaction system. Three patterns have been defined for mass action kinetics to disclose the catalytic mechanism of enzymes in enzymatic reactions and metabolism [voet]. There are three forms of mass action kinetics; the first rule is the one used in our model and is given below.

#### Mass Action I (MAI)

The MAI considers the mechanism by which the reactants act to form an active complex together with a substrate, to modify a substrate to decay a product, and to release the product within dissociation. To each property employed by MS1 a kinetic rate is associated to each singular reaction. The processes in which an enzyme interferes are summarized by the equation:



where  $S$  is a substrate which, together with the enzyme  $E$ , forms an active complex with the kinetic rate  $k_1$ ; with the rate  $k_2$  the intermediary form  $E/S$  is decomposed to the initial reactants; the product  $P$  is formed from  $E/S$  with the kinetic rate  $k_3$ .

The atomic component of a biochemical system can be considered the most simple reaction which occurs in the system (which is considered to be less than an enzymatic reaction described by any of the mass action kinetic rules). Two patterns for the atomic reactions have been established, one pattern for creating a specie out of, at least, two species, and one pattern for decomposing one specie into, at least, two species.

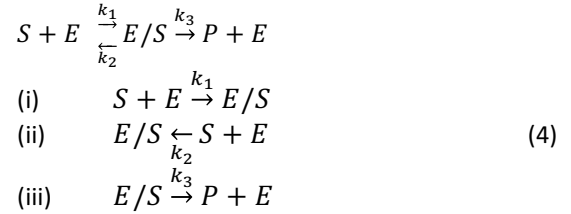
**Building pattern.** Two species are merged together to form a third one with a specific kinetic rate. Within this pattern, one of the input species is a substrate and the other one plays the role of an enzyme. The generically called product resulted between a substrate and an enzyme is an active complex. In equation (2),  $S_1$  is the substrate,  $S_2$  is the enzyme and  $S_3$  is the resulting active complex which it could have been noted as well as  $S_1/S_2$ .



**Decomposing pattern.** A product is dissociated back to its forming species with a certain kinetic rate. On such a pattern, from the two resulted reactants one is enzyme. In equation (3),  $S_3$  is the active complex,  $S_1$  is the substrate and  $S_2$  is the enzyme.



Thus, the enzymatic reaction for MA1, (given in equation (1)), can be decomposed into atomic components as follows:



With the above notation, the component (i) respects the building pattern, while the components (ii), and (iii) respect the decomposing pattern.

#### B. Petri nets

A Petri net is one of the existing mathematical modelling structures used for the description of distributed system, but mostly in biochemical systems as a reaction-system behaviour descriptor. One such net is composed of two types by nodes, places and transitions connected through edges. The usage of Petri nets in biological systems comes as a natural solution as biochemical reactions are inherently bipartite (species, interactions), concurrent (interactions occur independent or parallel) and stochastic [[1]]; such attributes de-describe best the structure of the Petri nets. The advantages of using Petri nets to model biological systems come from the intuitive and executable modelling style which is imposed. This structure offers true concurrency partial order semantics is involved. Another point is gained by Petri nets as it enforces the development of mathematical analysis techniques on it, hence over the modelled biological system [Breitling].

The Petri net can represent a number of biological compounds through its places, tokens and links. A place corresponds to a node of the net and tokens at a node can be used to represent concentration levels. Links between places represent links in a pathway.

### III. RELATED WORK

In [fogel] a basic evolutionary algorithm is used to infer biochemical systems. The entire solution is revolved around the syntax and semantics offered by the functional Petri nets. The input for the algorithm is limited to a sample of the targeted behaviour of the system. A limitation on the number of species that can participate in a reaction has been imposed (only two can take part); this is an application convention sustained by the fact that reaction involving more than two reactants and products are likely to occur and do not influence the system too much.

The representation of the solution (respectively of the individual) is an encoding mechanism of the corresponding functional Petri net (see Figure 1). The network is split into a set of encoding series, each one corresponding to an enzymatic reaction. In the encoding string only substrates and products are considered. The smallest part of the encoding represents specie and it contains the name of the specie, its id, and the kinetic rate associated to the enzymatic reaction in play. Figure 1 shows how from a reaction (represented as Petri net) the specific encoding is generated. Each candidate

solution is evaluated based on the associated fitness function. The fitness function is designed as the numerical difference between the behaviour of the generated system and the target behaviour given as input.

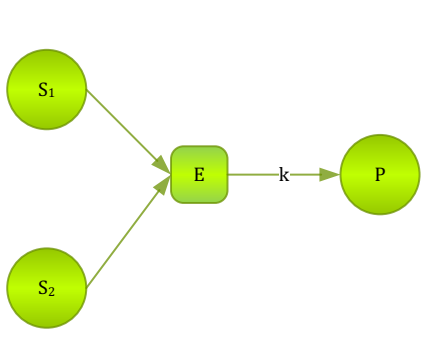


Fig. 1. Solution encoding used in [fogel]

In [sakamoto] a method for inferring biological systems characterized by differential equation is developed by the usage of genetic programming. Initially the application was creating for generating systems for gene regulation and due to the good performance it has been extended to biological systems. In the implementation has been used a system of differential equations to model the dynamics of the behaviour of the system; the generated equation system is given by

$$fractdX_i/dt = f_i(X_1, X_2, \dots, X_n), \quad i = (1, 2, \dots, n)$$

where  $X_i$  is a state variable and  $n$  is the number of components in the system.

The GP algorithm has been applied over the right-hand side of the equations in the systems. Each equation (seen as a gene) was manipulated as a function tree. The authors of [sukamoto] reported that better results have been achieved by hybridizing the GP algorithm with the least mean squares method.

The approach presented in [Kitagawa] uses evolutionary algorithms and Petri nets for modelling a biochemical system. The biochemical network is represented using a Petri net which is then translated into a string representation used by the evolutionary algorithm. The transformation allows the application of evolutionary operators in an easy manner. The method is applied in particular for metabolic pathways.

#### IV. LEARNING THE NETWORK ARCHITECTURE AND KINETIC RATES

In order to learn the architecture of the biochemical network and the associated kinetic rates, we propose a combination of two well-known intelligent systems: genetic programming and simulated annealing. We combine them in an integrated manner (as depicted in Figure 2) in the sense that the main thread of the algorithm is conducted by genetic programming while the simulated annealing is inserted for speeding up the kinetic rates optimization process.

The SA approach is fully dependent of the output generated by the GP iterations and it is not targeted to offer self-sufficient solution.

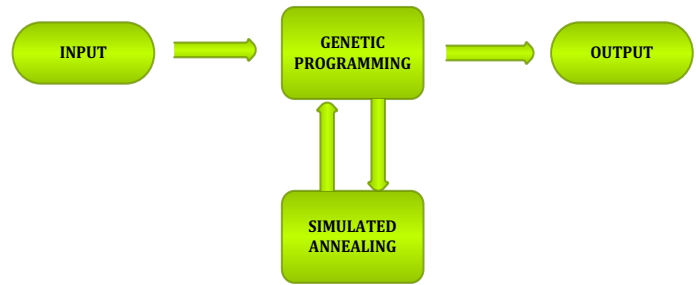


Fig. 2. Generic architecture of the integrated hybrid intelligent system.

#### $(S_1, 1, k)(S_2, 2, k)(P, 3, k)$ Implementation

The main thread of the algorithm is designed to respect almost accurately the structure of a generic GP algorithm which has access and full-rights for modifying the entire content of the representation of the solution.

The SA algorithm is involved within the GP code in order to overcome some of its short-comings generated by the large amount of parameters to optimize. The solution space for SA is much smaller than the one of GP as the access that SA gets to the solution representation is much smaller. The decision when to call SA is important as it is made on the assumption that until that point suitable solutions have been generated regarding the network topology. For the applications considered in this work, the decision has been taken on the basis of empirical data gathered during the executions: every 30 GP iterations, SA will come in to smooth the second part of the problem (appropriate kinetic rates).

The purpose of combining GP algorithm with an optimization heuristic (in our case SA) is not only to perform faster optimisations for one of the dimensions of the problem. SA can be seen as a new GP operator (not in general, but strictly for this particular problem of network inference) for increasing the diversity and encourage a global convergence of the algorithm.

As previously mentioned, the SA algorithm is called every 30 generation; at each call-point not only an SA is employed, but a considerable number of SA execution threads. Each SA thread is assigned to a GP chromosome, chosen using the survival selection operator. Once the SA optimization is over, the updated chromosomes are pushed back into the population. After being pushed back in the GP, the SA-optimized chromosomes serve as a base for another crossover/mutation session before a new GP iteration. The constants which influence the execution of any SA algorithm, initial and final temperature, and cycles per temperature, are generated randomly within a specific range for each chromosome on the execution thread in order to encourage the diversity of the solutions retrieved back.

The conceptual description of the hybridization of the solution-algorithm for inferring a biochemical network can be easily followed in the flowchart in Figure 3.

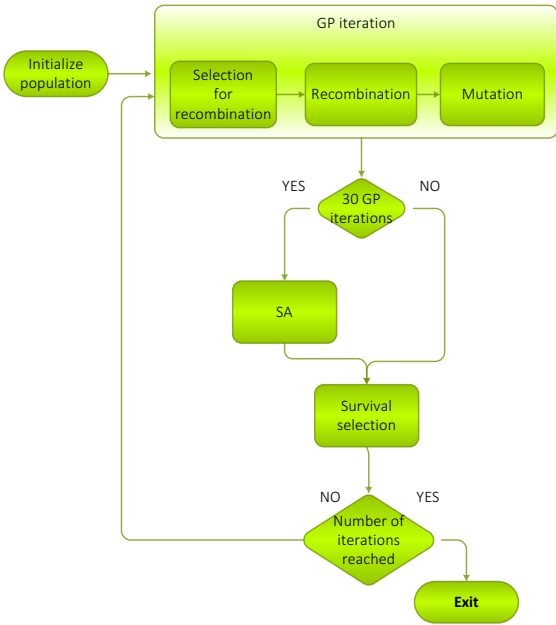


Fig. 3. Flowchart of the GP and SA algorithms interaction and application.

The following sections present how the components of the GP algorithm – chromosome representation, fitness function, selection, crossover and mutation operators - are modelled for biochemical networks and how SA is implemented.

### 1) GP Components

The components to be set for genetic programming are:

- chromosome, which encodes the solution of the problem
- fitness function, which evaluates the suitability of a solution
- mutation and crossover operators, that contribute to the generation of new solutions

Usually a GP solution is evolved using a single type of mutation and crossover; however, due to the complexity of problem and the number of parameters to optimize, there have been implemented multiple types of operators which will be presented in the following sections.

#### a) Representation

A solution for our problem is a network of reactions together with their kinetic rates.

Solutions space is defined as the set of possible reaction models involving the given enzymes, substrates and, possibly, products.

A chromosome consists of a set of reactions that occur in the corresponding real-life biochemical system, and having attached to each reaction a certain kinetic rate. The connection between this solution representation and Petri nets, which are mostly used for visualizing and simulating over a reaction model, is tight and it can be easily translated from one representation to another.

The chromosome encodes the solution by considering and containing a set of reactions. For each reaction there exist a set of reactants (input species), a set of products (output species),

and a kinetic rate attached to it. Figure 4 explains in a straightforward manner how a solution of the problem (visualized as a Petri net) is translated into a GP chromosome; the Petri net is the solution, while the table is the chromosome (each line in the table is a gene).

A validity condition per chromosome is employed: a chromosome is valid if it contains every complex specie (composed of several simple species) in the output of at least one reaction; in other words, a set of reaction in a valid chromosome should be able to generate all the species in the system. However, the validity condition should not be always accounted as it imposes a hard constraint on the candidate chromosomes; instead, the appreciation for validity is imported into the fitness function.

#### b) Fitness function

The behaviour for the chromosome is computed in the same manner described in [Kwang-Hyun], respectively by solving this system of ordinary differential equations (ODEs) associated to the network encoded in the chromosome. For example, the equations below represent the ODE system associated with the biochemical network in Figure 4.

$$\begin{aligned} \frac{ds_1}{dt} &= k_2 \cdot s_3 + k_5 \cdot s_4 - k_1 \cdot s_1 \cdot s_2 \\ \frac{ds_2}{dt} &= k_2 \cdot s_3 + k_{11} \cdot s_{11} - k_1 \cdot s_1 \cdot s_2 \\ \frac{ds_3}{dt} &= k_1 \cdot s_1 \cdot s_2 + k_4 \cdot s_4 - k_2 \cdot s_3 - k_3 \cdot s_3 \cdot s_9 \\ \frac{ds_4}{dt} &= k_3 \cdot s_3 \cdot s_9 - k_4 \cdot s_4 - k_5 \cdot s_4 \\ \frac{ds_5}{dt} &= k_5 \cdot s_4 + k_7 \cdot s_8 - k_6 \cdot s_5 \cdot s_7 \\ \frac{ds_6}{dt} &= k_5 \cdot s_4 + k_{10} \cdot s_{11} - k_9 \cdot s_6 \cdot s_{10} \\ \frac{ds_7}{dt} &= k_7 \cdot s_8 + k_8 \cdot s_8 - k_6 \cdot s_5 \cdot s_7 \\ \frac{ds_8}{dt} &= k_6 \cdot s_5 \cdot s_7 + k_7 \cdot s_8 - k_8 \cdot s_8 \\ \frac{ds_9}{dt} &= k_4 \cdot s_4 + k_8 \cdot s_8 - k_3 \cdot s_3 \cdot s_9 \\ \frac{ds_{10}}{dt} &= k_{10} \cdot s_{11} + k_{11} \cdot s_{11} - k_9 \cdot s_6 \cdot s_{10} \\ \frac{ds_{11}}{dt} &= k_9 \cdot s_6 \cdot s_{10} + k_{10} \cdot s_{11} - k_{11} \cdot s_{11} \end{aligned}$$

where:

- $s_i(t)$  represents the concentration of specie  $i$  at time step  $t$  and
- $r_j$  represents the kinetic rate associated to  $j^{\text{th}}$  reaction.

For evaluating the fitness, the difference between the target concentration and the concentrations obtained by evaluated model is calculated. The absolute values of these differences are summed up as given in equation (5).

The purpose is to minimize the value of the fitness function.

$$fitness = \sum_{i=0}^m \sum_{t=0}^n |s_i(t) - target_i(t)| \quad (5)$$

where:

- $m$  is the number of time steps for which exists an targeted behaviour
- $n$  is the number of species in the system
- $s_i(t)$  represents the concentration of specie  $i$  at time step  $t$  and
- $target_i(t)$  denotes the target concentration of specie  $i$  at time step  $t$ .

Other aggregation methods may be considered instead of simple summation of differences for each concentration, in order to ensure the model does not converge to a local optimum (most concentrations are fitted closely and the errors are only for a small number of species).

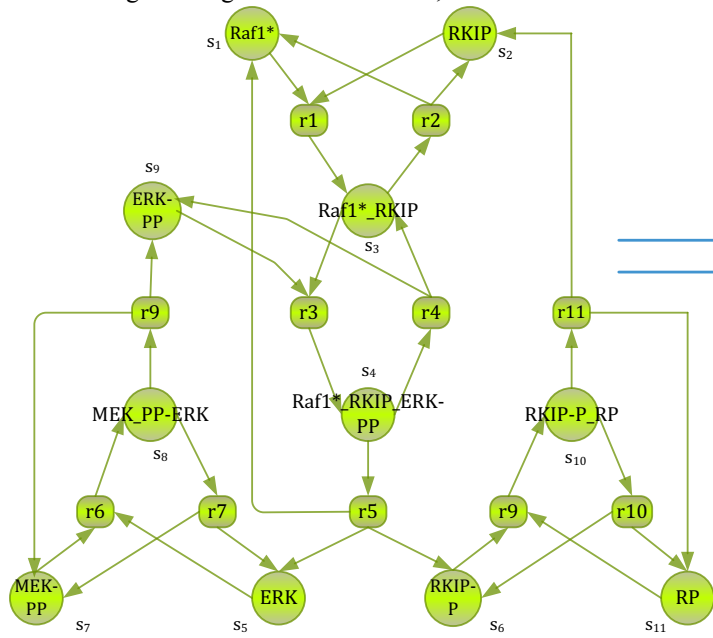
The fitness function given in equation (5) further includes a penalty function which adds a corrective value if the model does not generate certain species for which the target behaviour is given as input or generates species for which there is no given target behaviour. Thus, the fitness formula is

modified as in equation (6). By adding this penalty function, the check for validity of the chromosome, when created, is no longer required.

$$fitness = \sum_{t=0}^m \sum_{i=0}^n |s_i(t) - target_i(t)| - \sum_{t=0}^m \left( \sum_{i \in X} penalty\_missing_i + \sum_{j \in Y} penalty\_extra_j \right) \quad (6)$$

where:

- $X$  represents the set difference between target specie set and current specie set
- $Y$  represents the set difference between current specie set and target specie set
- $penalty\_missing$  is a constant representing penalty for missing specie from the current specie set
- $penalty\_extra$  is a constant representing penalty for extra specie in the current specie set
- $m, n, s_i(t)$  and  $target_i(t)$  are same notations as in equation (5)



Input species set	Output species set	Kinetic rates
{Raf1*, RKIP}	{Raf1*_RKIP}	r1 kinetic rate
{Raf1*_RKIP}	{Raf1*, RKIP}	r2 kinetic rate
{ERK-PP, Raf1*_RKIP}	{Raf1*_RKIP_ERK-PP}	r3 kinetic rate
{Raf1*_RKIP_ERK-PP}	{ERK-PP, Raf1*_RKIP}	r4 kinetic rate
{Raf1*_RKIP_ERK-PP}	{Raf1*, RKIP-PP, ERK}	r5 kinetic rate
{MEK-PP, ERK}	{MEK-PP_ERK}	r6 kinetic rate
{MEK-PP_ERK}	{MEK-PP, ERK}	r7 kinetic rate
{MEK-PP_ERK}	{MEK-PP, ERK-PP}	r8 kinetic rate
{RP, RKIP-P}	{RKIP-P_RP}	r9 kinetic rate
{RKIP-P_RP}	{RP, RKIP-P}	r10 kinetic rate
{RKIP-P_RP}	{RP, RKIP}	r11 kinetic rate

Fig. 4. Translation of a Petri net representation into a GP chromosome.

### c) Selection operators

The role of selection operators in an evolutionary computation algorithm is to select the parents for crossover and/or mutation and to select the chromosomes which will survive as solutions of the next generation. In our biochemical network inference work, we employ binary tournament, roulette wheel selection, and elitism.

In order to preserve a greater diversity in the population a composed selection operator for the next generation survivors has been designed. This operator will select the individuals in the following way:

- 5% elitism
- 45% binary tournament
- 50% roulette wheel selection.

### d) Mutation operators

We designed four mutation operators for our specific GP chromosome representation. The algorithm can use all of them at once or only some of them. Each operator affects only one gene of the chromosome, i.e. a reaction of the model. Consequently, the description of the first two operators and their associated images below refer to a single reaction in the biochemical network (which is the entire chromosome) while the last two are visualized in the context of the whole chromosome.

**Alteration of one kinetic rate.** This mutation is translated in mutation of a real representation. Thus, another real number within the specified range is generated to replace the current one as shown in Figure 5.

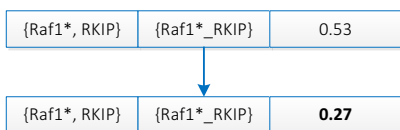


Fig. 5. Kinetic rate mutation in GP chromosome representation.

**Replacement of a specie.** A specie in the reaction is replaced randomly with another specie (as seen in Figure 6). However, the choice for the replacement specie is constrained in order not to make the chromosome contain duplicate reactions or lose the only reaction that generates certain specie.

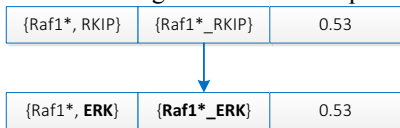


Fig. 6. Specie replacement as a form of mutation in a GP chromosome.

**Insertion of a reaction.** This operator entails adding a new reaction to the chromosome (see Figure 7). The reaction is picked randomly. The validity of the chromosome must be enforced to ensure that the new reaction does not already exist (in this case the mutation will simply be discarded).

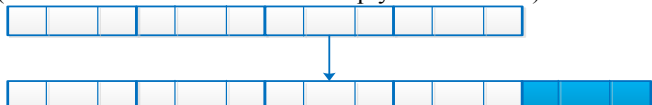


Fig. 7. Insertion of a new reaction in a GP chromosome.

**Deletion of a reaction.** This operator entails removing a reaction from the chromosome (see Figure 8). The reaction is chosen randomly. The validity of the chromosome should be checked in order not to eliminate the only reaction that produces certain specie.

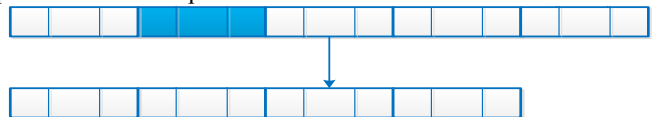


Fig. 8. Deletion of a reaction in a GP chromosome.

### e) Crossover operators

Two crossover operators have been implemented in order to increase the pallet for solution generation of the algorithm. We call them cut-and-splice and pick-and-replace and each of them is described below.

**Cut-and-splice.** This crossover operator works in two steps. First, the common genes are copied fully to the children. Afterwards, the remaining genes are assigned randomly and almost equally to the children. The order of the genes in the children does not matter as the chromosome is seen as a set of reactions. Figure 9 emphasizes how cut-and-splice crossover operator is employed.

**Pick-and-replace.** This operator, applied unidirectional, generates only one child; if applied twice on the same parents, two children may be generated. From the second parent, a reaction is chosen to replace a reaction in the first parent; the two chosen reaction from the parents must differ by one

specie, at most. For an exemplification of this operator see Figure 10.

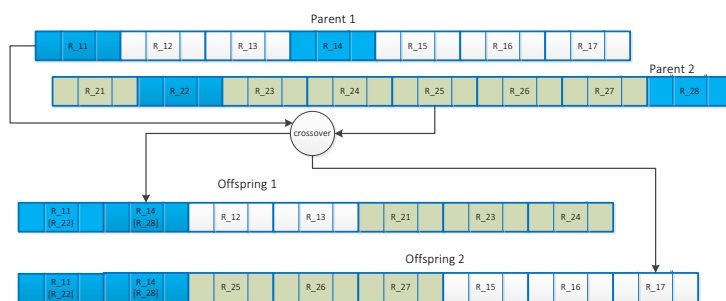


Fig. 9. Cut-and-splice crossover operator.

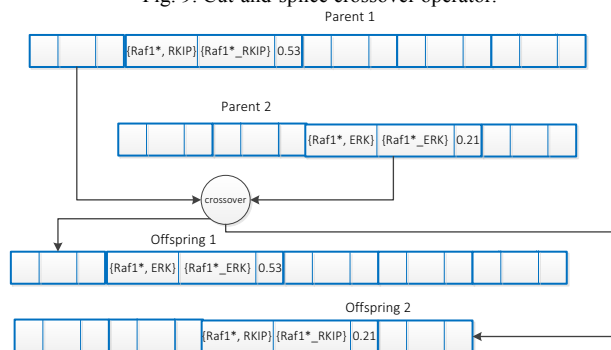


Fig. 10. Pick-and-replace crossover operator.

### 2) SA Implementation

Simulated annealing is a well known optimisation method. Since the mutation of the kinetic rates alone on the GP chromosome is not a powerful optimization method, additional measures should be taken in order to obtain good kinetic rates values. Thus, we decided to incorporate a dedicated optimization heuristic within the evolutionary process simulated by GP. The role of SA is to minimize the fitness of the chromosome defined as a genetic programming component; therefore the actual function to be optimized by the SA algorithm employed is the one encoded in the chromosome and its cost is equivalent the fitness formula in equation (6).

The chromosome generated by the GP algorithm and received as input by the SA algorithm is not entirely used for optimization; there is only one small part which is available for SA, more exactly the kinetic rates attached to each reaction (gene of the chromosome). The reasoning behind this decision consists of the rather limited capabilities of the SA in terms of computational complexity and of the speed SA has when the set of parameters to be optimized is small. The complex mutation and crossover operators have already a great power to generate new chromosomes, while the selection operators keep the population diverse enough; therefore the SA is not supposed to handle these operations again because it will make the algorithm heavier and out of its scope. Indeed, the employed mutation operators for GP contain the possibility of altering the kinetic rates, but this process may occur less often than desired and in concurrency with other operators; thus SA

is allowed to concentrate only on a small part of the chromosome.

The SA algorithm will not modify at one step too much the structure. At each iteration, SA will affect only one gene by modifying the corresponding kinetic rate with a random value within a specific range.

SA efficiency is directly proportional with the problem-specific settings of the algorithm parameters, temperatures, cycles per temperature, uphill probability. Although not favoured by theoreticians, the method trial-and-error works when establishing SA parameters.

The temperature updates have been made according to the recommendation in [heaton]. Thus the temperature updates are based on the equation (7):

$$T = e^{\frac{\ln\left(\frac{\min T}{\max T}\right)}{\text{cycles}-1}} \cdot T \quad (7)$$

where:

- T is the current temperature
- minT is a constant representing the end temperature of the algorithm
- maxT is a constant representing the start temperature of the algorithm
- cycles represents the number of annealing cycles through which the algorithm has passed.

The probability of accepting an uphill move should be proportional with the number of cycles which already ran in order to have at the beginning of the algorithm more uphill moves allowed and towards the end a transformation of the algorithm in a greedy-like approach. The function for probability of accepting an uphill move once with the number of runs was implemented according to the recommendations in [shoemaer] to respect the inequality (8):

$$\text{random} < e^{-\frac{\text{candidate}-\text{best}}{T}} \quad (8)$$

where:

- random is a uniformly distributed random number
- candidate is the candidate for uphill move
- best denotes the best solution so far and
- T is the current temperature.

### B. Output

The output of the application is intended to be as close as possible to the visual form of a biochemical network, either as a Petri net, or as a directed annotated graph. The rough output of the GP algorithmic thread will be a solution in the representation established as a GP particular component; therefore post-processing is required to bring the solution in a visualizer-simulation tool. The construction of the system will be based on Petri nets. In the recent years, a markup language for system biology has been developed. The language is called intuitively Systems Biology Markup Language (SBML). The SBML format will be used to be the actual output of the application. This type of file is supported as import file type by many Petri net simulation tools, in this way having recognized the applicability, scalability, and importance of it

in the systems biology field. The basic output of the application consists of a set of reactions, each of it having attached a kinetic rate (as specified in problem statement).

Taking into account the minimum requirements of the SBML format, the following sets must be constructed: species, parameters, and reactions. The species expected in SBML are mapped to the set of reactants in the solution system provided by the GP algorithm. The parameters are in fact the kinetic rates attached to the reactions. Finally, the list of reactions corresponds in meaning within the inner-application meaning; however, a more precise specification is required: separate reactants from products (in a wide-meaning) add assign them to the corresponding lists and write the kinetic law which in fact is the specification, by Id, of the parameter/kinetic rate used for the reaction.

Finally, when the SBML mappings are made and the file in the corresponding format is generated, the execution of the application ceases. In order to use the generated output, the actual end-user must perform a couple of steps in order to visualize the biochemical system. There are many Petri nets tools, but for testing it was used Snoopy []. So the steps for using the system are:

1. Import SBML file to Snoopy as a continuous Petri net.
2. Change concentrations of reactants (marking of places).
3. Simulate.

## V. TEST CASES

Signaling pathways play a pivotal role in many key cellular processes [Elli 02]. The abnormality of cell signaling can cause uncontrollable division of cells, which may lead to cancer. There is one of the most important and intensively studied signaling pathways: ERK pathway (the Ras/Raf-1/MEK/ERK signaling pathway) which transfers the mitogenic signals from the cell membrane to the nucleus [Yeun 00]. The ERK pathway is de-regulated in various diseases, ranging from cancer to immunological, inflammatory and degenerative syndromes and thus represents an important drug target.

A brief illustration of regulations among proteins and complex based on signaling transduction in the ERK pathway is given as follows. Ras is activated by an external stimulus, via one of many growth factor receptors; it then binds to and activates Raf-1 to become Raf-1\*, or activated Raf, which in turn activates MAPK/ERK Kinase (MEK) which in turn activates Extracellular signal Regulated Kinase (ERK). Cell differentiation is controlled by following cascade of protein interactions: Raf-1 → Raf-1\* → MEK → ERK.

The effect of regulation is dependent upon the activity of ERK. The Raf-1 kinase inhibitor protein (RKIP) inhibits the activation of Raf-1 by binding to it, disrupting the interaction between Raf-1 and MEK, thus playing a part in regulating the activity of the ERK pathway [Yeun 99]. A number of computational models have been developed in order to understand the role of RKIP in the pathway and ultimately to develop new therapies [Cho 03, Cald 04].



The RKIP pathway is used in the previous sections for the explanations of the proposed method where a simplified Petri net version of it is displayed.

(JAK)–signal transducer and activator of transcription (STAT) signaling pathway (also known as JAK-STAT) is another important and studied pathway. It is involved in signaling through multiple cell surface receptors such as receptor tyrosine kinases, Gprotein-coupled receptors, and erythropoietin receptor (EpoR). Binding of the hormone Epo to the receptor activates the receptor-bound tyrosine kinase JAK2 and it further conducts to the tyrosine phosphorylation of the EpoR cytoplasmic domain. The core module of the JAK-STAT pathway is represented in Figure 11 (the image is taken from []).

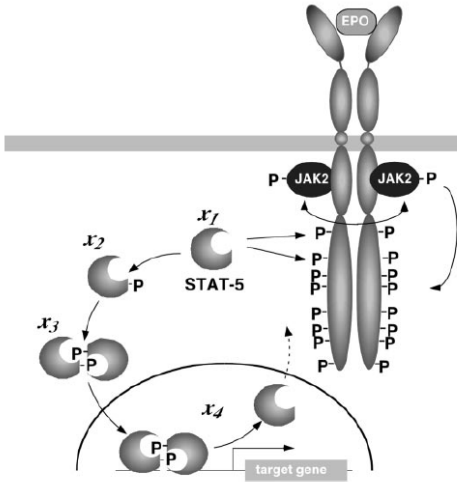


Figure 11. JAK-STAT pathway (taken from []).

Table I contains the parameters used by the proposed method in order to simulate the biochemical systems (network and kinetic rates) for these two models.

TABLE I. PARAMETRS USED IN EXPERIMENTS

Parameter s	Test cases	
	<i>RKIP</i>	<i>JAK-STAT</i>
No of independent runs	10	10
No of generations	1000	1000
Population size	500	500
Mutation probability	0.9	0.9
Mutation operators	Combination (with a random proportion) of alteration, replacement, insertion, deletion	Combination (with a random proportion) of alteration, replacement, insertion, deletion
Crossover probability	0.5	0.5
Crossover operators	Combination of cut and splice and pick and replace	Combination of cut and splice and pick and replace
Elitism	5% elitism 45% binary tournament 50% roulette wheel selection	5% elitism 45% binary tournament 50% roulette wheel selection

The parameters employed in the experiments have been tested individually as well as in various combinations. It appeared that a combination of all of them gives the best results.

All the experiments have been performed for 10 independent runs and the results have been statistically analyzed.

Figures 12 and 13 and Table II display the results of the simulations. Those results are analysed in terms of fitness value (lower values are preferable) and number of reactions matched with the initial model (the known biological model). They represent 10 independent simulations of the model implemented in Java and run on a Windows machine 2.33 GHz QC CPU and 8GB of RAM.

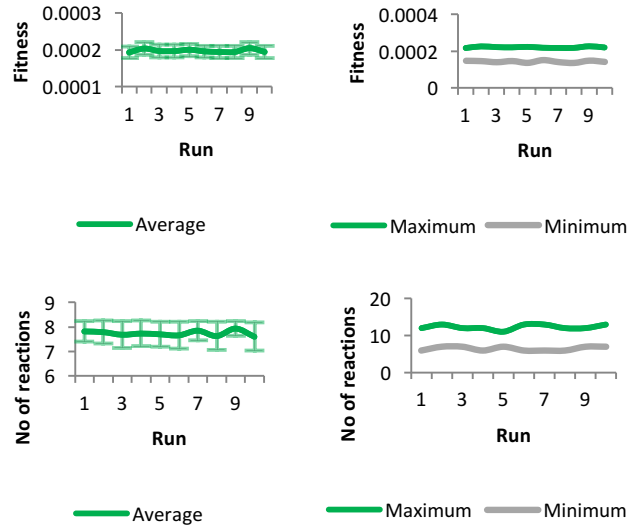


Fig. 12. Statistical analysis for the JAK-STAT system.

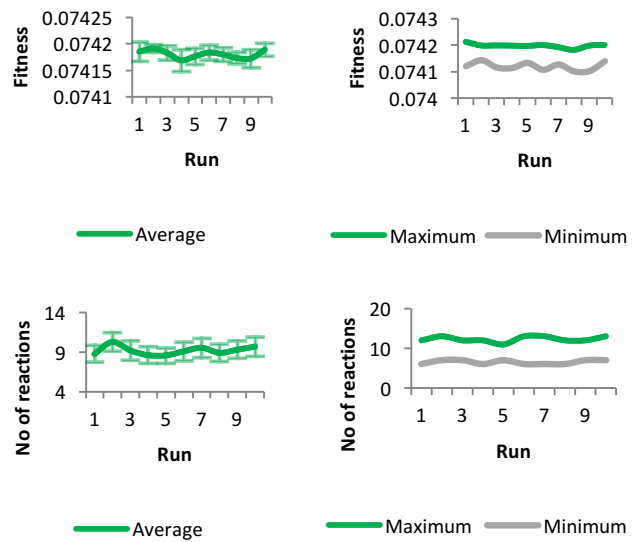


Fig. 13. Statistical analysis for the RKIP model.



TABLE II. STATISTICAL ANALYSIS

Statistical analysis	Test cases	
	<i>RKIP</i>	<i>JAK-STAT</i>
Best fitness	0.07414	1.35E-4
Average fitness	0.07418	1.97E-4
No of reactions (best solution)	10	8
Average no of reactions	9.18	7.73
Average stdev fitness	1.41E-5	1.66E-5
Average stdev no of reactions	1.11	0.48

From the above analysis it can be observed that the proposed method is able to find approximate solutions for the studied models. JAK-STAT has a lower complexity while compared with RKIP and, in this case, all the reactions existing in the initial model are evolved. Kinetic rates are close to the real values as well.

For the RKIP test case, the best individual found in all 10 runs contains 10 of the original reactions (there are 11 original reactions). However, from the graphs in Figure 13 it can be observed that in some of the runs the maximum number of reactions is 12 and 13. This, in fact, means that we find more reactions, but not all of them are contained in the original model. This is also reflected in the fitness value (the best individual, although containing only 10 reactions, has a lower fitness than the model containing 12 or 13 reactions). The results obtained show that the proposed method could be used for approximating topology and kinetic rates for complex biochemical systems. Its construction allows improvements and modifications which make it easy to adapt to similar (but not identical) biological problems (i.e. finding missing reactions in a network for instance).

## VI. CONCLUSIONS

The method proposed in this work targets both the topology and the kinetic rates design of a biological system. Genetic programming is suitable for generating network like topologies while simulated annealing is suitable for optimization. The proposed approach is able to generate the required topology (for small cases) or a good approximation for more difficult ones and a sufficient approximation of the kinetic rates. The algorithm was tested against fully specified networks. A next step, for bringing more generality to the system, would be to test it for networks in which some reaction components, rules and/or reactions are missing (or unknown even for biologists).

## REFERENCES

[1]. E. Aarts, J. Korst, W. Michiels, *Simulated Annealing and Boltzmann Machines: a stochastic approach to combinatorial optimization and neural computing*, Wiley, 1989, pp. 188-202

[2]. R. Breitling, D. Gilbert, M. Heiner, R. Orton, A structured approach for engineering of biochemical network models, illustrated for signalling pathways, *In Brief Bioinform*, 9(5):404-402, 2008

[3]. G. Fogel, D. Corne, *Evolutionary Computation in Bioinformatics*, Morgan Kaufmann, 2003, pp. 256-276

[4]. J.T. Heaton, *Introduction to Neural Networks with Java*, Heaton Research, Inc., pp. 245-266

[5]. M. Heiner, R. Donaldson, D. Gilbert, *Petri Nets for Systems Biology, Symbolic Systems Biology: Theory and Methods*, Jones & Bartlett Learning, LCC, pp. 61-97, 2010

[6]. S. Kirkpatrick, C.D. Gelatt, M.P. Vecchi, M.P., *Optimization by Simulated Annealing*, *Science*, 220(4598), pp. 671-680, 1983

[7]. J. Kitagawa, H. Iba, *Identifying Metabolic Pathways and Gene Regulation Networks with Evolutionary Algorithms*, *In Evolutionary Computation in Bioinformatics*, G. Fogel and D. Corne (Eds.), Elsevier, 2002

[8]. C. Kwang-Hyun et. al, *Mathematical modeling of the influence of RKIP on the ERK signaling pathway*, *In Computational Methods in Systems Biology*, C. Priami, editor, LNCS volume 2602, pp. 127-141, 2003

[9]. M. Oltean, C. Grosan, *A Comparison of Several Linear Genetic Programming Techniques*, *Complex Systems*, 14(4), pp. 285-313, 2003.

[10]. M. Oltean, C. Grosan, Laura Diosan, Cristina Mihaila: *Genetic Programming with Linear Representation: a Survey*. *International Journal on Artificial Intelligence Tools* 18(2): pp. 197-238 2009

[11]. E. Sakamoto, H. Iba, *Inferring a system of differential equations for a gene regulatory network by using genetic programming*, *In Proceedings of the IEEE Congress on Evolutionary Computation*, IEEE Service Center, Piscataway, N.J., 2000

[12]. S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. "Optimization by simulated annealing". *Science, Number 4598, 13 May 1983*, Vol. 220, 4598, pp. 671-680, 1983.

[13]. D. Voet, J. Voet, C.W. Pratt, *Fundamentals of Biochemistry: Life at the Molecular Level*, Wiley, 2006

[14]. Swameye, T.G. Muller, J. Timmer, O. Sandra, U. Klingmuller, *Identification of nucleocytoplasmic cycling as a remote sensor in cellular signaling by databased modelling*, *Proc Natl Acad Sci U S A*; 100(3), pp. 1028-33, 2003 (JAK-STAT)

[15]. Klingmueller, U., S. Bergelson, J. G. Hsiao, H. F. Lodish. *Multiple tyrosine residues in the cytosolic domain of the erythropoietin receptor promote activation of STAT5*. *Proc. Nat. Acad. Sci. USA*. 93, pp. 8324-8328, 1996. (JAK-STAT)

[16]. M. Calder, S. Gilmore, J. Hillston. "Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA". *In: Trans. Computational Systems Biology*, pp. 1-23, Springer, 2004.

[17]. K. H. Cho, S. Y. Shin, H. W. Kim, O. Wolkenhauer, B. McFerran, W. Kolch. "Mathematical modeling of the influence of RKIP on the ERK signaling pathway". *In: C. Priami, editor, Computational Methods in Systems Biology (CSMB '03)*, pp. 127-141, Springer, 2003.

[18]. W. Elliot, D. Elliot. *Biochemistry and molecular biology*. Oxford University Press, 2nd edition Ed., 2002.

[19]. K. Yeung, P. Janosch, B. McFerran, D.W. Rose, H. Mischak, J.M. Sedivy, W. Kolch. "Mechanism of suppression of the Raf/MEK/Extracellular signal regulated kinase pathway by the Raf kinase inhibitor protein". *Molecular and Cellular Biology*, 20(9), pp. 3079-3085, 2000.

[20]. K. Yeung, T. Seitz, S. Li, P. Janosch, B. McFerran, C. Kaiser, F. Fee, K. D. Katsanakis, D. W. Rose, H. Mischak, J. M. Sedivy, W. Kolch. "Suppression of Raf-1 kinase activity and MAP kinase signaling by RKIP". *Nature*, 401, pp. 173-177, 1999.