

# Computational models for inferring biochemical networks

Silvia Rausanu · Crina Grosan · Zujian Wu ·  
Ovidiu Parvu · Ramona Stoica · David Gilbert

Received: 15 December 2013 / Accepted: 13 May 2014 / Published online: 12 June 2014  
© Springer-Verlag London 2014

**Abstract** Biochemical networks are of great practical importance. The interaction of biological compounds in cells has been enforced to a proper understanding by the numerous bioinformatics projects, which contributed to a vast amount of biological information. The construction of biochemical systems (systems of chemical reactions), which include both topology and kinetic constants of the chemical reactions, is NP-hard and is a well-studied system biology 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 system. Simulations and analysis of an artificial model and three real models (two models and the noisy version of one of them) show promising results for the proposed method.

**Keywords** Systems biology · Biochemical systems · Genetic programming · Simulated annealing · Optimization · Petri nets

## 1 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 of 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 visualizing 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 constants), 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.

---

S. Rausanu · C. Grosan (✉) · R. Stoica  
Department of Computer Science, Babes-Bolyai University,  
Cluj-Napoca, Romania  
e-mail: crina.grosan@gmail.com; crina.grosan@brunel.ac.uk

C. Grosan · O. Parvu · D. Gilbert  
Department of Computer Science, Brunel University,  
London, UK

Z. Wu  
College of Information Science and Technology,  
Jinan University, Guangzhou, People's Republic of China

In this paper, we focus on the automatic identification of network (pathway) structures and their corresponding kinetic constants from observed time-domain concentrations alone (without assuming a given basic structure or any given reaction kinetics). The work in this paper represents an extension of our previous work [14] and includes a more detailed description of the computational system as well as two more experiments.

The paper is organized as follows: Sect. 2 describes the biological problem of biochemical systems (networks) representation and modelling. Section 3 presents existing work in this area. Section 4 describes, in details, our approach. Section 5 is dedicated to numerical experiments, and Sect. 6 concludes the work.

## 2 Biochemical systems

Mathematical formulations of metabolic pathways 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.

### 2.1 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 reactions: 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 catalysed) by an enzyme. The enzyme is simply a protein that facilitates a chemical reaction. In catalysed 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 [7]. 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 is 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 [16]. There are three forms of mass action kinetics; the first rule is the one used in our model and is given below.

#### 2.1.1 Mass action 1 (MA1)

The MA1 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 simplest 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 species out of, at least, two species and one pattern for decomposing one species into, at least, two species.

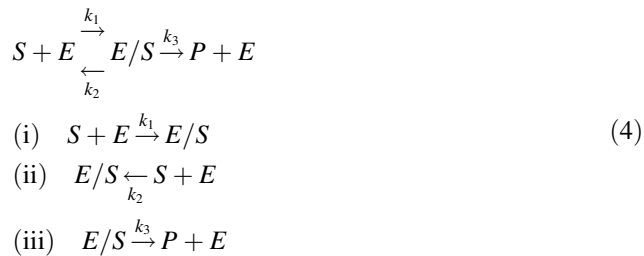
**2.1.1.1 Building pattern** Two species are merged together to form a third one with a specific kinetic rate. Within this pattern, one of the input species is a substrate and the other one plays the role of an enzyme. The generically called product resulted between a substrate and an enzyme is an active complex. In Eq. (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  $S_1/S_2$ .



**2.1.1.2 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 Eq. (3),  $S_3$  is the active complex,  $S_1$  is the substrate and  $S_2$  is the enzyme (mass action 1 is used).



Thus, the enzymatic reaction for MA1, [given in Eq. (1)], can be decomposed into atomic components as follows (mass action 1 is used):



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

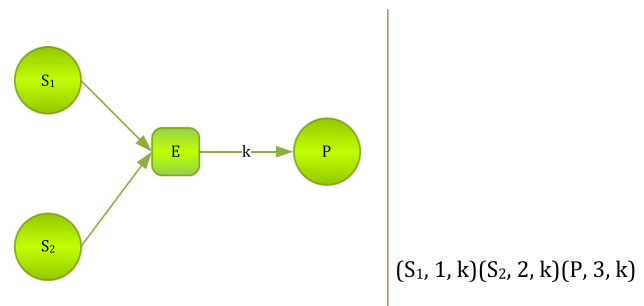
## 2.2 Petri nets

A Petri net is one of the mathematical modelling structures used for the description of distributed systems, 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 [2]; such attributes 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, and 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 [2].

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.

## 3 Related work

In [5], 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



**Fig. 1** Solution encoding used in [5]

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 Fig. 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 species, and it contains the name of the species, 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.

In [15], a method for inferring biological systems characterized by differential equation is developed using genetic programming (GP). Initially, the application was created for generating systems for gene regulation, and due to the good performance, it has been extended to biological systems. In the implementation, a system of differential equations to model the dynamics of the behaviour of the system has been used; the generated equation system is given by:

$$\frac{dX_i}{dt} = f_i X_1, X_2, \dots, X_n, \quad i = (1, 2, \dots, n)$$

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 [15] reported that better results have been achieved by hybridizing the GP algorithm with the least mean squares method.

The approach presented in [9] 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.

A combination of evolutionary strategies and simulated annealing is employed in [20, 21], but the model considers a piece-wise approach rather than a global construction.

#### 4 Learning the network architecture and kinetic constants

In order to learn the architecture of the biochemical network and the associated kinetic constants, we propose a combination of two well-known intelligent systems: genetic programming [12, 13] and simulated annealing [1, 8].

Genetic programming is an algorithmic method to evolve computer programs. This algorithm is categorized in the class of evolutionary algorithms, following closely, in a metaphoric paraphrasing, Darwin's theory of evolution, "survival of the fittest". The basic idea of genetic programming is that from a population of individuals (computer programs), randomly generated at the beginning, in which reproduction among them occurs; with the pass from generation to generation, only the fittest members will survive. Obviously, the appreciation of the fitness of one program is entirely dependent of the nature of the problem to solve. The aforementioned fitness is an important component of the GP technical structure, along with the representation on the individual, the reproduction operators—mutation and crossover—and the survival or reproduction selectors.

Simulated annealing (SA) is a heuristics for global optimum approximation inspired from process of annealing used in metallurgy for reducing defects. The basic idea in SA is the fact that during the run of the algorithm, worst moves are allowed taking into account the fact that there is a possibility that the initial heuristic may not be appropriate, and thus, a drastic change should be made. The number of accepted wrong moves is advised to decrease inversely proportional with the number of iteration through which the optimization has passed.

We combine GP and SA in an integrated manner (as depicted in Fig. 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 constants optimization process.

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

##### 4.1 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.

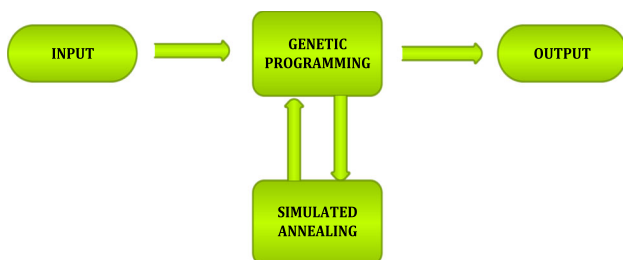
The SA algorithm is involved within the GP code in order to overcome some of its short-comings generated by the large amount of parameters to optimize. The solution space for SA is much smaller than the one of GP as the access that SA gets to the solution representation is much smaller. The decision 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: for every 30 GP iterations, SA will come into smooth the second part of the problem (appropriate kinetic constants).

The purpose of combining GP algorithm with an optimization heuristic (in our case SA) is not only to perform faster optimisation 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.

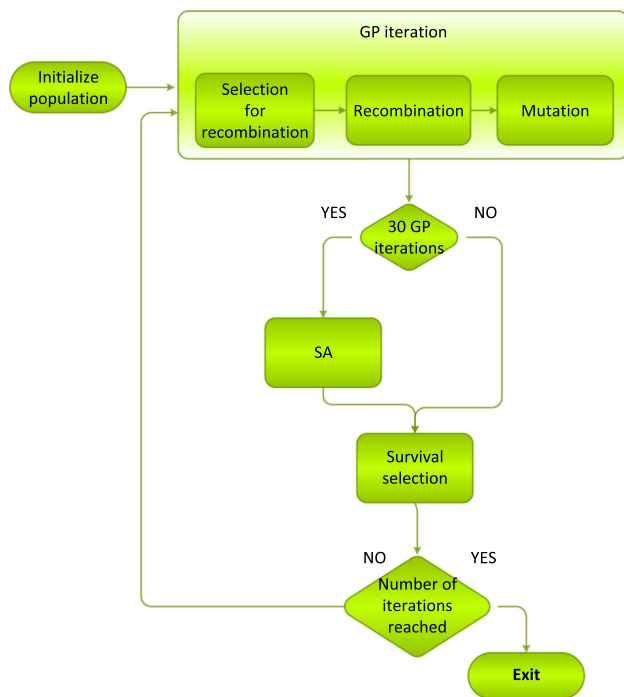
As previously mentioned, the SA algorithm is called every 30 generation; at each call-point, not only an SA is employed, but a considerable number of SA execution threads. Each SA thread is assigned to a GP chromosome, chosen using the survival selection operator. Once the SA optimization is over, the 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 hybrid method for inferring a biochemical network can be easily followed in the flowchart in Fig. 3.

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



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



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

#### 4.1.1 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 the problem and the number of parameters to optimize, there have been implemented multiple types of operators, which will be presented in the following sections.

**4.1.1.1 Representation** A solution for our problem is a network of reactions together with their kinetic constants.

Solutions space is defined as the set of all 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 straight-forward 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 species (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.

**4.1.1.2 Fitness function** The behaviour for the chromosome is computed in the same manner described in [11], 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 Fig. 4.

$$\frac{ds_1}{dt} = k_2 \cdot s_3 + k_5 \cdot s_4 - k_1 \cdot s_1 \cdot s_2$$

$$\frac{ds_2}{dt} = k_2 \cdot s_3 + k_{11} \cdot s_{11} - k_1 \cdot s_1 \cdot s_2$$

$$\frac{ds_3}{dt} = k_1 \cdot s_1 \cdot s_2 + k_4 \cdot s_4 - k_2 \cdot s_3 - k_3 \cdot s_3 \cdot s_9$$

$$\frac{ds_4}{dt} = k_3 \cdot s_3 \cdot s_9 - k_4 \cdot s_4 - k_5 \cdot s_4$$

$$\frac{ds_5}{dt} = k_5 \cdot s_4 + k_7 \cdot s_8 - k_6 \cdot s_5 \cdot s_7$$

$$\frac{ds_6}{dt} = k_5 \cdot s_4 + k_{10} \cdot s_{11} - k_9 \cdot s_6 \cdot s_{10}$$

$$\frac{ds_7}{dt} = k_7 \cdot s_8 + k_8 \cdot s_8 - k_6 \cdot s_5 \cdot s_7$$

$$\frac{ds_8}{dt} = k_6 \cdot s_5 \cdot s_7 + k_7 \cdot s_8 - k_8 \cdot s_8$$

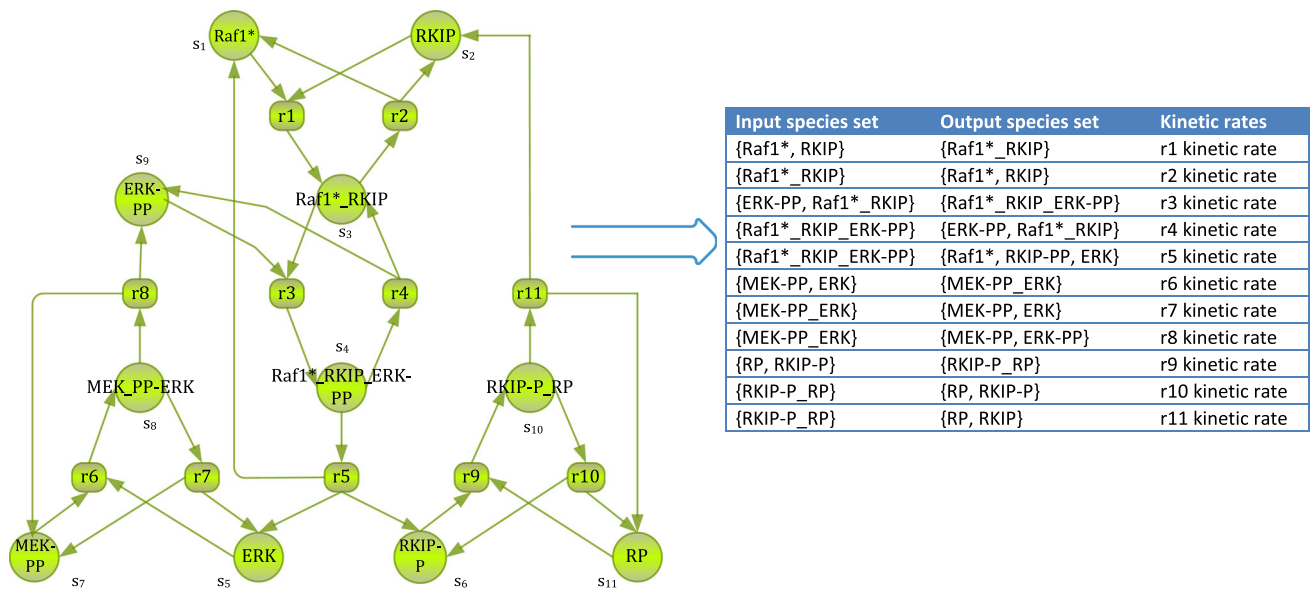
$$\frac{ds_9}{dt} = k_4 \cdot s_4 + k_8 \cdot s_8 - k_3 \cdot s_3 \cdot s_9$$

$$\frac{ds_{10}}{dt} = k_{10} \cdot s_{11} + k_{11} \cdot s_{11} - k_9 \cdot s_6 \cdot s_{10}$$

$$\frac{ds_{11}}{dt} = k_9 \cdot s_6 \cdot s_{10} + k_{10} \cdot s_{11} - k_{11} \cdot s_{11}$$

where:

- $s_i(t)$  represents the concentration of species  $i$  at time step  $t$  and
- $r_j$  represents the kinetic rate associated to  $j$ th reaction.



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

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 Eq. (5).

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

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

where:

- $m$  is the number of time steps for which exists a targeted behaviour
- $n$  is the number of species in the system
- $s_i(t)$  represents the concentration of species  $i$  at time step  $t$  and
- $\text{target}_i(t)$  denotes the target concentration of species  $i$  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 Eq. (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 Eq. (6). By adding this penalty function, the check for validity of the chromosome, when created, is no longer required.

$$\begin{aligned} \text{fitness} = & \sum_{t=0}^m \sum_{i=0}^n |s_i(t) - \text{target}_i(t)| \\ & - \sum_{t=0}^m \left( \sum_{i \in X} \text{penalty\_missing}_i + \sum_{j \in Y} \text{penalty\_extra}_j \right) \end{aligned} \quad (6)$$

where:

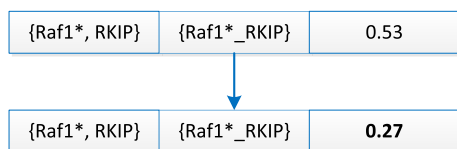
- $X$  represents the set difference between target species set and current species set
- $Y$  represents the set difference between current species set and target species set
- $\text{Penalty\_missing}$  is a constant representing penalty for missing species from the current species set
- $\text{Penalty\_extra}$  is a constant representing penalty for extra species in the current species set
- $m, n, s_i(t)$  and  $\text{target}_i(t)$  are same notations as in Eq. (5).

**4.1.1.3 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 selection, roulette wheel selection and elitism.

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

- 5 % elitism
- 45 % binary tournament selection





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

– 50 % roulette wheel selection.

**4.1.1.4 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 Fig. 5.

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

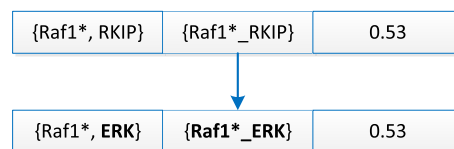
**Insertion of a reaction** This operator entails adding a new reaction to the chromosome (see Fig. 7). The reaction is picked randomly. The validity of the chromosome must be enforced to ensure that the new reaction does not already exist (in this case, the mutation will simply be discarded).

**Deletion of a reaction** This operator entails removing a reaction from the chromosome (see Fig. 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 species.

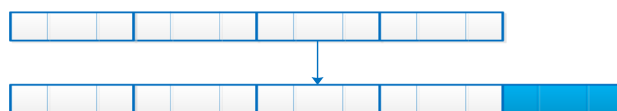
**4.1.1.5 Crossover operators** Two crossover operators have been implemented in order to increase the pallet for solution generation of the algorithm.

We use cut-and-splice and pick-and-replace crossover operators and each of them is described below.

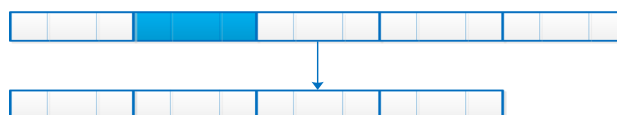
**Cut-and-splice** This crossover operator works in two steps. First, the genes common to both parents are copied into the children. Afterwards, the remaining genes are assigned randomly and almost equally to the children. The order of the genes in the children does not matter as the chromosome is seen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



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



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



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

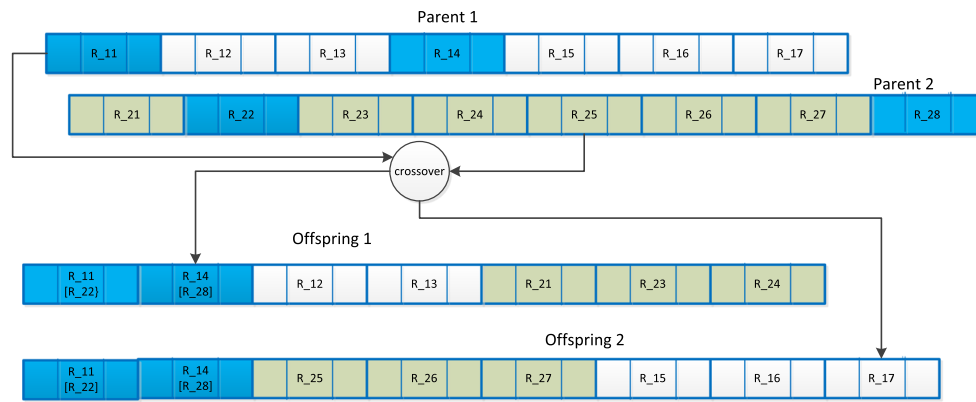
emphasizes how cut-and-splice crossover operator is employed.

**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 species, at most. For an exemplification of this operator see Fig. 10.

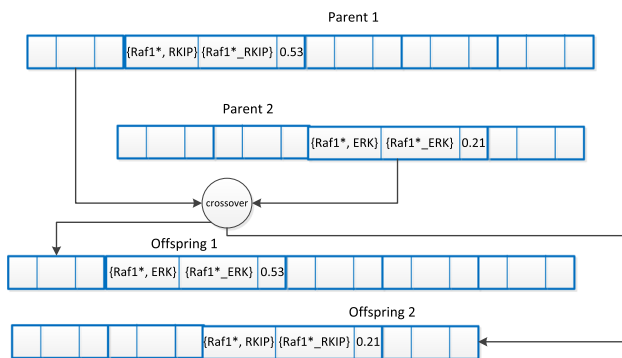
#### 4.1.2 SA implementation

Simulated annealing is a well-known optimisation method. Since the mutation of the kinetic constants alone on the GP chromosome is not a powerful optimization method, additional measures should be taken in order to obtain good kinetic rate 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 to the fitness formula in Eq. (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 constants 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



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



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

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 mutation operators employed for GP contain the possibility of altering the kinetic constants, 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 too much the structure at one step. 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 and uphill probability.

The temperature updates have been made according to the recommendation in [6]. Thus, the temperature updates are based on the Eq. (7):

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

where:

- $T$  is the current temperature
- $\min T$  is a constant representing the end temperature of the algorithm
- $\max T$  is a constant representing the start temperature of the algorithm
- $\text{cycles}$  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 to respect the inequality (8):

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

where:

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

## 4.2 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 them 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 constants 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) added 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 Snoopy tool [22] was used. 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.

## 5 Numerical experiments

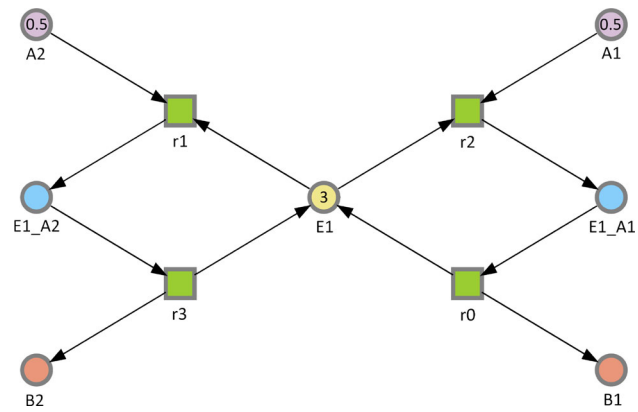
In order to test our approach, we consider four examples. One is artificial and is generated by the authors, and the others are well-known signalling pathways.

Signalling pathways play a pivotal role in many key cellular processes [4]. The abnormality of cell signalling can cause uncontrollable division of cells, which may lead to cancer.

### 5.1 Test cases

#### 5.1.1 An artificial network

The first experiment is a simple artificially created network as depicted in Fig. 11. We used this first example to test



**Fig. 11** The artificially created biochemical network

and improve the model before moving the more complex real ones.

#### 5.1.2 RKIP pathway

The RKIP pathway is one of the most important and intensively studied signalling pathways: ERK pathway (the Ras/Raf-1/MEK/ERK signalling pathway), which transfers the mitogenic signals from the cell membrane to the nucleus [18]. The ERK pathway is deregulated 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 signalling 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:  $\text{Raf-1} \rightarrow \text{Raf-1}^* \rightarrow \text{MEK} \rightarrow \text{ERK}$ .

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 [19]. 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 [3, 11].

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.

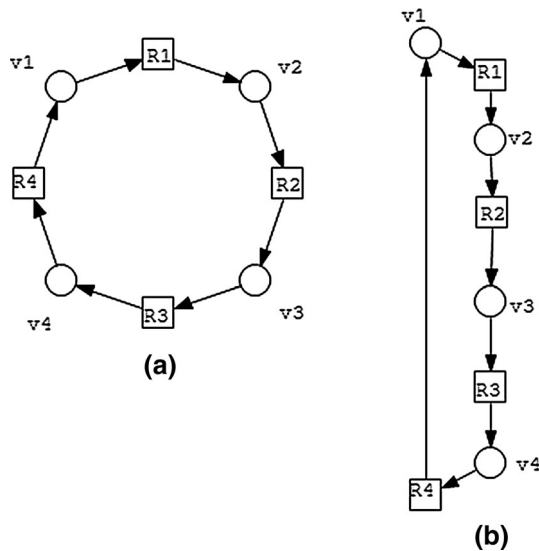
#### 5.1.3 Noisy RKIP pathway

This test is similar to standard RKIP pathway, the difference being that the input for the repository of reactions, the

reverse reactions of the original reactions, has been included as well. This approach doubles the number of reactions and tests the ability of the algorithm to adjust to noisy data.

#### 5.1.4 JAK–STAT pathway

JAK–signal transducer and activator of transcription (STAT) signalling pathway (also known as JAK–STAT) is



**Fig. 12** JAK–STAT pathway

another important and studied pathway [10, 17]. It is involved in signalling 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 two alternative ways (Fig. 12a, b).

#### 5.2 Parameter setting

Table 1 contains the parameters used by the proposed method in order to simulate the biochemical systems (network and kinetic constants) for the models.

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.

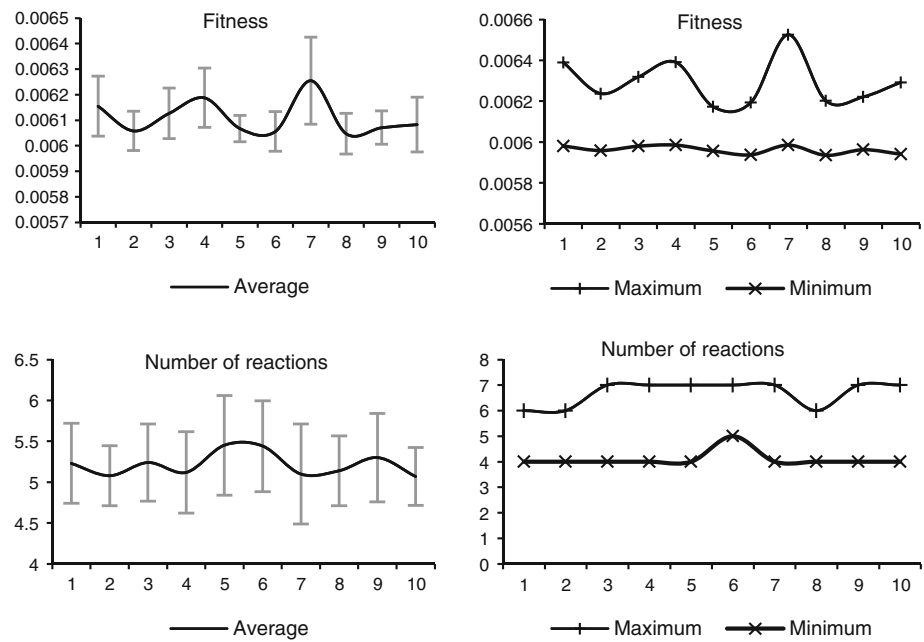
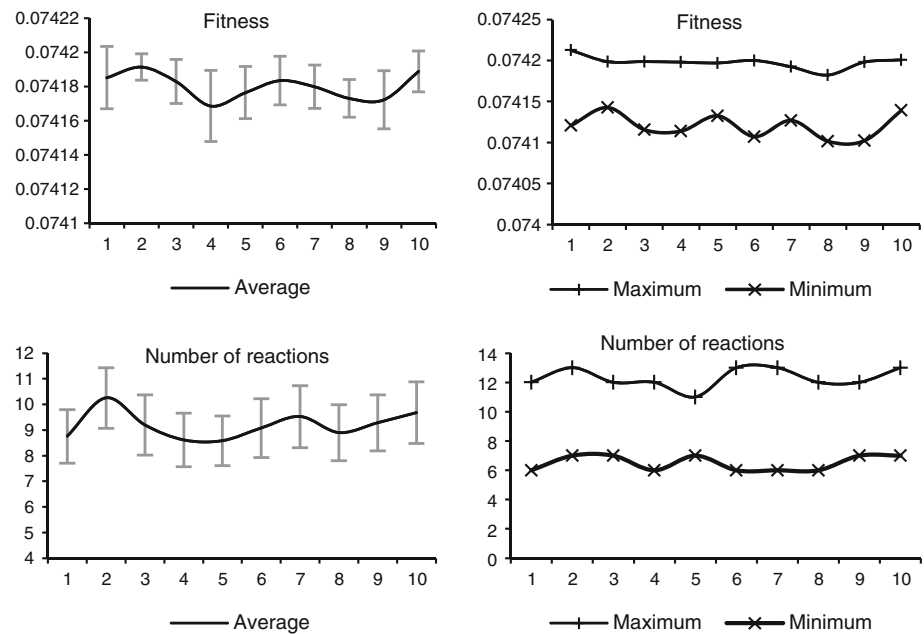
All the experiments have been performed for 10 independent runs, and the results have been statistically analysed.

#### 5.3 Results and discussion

Figures 13, 14, 15, 16 and Table 2 display the results of the simulations. Those results are analysed in terms of fitness value (lower values are preferable), and number of reactions

**Table 1** Parameters used in experiments

Parameters	Test cases			
	Artificial example	RKIP	Noisy RKIP	JAK–STAT
No of independent runs	10	10	10	10
No of generations	1,000	1,000	1,200	1,000
Population size	500	500	500	500
Mutation probability	0.3	0.9	0.3	0.9
Mutation operators	Combination (with a random proportion) of alteration, replacement, insertion, deletion	Combination (with a random proportion) of alteration, replacement, insertion, deletion	Combination (with a random proportion) of alteration, replacement, insertion, deletion	Combination (with a random proportion) of alteration, replacement, insertion, deletion
Crossover probability	0.1	0.5	0.1	0.5
Crossover operators	Combination of cut and splice and pick and replace	Combination of cut and splice and pick and replace	Combination of cut and splice and pick and replace	Combination of cut and splice and pick and replace
Elitism	5 % elitism	5 % elitism	5 % elitism	5 % elitism
	45 % binary tournament	45 % binary tournament	45 % binary tournament	45 % binary tournament
	50 % roulette wheel selection	50 % roulette wheel selection	50 % roulette wheel selection	50 % roulette wheel selection

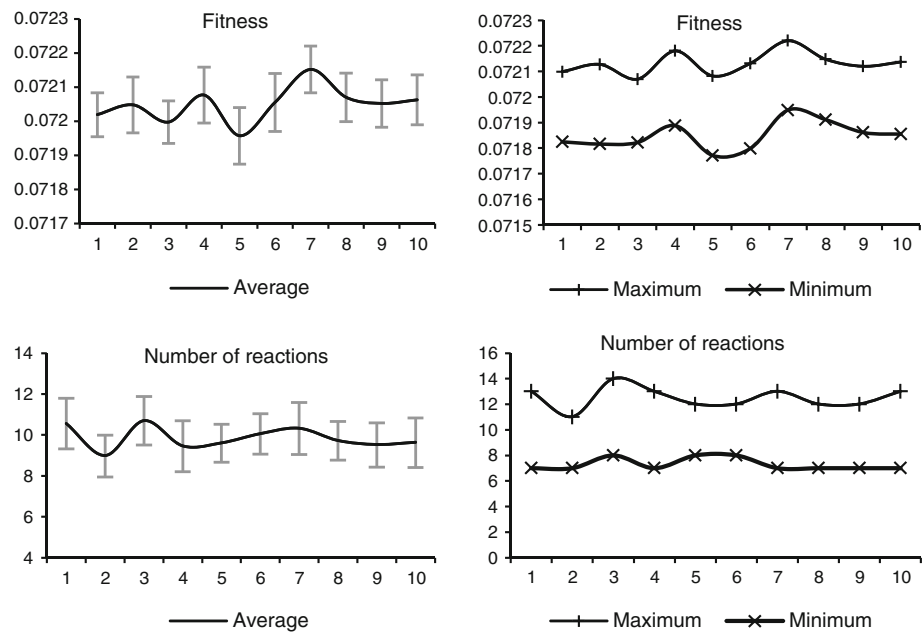
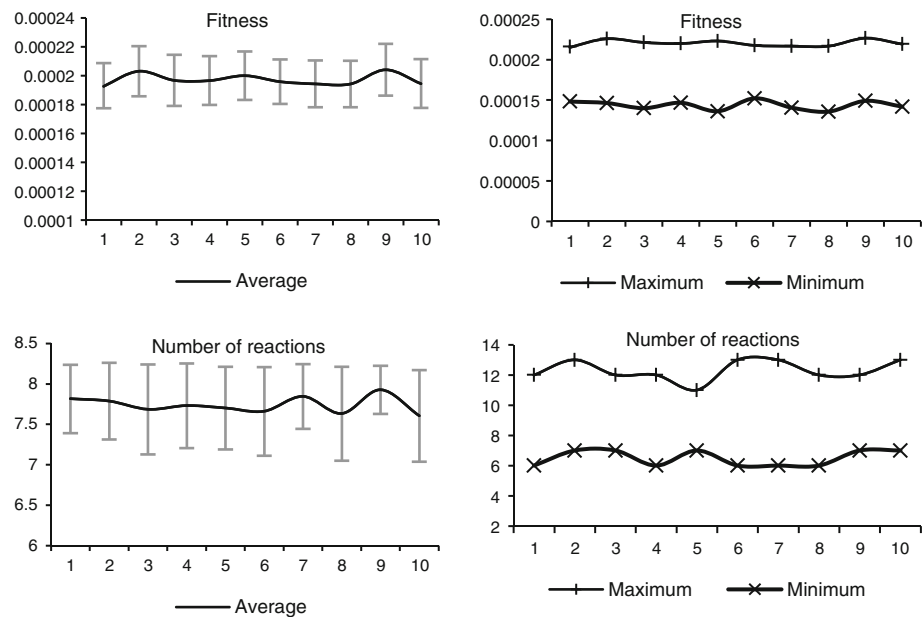
**Fig. 13** Statistical analysis for the artificial network**Fig. 14** Statistical analysis for the RKIP pathway

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 8 GB of RAM.

From the above analysis, it can be observed that the proposed method is able to find approximate solutions for the studied models. The artificial example is the simplest one, and the method obtains best results for this. The value of the fitness function is significantly lower as compared to that of the fitness for the real experiments. 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 constants 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 Fig. 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).

**Fig. 15** Statistical analysis for the Noisy RKIP pathway**Fig. 16** Statistical analysis for the JAK–STAT system**Table 2** Statistical analysis

Statistical analysis	Test cases			
	Artificial example	RKIP	Noisy RKIP	JAK–STAT
Best fitness	0.00593	0.07414	0.07177	1.35E–4
Average fitness	0.00611	0.07418	0.07204	1.97E–4
No of reactions (best solution)	6	10	10	8
Average no of reactions	5.21	9.18	9.84	7.73
Average SD fitness	9.62E–5	1.41E–5	7.4E–5	1.66E–5
Average SD no of reactions	0.49	1.11	1.14	0.48

In the case of Noisy RKIP biochemical system, the model obtained by our simulation respects the same topology and the difference of output comes from the kinetic constants flaw of approximation. We have also noticed that the number of required iterations for the fitness value to drop was greater in comparison with the simple case in previous section, still the algorithm has proven that it is able to adapt to noisy data.

The results obtained show that the proposed method could be used for approximating topology and kinetic constants 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).

## 6 Conclusions

The method proposed in this work targets both the topology and the kinetic constants 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 constants. 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). Another extension could be the investigation of other type of biochemical networks, more complex (such as cascades), not the signalling networks alone.

**Acknowledgments** S. Rausanu acknowledges support from ISDC Romania and C. Grosan acknowledges support from the Romanian National Authority for Scientific Research, CNDI-UEFISCDI, Project No. PN-II-PT-PCCA-2011-3.2-0917.

## References

1. Aarts E, Korst J, Michiels W (1989) Simulated annealing and Boltzmann machines: a stochastic approach to combinatorial optimization and neural computing. Wiley, New York, pp 188–202
2. Breitling R, Gilbert D, Heiner M, Orton R (2008) A structured approach for engineering of biochemical network models, illustrated for signalling pathways. *Brief Bioinform* 9(5):402–404
3. Calder M, Gilmore S, Hillston J (2004) Modelling the influence of RKIP on the ERK signalling pathway using the stochastic process algebra PEPA. In: Priami C et al (eds) *Transactions on computational systems biology*. Springer, Berlin, pp 1–23
4. Elliot W, Elliot D (2002) *Biochemistry and molecular biology*, 2nd edn. Oxford University Press, Oxford
5. Fogel G, Corne D (2003) *Evolutionary computation in bioinformatics*. Morgan Kaufmann, Los Altos, pp 256–276
6. Heaton JT (2008) *Introduction to neural networks with java*. Heaton Research Inc., Chesterfield, pp 245–266
7. Heiner M, Donaldson R, Gilbert D (2010) *Petri nets for systems biology, symbolic systems biology: theory and methods*. Jones & Bartlett Learning, Woods Hole, pp 61–97
8. Kirkpatrick S, Gelatt CD, Vecchi MP (1983) Optimization by simulated annealing. *Science* 220(4598):671–680
9. Kitagawa J, Iba H (2002) Identifying metabolic pathways and gene regulation networks with evolutionary algorithms. In: Fogel G, Corne D (eds) *Evolutionary computation in bioinformatics*. Elsevier, Amsterdam
10. Klingmueller U, Bergelson S, Hsiao JG, Lodish HF (1996) Multiple tyrosine residues in the cytosolic domain of the erythropoietin receptor promote activation of STAT5. *Proc Natl Acad Sci USA* 93:8324–8328 (JAK-STAT)
11. Kwang-Hyun C et al (2003) Mathematical modeling of the influence of RKIP on the ERK signaling pathway. In: Priami C (ed) *Computational methods in systems biology (CMSB)*. LNCS, vol 2602. Springer, Berlin, Heidelberg, pp 127–141
12. Oltean M, Grosan C (2003) A Comparison of several linear genetic programming techniques. *Complex Syst* 14(4):285–313
13. Oltean M, Grosan C, Diosan L, Mihaila C (2009) Genetic programming with linear representation: a survey. *Int J Artif Intell Tools* 18(2):197–238
14. Rausanu S, Grosan C, Wu Z, Parvu O, Gilbert D (2013) Evolving biochemical systems. In: *IEEE congress on evolutionary computation*, IEEE CS, pp 1602–1609
15. Sakamoto E, Iba H (2000) 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, NJ
16. Voet D, Voet J, Pratt CW (2006) *Fundamentals of biochemistry: life at the molecular level*. Wiley, New York
17. Swameye I, Muller TG, Timmer J, Sandra O, Klingmuller U (2003) Identification of nucleocytoplasmic cycling as a remote sensor in cellular signaling by databased modelling. *Proc Natl Acad Sci USA* 100(3):1028–1033
18. Yeung K, Janosch P, McFerran B, Rose DW, Mischak H, Sedivy JM, Kolch W (2000) Mechanism of suppression of the Raf/MEK/extracellular signal regulated kinase pathway by the Raf kinase inhibitor protein. *Mol Cell Biol* 20(9):3079–3085
19. Yeung K, Seitz T, Li S, Janosch P, McFerran B, Kaiser C, Fee F, Katsanakis KD, Rose DW, Mischak H, Sedivy JM, Kolch W (1999) Suppression of Raf-1 kinase activity and MAP kinase signaling by RKIP. *Nature* 401:173–177
20. Wu Z, Grosan C, Gilbert D (2013) Empirical study of computational intelligence strategies for biochemical systems modelling. In: *Nature inspired cooperative strategies for optimization (NICSO)*. Studies in computational intelligence, vol 512. Springer International Publishing, Switzerland, pp 245–260
21. Wu Z, Yang S, Gilbert D (2012) A hybrid approach to piece-wise modelling of biochemical systems. In: *12th international conference on parallel problem solving from nature*, LNCS 7491/2012, pp 519–528
22. <http://www-dssz.informatik.tu-cottbus.de/DSSZ/Software/Snoopy>