Efficient Computation of Alternative Structures for Large Kinetic Systems Using Linear Programming

János Rudan[1], Gábor Szederkényi[1,2], Katalin M. Hangos[2,3]

[1] *Faculty of Information Technology, Pázmány Péter Catholic University
Práter u. 50/a, H-1083 Budapest, Hungary*

[2] *Process Control Research Group, Systems and Control Laboratory,
Computer and Automation Research Institute,
Hungarian Academy of Sciences
H-1518, P.O. Box 63, Budapest, Hungary*

[3] *Department of Electrical Engineering and Information Systems,
University of Pannonia
Egyetem u. 10, H-8200 Veszprém, Hungary*

rudan.janos@itk.ppke.hu, szederkenyi@itk.ppke.hu, hangos@scl.sztaki.hu

(Received July 10, 2013)

Abstract

In this paper linear programming (LP)-based algorithms are presented to compute alternative structures (realizations) of biochemical reaction networks (CRNs) with mass action kinetics. The proposed algorithms have polynomial time complexity which enables us to handle large scale, biologically relevant problems. The main new contributions are the following: firstly, a new, effective LP-based method is presented that is guaranteed to compute the dense super-structure of a CRN, and secondly, it is shown that dynamically equivalent sparse structures can be computed efficiently and precisely by applying the theory of sparse nonnegative solutions of under-determined linear systems. It is shown through illustrative examples that the proposed methods outperform and thus can substitute the previously described MILP-based methods that are hard to tract computationally if the number of decision variables is high.

1   Introduction

The rigorous structural and dynamical analysis of biologically motivated kinetic systems such as intracellular signalling pathways and gene regulation networks has gained an

increased attention in recent years. The amount and quality of experimental data are continuously improving due to the fast development of sensors and computer systems. These trends naturally imply a need for the parallel improvement of modelling and computational methods to be able to handle the growing amount of data and to analyse more complex, possibly biologically relevant processes and networks.

In this paper, by Chemical Reaction Networks (CRNs) we mean deterministic kinetic systems obeying the mass action law. It is known that such systems form a wide class of smooth nonlinear systems that are able to produce all important qualitative phenomena in nonlinear dynamics such as oscillations, multiplicities and even chaos [1]. Thus, the kinetic system form can be useful for the description of nonnegative models outside of (bio)chemistry, e.g. for epidemic, transportation or economic models as well [2, 3].

The general applicability of kinetic models is definitely extended by the strong results of chemical reaction network theory (CRNT). CRNT was initiated in the 1970's and 80's with the first publications about the relations between the structure and qualitative dynamics of CRNs treated as a general nonlinear system class [4, 5, 6]. Since then, numerous deep and useful results have been published in this continuously developing field (see, e.g. [1, 7, 8])

It has been known at least from the 1970's that structurally/parametrically different reaction schemes might produce exactly the same dynamics in the concentration space. This phenomenon is usually called *macro-equivalence* [4] or *dynamical equivalence.* Necessary and sufficient geometric conditions for macro-equivalence were shown in [9] with a technical comment in [10]. For the computation of dynamically equivalent structures with preferred properties such as detailed or complex balance, (weak) reversibility, the inclusion of minimal/maximal number of reactions or complexes, optimization-based procedures have been published in [11, 12, 13, 14]. The concept of *linear conjugacy* is an extension of dynamical equivalence, allowing a positive diagonal state transformation between the solutions of linearly conjugate CRNs [15]. The previously mentioned computational methods were generalized for linearly conjugate systems in [16, 17]. Moreover, the important problem of finding a weakly reversible linearly conjugate CRN structure with minimal deficiency was solved in [18]. Biologically relevant examples for the structural non-uniqueness of CRNs were shown in [19].

Linear programming (LP) can be defined as the problem of minimizing (or maximizing)

a linear function with respect to linear constraints, where all the variables are real-valued. Constraints can be both equalities and inequalities because in this context they can be transformed into each other [20]. Solution techniques, such as the simplex algorithm and interior point based methods are widely investigated and implemented in freely available software tools. Currently applied solution methods usually guarantee the polynomial-time solution of the LP problems. There are numerous important problems in science and engineering that can be solved through LP [21, 22].

Mixed integer linear programming (MILP) differs from LP in that some of the decision variables are integer valued. This fact has a fundamental effect on the complexity of the solution. The corresponding MILP problem is a combinatorial optimization problem which is generally NP-hard. Several solution techniques are developed to search for the solution with heuristic-driven search methods (e.g. branch-and-bound, price-and-bound techniques) and some of them are implemented in freely available solvers. It should be noted that the size and the structure of the original problem together with the type of solver seriously influence the solution efficiency. MILP techniques are widely applied in the case of scheduling problems, process control, traffic control etc. [23].

Using the fact that certain propositional logic problems can be transformed into MILP problems by introducing appropriate logical variables [24, 25], the computation of pre-ferred kinetic realizations were straightforwardly written in an MILP framework in [11]. However, the original MILP approach seriously limits the network size that can be treated computationally within a reasonable time interval, and it is known that networks of bi-ological relevance often contain several hundred complexes and reactions. Therefore, the aim of this paper is to propose and analyse computationally efficient methods preferably not containing integer variables for determining dynamically equivalent sparse and dense realizations containing the minimal or maximal number of reactions, respectively.

The structure of the paper is the following: in Section 2, the modelling framework is described. In Section 3, the existing computational methods are reviewed and the proposed new methods are introduced. In Section 4, the correctness of the proposed methods are shown through randomly generated and biologically meaningful examples, too. In Section 5, the main results are summarized.

2 Structural and dynamical description of CRNs

In this Section, the general notations and definitions for representing CRNs are introduced. The applied notations are based on the introductory parts of [11, 12, 13].

2.1 Basic notions

We consider CRNs as closed deterministic chemical systems under isothermal and isobaric conditions. The fundamental elements of the system are the so-called chemical species X_i, $i = 1, ..., n$ taking part in r reactions that obey the mass action law. The state vector is built up from the concentrations of the species, i.e. $x_i = [X_i]$, $i = 1, ..., n$ the values of which are nonnegative.

Complexes C_i, $i = 1, ..., m$ are formally nonnegative linear combinations of the species, i.e. $C_k = \sum_{i=1}^{n} \alpha_{ik} X_i$ for $k = 1, ..., m$ where α_{ik} are the nonnegative integer stoichiometric coefficients. An elementary reaction step between the source and product complexes, C_i and C_j, respectively, is denoted by

$$C_i \rightarrow C_j \tag{1}$$

According to the mass action law, the reaction rate corresponding to reaction (1) is given by

$$\rho_{ij}(x) = k_{ij} \prod_{l=1}^{n} x_l^{\alpha_{li}}, \tag{2}$$

where $k_{ij} > 0$ is the reaction rate coefficient.

It can happen that both reactions $C_i \rightarrow C_j$ and $C_j \rightarrow C_i$ are present in the network. Such pairs of reactions are called reversible reactions. In this particular framework, these reactions are handled as separate elementary reactions.

2.2 Graph representation

We can assign the following directed graph to the reaction network: the directed graph $D = (V_d, E_d)$ consists a finite nonempty set V_d of vertices and finite set of E_d of ordered pairs of distinct vertices called directed edges. The vertices represent the complexes: $V_d = \{C_1, ...C_m\}$ and the edges stand for the reactions: $(C_i, C_j) \in E_d$ if complex C_i is transformed to C_j in one of the reactions in the network. The reaction rates appear as nonnegative weights on the edges in the directed graph. This graph-based representation described in the form of matrices will be used in the forthcoming parts of this paper.

By the structure of a given CRN we mean the unweighted graph of the reaction network.

2.3 ODE-based description

CRN systems can be represented with differential equations and vica versa. In this paper - similar to [11] - the following representation is used which is defined in [26].

$$\dot{x} = Y \cdot A_k \cdot \psi(x) \tag{3}$$

where $x \in \mathbb{R}^n$ is the vector of the concentrations of the species, $Y \in \mathbb{R}^{n \times m}$ is the matrix containing the stoichiometric coefficients of the complexes namely $Y_{ij} = \alpha_{ij}$, $A_k \in \mathbb{R}^{m \times m}$ contains the edge weights of the weighted directed graph of the CRN and $\psi : \mathbb{R}^n \to \mathbb{R}^m$ is a vector mapping defined by:

$$\psi_j(x) = \prod_{i=1}^{n} x_i^{Y_{ij}}, \quad j = 1, ..., m \tag{4}$$

The structure of matrix Y is the following: the ith column of Y contains the composition of complex C_i, so the Y_{ij} value is the stoichiometric coefficient of C_i corresponding to the specie X_j.

The matrix A_k has the following elements:

$$[A_k]_{ij} = \begin{cases} -\sum_{l=1, l \neq i}^{m} k_{il} & if \ i = j \\ k_{ji} & if \ i \neq j \end{cases} \tag{5}$$

meaning that the diagonal elements of A_k contain the negative sum of the weights of the edges starting from the complex C_i while the off-diagonal elements $[A_k]_{ij}$, $i \neq j$ contain the weights of the directed edges (C_j, C_i) coming into C_i. Therefore A_k is a column conservation matrix: the sum of the elements in each column is zero. Based on these A_k is often called as the Kirchhoff matrix of the CRN.

Using the notation

$$M = Y \cdot A_k, \tag{6}$$

eq. (3) becomes

$$\dot{x} = M \cdot \psi(x) \tag{7}$$

where M contains the coefficients of the monomials in the polynomial ODE describing the time-evolution of the state variables.

2.4   Dynamically equivalent realizations of CRNs

The matrix pair (Y, A_k) is called a realization of a CRN described by M if the following conditions hold: eq. (6) is fulfilled while all the elements of Y are nonnegative integers, and A_k is a column conservation matrix having nonpositive diagonal and nonnegative off-diagonal elements.

It can be shown that multiple alternative realizations of a reaction network are possible since M can often be factorized on several ways into an (Y, A_k) form. If (Y^1, A_k^1) and (Y^2, A_k^2) exist as alternative realizations and

$$M = Y^1 \cdot A_k^1 = Y^2 \cdot A_k^2 \tag{8}$$

is fulfilled, then these realizations are called dynamically equivalent.

Based on these, one can define structural properties such as minimal or maximal number of nonzero elements in matrix A_k and search for alternative representations of the given network which fulfill these criteria.

In this particular case, we want to determine the sparse and dense realizations denoted by (Y^s, A_k^s) and (Y^d, A_k^d), respectively. We call a realization *sparse* if the number of the non-zero elements in matrix A_k^s is minimal. A realization is called *dense* if it contains the maximal number of non-zero elements in matrix A_k^d. It was shown in [11] that the dense realization is a unique superstructure containing all mathematically possible reactions.

It should be noted, that while the dense realization is necessarily unique the sparse realization may not be unique: possibly several different sparse structures and/or parameter set can represent the same dynamical behaviour.

In general the alternative realizations may contain complexes that do not appear in the original network. In this paper we restrict the focus of our search to alternative CRNs where the set of complexes are the subset of the set of complexes of the original network.

Example   To illustrate the notion of dynamical equivalence on a small-scale example, we will use a classical 3-dimensional kinetic Lorenz system that was introduced in [27]. The classical Lorenz system can be transformed to kinetic form through coordinates-shifting and an appropriate time-scaling (see [27] for the computation details). After

these transformations, the kinetic ODEs of the system are

$$
\begin{aligned}
\dot{x}_1 &= \sigma x_1 x_2^2 x_3 - \sigma x_1^2 x_2 x_3 + \sigma(w_1 - w_2)x_1 x_2 x_3 \\
\dot{x}_2 &= (\rho + c_3)x_1^2 x_2 x_3 + (w_2 - w_1\rho - w_1 w_3)x_1 x_2 x_3 - x_1 x_2^2 x_3 - x_1^2 x_2 x_3^2 + w_1 x_1 x_2 x_3^2 \quad (9) \\
\dot{x}_3 &= x_1^2 x_2^2 x_3 - w_2 x_1^2 x_2 x_3 - w_1 x_1 x_2^2 x_3 + (w_1 w_2 + \beta w_3)x_1 x_2 x_3 - \beta x_1 x_2 \bar{x}_3^2
\end{aligned}
$$

where $\sigma = 10, \rho = 28, \beta = 8/3$, and $W = [w_1 \ w_2 \ w_3] = [24 \ 25 \ 26]$. It is shown in [27] that using these parameters, (9) shows chaotic behaviour with an attractor that is very similar to the attractor of the classical non-kinetic Lorenz system.

The complex composition matrix of the system is given by:

$$
Y^l = \begin{bmatrix}
1 & 0 & 2 & 1 & 2 & 1 & 1 & 1 & 2 & 2 & 2 & 1 & 2 \\
1 & 1 & 1 & 2 & 2 & 0 & 1 & 2 & 1 & 0 & 1 & 2 & 2 \\
1 & 1 & 1 & 1 & 1 & 1 & 2 & 2 & 2 & 2 & 0 & 0 & 2
\end{bmatrix},
$$

where the ith column contains the composition of complex C_i (i.e. $C_1 = X_1 + X_2 + X_3$, $C_2 = X_2 + X_3$ etc.). The non-zero off-diagonal elements of the network's Kirchhoff matrix A_k^l are the following:

$[A_k^l]_{2,1} = 679.3324, \ [A_k^l]_{6,1} = 1940.3342, \ [A_k^l]_{13,1} = 669.3342, \ [A_k^l]_{11,3} = 59, \ [A_k^l]_{12,3} = 10,$
$[A_k^l]_{13,3} = 44, \ [A_k^l]_{10,4} = 0.5, \ [A_k^l]_{12,4} = 34, \ [A_k^l]_{13,4} = 9.5, \ [A_k^l]_{13,5} = 1, \ [A_k^l]_{8,7} = 22.6666,$
$[A_k^l]_{12,7} = 1.3334, \ [A_k^l]_{10,9} = 1.$

Then the monomial coefficient matrix can be written as

$$
M = Y^l \cdot A_k^l = \begin{bmatrix}
-10 & 0 & -10 & 10 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\
-1271 & 0 & 54 & -1 & 0 & 0 & 24 & 0 & -1 & 0 & 0 & 0 & 0 \\
669.3342 & 0 & -25 & -24 & 1 & 0 & -2.6667 & 0 & 0 & 0 & 0 & 0 & 0
\end{bmatrix}.
$$

The graph representation of this network can be seen in Fig. 1a. An alternative sparse and dense realization can be seen in Figs 1b and 1c, respectively. The reaction rates of the sparse realizations are depicted as edge weights in Figs 1a and 1b. In Fig. 1c, the structure of the dense realization can be seen, while the non-zero off-diagonal elements of the corresponding Kirchhoff matrix A_k^d (containing 51 reactions) are the following

$[A_k^d]_{2,1} = 679.6342, \ [A_k^d]_{3,1} = 0.1, \ [A_k^d]_{4,1} = 0.1, \ [A_k^d]_{5,1} = 0.1, [A_k^d]_{6,1} = 602.3658, \ [A_k^d]_{7,1} = 0.1,$
$[A_k^d]_{8,1} = 0.1, \ [A_k^d]_{9,1} = 0.1, \ [A_k^d]_{10,1} = 669.1342, \ [A_k^d]_{11,1} = 0.1, [A_k^d]_{12,1} = 0.1, \ [A_k^d]_{13,1} = 0.1,$
$[A_k^d]_{1,3} = 0.1, \ [A_k^d]_{2,3} = 0.1, \ [A_k^d]_{4,3} = 0.1, \ [A_k^d]_{5,3} = 44.6, \ [A_k^d]_{6,3} = 0.1, \ [A_k^d]_{7,3} = 0.1,$
$[A_k^d]_{8,3} = 0.1, \ [A_k^d]_{9,3} = 0.1, \ [A_k^d]_{10,3} = 0.1, \ [A_k^d]_{11,3} = 16.2, \ [A_k^d]_{12,3} = 9.3, \ [A_k^d]_{13,3} = 0.1,$
$[A_k^d]_{1,4} = 0.1, \ [A_k^d]_{2,4} = 0.1, \ [A_k^d]_{3,4} = 0.1, \ [A_k^d]_{5,4} = 9.6, \ [A_k^d]_{6,4} = 0.1, \ [A_k^d]_{7,4} = 0.1,$
$[A_k^d]_{8,4} = 0.1, \ [A_k^d]_{9,4} = 0.1, \ [A_k^d]_{10,4} = 0.1, \ [A_k^d]_{11,4} = 0.1, \ [A_k^d]_{12,4} = 24.4, \ [A_k^d]_{13,4} = 0.1,$
$[A_k^d]_{13,5} = 1, \ [A_k^d]_{1,7} = 0.1, \ [A_k^d]_{2,7} = 0.6, \ [A_k^d]_{3,7} = 0.1, \ [A_k^d]_{4,7} = 0.1, \ [A_k^d]_{5,7} = 0.1,$
$[A_k^d]_{6,7} = 0.1, \ [A_k^d]_{7,7} = -25.4, [A_k^d]_{8,7} = 23.2166, \ [A_k^d]_{9,7} = 0.1, \ [A_k^d]_{10,7} = 0.1, \ [A_k^d]_{11,7} = 0.1,$
$[A_k^d]_{12,7} = 0.68335, \ [A_k^d]_{13,7} = 0.1, \ [A_k^d]_{10,9} = 1.1, \ [A_k^d]_{13,9} = 0.1.$
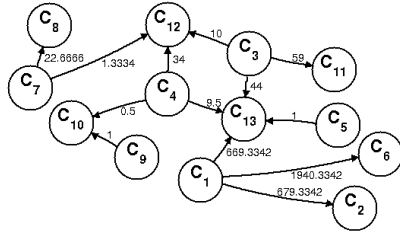
(a) A possible sparse CRN structure realizing (9). The corresponding matrices are (Y^l, A_k^l). Reaction rates appear as edge weights.
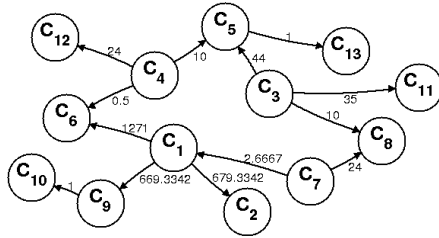
(b) Another possible sparse CRN structure realizing (9). Reaction rates appear as edge weights.
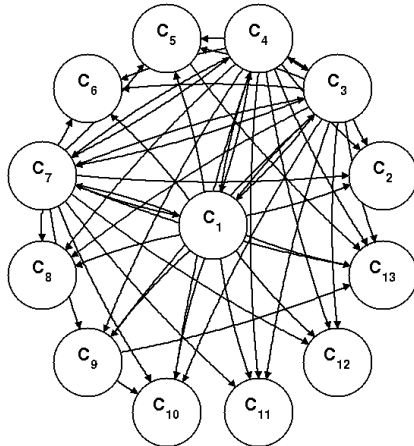
(c) The dense realization of (9). Edge weights are listed in A_k^d.

Figure 1: Alternative realizations of a CRN defined in Example 2.4.

2.5 Optimization framework for computation of dynamically equivalent structures

The original method of computing alternative realizations is based on the natural formulation of the problem as a MILP problem [28]. However, some alternative LP-based methods (see e.g. [19] or [29]) have also been published in the literature recently. We will shortly review these known solutions to be able to compare them with our new methods.

2.5.1 Mixed Integer Linear Programming approach

An MILP problem can be stated as follows [28]:

$$
\begin{cases}
\min_{x} c^T x \\
Ax \leq b \\
x \geq 0 \\
x_i \in \mathbb{R}, \ i = 1, ..., k \\
x_j \in \mathbb{Z}, \ j = k+1, ..., l
\end{cases}
\tag{10}
$$

where x is the l-dimensional vector of decision variables consisting of k real and $l - k$ integer elements. Matrix A and vector b define the linear inequality constraints. With the above formulation, equality constraints can also be treated by rewriting the problem to contain purely inequality constraints [20]. The linear function $c^T x$ with $c \in \mathbb{R}^l$ is the objective function to be minimized. If all decision variables are real (i.e. $k = l$), then eq. (10) defines a standard linear programming (LP) problem.

Now let us formulate the CRN alternative realization problem as an MILP. This formulation has also a crucial role in one of the new LP-based methods presented later in Section 3.2. We will represent the Kirchhoff matrix defined in Section 2.3 as:

$$
A_k = \begin{bmatrix}
-a_{11} & a_{12} & \cdots & a_{1m} \\
a_{21} & -a_{22} & & a_{2m} \\
\vdots & & & \vdots \\
a_{m1} & a_{m2} & \cdots & -a_{mm}
\end{bmatrix}.
\tag{11}
$$

The Kirchhoff property of A_k can be expressed by the following linear constraints:

$$
\sum_{i=1}^{m} [A_k]_{ij} = 0, \quad j = 1, \ldots, m
\tag{12}
$$

$$
[A_k]_{ij} \geq 0, \quad i, j = 1, \ldots, m, \quad i \neq j
\tag{13}
$$

$$
[A_k]_{ii} \leq 0, \quad i = 1, \ldots, m, .
\tag{14}
$$

Let us call eq. (6) and eq. (12)-(14) altogether as *kinetic constraints* because they clearly characterize the kinetic system structure of the model, and they can be considered as a constraint set in an optimization problem.

To formulate a MILP structure, where the real-valued decision variables corresponds to the off-diagonal elements of A_k, the following additional bounds for these elements are given:

$$[A_k]_{ij} \leq l_{ij}, \quad i, j = 1, \ldots, m, \quad i \neq j \tag{15}$$

$$l_{ii} \leq [A_k]_{ii}, \quad i = 1, \ldots, m. \tag{16}$$

where l_{ij} for $i, j = 1, \ldots, m$ are appropriately chosen bounds (with sufficiently large absolute values). Then the nonzero property of the individual reaction rate coefficients can now be written as

$$\delta_{ij} = 1 \leftrightarrow [A_k]_{ij} > \epsilon, \quad i, j = 1, \ldots, m, \quad i \neq j. \tag{17}$$

where δ_{ij} are binary decision variables, '$\leftrightarrow$' denotes the 'if and only if' relation, and ϵ is a small positive number used for distinguishing between practically zero and non-zero values.

The linear inequalities corresponding to eq. (17) can be translated to the following linear inequalities [28]

$$0 \leq [A_k]_{ij} - \epsilon \delta_{ij}, \quad i, j = 1, \ldots, m, \ i \neq j \tag{18}$$

$$0 \leq -[A_k]_{ij} + l_{ij} \delta_{ij}, \quad i, j = 1, \ldots, m, \ i \neq j. \tag{19}$$

Finally, minimization or maximization of the objective function

$$C(\delta) = \sum_{\substack{i,j=1 \\ i \neq j}}^{m} \delta_{ij} \tag{20}$$

leads to the computation of sparse or dense realizations, respectively.

It is important to mention that the solution of MILP problems is generally NP-hard, hence MILP problems are usually solved by computationally very intensive heuristics-driven techniques which are sometimes unreliable in case of a large-scale problem. In summary, we can conclude that MILP problems do not scale well to many networks which are large enough to describe complex real-life systems. Meanwhile, pure LPs can be solved in polynomial time.

2.5.2 Computing sparse realizations with a known iterative LP-based technique

In [29], an LP-based algorithm is presented to search for sparse linear models of gene regulation networks. The algorithm uses an iterative method to approximate the elements of the A_k matrix. The approach is quite efficient: as it is mentioned in [29], the method usually doesn't need more than a few (less than 10) iterations to converge.

We briefly describe how to apply this method for the computation of sparse CRN realizations. Let A_k^p denote the A_k in the pth step. Firstly set all off-diagonal elements of A_k^0 to 1. Let us define a weight w_{ij}^p for each element of A_k^p and initialize them as $w_{ij}^p = 1$ for each $i, j = 1, \ldots, m$. In the pth iteration step, let us recalculate the weights w_{ij}^p as follows:

$$w_{ij}^p = \frac{\beta}{\beta + |[A_k]_{i,j}^{p-1}|} \quad \text{for } i, j = 1, \ldots m \tag{21}$$

for an appropriate $\beta > 0$ value. The off-diagonal elements of A_k^p are obtained by solving a linear programming problem constrained by eq. (6), eq. (12)-(14) with the following objective function:

$$J = \min \sum_{i,j=1}^{m} w_{ij}^p |[A_k]_{i,j}^p| \tag{22}$$

The incorporated *kinetic constraints* ensure that in each iteration the obtained matrix A_k^p represents a dynamically equivalent realization and it has the Kirchoff property. The algorithm repeats eq. (22)-(21) until the objective value J will not change significantly between two successive steps. At the end of the iteration process the algorithm returns with A_k^p obtained in the last iteration step as a sparse realization of the network. In the following, we will refer to this algorithm as *Iterative LP*.

2.5.3 Computing dense realizations using multiple LP steps

This method was published in [19], and it can find dense realizations in polynomial time. The method uses $m \cdot (m-1)$ LP computation steps to generate the result. We will use the method for comparison with our new approach, therefore we summarize it for convenience (more details can be found in [19]).

For each $p, q = 1, \ldots m, p \neq q$, solve the optimization problem:

$$\max f_{pq} = [A_k]_{p,q} \tag{23}$$

with respect to the eq. (6), eq. (12)-(14) together with appropriate bounds eq. (15)-(16). The reaction $C_q \to C_p$ is present in the dense realization only if $\max f_{pq} > 0$. Let us

denote the solution corresponding to $(p, q), p \neq q$ by $\bar{A}_k^{pq}$. Now we will use these solutions to compute the final optimization step to generate the dense realization using the following quantities:

$$\epsilon_{ij} = \left[\frac{1}{m(m-1)} \sum_{p,q=1; p \neq q}^{m} \bar{A}_k^{pq} \right]_{i,j}, i \neq j \tag{24}$$

As we can see, $\epsilon_{ij} \geq 0 \, \forall i \neq j$ and $\epsilon_{ij} > 0$ *iff* the reaction $C_j \rightarrow C_i$ is in the dense realization. By solving the following LP feasibility problem for A_k with arbitrary linear cost function, the dense realization can be obtained:

$$\begin{cases} Y \cdot A_k = M \\ \sum_{i=1}^{m} [A_k]_{i,j} = 0 & j = 1, \dots m \\ \epsilon_{ij} \leq [A_k]_{i,j} \leq U_{i,j} & i,j = 1, \dots m, \, i \neq j \\ [A_k]_{i,i} \leq 0 & i = 1, \dots m \end{cases} \tag{25}$$

where $U_{i,j}$ are proper upper bounds with sufficiently large positive value. In the following, we will refer to this algorithm as *Element-wise LP*.

3   Improved methods for computing CRN structures

In this Section two new methods will be proposed that are fast, LP-based techniques to generate sparse and dense realizations of CRNs.

3.1   Computing sparse realizations

The results in [30, 31] show that in the case of large size, underdetermined system of linear equations (even in the case of non-negative decision variables) formulated as $Ax = b$, the l^1-norm based minimization can produce a sparse solution out of the infinitely many possible solutions of the problem. In this context, sparse solution means that the given x vector contains the minimal number of non-zero elements. In other words, under specific conditions, the combinatorial optimization problem of finding the sparse realization can be efficiently approached as a convex optimization problem. This is possible if the solution vector x with length n is sparse enough: it should not have more than $\rho \cdot n$ nonzero elements. In [32] an empirical limit $\rho = 0.3$ is suggested, we also used this value.

The l^1 minimization method is applied in our case as follows. Recall that in a given CRN there are n species and m complexes. The equality constraints of the optimization problem are the *kinetic constraints* (see eq. (6), (12)-(14)). For a single column of A_k containing m elements, the number of kinetic constrains will be $n + 1$. Hence, in case of

$n + 1 < m$ the emerging equality-type constraints as a system of linear equations remains underdetermined. Therefore, a column-wise l^1-norm based minimization is completed and the resulting vectors are considered as the column vectors of the sparse realization of the CRN:

$$\min \sum_{i=1}^{m} abs([A_k]_{i,j}) \ for \ \forall j = 1, \ldots m, i \neq j \tag{26}$$

Let us denote the ratio of non-zero and zero elements in $[A_k]_{.,j}$ as τ. If $\tau < \rho \cdot m$, then the minimization successfully finds the sparse solution. In the following, we will refer to this algorithm as the l^1-norm based algorithm.

3.2   Computing dense realizations

The proposed method is based on the construction of the MILP problem formulation given in sub-section 2.5.1. The main idea was to formulate the problem by relaxing the integrality constraints and then solve the remaining LP problem. By maximizing the sum of the relaxed auxiliary variables, the number of directed edges (i.e. the number of non-zero off-diagonal elements of A_k) is maximized, too, in the reaction graph of the CRN.

The constraint matrices were built up again using the *kinetic constraints*. Similarly to the MILP case, a set of auxiliary variables were defined: a σ_i variable for each state variable x_i. All these auxiliary variables are real valued. The constraints keeping track of nonzero reaction rate coefficients are given by

$$\sigma_i = 1 \longleftrightarrow x_i > \epsilon_e \tag{27}$$

where ϵ_e is a minimal (naturally positive) edge weight under which the edge is excluded from the network (i.e. the corresponding reaction rate coefficient is considered zero). After the relaxation of the integrality constrains the relation in eq. (27) becomes:

$$\epsilon \cdot \sigma_i \leq x_i \tag{28}$$

where $\sigma_i \in [0; 1]$ and $\epsilon > 0$ is a sufficiently small number, but $\epsilon > \epsilon_e$. By considering ϵ as a scaling factor, after short reformulation we obtain:

$$-x_i + \sigma_i \leq 0 \tag{29}$$

$$\sigma_i \in [0; \epsilon] \tag{30}$$

where eq. (29) is formulated as a constraint and eq. (30) is formulated as lower and upper bounds in the LP problem.

Now the optimization problem is defined as follows:

$$\max \sum_{i=1}^{m} \sigma_i \tag{31}$$

$$\begin{cases} Y \cdot A_k = M \\ \sum_{i=1}^{m} [A_k]_{i,j} = 0 & j = 1, \dots m \\ 0 \leq \sigma_i \leq \epsilon & i = 1, \dots m \\ -x_i + \sigma_i \leq 0 & i = 1, \dots m \end{cases} \tag{32}$$

Let us show that the solution of the above optimization problem indeed determines the dense network: suppose that a given edge (corresponding to the decision variable x_i) can be present in the network which means that there is no such kinetic constraint which forbids that x_i could be larger than zero. In this case, according to the constraint formulated in eq. (29), $\sigma_i > 0$ will occur which yields $x_i \geq \epsilon$. Because of $\epsilon > \epsilon_e$, the edge will be present in the network. In the opposite case, when the edge represented by x_i should be excluded from the network according to the kinetic constraints, $x_i = 0$ leading to $\sigma_i = 0$ according to eq. (29). As a result of the maximization, a dense realization with edge weights larger than ϵ_e is obtained. In the following, we will refer to this algorithm as the *LP-MAX* algorithm.

4   Results

Both the l^1-norm based search and the *LP-MAX* methods were involved in a comparative study in which extensive simulations were completed to evaluate the performance of the proposed methods. Large scale problems were investigated to present the capability of dealing with biologically relevant problems, too. The MILP-based methods presented in Section 2.5.1 and the LP-based methods presented in Section 2.5.2 and Section 2.5.3 were used as a basis of comparison. To solve the LP problems, we used the CLP solver [33] from the COIN-OR community. The formulated MILP problems were solved by the DIP solver [34] from the COIN-OR community [35] because it can solve these type of problems quite efficiently. All the simulations were done on a 8-core 3.0GHz computer. The computational problems were handled in MATLAB environment with the help of the CRNReals Toolbox [36].

4.1 Computing alternative realizations of a CRN describing a classical 3-dimensional Lorenz system

Using the 3-dimensional Lorenz system introduced in Section 2.4, the validation of the proposed methods on a small-scale CRN can be completed. It is known from [27] that this system can be represented by CRNs having 3 species and 13 complexes with sparse realizations containing 13 off-diagonal non-zero elements, and its dense realization contains 51 off-diagonal non-zero elements.

Computing the sparse realization took 17.172 seconds in case of the MILP based method. *Iterative LP* were able to find a valid sparse realization in 0.0144 seconds while the proposed l^1-norm based sparse search can complete this task in 0.0166 seconds.

The dense realization was also successfully computed by all three algorithms. The original MILP based algorithm completed it in 5.3477 seconds, *Element-wise LP* consumed 0.2524 seconds, and the *LP-MAX* algorithm needs 0.0446 seconds to compute the solution.

4.2 Performance analysis of the proposed algorithms on large random networks

All the methods were tested on several large, randomly generated CRNs to be able to test their performance and time consumption of the solution. n as the number of species and m as the number of complexes were defined as parameters. First, a polynomial representation of the system with the given parameters was generated. The coefficient matrix $M_c \in \mathbb{R}^{n \times m}$ of the monomials of the polynomial system is a random matrix where $10 \leq [M_c]_{i,j} \leq 110, \forall i = 1, ..., n; j = 1, ..., m$. Rows of the exponent matrix $B \in \mathbb{R}^{m \times n}$ correspond to monomials of the system, namely row i stores the exponents of the i-th monomial. The system (M, B) is converted to a so-called canonical CRN representation (Y, A_k) as it is described in [37].

Altogether more than 100 different scenarios were used to examine the performance of the proposed algorithms. The networks used during these tests were different both in their structure and in their size. In the following, the numerical results of some scenarios are presented. More results can be found in an electronic supplement at
http://daedalus.scl.sztaki.hu/PCRG/works/CRN_Alter_Struct.zip.

In all cases the following numerical values were used: reactions having reaction rate smaller than 10^{-6} were omitted from the networks. Let us consider two different realiza-

Number of complexes	Number of reactions in the sparse realization		
	l^1-norm based	MILP-based	*Iterative LP*
220	200	200	200
330	300	300	300
440	400	400	400
550	500	500	500
	Computational time (s)		
220	0.49	603.32	29.16
330	1.55	1597.21	134.17
440	3.35	2784.53	447.85
550	4.22	4330.62	1028.59

Table 1: Comparing the LP- and MILP-based methods while searching for the sparse realization. The size of the computational problem grows as the number of complexes increases.

tions of the same CRN, namely (Y, A_k^1) and (Y, A_k^2) and define the R matrix as follows: $R = \text{abs}(Y \cdot A_k^1 - Y \cdot A_k^2)$. These two realizations were considered as dynamically equivalent representations if $R < 10^{-3}$.

In Table 1, the summary of the search for the sparse realization can be found. Each row represents a different problem size: the number of species was fixed to 10 but the number of complexes was increased to enlarge the computational task. In the first row block the number of off-diagonal non-zero elements can be found in the resulting A_k matrices. One can see the match in the values which implies that all the algorithms successfully found the realization containing the minimal number of reactions. In the second row block, the running time of the algorithms can be seen while evaluating the previously presented scenarios. In Table 2, the result of the search for the dense superstructure is summarized. The structure of the table is similar to Table 1. To summarize the above presented results we can say, that the presented new, LP-based algorithms successfully compute both the dense and sparse realizations while outperform all the previously presented methods. With the help of these methods the computation of the alternative realizations of biologically relevant, large size networks can be done efficiently.

4.3 Computing alternative realizations of the *ErbB* network

As it was mentioned before, we would like to use our methods to study the possible structures of large scale, biologically relevant networks. As a case study the *ErbB* network described in [38] was investigated. The *ErbB* signalling pathways regulate several phys-

Number of complexes	Number of reactions in the dense realization		
	LP-MAX	MILP-based	*Element-wise LP*
220	4380	4380	4380
330	10528	10528	10528
440	18438	18438	18438
550	30936	30936	30936
	Computational time (s)		
220	6.25	1413.16	156.41
330	35.53	2251.98	703.77
440	66.71	3828.73	1736.76
550	304.75	5422.87	4093.58

Table 2: Comparing the LP- and MILP-based methods while searching for the dense realization. The size of the computational problem grows as the number of complexes increases.

iological responses such as cell survival, proliferation and motility. The malfunction or hyperfunction of these pathways are involved in the explanation of various types of human cancers. These types of pathways are under active examination as possible drug targets.

In our representation the *ErbB* signalling pathway model consists of 504 species, 1082 complexes and 1654 reactions. The model description was originally a sparse representation. With the help of the *LP-MAX* algorithm introduced in Section 3.2 the dense realization is computed. The algorithm using the MILP approach was unable to complete the optimization within reasonable time. The resulting dense realization contains 1683 reactions: 29 mathematically possible extra reactions originating from 15 different complexes compared to the published model. The overall computational time was 4993 seconds. The extra reactions introduced into the network can be seen in Fig. 2. The list of the extra reactions can be found in Table 3. The notations of the complexes in Table 3. are the same as in the original model description published at `http://www.ebi.ac.uk/compneur-srv/biomodels-main/publ-model.do?mid=BIOMD0000000255`.

The sparse realization was also extracted from the dense realization with the help of the l^1-norm based sparse search. The resulting network had the same structure as the original sparse representation. The computational time was around 430 seconds.

	Starting complex → ending complex
1.	(ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K + PIP3 → PIP3 + (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:PIP2
2.	(ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:PIP2 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K → PIP3 + (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:PIP2
3.	PIP3 + (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:PIP2 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:PIP2 → PIP3 + (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)2
4.	(ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)2 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:PIP2 → PIP3 + (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)2
5.	PIP3 + (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)2 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)2 → PIP3 + (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)3
6.	(ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)3 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)2 → PIP3 + (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)3
7.	PIP3 + (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)3 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)3 → PIP3 + (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)4
8.	(ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)4 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)3 → PIP3 + (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)4
9.	PIP3 + (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)4 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)4 → PIP3 + (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)5
10.	(ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)5 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)4 → PIP3 + (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)5
11.	(ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K + PIP2 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:PIP2 + PIP2
12.	(ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:PIP2 + PIP2 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:PIP2 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)2 + PIP2
13.	(ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)2 + PIP2 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)2 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)3 + PIP2
14.	(ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)3 + PIP2 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)3 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)4 + PIP2
15.	(ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)4 + PIP2 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)4 → (ErbB3:ErbB2)_P:GAP:Grb2:Gab1_P:PI3K:(PIP2)5 + PIP2

Table 3: List of all the extra reactions in the dense realization of the *ErbB* signaling pathway network. 29 mathematically possible extra reactions originating from 15 different complexes were found.
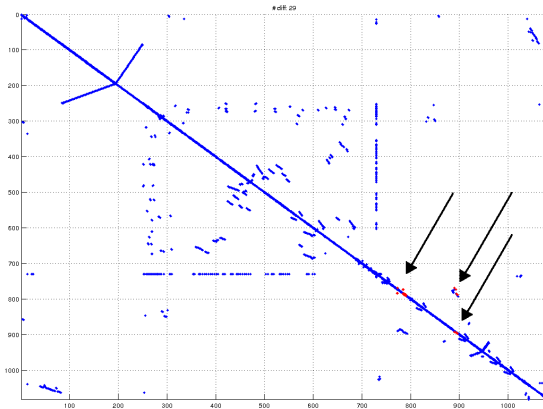
Figure 2: Structure of the A_k matrix representing the dense realization of the *ErbB* network. The differences between the sparse and dense realizations are coloured and marked by the arrows.

4.4 Possible parallelization

As the size of the emerging computational problem grows during the search for the alternative realizations, the need for possible parallelization is increasing. Fortunately, as the reader can notice, all the proposed algorithms can be parallelized easily because all the optimization problems are built up by taking the A_k matrix column-wise. Moreover all the results of the optimizations are incorporated into the final result independently of each other.

This means that in case of having an A_k matrix with size $m \times m$ the search for the alternative realizations (both in case of sparse and dense realizations) can be split up into m independent, parallel solvable problems. This gives the opportunity to gain even more speedup in the solution process.

5 Conclusion

In the present paper linear programming based methods were presented to compute alternative realizations of CRNs. By analysing the properties of the system model and the MILP based description, simplified algorithms were developed which have polynomial complexity.

The classical, combinatorial optimization based techniques were compared to the pro-

posed, Linear Programming based methods to show the correctness of them. It is shown that these methods successfully solve the problem while the possible computational problems which could emerge in case of an MILP problem are avoided. The proposed techniques give the opportunity to find dense and sparse realizations of large scale networks in limited time which enables us to deal with real biological systems. The algorithms can be easily applied in a parallel framework, too.

In the future, the LP-based methods can be further developed to be able to incorporate additional constraints constructed for the state variables and to handle additional prescribed properties regarding the structure of the generated network. With the help of these developments it would be possible to eliminate the biologically meaningless networks from the set of mathematically possible solutions.

References

[1] P. Érdi, J. Tóth, *Mathematical Models of Chemical Reactions. Theory and Applications of Deterministic and Stochastic Models*, Manchester Univ. Press, Manchester, 1989.

[2] W. M. Haddad, V. S. Chellaboina, Q. Hui, *Nonnegative and Compartmental Dynamical Systems*, Princeton Univ. Press, 2010.

[3] N. Samardzija, L. D. Greller, E. Wassermann, Nonlinear chemical kinetic schemes derived from mechanical and electrical dynamical systems, *J. Chem. Phys.* **90** (1989) 2296–2304.

[4] F. Horn, R. Jackson, General mass action kinetics, *Arch. Rat. Mech. Anal.* **47** (1972) 81–116.

[5] M. Feinberg, Chemical reaction network structure and the stability of complex isothermal reactors – I. The deficiency zero and deficiency one theorems, *Chem. Engin. Sci.* **42** (1987) 2229–2268.

[6] M. Feinberg, Chemical reaction network structure and the stability of complex isothermal reactors – II. Multiple steady states for networks of deficiency one, *Chem. Engin. Sci.* **43** (1988) 1–25.

[7] D. Angeli, A tutorial on chemical network dynamics, *Eur. J. Control* **15** (2009) 398–406.

[8] E. Sontag, Structure and stability of certain chemical networks and applications to the kinetic proofreading model of T-cell receptor signal transduction, *IEEE Transact. Autom. Control* **46** (2001) 1028–1047.

[9] G. Craciun, C. Pantea, Identifiability of chemical reaction networks, *J. Math. Chem.* **44** (2008) 244–259.

[10] G. Szederkényi, Comment on "Identifiability of chemical reaction networks" by G. Craciun and C. Pantea, *J. Math. Chem.* **45** (2009) 1172–1174.

[11] G. Szederkényi, Computing sparse and dense realizations of reaction kinetic systems, *J. Math. Chem.* **47** (2010) 551–568.

[12] G. Szederkényi, K. M. Hangos, Finding complex balanced and detailed balanced realizations of chemical reaction networks, *J. Math. Chem.* **49** (2011) 1163–1179.

[13] G. Szederkényi, K. M. Hangos, T. Péni, Maximal and minimal realizations of reaction kinetic systems: computation and properties, *MATCH Commun. Math. Comput. Chem.* **65** (2011) 309–332.

[14] G. Szederkényi, K. M. Hangos, Z. Tuza, Finding weakly reversible realizations of chemical reaction networks using optimization, *MATCH Commun. Math. Comput. Chem.* **67** (2012) 193–212.

[15] M. D. Johnston, D. Siegel, Linear conjugacy of chemical reaction networks, *J. Math. Chem.* **49** (2011) 1263–1282.

[16] M. D. Johnston, D. Siegel, G. Szederkényi, A linear programming approach to weak reversibility and linear conjugacy of chemical reaction networks, *J. Math. Chem.* **50** (2012) 274–288.

[17] M. D. Johnston, D. Siegel, G. Szederkényi, Dynamical equivalence and linear conjugacy of chemical reaction networks: new results and methods, *MATCH Commun. Math. Comput. Chem.* **50** (2012) 274–288.

[18] M. D. Johnston, D. Siegel, G. Szederkényi, Computing weakly reversible linearly conjugate chemical reaction networks with minimal deficiency, *Math. Biosci.* **241** (2013) 88–98.

[19] G. Szederkényi, J. R. Banga, A. A. Alonso, Inference of complex biological networks: distinguishability issues and optimization–based solutions, *BMC Sys. Biol.* **5** (2011) 177:1–15.

[20] J. W. Chinneck, *Practical Optimization: A Gentle Introduction*, Carleton Univ., Ottawa, 2009.

[21] G. B. Dantzig, M. N. Thapa, *Linear Programming 1: Introduction*, Springer–Verlag, New York, 1997.

[22] G. B. Dantzig, M. N. Thapa, *Linear Programming 2: Theory and Extensions*, Springer–Verlag, 2003.

[23] C. A. Floudas, *Nonlinear and Mixed–Integer Optimization*, Oxford Univ. Press, Oxford, 1995.

[24] R. Raman, I. E. Grossmann, Modelling and computational techniques for logic based integer programming, *Computers Chem. Engin.* **18** (1994) 563–578.

[25] A. Bemporad, M. Morari, Control of systems integrating logic, dynamics, and constraints, *Automatica* **35** (1999) 407–427.

[26] M. Feinberg, *Lectures on Chemical Reaction Networks*, Univ. Wisconsin, Madison, 1979.

[27] Z. A. Tuza, G. Szederkényi, K. M. Hangos, A. A. Alonso, J. R. Banga, Computing all sparse kinetic structures for a Lorenz system using optimization methods, *Int. J. Bifurcation Chaos*, in press.

[28] S. P. Bradley, A. C. Hax, T. L. Magnanti, *Applied Mathematical Programming*, Addison–Wesley, Reading, 1977.

[29] M. M. Zavlanos, A. A. Julius, S. P. Boyd, G. J. Pappas, Inferring stable genetic networks from steady–state data, *Automatica* **47** (2011) 1113–1122.

[30] D. L. Donoho, Compressed sensing, *IEEE Trans. Inform. Theory* **52** (2006) 1289–1306.

[31] D. L. Donoho, J. Tanner, Sparse nonnegative solutions of underdetermined linear equations by linear programming, *Proc. Nat. Acad. Sci.* **102** (2005) 9446–9451.

[32] D. L. Donoho, For most large underdetermined systems of linear equations the minimal L^1-norm solution is also the sparsest solution, *Commun. Pure Appl. Math.* **59** (2004) 797–829.

[33] *CLP – Coin-or linear programming*, https://projects.coin-or.org/Clp .

[34] M. V. Galati, T. K. Ralphs, DIP: A framework for decomposition in integer programming, Technical report, COR@L Laboratory, Lehigh Univ., 2012.

[35] *COmputational INfrastructure for Operations Research*, http://coin-or.org .

[36] G. Szederkényi, J. R. Banga, A. A. Alonso, CRNreals: a toolbox for distinguishability and identifiability analysis of biochemical reaction networks, *Bioinformatics* **28** (2012) 1549–1550.

[37] V. Hárs, J. Tóth, On the inverse problem of reaction kinetics, in: M. Farkas, L. Hatvani (Eds.), *Qualitative Theory of Differential Equations*, North–Holland, Amsterdam, 1981, pp. 363–379.

[38] W. W. Chen, B. Schoeberl, P. J. Jasper, M. Niepel, U. B. Nielsen, D. A. Lauffenburger, P. K. Sorger, Input–output behavior of ErbB signaling pathways as revealed by a mass action model trained against dynamic data, *Mol. Syst. Biol.* **5** (2009) 239:1–19.