DOI:10.15774/PPKE.ITK.2018.004

Pázmány Péter Catholic University

Doctoral Thesis

Structural analysis of biochemically motivated nonlinear systems

Author:
Bernadett Ács

Thesis Advisor:
Dr. Gábor Szederkényi, D.Sc.

A thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

in the

Roska Tamás Doctoral School of Sciences and Technology
Faculty of Information Technology and Bionics

2018

DOI:10.15774/PPKE.ITK.2018.004

DOI:10.15774/PPKE.ITK.2018.004

Abstract

Kinetic systems form a special class among nonnegative dynamical systems that are suitable to describe all kinds of dynamical behaviour. Kinetic systems can be applied for modelling various kinds of systems, such as biochemical reaction networks, electronic networks, transportation problems or the spreading of epidemics. The aim of this thesis is to study the structural properties of mass action kinetic systems and introduce novel algorithms for the computation of their realizations. During the computations the linear programming optimization framework is applied. It is proven that the dense linearly conjugate realization of a kinetic system with a fixed set of complexes and also fulfilling a finite set of additional linear constraints has the superstructure property. This plays an essential part in the presented methods. A polynomial-time algorithm is given for computing the dense linearly conjugate realization of a kinetic system that is significantly more efficient than the previously known methods. An already existing algorithm for determining a weakly reversible dynamically equivalent realization of a kinetic system is extended to the case of linearly conjugate realizations. Two methods are presented for computing all possible reaction graph structures representing linearly conjugate – and as special case dynamically equivalent – realizations of a kinetic system. Both methods are suitable for parallel implementation. A generalized form of kinetic systems is introduced that allows uncertain parameters and additional linear constraints as well. The methods for computing a dense realization, the set of core reactions and all possible reaction graph structures are extended to this model as well. The correctness of each presented algorithm is proven, and their working is demonstrated on several examples.

DOI:10.15774/PPKE.ITK.2018.004

Acknowledgements

First of all, I would like to thank my thesis advisor Prof. Gábor Szederkényi for introducing me into this interesting topic and guiding me through its various and details by always giving me such problems that I was actually able to solve. I am also very grateful for his patience and understanding during our work together.

I would like to thank Prof. Zsolt Tuza for the opportunity of working with him. He has shown us some of his beautiful ideas, and beside that he has improved the presentation of our results considering even the smallest details, from which I have learnt a great deal.

I am thankful for the collaboration with former and present members of our group, especially to Dr. Zoltán A. Tuza, Dr. Dávid Csercsik and Gergely Szlobodnyik for designing implementations for the algorithms introduced in this thesis. Without their work these would remain only theoretical results.

I am grateful to Prof. Katalin M. Hangos for the chance of taking part in and in some cases even delivering talks on seminars organized by her group at SZTAKI. It was not just a great opportunity for observing and participating, but her comments have also made a great impact on my documents as well as my presentation skills.

I would also like to thank Dr. János Tóth for inviting me to seminars organized by him at BME. It was a great opportunity to present our results as well as to meet interesting people and new ideas.

I am really thankful to Dr. Ágnes Bércesné Novák for encouraging me to apply to the doctoral school. She has opened up new perspectives to me.

I am also very thankful to my family for their patience and support in various fields of life during the last four years.

I am grateful to the Roska Tamás Doctoral School for accepting and supporting me during my road to graduation. I would like to specially thank Tivadarné Vida for kindly giving me all the help with administrative problems.

I would also like to acknowledge the support of the grants OTKA 104706, KAP 1.1 14/029, 1.1 15/052, 16-71005-1.1 and 17-61008-1.2. The doctoral school was also supported by the European Union, co-financed by the European Social Fund (EFOP-3.6.3-VEKOP-16-2017-00002).

DOI:10.15774/PPKE.ITK.2018.004

Contents

DOI:10.15774/PPKE.ITK.2018.004

DOI:10.15774/PPKE.ITK.2018.004

Notations and Symbols

$\mathbb{R}$	the set of real numbers
$\mathbb{R}_+$	the set of nonnegative real numbers
$\mathbb{N}$	the set of natural numbers, including 0
$\mathbb{R}^n$	the n-dimensional Euclidean space
$\mathbb{R}^n_+$	the nonnegative orthant in $\mathbb{R}^n$
$\lvert H\rvert$	the number of elements in the set H
$H^{n\times m}$	the set of matrices having entries from a set H with n rows and m columns
$[M]_{ij}$	the entry of a matrix M with row index i and column index j
$[M]_{.j}$	the column j of a matrix M
$vec(M)$	the concatenation of the columns of the matrix M $vec(M) = [[M]_{.1}^\top, \ldots, [M]_{.m}^\top]^\top$
e_i^n	the ith unit vector in $\mathbb{R}^n$, where the coordinate i is equal to 1 and all other coordinates are zero
$\mathbf{0}^n$	the zero vector in $\mathbb{R}^n$
$\mathbf{1}^n$	the vector in $\mathbb{R}^n$ with all coordinates equal to 1
$\mathbf{0}^{n\times m}$	the zero matrix in $\mathbb{R}^{n\times m}$
$\mathbf{I}^n$	the unit matrix in $\mathbb{R}^{n\times n}$
$\mathbf{0}_q$	the binary sequence of length q with all coordinates equal to zero
$\mathbf{1}_q$	the binary sequence of length q with all coordinates equal to 1
$sgn(x)$	the sign function
$\mathcal{S}$	the set of species of a CRN
$\mathcal{C}$	the set of complexes of a CRN
$\mathcal{R}$	the set of reactions of a CRN
$C_i \to C_j$	the reaction from complex C_i to complex C_j
(C_i, C_j)	the reference of reaction $C_i \to C_j$ in formulas
k_{ij}	reaction rate coefficient of the reaction $C_i \to C_j$
M	the coefficient matrix of a polynomial system
Y	the complex composition matrix of a CRN
A_k	the Kirchhoff matrix of a CRN
$\psi^Y(x)$	the monomial function of the CRN (Y, A_k) (independent from A_k)
T	positive definite diagonal transformation matrix
Φ_T	the transformation matrix of the monomial function φ $[\Phi_T]_{ii} = \varphi_i(T \cdot \mathbf{1}^p)$
Ψ_T	the transformation matrix of the monomial function ψ_Y of the CRN (Y, A_k) $[\Psi_T]_{ii} = \psi_i^Y(T \cdot \mathbf{1}^m)$

DOI:10.15774/PPKE.ITK.2018.004

A_b	the transformed Kirchhoff matrix of a linearly conjugate realization $A_b = A_k \cdot \Psi_T^{-1}$
$[M, Y]$	the kinetic system defined by the coefficient matrix M and the complex composition matrix Y
$[\mathcal{P}, L, Y]$	the uncertain kinetic system defined by the polyhedron $\mathcal{P}$ of uncertain parameters, the set L of additional linear constraints and the complex composition matrix Y
(Y, A_k)	the CRN characterized by the matrices Y and A_k
(T^{-1}, A_b)	the linearly conjugate realization defined by the matrices T^{-1} and A_b
$\mathcal{Q}$	the polyhedron representing the solutions of an optimization model
$\overline{\mathcal{Q}}$	the closure of the polyhedron $\mathcal{Q}$
$E_c[M, Y]$	the set of core edges of the kinetic system $[M, Y]$
$G(Y, A_k)$	the reaction graph of the CRN (Y, A_k)
$V(G)$	the set of vertices of the graph G
$E(G)$	the set of edges of the graph G
K_m	complete directed graph on m vertices
R	a realization and its representation as a binary sequence
D	the dense realization and its representation as a binary sequence
G_R	the reaction graph structure of the realization R
$e(R)$	the number of edges of the graph G_R

DOI:10.15774/PPKE.ITK.2018.004

Chapter 1

Introduction

The development and/or the maintenance of any kind of device or system requires some kind of knowledge about its possible states. To reveal the connections among properties of different events and to try to predict certain characteristics of the future quantitative mathematical models are successfully applied. These models usually describe only selected properties of the real process, but from an application point of view in most cases it is enough.

The operation of more complicated systems such as living organisms can often be described by complex phenomena, and for the modelling of quantities changing in space and/or in time dynamical systems are the most commonly applied tools. Therefore, this type of modelling has become an intensively studied and frequently applied tool in systems biology.

In many real life problems for example in economic systems, population dynamics or biochemical systems the variables are physically constrained to have only nonnegative values, and therefore the theory of nonnegative systems [1] needs to be applied for their characterization. A dynamical system is called nonnegative if its trajectories remain in the nonnegative orthant whenever the initial value is nonnegative. (If strict positivity is required then it is called a positive system.) A wide subclass of dynamical systems can be transformed into nonnegative form by shifting the coordinates into the nonnegative orthant and then in a further transformed version of the model the trajectories can be kept in a given region, see [2].

A more special class of nonnegative dynamical systems is formed by the quasi polynomial (QP) systems, which was first introduced in [3]. The author has also shown that most smooth dynamical models can algorithmically be transformed into QP form, which property makes such systems suitable for the modelling of dynamical systems belonging to a much wider class.

If the right hand side of the ordinary differential equations of the system can be given in the form of a multivariate polynomial as well, then it is called a polynomial system. The aim of this thesis is the structural and computational analysis of a certain type of nonnegative polynomial systems called kinetic systems, that can describe the dynamics

DOI:10.15774/PPKE.ITK.2018.004

of chemical reaction network (CRN) models obeying the mass action law. Despite the fact that kinetic systems are rather special polynomial systems, these models are versatile tools in modelling. Furthermore, by using suitable model transformations the majority of nonnegative dynamical systems can be transformed into kinetic form [3, 4].

The different types of dynamical systems and possible transformations between them are shown on the diagram in Figure 1.1. It has to be mentioned however, that the examination of model types different from kinetic systems and of the transformations between these models is not included in this thesis.

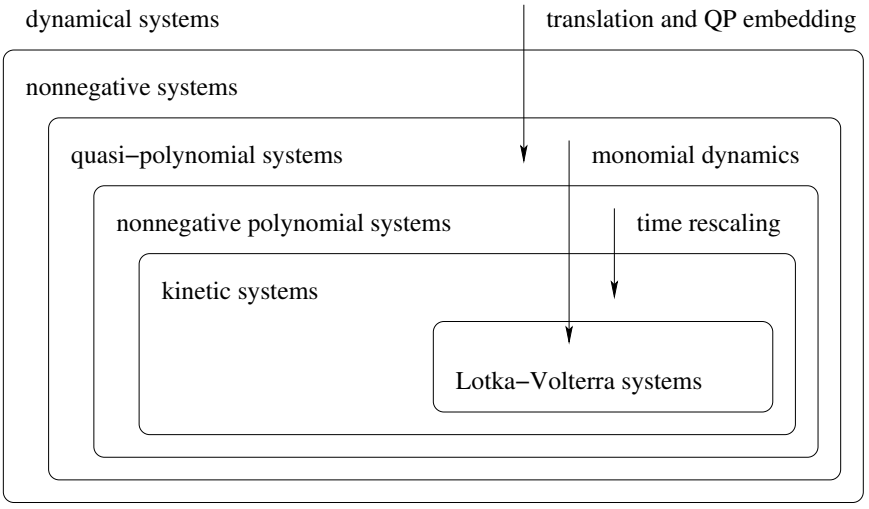


FIGURE 1.1: Classes and transformations of dynamic systems.

Chemical reaction networks obeying the mass action law can be originated from the dynamical modelling of chemical and biochemical processes, but they can be applied to describe various kinds of dynamical phenomena. Their applications appear in several different fields of science and engineering, such as the modelling of electrical networks, transportation problems or the spreading of epidemics, therefore these models are so-called universal descriptors [5, 2].

The class of kinetic systems is defined by chemical reaction network models, but for the verification of the kinetic property it is not necessary to compute a suitable CRN, it is enough to examine just the sign pattern of the monomial coefficients, see [6].

It is known that in general there are many realizations and different reaction graph structures corresponding to a given kinetic dynamics. This phenomenon is called macro-equivalence or dynamical equivalence [7, 8]. There is also a generalization of dynamical equivalence called linear conjugacy, where a positive definite diagonal linear transformation is applied to the state variables working as if the units of measurement were individually scaled [9]. It is easy to see that linear conjugacy preserves the kinetic property of the system and also the main qualitative dynamical properties like stability, multiplicities or the boundedness of solutions. However, due to the larger degree of

DOI:10.15774/PPKE.ITK.2018.004

freedom introduced by the transformation parameters, in general, it allows a larger set of possible structures compared to dynamical equivalence.

There is a widely applied structure oriented representation of CRNs that is a weighted directed graph called the Feinberg-Horn-Jackson graph. It depicts the reactions which are present in the network, and some other parameters of the network as well that are easier to describe with graph properties. Furthermore, in some cases there is a relation between the dynamics of the network and the reaction graph structure, without considering the actual reaction rates. This has become an important research area in chemical reaction network theory since the 1970s, see [7, 10]. In this topic there are several practice oriented results [11] as well as beautiful mathematical designs [12].

To determine a possible reaction network structure of a given kinetic system a symbolic method was proposed in [6]. Since this method returns only one particular dynamically equivalent realization called the canonical realization, a different approach must be applied to determine others.

Chemical reaction networks have a simple algebraic characterization, which makes it particularly appealing to develop computational methods for their dynamical and structural analysis [13, 1] or even control [14]. Realizations of a given kinetic dynamics can be defined by linear constraints, that suggests the application of linear optimization methods. Since this is in general a very simple model, several computational methods have already been developed to find linearly conjugate or, as a special case, dynamically equivalent realizations of kinetic systems [15, 16] and also having preferred properties such as density/sparsity [17, 18], maximal or minimal realizations [19], complex or detailed balance [20, 21], weak reversibility [22], [51] or minimum deficiency [23, 24].

The general form of these problems can be written as a linear programming (LP) model using only continuous decision variables. For solving an LP problem there are several polynomial time algorithms, the first provably correct solution is the Simplex Algorithm that was developed by Dantzig in 1947 [25, 26]. This algorithm works in most of the practical applications very efficiently, however in 1972 Klee and Minty gave an example, the so-called Klee-Minty cube, for proving that the simplex algorithm in the worst case might require exponential time [27].

In practical applications an algorithm is considered to be efficient if it runs in polynomial time, i.e. the number of required computation steps can be given as a polynomial function of the size of the input. These algorithms run in real time even in the case of larger inputs, while the required time of exponential methods increases very fast with the size of the input.

After the introduction of the simplex method several different algorithms have also been developed for the efficient solution of linear optimization problems, such as the criss-cross method or the ellipsoid method [28]. Despite that, in most application still the simplex method is used. It has to be mentioned though, that the efficiency of the chosen method can highly depend on the implementation.

In the cases of some special realizations the computation requires the application of integer and continuous variables at the same time, which transforms the model into a

DOI:10.15774/PPKE.ITK.2018.004

mixed integer linear programming (MILP) problem. This problem is known to be NP-complete, which means in practical applications that there is no efficient method for solving it. There are several approximative methods that include the solving of the the LP-relaxed version of the problem, but also exact methods such as the cutting plane method developed by Gomory [29] and its improved version, the Branch and Bound method proposed by Land and Doig in 1960 [30]. Despite the many possible solutions it is still desired to avoid the application of integer variables, since there are much more efficient methods for solving linear optimization problems defined on continuous variables.

In the case of several problems it makes a great difference if one applies an unusual approach. For example the problem of computing dense realizations can be formed at first sight as a MILP problem. For a long time in the literature there have been only non-polynomial time solutions or ones that work in polynomial-time but for special cases only. However, by the application of convex geometry it was possible to give a simple and efficient polynomial-time solution of this problem. The method works for any given kinetic system, even if a dense realization with some special property needs to be determined. This approach was applied also in the proof of the superstructure property of dense realizations, and the developed computational method was used as a subroutine in other algorithms presented in this thesis as well. The results are introduced in detail in [51] and in Chapter 3 of this thesis.

Weakly reversible realizations form an intensively studied class of CRN realizations where there is a connection between structure and dynamics. In the language of graph theory weak reversibility means that the components of the directed reaction graph are strongly connected. One of the most important results in this area is the Deficiency Zero Theorem [31, 10]. It says that a weakly reversible CRN having zero deficiency for any choice of positive reaction rate coefficients has exactly one locally asymptotically stable equilibrium point in every positive stoichiometric compatibility class. According to the Global Attractor Conjecture this stability is actually global (with respect to the positive orthant) not just for deficiency zero weakly reversible CRNs, but for a wider class of systems called complex balanced networks, see, e.g. [32, 33]. The Global Attractor Conjecture has been proven for one linkage class networks in [34], and recently a general proof was also proposed in [12].

Due to the importance of the weak reversibility property of CRN realizations there have been several attempts to design efficient computation methods for determining such realizations, using both algebraic [35, 36] and graph theory based solutions [22], MILP and LP programming methods. Based on the superstructure property it was possible to generalize the polynomial-time algorithm presented in [22] for computing linearly conjugate weakly reversible realizations of a given kinetic dynamics that applies the minimal necessary number of variables, and also to prove the correctness of this method. The results related to this topic are presented in [51] and in Chapter 4 of this thesis.

After treating the above mentioned important special cases a question arises naturally: Is it possible to give a computationally efficient algorithm for determining all possible

DOI:10.15774/PPKE.ITK.2018.004

reaction graph structures representing linearly conjugate CRN realizations of a given kinetic system?

Based on the idea of Prof. Zsolt Tuza it was possible to propose an algorithm for the complete generation of reaction graph structures representing linearly conjugate realizations of a given kinetic system. It is the first provably correct solution of the problem. Due to the possible large number of solutions one cannot expect to find a polynomial-time algorithm for the overall problem. But it can be shown that between the returning of two consecutive reaction graph structures the time elapsed is always polynomial. This method was published in [52]. The algorithm is suitable for parallel implementation as well, which might highly increase its efficiency. The improved computation method was proposed in [55], and all the results considering this algorithm are presented in detail in Section 5.1 of this thesis.

Later an other algorithm was developed as well for the characterization of the possible reaction graph structures. The great advantage of this method is that during the computation every existing reaction graph structure is returned exactly once, and compared to the other algorithm it works more efficiently. The results are demonstrated in [53] and in Section 5.2 of this thesis.

In some cases the dynamics of the system might not be precisely known, for example if the coefficients of the kinetic system are given as the results of some noisy measurements. For modelling such dynamics a generalization of kinetic systems has been introduced that is suitable for handling uncertain parameters and also additional linear constraints, whenever the possible values of the unknown parameters can be represented as points of a convex polyhedron. Due to the similar model structure it can be proven that the dense realization of the generalized uncertain model with possible additional linear constraints also has the superstructure property, and all the algorithms introduced previously for computing certain realizations of non-uncertain kinetic systems can be applied with appropriate modifications to the case of this type of systems. Furthermore, the method demonstrated in [53] for computing the set of possible reaction graph structures of an uncertain kinetic system can be modified to apply parallel computations. The results related to uncertain models are demonstrated in [56], [54] and in Section 6 of this thesis.

DOI:10.15774/PPKE.ITK.2018.004

Chapter 2

Notations and computational background

In this chapter the notations of kinetic systems as special type of nonnegative polynomial systems and their realizations as chemical reaction networks are summarized. The realizations can be determined by the application of a linear programming based computational model, which is also demonstrated here in Section 2.4.

2.1 Nonnegative polynomial systems

Polynomial systems are dynamical systems where the dynamic equations can be written in the form of a (multivariate) polynomial.

Definition 2.1. *Let $x : \mathbb{R} \to \mathbb{R}^n$ be a function, $M \in \mathbb{R}^{n\times p}$ a coefficient matrix and $\varphi : \mathbb{R}^n \to \mathbb{R}^p$ a monomial-type vector-mapping with coordinate functions of the form $\varphi_j(x) = \sum x_i^{\beta_{ij}}$, where $\beta_{ij} \in \mathbb{N}$ for all $i \in \{1,\dots,n\}$ and $j \in \{1,\dots,p\}$. Then the following system is called a* ***polynomial system****:*

$$\dot{x} = M \cdot \varphi(x) \tag{2.1}$$

A polynomial system is **nonnegative** if in the case of any nonnegative initial value the trajectories remain in the nonnegative orthant. To this property a necessary and sufficient condition can be given, see [37]. If the polynomial system is given in the form $\dot{x} = f(x)$, then it is nonnegative if and only if the function f is essentially nonnegative. A function $f : [0,\infty)^n \to \mathbb{R}^n$ is called **essentially nonnegative** if for its coordinate functions $f_i : [0,\infty)^n \to \mathbb{R}$ with every index $i \in \{1,\dots,n\}$ the inequality $f_i(x) \geq 0$ holds, whenever x is in the nonnegative orthant and the coordinate x_i is zero.

By definition a nonnegative polynomial system is called kinetic if there is a chemical reaction network (CRN) with the given dynamical behaviour.

DOI:10.15774/PPKE.ITK.2018.004

2.2 Algebraic and dynamical characterization of chemical reaction networks

Definition 2.2. *A **chemical reaction network** can be characterized by three sets, see e.g.* [31, 10].

species: $\mathcal{S} = \{X_i \mid i \in \{1, \dots, n\}\}$

complexes: $\mathcal{C} = \{C_j = \sum_{i=1}^{n} \alpha_{ji} \cdot X_i \mid \alpha_{ji} \in \mathbb{N},\ j \in \{1, \dots, m\},\ i \in \{1, \dots, n\}\}$

reactions: $\mathcal{R} \subseteq \{(C_i, C_j) \mid C_i, C_j \in \mathcal{C}\}$

*The reaction $C_i \to C_j$ for $i, j \in \{1, \dots m\}$, $i \neq j$ is represented by the ordered pair (C_i, C_j), and the rate of the reaction is determined by the corresponding **reaction rate coefficient** $k_{ij} \in \mathbb{R}_+$. This reaction is present in the reaction network if and only if $k_{ij} > 0$ (i.e. $k_{ij} \neq 0$) holds.*

The numerical properties of chemical reaction networks can be characterized by special matrices. The linear combinations defining the structures of the complexes are included by the **complex composition matrix** $Y \in \mathbb{N}^{n \times m}$, where

$$[Y]_{ij} = \alpha_{ji} \qquad i \in \{1, \dots, n\},\ j \in \{1, \dots, m\} \tag{2.2}$$

The structure of the reaction network is described through the reaction rates by the **Kirchhoff matrix** $A_k \in \mathbb{R}^{m \times m}$ of the CRN. It is a Metzler-matrix, i.e. all its off-diagonal entries are nonnegative. Furthermore, the sums of the entries in each column are zero, therefore the Kirchhoff matrix is also called a column-conservation matrix. The entries of the matrix are defined by the following equation:

$$[A_k]_{ij} = \begin{cases} k_{ji} & \text{if } i \neq j \\ -\sum_{l=1, l \neq i}^{m} k_{il} & \text{if } i = j \end{cases} \tag{2.3}$$

Let the function $x : \mathbb{R} \to \mathbb{R}^n_+$ describe the concentrations of the species depending on time. Assuming mass-action kinetics the dynamics of the concentrations can be characterized by dynamical equations of the form of a polynomial system:

$$\dot{x} = Y \cdot A_k \cdot \psi^Y(x) \tag{2.4}$$

where $\psi^Y : \mathbb{R}^n \to \mathbb{R}^m$ is the **monomial function** of the CRN. The monomials, i.e. the coordinate functions correspond to the complexes and they are defined as

$$\psi^Y_j(x) = \prod_{i=1}^{n} x_i^{Y_{ij}} \qquad j \in \{1, \dots, m\} \tag{2.5}$$

It can be seen that the structural and dynamical properties of a CRN are uniquely determined by the matrices Y and A_k, consequently a reaction network can be referred to as the matrix pair (Y, A_k).

Now, the accurate definition of kinetic systems can be given.

DOI:10.15774/PPKE.ITK.2018.004

2.2.1 Kinetic systems

Definition 2.3. *A polynomial system* $\dot{x} = M \cdot \varphi(x)$ (2.1) *with a function* $x : \mathbb{R} \to \mathbb{R}^n$, *a coefficient matrix* $M \in \mathbb{R}^{n \times p}$ *and a monomial function* $\varphi : \mathbb{R}^n \to \mathbb{R}^p$ *is called* ***kinetic*** *if there exists a chemical reaction network* (Y, A_k) *so that the following equation holds.*

$$M \cdot \varphi(x) = Y \cdot A_k \cdot \psi^Y(x) \tag{2.6}$$

A necessary and sufficient condition of the kinetic property can be given as follows by prescribing the sign pattern of the matrix M, see [6].

Proposition 2.4. *Let a polynomial system* $\dot{x} = M \cdot \varphi(x)$ *be characterized by a coefficient matrix* $M \in \mathbb{R}^{n \times p}$ *and a monomial function* $\varphi : \mathbb{R}^n \to \mathbb{R}^p$ *with coordinate functions of the form* $\varphi_j(x) = \sum_{i=1}^{n} x_i^{\beta_{ij}}$, *where* $\beta_{ij} \in \mathbb{N}$ *for all* $i \in \{1, \dots, n\}$ *and* $j \in \{1, \dots, p\}$. *This polynomial system is kinetic if and only if the following holds:*

$$\Big[[M]_{ij} < 0 \implies \beta_{ij} > 0 \Big] \qquad i \in \{1, \dots, n\},\ j \in \{1, \dots, p\} \tag{2.7}$$

If the polynomial system (2.1) is kinetic and the CRN (Y, A_k) fulfils Equation (2.6), then this CRN is called a **dynamically equivalent realization** of the kinetic system (2.1). In general, a kinetic system has several dynamically equivalent realizations, there might be chemical reaction networks with different reactions or even different sets of complexes that are governed by the same dynamics, see e.g. [7, 8, 17].

2.2.2 The canonical realization

In most of the cases the dynamics of the kinetic system cannot be realized on the set of complexes determined by the monomial function φ in Equation (2.1). A possible set can be determined using the method presented in [6], that also provides a dynamically equivalent realization of the kinetic system called the **canonical realization**. During the computation the reactions are defined one by one from the dynamical equations. The monomial $\prod_{i=1}^{n} x_i^{\beta_{ij}}$ with coefficient c in the equation of $\dot{x}_l$ characterizes the reaction:

$$\sum_{i=1}^{n} \beta_{ij} \cdot X_i \xrightarrow{c} \left(\sum_{i=1}^{n} \beta_{ij} \cdot X_i \right) + sgn(c) \cdot X_l, \tag{2.8}$$

where *sgn* is the sign function. The set of complexes in the canonical realization can be complemented as well by more complexes, and realizations of the kinetic system with a different set of complexes can also be determined, as it will be shown in Section 2.4

DOI:10.15774/PPKE.ITK.2018.004

2.2.3 Linearly conjugate realizations of kinetic systems

The notion of dynamical equivalence can be extended to the case when the state space is subject to a positive definite diagonal linear transformation. It works as if the species concentrations were individually scaled.

Such a transformation preserves the kinetic property of the system, as it was proven in [38] and [9]. If a polynomial system is kinetic then by using any transformation of the form (2.9) the transformed model will also be kinetic.

The transformation is defined by a positive definite diagonal matrix $T \in \mathbb{R}^{n\times n}$ so that the state variable x is transformed to the form $\bar{x} = T^{-1} \cdot x$ (i.e. $x = T \cdot \bar{x}$). The time derivative of the transformed variable can be written as follows:

$$\dot{\bar{x}} = T^{-1} \cdot \dot{x} = T^{-1} \cdot M \cdot \varphi(x) = T^{-1} \cdot M \cdot \varphi(T \cdot \bar{x}) = T^{-1} \cdot M \cdot \Phi_T \cdot \varphi(\bar{x}), \tag{2.9}$$

where $\Phi_T \in \mathbb{R}^{p\times p}$ is a positive definite diagonal matrix, the diagonal entries are $[\Phi_T]_{ii} = \varphi_i(T \cdot \mathbf{1}^p)$ for $i \in \{1, \dots, p\}$ and $\mathbf{1}^p \in \mathbb{R}^p$ is a column vector with all coordinates equal to 1.

The realizations of the generalized model can be defined as follows:

Definition 2.5. *The reaction network (Y, A_k) is a **linearly conjugate realization** of the kinetic system (2.1) if there exists a positive definite diagonal matrix $T \in \mathbb{R}^{n\times n}$ so that*

$$Y \cdot A_k \cdot \psi^Y(x) = T^{-1} \cdot M \cdot \Phi_T \cdot \varphi(x) \tag{2.10}$$

It is easy to see that dynamically equivalent realizations are also linearly conjugate realizations of the kinetic system with the transformation matrix T and Φ_T being equal to the unit matrix $\mathbf{I}^n$.

2.3 Graph representation

Chemical reaction networks can also be represented as an edge-weighted directed graph, which is a more insightful description of the structural properties.

Definition 2.6. *The directed graph $G(V, E)$ with weight function $w : E(G) \to \mathbb{R}_+$ is called **Feinberg- Horn-Jackson graph** or **reaction graph** of the CRN defined by the sets $\mathcal{S}, \mathcal{C}, \mathcal{R}$ and reaction rate coefficients k_{ij} for all $i, j \in \{1, \dots, m\}, i \neq j$ if*

- *the **vertices** correspond to the complexes* – $V(G) = \mathcal{C}$
- *the **directed edges** represent the reactions* – $E(G) = \mathcal{R}$
- *and the **weights** are the reaction rate coefficients* – $w((C_i, C_j)) = k_{ij}$

Let the vertices $v_i, v_j \in V(G)$ represent the complexes C_i and C_j, respectively. Then there is a directed edge in the reaction graph from vertex v_i to v_j if and only if the reaction $C_i \to C_j$ takes place in the CRN, and the weight of this edge is the corresponding reaction

DOI:10.15774/PPKE.ITK.2018.004

rate coefficient k_{ij}. From the definition of reactions it follows that in the reaction graph there are no loops or multiple edges with identical directions.

If the reaction network is given as the pair (Y, A_k) the reaction graph representing it is referred to as $G(Y, A_k)$.

In some cases the reaction rates are not relevant, therefore there is no need to indicate them in the reaction graph. The reaction graph without considering the edge weights is called a **reaction graph structure**. It is possible that a kinetic system has multiple realizations with identical reaction graph structures, such realizations are called **structurally identical**. If there are two structurally identical realizations of a kinetic system then there are infinitely many. But if two realizations correspond to different reaction graph structures, then these are called **structurally different**.

2.4 Linear programming based computational model

Dynamically equivalent and linearly conjugate realizations of kinetic system models can be computed by applying linear optimizations methods. Dynamical equivalence is a special case of linear conjugacy, therefore the computational model presented here is only for the general case.

2.4.1 The general form of LP models

The general form of a linear optimization problem is

$$\begin{aligned} &\max c^\top \cdot x \\ &A \cdot x \leq b \\ &x \geq 0 \end{aligned}$$

where $x \in \mathbb{R}^n$ represents the decision variables, $b \in \mathbb{R}^m$, $c \in \mathbb{R}^n$ are parameter vectors and $A \in \mathbb{R}^{m \times n}$ is a coefficient matrix, which is also known.

By definition all the constraints must be nonstrict inequalities since the set of possible solutions should be a closed polyhedron. However, equations of the form $a \cdot x = b_i$ can be included in the model using an equivalent set of two inequalities:

$$a \cdot x = b_i \quad \Longleftrightarrow \quad a \cdot x \leq b_i, \quad -a \cdot x \leq -b_i$$

The first provably correct solution is the Simplex Algorithm developed by Dantzig in 1947 [25, 26]. This algorithm works in most of the practical applications in polynomial time, but there are also exceptions. Later several different algorithms were also designed for the efficient computation of LP problems, such as the criss-cross method or the ellipsoid method [28].

It is possible to include integer variables in the optimization model as well. It is possible that all the variables are integers, or both integer and continuous variables are present

DOI:10.15774/PPKE.ITK.2018.004

in the model, then the model is called an integer linear programming (ILP) problem, or a mixed integer linear programming (MILP) problem, respectively. Both models are known to be NP-hard, consequently there exist no polynomial solution to these problems. There are several approximative and also exact methods, such as the cutting plane method developed by Gomory [29] and the Branch and Bound method proposed by Land and Doig [30]. However, it is not possible to solve MILP problems as efficiently as LP problems defined with only continuous variables.

2.4.2 LP model description of linearly conjugate realizations

By definition a linearly conjugate realization of the kinetic system $\dot{x} = M \cdot \varphi(x)$ must fulfil Equation (2.10). For each equation of the system the two sides are multivariate polynomials that are identical if and only if the sets of monomials are the same and the coefficients corresponding to identical monomials are pairwise the same. Consequently, the monomial functions ψ^Y and φ must be equal. Since the kinetic dynamical equations are assumed to be given, the complexes corresponding to the monomials of φ that have non-zero coefficients in any of the equations must be in the set of complexes of the kinetic systems. The method presented in Section 2.2.2 and originally in [6] can provide a possible set of complexes. It can be applied without modification or it can be complemented with other complexes as well. However, before the computation it is necessary to fix the set of complexes on which the realizations that we are looking for are defined.

Let the fixed set of complexes be the one that is characterized by the matrix $Y \in \mathbb{R}^{n\times m}$, and the new monomial function be $\psi^Y : \mathbb{R}^n \to \mathbb{R}^m$. Each coordinate function of ψ^Y corresponds to a complex in the new set, which is an extension of the set of complexes defined by the monomials of the function φ (none of the original elements are removed). In order to describe the dynamic equations of the original kinetic system using the monomial function ψ^Y as

$$\dot{x} = M \cdot \psi^Y(x), \tag{2.11}$$

the coefficient matrix needs to be modified. In the matrix $M \in \mathbb{R}^{n\times m}$ this modification will result in zero columns corresponding to the new monomials. For simplicity the modified coefficient matrix is also denoted by M.

Using these notations Equation (2.10) changes as:

$$Y \cdot A_k \cdot \psi^Y(x) = T^{-1} \cdot M \cdot \Psi_T \cdot \psi^Y(x), \tag{2.12}$$

where the matrix $\Psi_T \in \mathbb{R}^{m\times m}$ is a positive definite diagonal matrix with diagonal entries $[\Psi_T]_{ii} = \psi_i^Y(T \cdot \mathbf{1}^m)$ for all $i \in \{1, \ldots, m\}$. (Ψ_T is defined by the function ψ^Y similarly as Φ_T is defined by φ.)

The polynomials on the two sides are equal if and only if the corresponding coefficients are pairwise identical. By using this fact and the notation $A_b = A_k \cdot {\Psi_T}^{-1}$ Equation (2.10) can be written as:

$$Y \cdot A_b = T^{-1} \cdot M \tag{2.13}$$

DOI:10.15774/PPKE.ITK.2018.004

The matrix $A_b \in \mathbb{R}^{m\times m}$ is obtained by scaling the columns of A_k by positive scalars, consequently it is also a Kirchhoff matrix and it represents the same reaction graph structure as the matrix A_k. The simplified form of the linear conjugacy equation (2.13) can be applied only if the set of complexes is fixed, therefore in this work the kinetic systems are considered on a fixed set of complexes. Then the kinetic system $\dot{x} = M \cdot \psi^Y$ can be referred to as the matrix pair $[M, Y]$. From Equation (2.13) the matrices T^{-1} and A_b can be obtained, therefore in this work the linearly conjugate realizations are referred to as the corresponding matrix pair (T^{-1}, A_b). These parameters uniquely characterize the CRN (since the matrix Y is unchanged) by the matrix T^{-1} its inverse T and the matrix Ψ_T are uniquely defined, and from these the Kirchhoff matrix A_k can be computed as $A_k = A_b \cdot \Psi_T$.

The aim of the computation is to determine a linearly conjugate realization (T^{-1}, A_b) of the kinetic system $[M, Y]$, consequently the known parameters of the model are the matrices M and Y. The variables are the off-diagonal entries of the matrix $A_b \in \mathbb{R}^{m\times m}$. The diagonal ones are determined by the off-diagonals, therefore it is not necessary to consider them as variables. Since T is diagonal, $T^{-1} \in \mathbb{R}^{n\times n}$ is also a diagonal matrix, therefore only its diagonal entries are decision variables. It follows that the number of variables in the optimization model is $m^2 - m + n$. The constraints of the model are the following:

$$Y \cdot A_b - T^{-1} \cdot M = \mathbf{0}^{n\times m} \tag{2.14}$$

$$\sum_{\substack{j=1\\ j\neq i}}^{m} [A_b]_{ji} = -[A_b]_{ii} \qquad i \in \{1, \dots, m\} \tag{2.15}$$

$$[A_b]_{ij} \geq 0 \qquad i, j \in \{1, \dots, m\},\ i \neq j \tag{2.16}$$

$$[T^{-1}]_{ll} > 0 \qquad l \in \{1, \dots, n\} \tag{2.17}$$

To ensure linear conjugacy Equation (2.14) must be fulfilled. It is equivalent to Equation (2.10), since $\mathbf{0}^{n\times m} \in \mathbb{R}^{n\times m}$ is a zero matrix. The Equations (2.15), (2.16) and (2.17) are necessary to ensure that the matrices A_b and T^{-1} meet their definitions.

Further linear constraints can be added to the model if a realization with special properties is required to be determined. For example the exclusion of some set $\mathcal{H} \subset \mathcal{R}$ of reactions can be written as

$$[A_b]_{ji} = 0 \qquad (C_i, C_j) \in \mathcal{H} \tag{2.18}$$

The objective function of the optimization can be defined in several ways according to the additional requirements, as it will be seen later.

2.5 Illustration of the basic notions

In this section examples are presented to demonstrate the notations and properties of kinetic systems and chemical reaction networks introduced in Chapter 2.

DOI:10.15774/PPKE.ITK.2018.004

2.5.1 Example 1 – Basic properties of reaction networks

Let us consider a simple chemical reaction network with two reactions. It is a realization of the kinetic system presented in [19].

$$3X_2 \xrightarrow{1} 3X_1$$
$$3X_1 \xrightarrow{2} 2X_1 + X_2$$

The characterizing sets of this model are the following:

the set of species:	$\mathcal{S} = \{X_1, X_2\}$
the set of complexes:	$\mathcal{C} = \{C_1 = 3X_2, C_2 = 3X_1, C_3 = 2X_1 + X_2\}$
the set of reactions:	$\mathcal{R} = \{(C_1, C_2), (C_2, C_3)\}$
the reaction rate coefficients:	$k_{12} = 1, k_{23} = 2$

The complex composition matrix Y, the monomial function $\psi^Y(x)$ and the Kirchhoff matrix A_k of the CRN are defined as follows:

$$Y = \begin{bmatrix} 0 & 3 & 2 \\ 3 & 0 & 1 \end{bmatrix} \qquad \psi^Y(x) = \begin{bmatrix} x_2^3 \\ x_1^3 \\ x_1^2 x_2 \end{bmatrix} \qquad A_k = \begin{bmatrix} -1 & 0 & 0 \\ 1 & -2 & 0 \\ 0 & 2 & 0 \end{bmatrix}$$

The equations describing the dynamical behaviour of the system can be written in the form of Equation (2.4), that prescribes the kinetic system as in Equation (2.11). The dynamical system and its coefficient matrix M are:

$$\dot{x}_1 = 3x_2^3 - 2x_1^3$$
$$\dot{x}_2 = -3x_2^3 + 2x_1^3$$
$$M = \begin{bmatrix} 3 & -2 & 0 \\ -3 & 2 & 0 \end{bmatrix}$$

By definition it is a kinetic system that is referred to as $[M, Y]$ and the reaction network (Y, A_k) is a dynamically equivalent realization of it. The reaction graph $G(Y, A_k)$ and the reaction graph structure representing the realization (Y, A_k) can be seen in Figures 2.1 and 2.2, respectively.

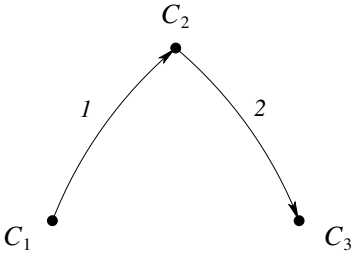


FIGURE 2.1: The reaction graph $G(Y, A_k)$

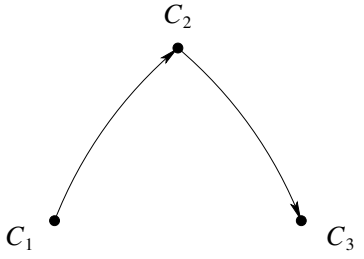
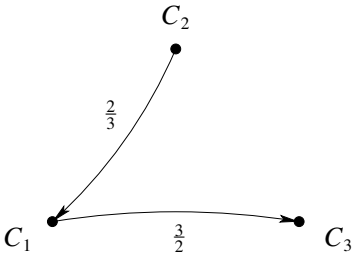


FIGURE 2.2: The reaction graph structure of the CRN (Y, A_k)

DOI:10.15774/PPKE.ITK.2018.004

It will be shown later in Section 5.3.1 that the kinetic system $[M, Y]$ has several dynamically equivalent realizations, one of them is the CRN (Y, A'_k). The reaction graphs of the realizations (Y, A_k) and (Y, A'_k) are very similar to each other, although these realizations are structurally different. Besides the isomorphism of the graphs the labels of the corresponding edges must be the same as well, and this latter property is not fulfilled by the graphs $G(Y, A_k)$ and $G(Y, A'_k)$.



$$A'_k = \begin{bmatrix} -\frac{3}{2} & \frac{2}{3} & 0 \\ 0 & -\frac{2}{3} & 0 \\ \frac{3}{2} & 0 & 0 \end{bmatrix}$$

FIGURE 2.3: The reaction graph $G(Y, A'_k)$

2.5.2 Example 2 – Canonical realization of a kinetic system

The following kinetic system was introduced previously as Example 3 in [35].

$$\begin{aligned} \dot{x}_1 &= x_1 x_2^2 - 2x_1^2 + x_1 x_3^2 \\ \dot{x}_2 &= -x_1^2 x_2^2 + x_1 x_3^2 \\ \dot{x}_3 &= x_1^2 - 3x_1 x_3^2 \end{aligned} \tag{2.19}$$

This polynomial system can be originated from the matrix equation $\dot{x} = M_1 \cdot \varphi(x)$, where

$$M_1 = \begin{bmatrix} 1 & 0 & -2 & 1 \\ 0 & -1 & 0 & 1 \\ 0 & 0 & 1 & -3 \end{bmatrix} \qquad \varphi(x) = \begin{bmatrix} x_1 x_2^2 & x_1^2 x_2^2 & x_1^2 & x_1 x_3^2 \end{bmatrix}^\top$$

It can be seen that this polynomial system fulfils the condition of Proposition 2.4, therefore it is a kinetic system. But there is no realization on the set $\mathcal{C}_1 = \{X_1 + 2X_2, 2X_1 + 2X_2, 2X_1, X_1 + 2X_3\}$ of complexes characterized by the monomial function φ. It can be checked for example by trying to find a dense realization using Algorithm 1. There exists at least one realization with a given set of complexes if and only it there is a dense realization on the same set of complexes. From the properties of polynomials it follows that these complexes must be included in the set of complexes of any realization, but in this case other elements are also necessary.

By the application of the method presented in [6] one can obtain a dynamically equivalent realization, and at the same time a suitable set of complexes. During this method every monomial in every dynamical equation of the polynomial system determines a reaction. For example, in the case of the equation $\dot{x}_2 = -x_1^2 x_2^2 + x_1 x_3^2$ the obtained reactions are:

DOI:10.15774/PPKE.ITK.2018.004

$$
\begin{aligned}
-x_1^2x_2^2\text{:} &\qquad 2X_1 + 2X_2 \xrightarrow{1} 2X_1 + X_2\\
+x_1x_3^2\text{:} &\qquad X_1 + 2X_3 \xrightarrow{1} X_1 + X_2 + 2X_3
\end{aligned}
$$

The elements of the defined set $\mathcal{C}_2$ of complexes are the following:

$$
\begin{array}{llll}
C_1 = X_1 + 2X_2 & C_2 = 2X_1 + 2X_2 & C_3 = 2X_1 + X_2 & C_4 = 2X_1\\
C_5 = X_1 & C_6 = 2X_1 + X_3 & C_7 = X_1 + 2X_3 & C_8 = 2X_1 + 2X_3\\
C_9 = X_1 + X_2 + 2X_3 & C_{10} = X_1 + X_3 &&
\end{array}
$$

The set $\mathcal{C}_2$ characterizes the complex composition matrix Y and the monomial function ψ^Y of the canonical realization:

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

$$
\psi^Y(x) = \begin{bmatrix} x_1x_2^2 & x_1^2x_2^2 & x_1^2x_2 & x_1^2 & x_1 & x_1^2x_3 & x_1x_3^2 & x_1^2x_3^2 & x_1x_2x_3^2 & x_1x_3 \end{bmatrix}^\top
$$

The computed reactions define the Kirchhoff matrix and the reaction graph as well.

$$
A_k = \begin{bmatrix}
-1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
1 & -1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & -3 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & -5 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\
0 & 0 & 0 & 0 & 0 & 0 & 3 & 0 & 0 & 0
\end{bmatrix}
$$

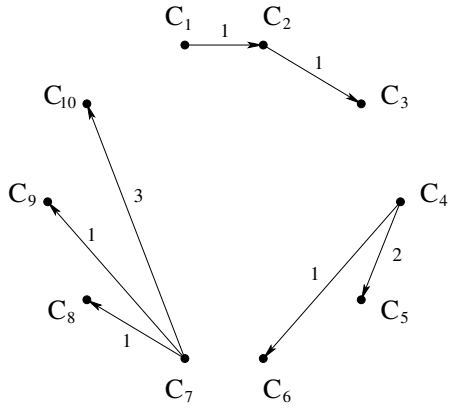


FIGURE 2.4: The reaction graph of the canonical realization

In order to obtain the form of Equation (2.12) the equation $\dot{x} = M_1 \cdot \varphi(x)$ defining the dynamics of the kinetic system needs to be rewritten as $\dot{x} = M_2 \cdot \psi^Y(x)$. The additional monomials appear in these equations with zero coefficients, therefore the defined dynamics is unchanged. With these notations the equation $M_2 \cdot \psi^Y(x) = Y \cdot A_k \cdot \psi^Y(x)$ holds, that is the special case of Equation (2.12) for dynamically equivalent realizations, where the matrices T^{-1} and Ψ_T are identity matrices. The coefficient matrix M_2 is as follows:

DOI:10.15774/PPKE.ITK.2018.004

$$M_2 = \begin{bmatrix} 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -3 & 0 & 0 & 0 \end{bmatrix}$$

2.5.3 Example 3 – Linear conjugacy of kinetic systems

The kinetic system examined in this section was introduced in [24] as Example 2.

$$\begin{aligned} \dot{x}_1 &= 1 - x_1 - x_1^2 + x_2 x_3 \\ \dot{x}_2 &= 2x_1 - 2x_2 x_3 - 2x_2^2 + 2x_3^2 \\ \dot{x}_3 &= x_1 - x_2 x_3 + x_2^2 - x_3^2 \end{aligned} \tag{2.20}$$

The linear optimization based method for computing realizations presented in Section 2.4 can be applied only if the set of complexes is fixed. Therefore, the set is fixed to contain only the complexes defined by the monomials, and this kinetic system is denoted as $[M, Y]$, where

$$M = \begin{bmatrix} 1 & -1 & -1 & 1 & 0 & 0 \\ 0 & 2 & 0 & -2 & -2 & 2 \\ 0 & 1 & 0 & -1 & 1 & -1 \end{bmatrix} \qquad Y = \begin{bmatrix} 0 & 1 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 2 & 0 \\ 0 & 0 & 0 & 1 & 0 & 2 \end{bmatrix}$$

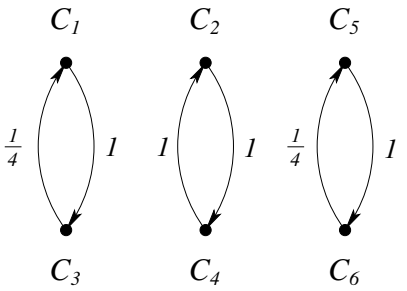
Despite the fact that the polynomial system (2.20) fulfils the conditions of Proposition 2.4, the computation returns that there is no dynamically equivalent realization of $[M, Y]$. Indeed, the kinetic property means only that there is a realization with some set of complexes, that is not necessarily the actually considered one. Furthermore, in the case of any kinetic system the canonical realization can be generated and it is a dynamically equivalent realization of the model.

Interestingly, the computation of linearly conjugate realizations returns that there exists such a realization with the given set of complexes.

$$T = \begin{bmatrix} \frac{1}{2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \frac{1}{2} \end{bmatrix} \qquad \Psi_T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{1}{4} & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & \frac{1}{4} \end{bmatrix}$$

The state transformation is defined by the matrix T and it also characterizes the diagonal matrix Ψ_T. By definition the CRN (Y, A_k) has to fulfil the equation $Y \cdot A_k = T^{-1} \cdot M \cdot \Psi_T$.

DOI:10.15774/PPKE.ITK.2018.004



$$A_k = \begin{bmatrix} -1 & 0 & \frac{1}{4} & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & 0 & 0 \\ 1 & 0 & -\frac{1}{4} & 0 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & \frac{1}{4} \\ 0 & 0 & 0 & 0 & 1 & -\frac{1}{4} \end{bmatrix}$$

FIGURE 2.5: The reaction graph $G(Y, A_k)$

.

In order to have a linear matrix equation in the optimization model, the equation is transformed to the form $Y \cdot A_k \cdot \Psi_T^{-1} = T^{-1} \cdot M$ and the notation $A_b = A_k \cdot \Psi_T^{-1}$ is applied. For this reason linearly equivalent realizations of a kinetic system $[M, Y]$ with a fixed set of complexes are referred to as the pair (T^{-1}, A_b). In this case the characterizing matrices are

$$T^{-1} = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \qquad A_b = A_k \cdot \Psi_T^{-1} = \begin{bmatrix} -1 & 0 & \frac{1}{16} & 0 & 0 & 0 \\ 0 & -\frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 1 & 0 & \frac{-1}{16} & 0 & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & \frac{1}{16} \\ 0 & 0 & 0 & 0 & 1 & -\frac{1}{16} \end{bmatrix}$$

It can be seen that A_b is also a Kirchhoff matrix and it defines the same set of reactions as the matrix A_k.

DOI:10.15774/PPKE.ITK.2018.004

Chapter 3

Dense realizations

There are realizations that have great importance regarding the results presented in this dissertation.

Definition 3.1. *A realization of a CRN is called a* ***dense realization*** *if it has the maximum number of reactions.*

Dense realizations can be defined in the sets of dynamically equivalent, linearly conjugate or any other kind of realizations of a kinetic system assuming a fixed set of complexes.

The operation of the methods presented in this dissertation depends on a special property of dense realizations. It is drawn up in Proposition 3.3, and was proven in [51].

3.1 Superstructure property

Definition 3.2. *Let $\mathcal{G}$ be a set of directed graphs defined on a fixed set of labelled vertices. A directed graph is called a* ***superstructure*** *considering the set $\mathcal{G}$ if it contains every graph in the set as a subgraph and it is minimal under inclusion.*

It follows from the definition that for every set $\mathcal{G}$ there exists a superstructure and it is unique, as it is the graph whose set of edges is the union of the edges of all graphs in $\mathcal{G}$.

It has been proven in [15] that for any kinetic system the dense linearly conjugate realization defines a superstructure among all linearly conjugate realizations of a kinetic system with a fixed set of complexes. This property holds also for dynamically equivalent realizations. However, during the computations presented in this work a little more is required, and it is proven in Proposition 3.3.

Proposition 3.3. *Among all the realizations linearly conjugate to a given kinetic system $[M, Y]$ on a fixed set of complexes and fulfilling a finite set of additional linear constraints the dense realization with the prescribed properties determines a superstructure.*

DOI:10.15774/PPKE.ITK.2018.004

Proof. In the proof the geometric properties of the set of possible realizations are utilized. As it was demonstrated in Section 2.4 the variables of the optimization model defining the linearly conjugate realizations of a kinetic system with a fixed set of complexes are the off-diagonal entries of the matrix $A_b \in \mathbb{R}^{m\times m}$ and the diagonal entries of the matrix $T^{-1} \in \mathbb{R}^{n\times n}$. If an ordering is defined on the set of variables the realizations can be represented as points in the Euclidean space $\mathbb{R}^{m^2-m+n}$. Let the first m^2-m coordinates correspond to the off-diagonal entries of matrix A_k ordered column-wise, and the diagonal entries of matrix T^{-1} be equal to the remaining n coordinates. In this approach the linear constraints of the forms of inequalities and equations are equivalent to halfspaces and hyperplanes, respectively. The set of possible solutions is the intersection of these halfspaces and hyperplanes, consequently it is a convex polyhedron $\mathcal{Q} \subseteq \mathbb{R}^{m^2-m+n}$ and every point of it corresponds to a realization.

Let us assume that the point $D = (d_1, \dots, d_{m^2-m}, \dots, d_{m^2-m+n}) \in \mathcal{Q}$ represents the dense linearly conjugate realization (T^{-1}, A_b) of the kinetic system $[M, Y]$ with the set of prescribed linear constraints.

By the definition of density the point D must have the maximum number of positive values among the coordinates $d_1, \dots, d_{m^2-m}$, and since D represents a real linearly conjugate realization, the coordinates $d_{m^2-m+1}, \dots, d_{m^2-m+n}$ are also positive.

Let us assume that the point $R = (r_1, \dots, r_{m^2-m}, \dots, r_{m^2-m+n}) \in \mathcal{Q}$ represents another linearly conjugate realization that has more positive coordinates than D. It means that there is an index $i \in \{1, \dots m^2-m\}$ for which $d_i = 0$ and $r_i > 0$ hold.

Since the polyhedron $\mathcal{Q}$ is convex, the interval (D, R) is also in $\mathcal{Q}$, and every interior point S of this interval corresponds to a linearly conjugate realization of the kinetic system $[M, Y]$ and fulfils the added constraints as well.

$$(D, R) = \{S \in \mathbb{R}^{m^2-m+n} \mid S = c \cdot D + (1-c) \cdot R,\ c \in (0, 1)\} \tag{3.1}$$

From the properties of positive linear combination it follows that all the coordinates that are positive (not zero) in D or R must be positive in the point S as well. Therefore S has more positive coordinates than D, which is a contradiction.

Consequently, there cannot exist such a point $R \in \mathcal{Q}$, and all points in the polyhedron $\mathcal{Q}$ can have positive values in only those coordinates where the point D has. The realization characterized by D defines a superstructure among the linearly conjugate realizations of the kinetic system $[M, Y]$ that fulfil the additional linear constraints, and the reaction graphs of these realizations are subgraphs of the reaction graph representing D. □

Corollary 3.4. *The superstructure property holds for dynamically equivalent realizations considering a set of additional constraints as well. It is because dynamically equivalent realizations are linearly conjugate realizations fulfilling the linear constraints*

$$[T^{-1}]_{ii} = 1 \qquad i \in \{1, \dots, n\} \tag{3.2}$$

Therefore the set of possible realizations of this model can also be represented as a convex polyhedron and the proof of Proposition 3.3 can be applied.

DOI:10.15774/PPKE.ITK.2018.004

3.2 Efficient algorithm for computing dense realizations

In the algorithms demonstrated in this thesis the computation of dense realizations fulfilling an additional set of non-strict linear inequalities is applied as a subroutine. It is invoked many times therefore it is essential to choose an efficient and accurate computation method.

In the literature there exist several alternative solutions for the computation of dense realizations. The method presented in [15] applies binary variables assigned to the reactions to track their presence, and by maximizing the sum of these binary variables can the dense realization be obtained. However, the application of binary variables requires the solution of a MILP problem that is known to be NP- hard. In [16] the binary variables are relaxed to the interval $[0, 1]$ and the problem is reformulated into the framework of linear programming, but it can be applied only in the case of dynamical equivalence. In [39] an iterative method was proposed that includes $m(m-1)+1$ linear programming steps, i.e. it is a polynomial time method but the number of required LP optimization steps is still quite large.

In this chapter an other method is proposed for computing the dense linearly conjugate realization of a kinetic system that fulfils a finite set of additional linear constraints as well. This method is also LP based and iterative, but the number of required optimization steps is in general significantly less than $m^2 - m$.

The source of the difficulties considering the computation of a dense linearly conjugate realization is that it cannot directly be written as a linear optimization problem. It is required that there is the maximal number of variables having positive values. As it was mentioned earlier, this task can be solved by adding integer variables, but in this case the aim is to apply only continuous variables. In addition to that in the model there are strict inequalities, since the matrix T^{-1} must be positive definite, but such inequalities are not allowed in an LP model.

The idea presented here solves both problems. The optimization problem is modified so that instead of strict inequalities non-strict ones are considered, and no integer variables are applied. This changes the set of possible solutions to be the closure of the polyhedron $\mathcal{Q}$, that is denoted by $\overline{\mathcal{Q}}$. The constraints of the LP optimization steps are very similar to Equations (2.14)-(2.17) characterizing linear conjugacy, the only difference is in the last equation.

$$Y \cdot A_b - T^{-1} \cdot M = \mathbf{0}^{n \times m} \tag{3.3}$$

$$\sum_{\substack{j=1 \\ j \neq i}}^{m} [A_b]_{ji} = -[A_b]_{ii} \qquad i \in \{1, \ldots, m\} \tag{3.4}$$

$$[A_b]_{ij} \geq 0 \qquad i, j \in \{1, \ldots, m\},\ i \neq j \tag{3.5}$$

$$[T^{-1}]_{ll} \geq 0 \qquad l \in \{1, \ldots, n\} \tag{3.6}$$

DOI:10.15774/PPKE.ITK.2018.004

To formulate the additional linear constraints it is easier to use the notation of a point $R = (r_1, \ldots, r_{m^2-m+n})$ as a linearly conjugate realization. Then the set of he additional constraints can be written as

$$\sum_{i=1}^{m^2-m+n} \delta_{ij} \cdot r_i \leq D_j \qquad \delta_{ij}, D_j \in \mathbb{R},\ j \in \{1, \ldots, |L|\} \tag{3.7}$$

From now on the LP model defined by the Equations (3.3)-(3.7) is called the modified model. In the algorithm to each variable a solution of the modified model is assigned where this variable is positive, if it is possible. A solution can belong to several variables, therefore in general only a small number of optimization steps are required. Then the dense realization is determined as a convex combination of the points representing the assigned solutions of the modified model.

In the algorithm the following procedure is applied repeatedly:

FindPositive$([M, Y], L, H)$ returns a pair (R, B), where R is a point in $\overline{\mathcal{Q}}$ that fulfils the constraints of the modified model characterized by the kinetic system $[M, Y]$ and the finite set L of non-strict linear inequalities, so that taking the set $H \subseteq \{1, \ldots, m^2-m+n\}$ of indices into account the value of the objective function $\sum_{j \in H} r_j$ is maximal. While $B \subseteq \{1, \ldots, m^2 - m + n\}$ is a set of indices for which $k \in B$ if and only if $r_k > 0$. If there exist no solution of the optimization problem, then the pair $(\mathbf{0}^{m^2-m+n}, \emptyset)$ is returned. The computation can be performed in polynomial time since it requires the solution of an LP optimization problem and the examination of positivity considering every element in a set of size $m^2 - m + n$.

Algorithm 1 Computes a dense linearly conjugate realization
Inputs: $[M, Y], L$
Output: *Result*

$H := \{1, 2, \ldots, m^2 - m + n\}$
$B := H$
$Result := \mathbf{0} \in \mathbb{R}^{m^2-m+n}$
$loops := 0$
while $B \neq \emptyset$ **do**
    $(R, B) :=$ **FindPositive**$([M, Y], L, H)$
    $Result := Result + R$
    $H := H \setminus B$
    $loops := loops + 1$
end while
$Result := Result/loops$
if $\exists i \in \{m^2 - m + 1, \ldots, m^2 - m + n\} \cap H$ or $Q = \mathbf{0}^{m^2-m+n}$ **then**
    There is no linearly conjugate realization of the kinetic system $[M, Y]$
    fulfilling the set L of constraints.
else
    Result determines a dense linearly conjugate realization of the kinetic
    system $[M, Y]$ fulfilling the set L of constraints.
end if

DOI:10.15774/PPKE.ITK.2018.004

Proposition 3.5. Algorithm 1 *returns a dense linearly conjugate realization of the kinetic system* $[M, Y]$ *on a given set of complexes and fulfilling a finite set* L *of additional constraints that are of the form of non-strict linear inequalities, if it exists. The computation runs in polynomial time.*

Proof. In the first step of the while loop the sum of all variables is maximized, then in the next step the variables that had positive value before are not considered in the objective function. This step is repeated until no point in $\overline{\mathcal{Q}}$ has positive value in the coordinates that remain in the set H. This property is equivalent to that the objective function of the optimization is zero. The computation stops after finitely many steps since the size of the set H is finite and it gets smaller in every step. The computed points of $\overline{\mathcal{Q}}$ are referred to as $R^1, R^2, \ldots, R^k$.

If for an index $j \in \{1, 2, \ldots, m^2 - m\}$ there is a point $P \in \overline{\mathcal{Q}}$ so that $p_j > 0$ holds, then there must be a point $R^i \in \overline{\mathcal{Q}}$ retuned by the procedure **FindPositive**$([M, Y], L, H)$ for which $r^i_j > 0$. Otherwise in the last step of the while loop the objective function would not be zero.

For all the indices $i \in \{m^2 - m + 1, \ldots, m^2 - m + n\}$ corresponding to the diagonal entries of the matrix T^{-1} there must be a point of $\overline{\mathcal{Q}}$ where it has a positive value. Otherwise T cannot be a positive definite matrix, consequently there is no linearly conjugate realization of the kinetic system.

The variable *Result* is computed as the arithmetic mean of the points $R^1, R^2, \ldots, R^k \in \overline{\mathcal{Q}}$ – but any convex combination with positive coefficients is also suitable – therefore it is also in $\overline{\mathcal{Q}}$.

For all $j \in \{1, \ldots, m^2 - m\}$ the value of $Result_j$ is positive if and only if there is an index $i \in \{1, \ldots k\}$ so that $R^i_j > 0$ holds. It follows from the computation that the point *Result* is positive in all coordinates, where it is possible, consequently the variable *Result* has the maximum number of positive coordinates.

It still needs to be proven that the point *Result* represents a valid linearly conjugate realization, i.e. it is not a point of $\overline{\mathcal{Q}} \setminus \mathcal{Q}$. It would be invalid only if it was a point of a facet corresponding to a strict inequality. Such inequalities are defined only in the case of the positive diagonal entries of the matrix T^{-1}. According to the algorithm the coordinates of *Result* corresponding to these variables are positive, therefore $Result \in \mathcal{Q}$ holds.

The number of optimization steps performed during the computation is at most $m^2 - m + n$, since the set determining the objective function is decreasing in every step. The optimization is in the form of an LP problem that can be solved in polynomial time, besides that just some minor computation is applied, thus the algorithm runs in polynomial time. □

Remark 3.6. During the actual computations a reaction $C_i \to C_j$ is considered to be present in the reaction network if and only if $[A_k]_{ji} > \varepsilon$, where ε is a sufficiently small positive threshold value for distinguishing between practically zero and non-zero reaction

DOI:10.15774/PPKE.ITK.2018.004

rate coefficients. In the applied implementation the value of ε was 10^{-6}.
It is important to remark as well that all variables of the computed realization have a value greater than ε, since the convex combination of numbers that are all larger than a fixed lower bound is also larger than this number.

Corollary 3.7. *If during the computation according to* Algorithm 1 *there are several solutions of the modified model assigned to variables then the number of dense realizations is continuum, since the coefficients of the convex combination can be chosen arbitrarily from the interval* $(0, 1)$. *Naturally, these realizations are structurally identical to each other, but might be different in parameters.*

3.2.1 Boundedness of variables

If the computation is performed by a computer program it is necessary to apply bounded optimization variables, and this property in general might restrict the set of possible solutions. However, in the case of linearly conjugate realizations the boundedness of variables can be ensured so that the set of possible reaction graph structures remains the same as in the original problem, and no solution is lost.

Since all variables are nonnegative by definition it is enough to show that upper bounds can be added, as it is stated in Proposition 3.8 and was originally presented in [51].

Proposition 3.8. *For any linearly conjugate realization* (T^{-1}, A_b) *of a kinetic system* $[M, Y]$ *there is another linearly conjugate realization* $(\hat{T}^{-1}, \hat{A}_b)$ *with all variables smaller than the given upper bound(s) so that the two realizations are structurally identical.*

This property holds even if just those realizations are considered that fulfil a finite set of additional homogeneous linear constraints. The general form of a homogeneous linear constraint on the variables $v_1, \ldots, v_p$ is

$$\sum_{i=1}^{p} c_i \cdot v_i \leq 0 \qquad\qquad c_1, \ldots c_p \in \mathbb{R} \tag{3.8}$$

Proof. If (T^{-1}, A_b) is a linearly conjugate realization of the kinetic system, then Equation (2.14) must hold. By multiplying this equation by a positive constant $c \in \mathbb{R}_+ \setminus \{0\}$

$$c \cdot T^{-1} \cdot M - c \cdot Y \cdot A_b = \mathbf{0}^{n \times m} \tag{3.9}$$

the equation becomes the condition corresponding to the linearly conjugate realization $(c \cdot T^{-1}, c \cdot A_k)$ of the same kinetic system.

$$(c \cdot T^{-1}) \cdot M - Y \cdot (c \cdot A_b) = \mathbf{0}^{n \times m} \tag{3.10}$$

The multiplication of the matrices by a constant does not change their essential properties. The matrix $c \cdot T^{-1} = \hat{T}^{-1}$ is a positive definite diagonal matrix and $c \cdot A_b = \hat{A}_b$

DOI:10.15774/PPKE.ITK.2018.004

is a column conservation matrix. Furthermore, $[\hat{A}_b]_{ij}$ is zero if and only if $[A_b]_{ij}$ is zero for all $i, j \in \{1, \ldots, m\}$. Consequently the linearly conjugate realizations $(\hat{T}^{-1}, \hat{A}_b)$ and (T^{-1}, A_b) of the kinetic system are structurally identical.

The value of the positive constant c can be determined so that all variables are below the given upper bound(s). The matrix equation can be considered as nm homogeneous linear equations. For a single equation it is easy to determine a possible constant c, and all the smaller positive values are also suitable.

$$c_1 > c_2 > 0 \quad \Longrightarrow \quad [c_1 \cdot A_b]_{ij} > [c_2 \cdot A_b]_{ij},\ [c_1 \cdot T^{-1}]_{ll} > [c_2 \cdot T^{-1}]_{ll} \qquad \forall i, j, l \quad (3.11)$$

Therefore the minimum of the finitely many constants computed for the individual equations is a suitable value for the constant c. □

Corollary 3.9. *Proposition 3.8 holds even if just those realizations are considered that fulfil a finite set of additional homogeneous linear constraints. The general form of a homogeneous linear constraint on the variables $v_1, \ldots, v_p$ is $\sum_{i=1}^{p} \gamma_i \cdot v_i \leq 0$ where the coefficients $\gamma_1, \ldots \gamma_p$ are real numbers, and such constraints hold even if the variables are scaled by a positive scalar c.*

$$\sum_{i=1}^{p} \gamma_i \cdot v_i \leq 0 \quad \Longleftrightarrow \quad \sum_{i=1}^{p} \gamma_i \cdot (c \cdot v_i) = c \cdot \sum_{i=1}^{p} \gamma_i \cdot v_i \leq 0 \qquad (3.12)$$

Corollary 3.10. *From Corollary 3.9 it follows that it is possible to apply bounded variables during the computing of linearly conjugate realizations that fulfil the constraints $[T]_{11} - [T]_{ii} = 0$ for all $i \in \{2, \ldots, n\}$. Every dynamically equivalent realization fulfils these constraints, and for every linearly conjugate realization that fulfils the constraints there is a dynamically equivalent realization so that the two realizations are structurally identical. If a linearly conjugate realization is referred to as the matrix pair (T^{-1}, A_b) where $T^{-1} = \frac{1}{d} \cdot I^n$, then by rearranging the equation $T^{-1} \cdot M = Y \cdot A_b$ characterizing linear conjugacy it becomes*

$$M = T \cdot Y \cdot A_b = d \cdot I^n \cdot Y \cdot A_b = Y \cdot (d \cdot A_b) \qquad (3.13)$$

that is the equation defining the dynamical equivalence of the realization $(Y, d \cdot A_b)$. The matrix $d \cdot A_b$ is obtained by scaling the Kirchhoff matrix A_b, therefore it is also a Kirchhoff matrix. Consequently, it is possible to apply bounded variables at the computation of dynamically equivalent realizations of a kinetic system with a fixed set of complexes.

DOI:10.15774/PPKE.ITK.2018.004

3.3 Examples

In this section the dense dynamically equivalent and linearly conjugate realizations of kinetic systems are examined, in some cases with additional linear constraints.

3.3.1 Example 4

The kinetic system examined in this section $[M, Y]$ was first presented in [40] as example $A1$.

$$Y = \begin{bmatrix} 0 & 1 & 0 & 2 & 2 & 3 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix} \qquad M = \begin{bmatrix} 0 & -k_2 & k_3 & -2k_4 & k_5 & 0 \\ k_1 & 0 & -k_3 & k_4 & -k_5 & 0 \end{bmatrix}$$

The parameters of the coefficient matrix are $k_1 = 1,\ k_2 = 1,\ k_3 = 0.05,\ k_4 = 0.1$ and $k_5 = 0.1$. With these parameter values the system shows oscillatory behaviour.

In the case of the previously known iterative polynomial-time method presented in [39] the number of applied optimization steps would have been $m^2 - m + 1 = 6^2 - 6 + 1 = 31$. The following realizations were computed using the Algorithm 1, and in each case the number of required optimization steps was significantly smaller.

$(T_{ld} - 1, A_b^{ld})$ is a dense linearly conjugate realization of the kinetic system $[M, Y]$, during its computation 4 optimization steps were applied.

$$A_b^{ld} = \begin{bmatrix} -80 & 1.167e7 & 3.083 & 3.333e6 & 0.333 & 0 \\ 0 & -2e7 & 0.5 & 5e6 & 0.5 & 0 \\ 80 & 0 & -4.25 & 4 & 0.5 & 0 \\ 0 & 5e6 & 0.25 & -2e7 & 1 & 0 \\ 0 & 0 & 0.25 & 4 & -8.5 & 0 \\ 0 & 3.333e6 & 0.167 & 1.167e7 & 6.167 & 0 \end{bmatrix} \qquad T_{ld}^{-1} = \begin{bmatrix} 40 & 0 \\ 0 & 80 \end{bmatrix}$$

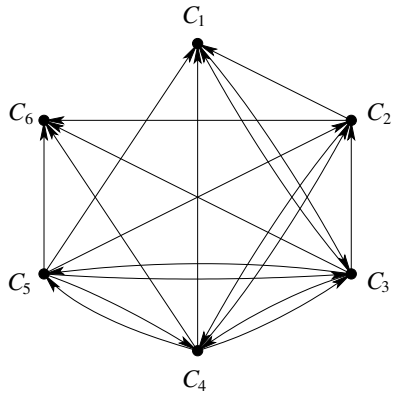


FIGURE 3.1: The reaction graph structures of the dense linearly conjugate realization (T_{ld}^{-1}, A_b^{ld}).

DOI:10.15774/PPKE.ITK.2018.004

The dense dynamically equivalent realization can be considered as a constrained linearly conjugate realization of the same kinetic system $[M, Y]$. It is the CRN (Y, A_k^{dd}), or written in the form of a linearly conjugate realization $(\mathbf{I}^2, A_k^{dd})$. The number of required iterations was only 4.

$$A_k^{dd} = \begin{bmatrix} -1 & 5000.5 & 0.027083 & 0.025 & 0 & 0 \\ 0 & -8750.125 & 0.0125 & 2500 & 0 & 0 \\ 1 & 0 & -0.05625 & 0.075 & 0 & 0 \\ 0 & 2499.75 & 0.00625 & -5000.125 & 0 & 0 \\ 0 & 0 & 0.00625 & 0.025 & -0.1 & 0 \\ 0 & 1249.875 & 0.004167 & 2500 & 0.1 & 0 \end{bmatrix}$$

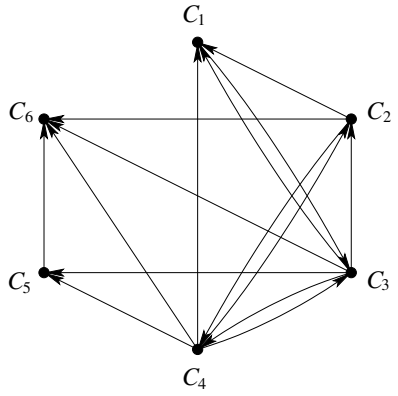


FIGURE 3.2: The reaction graph structure of the CRN (Y, A_k^{dd}).

By computing constrained dense realizations other structure-related questions can be answered as well, such as: Is it possible that the complexes C_5 and C_6 are not connected to the other complexes in a dynamically equivalent realization? To get the answer it is enough to compute the dense dynamically equivalent realization fulfilling the constraints that exclude all the reaction between the sets $\{C_1, C_2, C_3, C_4\}$ and $\{C_5, C_6\}$. If the unconstrained dense dynamically equivalent realization is known, then based on the superstructure property it is enough to define constraints for those reactions that are present in this realization.

The necessary constraints are $[A_k]_{53} = 0,\ [A_k]_{54} = 0$ considering the complex C_5, and $[A_k]_{62} = 0,\ [A_k]_{63} = 0,\ [A_k]_{64} = 0$ for C_6. Then the obtained constrained dense dynamically equivalent realization is (Y, A_k^{cdd}). The number of required iterations optimization steps was 3.

$$A_k^{cdd} = \begin{bmatrix} -1 & 3334 & 0.0166667 & 0 & 0 & 0 \\ 0 & -6667 & 0.0166667 & 0 & 0 & 0 \\ 1 & 0 & -0.05 & 0.1 & 0 & 0 \\ 0 & 3333 & 0.0166667 & -0.1 & 0 & 0 \\ 0 & 0 & 0 & 0 & -0.1 & 0 \\ 0 & 0 & 0 & 0 & 0.1 & 0 \end{bmatrix}$$

DOI:10.15774/PPKE.ITK.2018.004

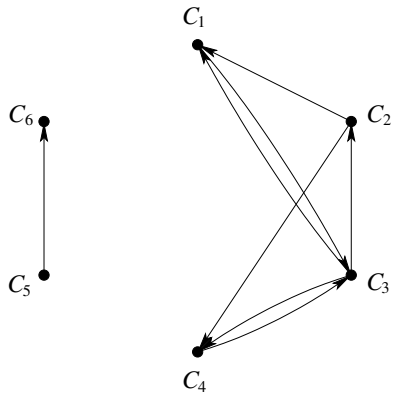


FIGURE 3.3: The reaction graph structure of the constrained dense dynamically equivalent realization (Y, A_k^{cdd}).

3.3.2 Example 5

The kinetic system examined in this section was published in [41]. It can be originated from the reaction network that is modelling the glyoxylate bypass.

The species of the network are:

I	active IDH (isocitrate dehydrogenase)
I_p	phosphorilated IDH
E	bifunctional enzyme IDHKP (IDH Kinase/Phosphatase)
EI	binding of the enzyme E and I
EI_p	binding of the enzyme E and I_p
EI_pI	binding of the enzyme E and both I_p and I

In the model 9 complexes are formed from these species, let us assume that this set is fixed. The structure of the complexes defines the following complex composition matrix.

$$C_1 = EI \qquad C_2 = EI_p$$
$$C_3 = EI_pI \qquad C_4 = I + E$$
$$C_5 = I_p + E \qquad C_6 = EI + I$$
$$C_7 = EI_p + I_p \qquad C_8 = EI + I_p$$
$$C_9 = I + EI_p$$

$$Y = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

The reaction graph structure of the kinetic system can be seen in Figure 3.4, and the reaction rate coefficients are as follows:

$$k_{41} = 1.6 \quad k_{14} = 0.3 \quad k_{15} = 1.06 \quad k_{52} = 4.62 \quad k_{25} = 0.94 \quad k_{24} = 0.12$$
$$k_{93} = 33 \quad k_{39} = 0.77 \quad k_{37} = 0.9 \quad k_{83} = 0.6 \quad k_{38} = 3 \quad k_{36} = 0.48297$$

DOI:10.15774/PPKE.ITK.2018.004

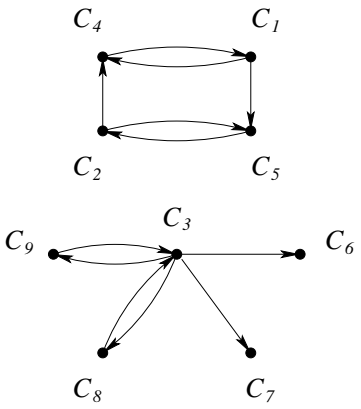


FIGURE 3.4: The reaction graph structure of the original reaction network.

This uniquely determines the kinetic system $[M, Y]$ where the coefficient matrix is

$$M = \begin{bmatrix} 0.3 & 0.12 & 1.25297 & -1.6 & 0 & 0 & 0 & 0 & -33 \\ 1.06 & 0.94 & 3.9 & 0 & -4.62 & 0 & 0 & -0.6 & 0 \\ 1.36 & 1.06 & 0 & -1.6 & -4.62 & 0 & 0 & 0 & 0 \\ -1.36 & 0 & 3.48297 & 1.6 & 0 & 0 & 0 & -0.6 & 0 \\ 0 & -1.06 & 1.67 & 0 & 4.62 & 0 & 0 & 0 & -33 \\ 0 & 0 & -5.15297 & 0 & 0 & 0 & 0 & 0.6 & 33 \end{bmatrix}$$

The dense realization (Y, A_k) of this kinetic system can be determined using only one optimization step in the algorithm, and it is structurally identical to the original reaction network depicted in Figure 3.4. Its Kirchhoff matrix A_k is as follows:

$$A_k = \begin{bmatrix} -1.36 & 0 & 0 & 1.6 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1.06 & 0 & 0 & 4.62 & 0 & 0 & 0 & 0 \\ 0 & 0 & -5.15297 & 0 & 0 & 0 & 0 & 0.6 & 33 \\ 0.3 & 0.12 & 0 & -1.6 & 0 & 0 & 0 & 0 & 0 \\ 1.06 & 0.94 & 0 & 0 & -4.62 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.48297 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.9 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 3 & 0 & 0 & 0 & 0 & -0.6 & 0 \\ 0 & 0 & 0.77 & 0 & 0 & 0 & 0 & 0 & -33 \end{bmatrix}$$

It follows that in the case of the given set of complexes no further reaction can be included in any realization of the kinetic model without changing its dynamics.

DOI:10.15774/PPKE.ITK.2018.004

3.4 Summary

I have proven new results regarding to dense realizations of kinetic systems, using a geometric approach. I have shown that a dense linearly conjugate realization of a kinetic system with a fixed set of complexes and fulfilling an additional finite set of linear constraints determines a superstructure considering all realizations of the constrained model. The correctness of the algorithms presented in the dissertation depends on this property. The results are described in detail in Section 3.1 and summarized in Thesis I.a.

I have developed a novel polynomial-time algorithm to compute a dense linearly conjugate realization of constrained kinetic models. The advantage of the method is that it applies linear optimization methods, and it works for every kinetic system without restrictions on the variables. This algorithm is applied as a subroutine in the algorithms presented in Theses II, III.a, III.b and IV.b. I have also shown that even if there are arbitrarily predefined upper bounds considering the variables, the set of possible reaction graph structures representing linearly conjugate realizations is the same as in the unbounded case. Therefore the computer implementations of the algorithms presented in this dissertation can work accurately. The results are described in detail in Section 3.2 and summarized in Thesis I.b.

The related publications are [51], [57] and [58].

DOI:10.15774/PPKE.ITK.2018.004

Chapter 4

Computing weakly reversible realizations

In a **reversible** reaction network every reaction is reversible, i.e. the reaction $C_i \to C_j$ takes place in the reaction network if and only if there is a reaction $C_j \to C_i$ as well. The property investigated in this section is a generalized version of reversibility, which has an influence on the dynamical properties.

Definition 4.1. *A chemical reaction network $(\mathcal{S}, \mathcal{C}, \mathcal{R})$ is called **weakly reversible** if for all complexes C_i, C_j where $i, j \in \{1, \ldots, m\}$ and $i \neq j$ it holds that the complex C_j is reachable from complex C_i through a series of reactions if and only if C_i is also reachable from C_j.*

The purpose of this section is to demonstrate a computation method for determining weakly reversible realizations, however, it is important to mention the most important dynamical properties of these realizations as well.

4.1 Dynamical properties of special weakly reversible reaction networks

For the demonstration of these relations further notions are required. Let the reaction network (Y, A_k) be defined by the sets $\mathcal{S}, \mathcal{C}, \mathcal{R}$ where $|\mathcal{S}| = n, |\mathcal{C}| = m$ and $|\mathcal{R}| = r$ hold. A **linkage class** of a directed graph is a weakly connected component, which is a maximal connected component of the graph not considering the directions of the edges. Let l denote the number of linkage classes of the reaction graph $G(Y, A_k)$.

For every reaction $C_i \to C_j$ a vector $v_k = [Y]_{.j} - [Y]_{.i} \in \mathbb{R}^n$ is defined, where k is in $\{1, \ldots, r\}$. The generated subspace of the vectors $v_1, \ldots, v_r$ is the **stoichiometric subspace** S corresponding to the reaction network, and its dimension is s. If $x_0 \in \mathbb{R}^n_+$ refers to a concentration then $(x_0 + S) \cap \mathbb{R}^n_+$ is called a positive **stoichiometric compatibility class** that contains the concentration x_0.

DOI:10.15774/PPKE.ITK.2018.004

Using these notations the **deficiency** d of the reaction network (Y, A_k) can de defined as:

$$d = m - l - s \tag{4.1}$$

The point $x^* \in \mathbb{R}^n$ is a **complex balanced equilibrium concentration** of the mass action system $\dot{x} = Y \cdot A_k \cdot \psi^Y(x)$ if $A_k \cdot \psi^Y(x^*)$ is equal to the zero vector $\mathbf{0}^m$. If this property holds for every equilibrium concentration of the mass action system then the system is called **complex balanced**. It is known from [32] that complex balance implies weak reversibility.

The following theorems summarize the most important dynamical properties of special weakly reversible reaction networks.

Theorem 4.2. Deficiency Zero Theorem [32] – original form, for weakly reversible reaction networks
A mass action system is complex balanced in the case of any positive values of reaction rate coefficients if and only if the corresponding chemical reaction network is weakly reversible and it has deficiency zero.

Theorem 4.3. Deficiency Zero Theorem [7] – application oriented form
Let us consider a mass action system for which the corresponding chemical reaction network is weakly reversible and has deficiency zero. Then, in the case of all positive values of reaction rate constants the system has in every positive stoichiometric compatibility class exactly one equilibrium concentration and it is locally asymptotically stable.

Theorem 4.4. Deficiency One Theorem [42]
Let us consider a chemical reaction network where the deficiency is δ and the number of linkage classes is l. Let δ_i for $i \in \{1, \ldots, l\}$ refer to the deficiency of the ith linkage class considered as a reaction network. Assume that $\sum_{i=1}^{l} \delta_i = \delta$ and for all indices $i \in \{1, \ldots, l\}$ $\delta_i \leq 1$ holds. If the reaction network is weakly reversible, then for any fixed values of reaction rate coefficients there exists in every positive stoichiometric compatibility class exactly one equilibrium point and at least one of them is positive.

It is presumed that weakly reversible realizations have other important properties. The **Boundedness conjecture** states that a weakly reversible mass action system has bounded trajectories in the case of any nonnegative initial state $x(0)$. And the so called **Global Attractor Conjecture** states that the equilibrium point of a complex balanced kinetic system in the corresponding positive stoichiometric compatibility class is a global attractor. The proofs of the conjectures in the general case has not been published yet, although their correctness is proven for the case of one linkage class reaction networks in [43] and [34] by the same author. It is very likely that the Global Attractor Conjecture has finally been proven, its description can be read in [12], but it has not been published yet.

DOI:10.15774/PPKE.ITK.2018.004

4.2 Structural properties

A series of reactions that is applied in Definition 4.1 is a directed path in the reaction graph, therefore it is more convenient to apply graph properties to formulate an equivalent of weak reversibility.

Definition 4.5. *A directed graph $G(V, E)$ is called **strongly connected** if for any vertices $v \in V(G)$ there is a directed path to every other vertex $w \in V(G)$ in the graph. A maximal strongly connected subgraph of a directed graph is called a **strong component** of the graph. If a strong component contains only one vertex, then it is called a **trivial strong component**.*

The vertex set of every directed graph can be uniquely partitioned into strong components, since mutual reachability defines an equivalence relation on the set of vertices, with the strong components as equivalence classes. It is trivial that this relation is reflexive and symmetric, only transitivity needs to be justified. For every pair of vertices $v_i, v_j \in V(G)$ let p_{ij} be a directed path from v_i to v_j, if there is any. (If there are several such paths, then any of them can be chosen.) Let us assume that the vertices v_i and v_j as well as the vertices v_j and v_k can be reached from each other via directed paths. Then by taking p_{ij} and then p_{jk} a directed path from v_i to v_k can be created. Similarly, a path from v_k to v_i can be obtained by concatenating the paths p_{kj} and p_{ji}. Consequently, the vertices v_i and v_k can be reached from each other and transitivity holds.

The following lemma gives a necessary and sufficient condition for weak reversibility.

Lemma 4.6. *A reaction network (Y, A_k) is weakly reversible if and only if there are no edges between different strong components of the reaction graph $G(Y, A_k)$.*

Proof. If the reaction graph has one strong component then the lemma is automatically fulfilled. Let G_1 and G_2 be two disjoint strong components of the reaction graph. Assume indirectly that there is a directed edge $\overrightarrow{v_1 v_2}$ connecting them, $v_1 \in V(G_1)$ and $v_2 \in V(G_2)$ hold, and the reaction network is weakly reversible. Then by definition there is a directed path from v_2 to v_1 as well. In this case the vertices v_1 and v_2 are equivalent according to the mutual reachability equivalence relation, the equivalence classes of which are the strong components. But it was assumed that they are in different equivalence classes, which is a contradiction.

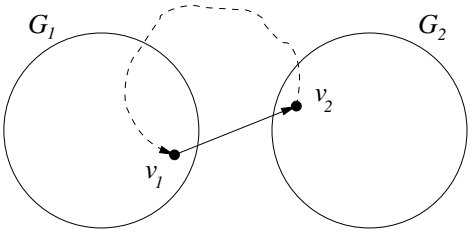


FIGURE 4.1: There are no edges between strong components.

Let us assume that there are no edges between different strong components of the reaction graph $G(Y, A_k)$. For any two vertices v_1 and v_2 it holds that if there is a directed

DOI:10.15774/PPKE.ITK.2018.004

path between them in either direction then v_1 and v_2 must be in the same strong component. Then there exists a directed path in the opposite direction as well. □

For Algorithm 2 presented in this chapter it is necessary to determine the strong components of given graphs. There are several methods for solving this problem, the most widely known are Kosaraju's algorithm (also called as Kosaraju-Sharir algorithm) see [44] and [45] and Tarjan's algorithm see [46]. These are polynomial-time algorithms based on Depth First Search.

4.3 Algorithm for computing weakly reversible realizations

There are several methods in the literature that aim to compute weakly reversible realizations of a kinetic system. Most of them apply a necessary and sufficient algebraic condition for weak reversibility, which states that a CRN is weakly reversible if and only if there is a strictly positive vector in the kernel of the Kirchhoff matrix A_k of the network. In [35] a MILP based method was introduced for computing linearly conjugate weakly reversible realizations using this condition. Later, in [36] a linear programming based method was also proposed for computing linearly conjugate weakly reversible CRN realizations. This algorithm however can be applied only within the predefined intervals of each variable and the number of required decision variables increases very fast with the number of complexes.

The motivation of the algorithm presented in this section was published by Szederkényi et al. in [22], and it computes a dense dynamically equivalent weakly reversible realization of a given kinetic system with a fixed set of complexes. It is an iterative polynomial-time method that applies graph properties and the LP optimization framework. The great advantage of this algorithm is that it requires the minimum number of variables, just the ones that are necessary to characterize a reaction network, and these variables are not necessarily considered to be bounded. This method can be extended to find a dense linearly conjugate weakly reversible realization of a kinetic system that is supplemented with a finite set of linear constraints as well. The main result of this section is the proof of correctness of the generalized algorithm, that has been proposed in [51].

The basic idea of the method is that edges between different strong components cannot occur in the reaction graph of a weakly reversible realization, see Lemma 4.6.

There are two procedures applied repeatedly during the algorithm:

FindLinConjDense$([M, Y], L, G)$ computes the dense linearly conjugate realization (T^{-1}, A_b) of the kinetic system $[M, Y]$ that also fulfils the set L of additional linear constraints and its reaction graph is a subgraph of the given directed graph G. This latter property can also be given by linear constraints, so according to Proposition 3.5 it can be computed using a polynomial-time algorithm. This method has been presented in Chapter 3 as Algorithm 1.

DOI:10.15774/PPKE.ITK.2018.004

FindCrossedges(G) returns the set of edges between the strong components of the graph G. Kosaraju's or Tarjan's algorithm is suitable, but any other polynomial-time method can also be applied to compute the strongly connected components of the graph G. Then the edges with endpoints in different strong components have to be determined. This computation requires just the checking of every edge, i.e. it can be done in $|E(G)|$ steps.

In the algorithm $G(T^{-1}, A_b)$ refers to the reaction graph representing the linearly conjugate realization (T^{-1}, A_b), $E(G)$ is the edge set of the graph G, and K_m denotes the complete directed graph on m vertices, i.e. for each pair of vertices there are edges in both directions. The complete graph is necessary only in the first application of the procedure **FindLinConjDense**, it represents that there are no restricted reactions, no additional constraints during the computation of the dense realization.

Algorithm 2 Computes a weakly reversible linearly conjugate realization
Inputs: $[M, Y], L$
Output: (T^{-1}, A_b)

1: (T^{-1}, A_b) :=**FindLinConjDense**$([M, Y], L, K_n)$
2: $G := G(T^{-1}, A_b)$
3: **while FindCrossedges**$(G) \neq \emptyset$ **do**
4: $\quad E(G) := E(G) \setminus$ **FindCrossedges**(G)
5: $\quad (T^{-1}, A_b) :=$ **FindLinConjDense**$([M, Y], L, G)$
6: $\quad G := G(T^{-1}, A_b)$
7: **end while**
8: **if** $E(G) = \emptyset$ **then**
9: $\quad$ There is no weakly reversible linearly conjugate realization.
10: **else**
11: $\quad (T^{-1}, A_b)$ is a weakly reversible linearly conjugate realization.
12: **end if**

In the first step of the algorithm a dense linearly conjugate realization is computed. It is known from [15] and the description on Section 3.1 that all other realizations can be represented by subgraphs of the reaction graph of the dense realization. If this realization is not weakly reversible, then it has edges between its strong components. Consequently, if there is a weakly reversible realization of this kinetic system, then its reaction graph cannot contain the edges connecting the strong components of the dense realization. Then a dense linearly conjugate realization without these edges is computed. If the result is again not weakly reversible, then the dense realization restricting the cross-component edges is considered in the next step. During these steps the number of possible reactions might decrease very fast, however, according to Proposition 3.3 this computation returns the correct the answer.

Proposition 4.7. *For any kinetic system, the dense weakly reversible linearly conjugate realization determines a superstructure among weakly reversible linearly conjugate realizations, and this realization can be computed in polynomial time by the application of* Algorithm 2 .

DOI:10.15774/PPKE.ITK.2018.004

Proof. Let G be the reaction graph of a weakly reversible linearly conjugate realization of the kinetic system $[M, Y]$. The graph G must be a subgraph of the reaction graph representing the dense linearly conjugate realization, which is denoted by G_0. Since there cannot be any edges between the strong components of G, and each strong component of G is a subgraph of a strong component of G_0, the cross-component edges of G_0 cannot be in the set $E(G)$ of edges in G. Since the realization described by the graph G is a linearly conjugate realization, according to Proposition 3.3 G must be a subgraph of the constrained dense linearly conjugate realization, where these edges are restricted. Let the graph G_1 represent this realization.

If there are edges connecting different strong components in the graph G_1, then these cannot be in $E(G)$ either. Therefore another realization is computed that is represented by the graph G_2 as reaction graph. The graph G_2 cannot contain the cross-component edges of G_1 and it must be a subgraph of G_1. According to its properties, G must be a subgraph of G_2.

The computation goes on until such a realization is found where there are no cross-component edges, or no edges at all, as written in Algorithm 2. If there is any weakly reversible realization, then the first case must occur, and G must be a subgraph of the graph computed by the algorithm, therefore this result determines a superstructure among weakly reversible linearly conjugate realizations, and it must be the dense one. If the second case occurs, then there exist two weakly reversible linearly conjugate realization of the kinetic system with the given set of complexes.

The computation requires the solution of at most $m^2 - m$ LP optimization problems and the same number of times the characterization of the strong components. Both methods require polynomial time, consequently the whole computation can be done in polynomial time. □

4.4 Examples

In this section the working of Algorithm 2 is demonstrated for the linearly conjugate and dynamically equivalent cases of an example from [35], that has been mentioned in [51] and in Section 2.5.2 of this thesis as Example 2 as well. The kinetic system $[M, Y]$ is defined by the matrices M and Y:

$$M = \begin{bmatrix} 1 & 0 & 0 & -2 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & -3 & 0 & 0 & 0 \end{bmatrix}$$

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

DOI:10.15774/PPKE.ITK.2018.004

4.4.1 Weakly reversible dynamically equivalent realization

For finding a dense weakly reversible dynamically equivalent realization, first the algorithm determines the dense dynamically equivalent realization $[Y, A_{k1}]$ of the kinetic system $[M, Y]$, where

$$A_{k1} = \begin{bmatrix} -1 & 0 & 0 & 0 & 0 & 0 & 5e-1 & 0 & 0 & 0 \\ 1 & -1 & 1e7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & -2e7 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1e7 & -3 & 0 & 1e7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 & 0 & 0 & 0 & 0 & 1e7 \\ 0 & 0 & 0 & 1 & 0 & -2e7 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & -1.00000035e7 & 0 & 1e7 & 1e7 \\ 0 & 0 & 0 & 0 & 0 & 1e7 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1e7 & 0 & -1e7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 & -2e7 \end{bmatrix}$$

Then the strong components of its reaction graph are determined. The reaction graph structure can be seen in Figure 4.2, where the edges between different strong components are indicated with dashed lines.

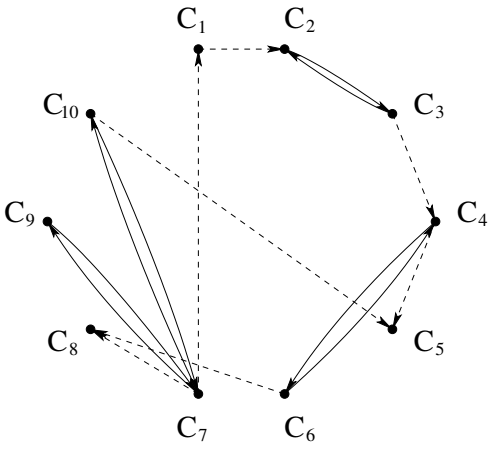


FIGURE 4.2: Reaction graph structure of the dense dynamically equivalent realization $[Y, A_{k1}]$.

The edges connecting different strong components cannot be included in any weakly reversible realization of this kinetic system due to the superstructure property of the dense realization. Therefore in the second step a constrained dense realization is computed, the constraints are defined in order to exclude the reactions $C_1 \to C_2$, $C_3 \to C_4$, $C_4 \to C_5$, $C_6 \to C_8$, $C_7 \to C_1$, $C_7 \to C_8$ and $C_{10} \to C_5$. However, it turns out that the constrained optimization problem is infeasible, therefore there is no weakly reversible dynamically equivalent realization of the kinetic system $[M, Y]$.

DOI:10.15774/PPKE.ITK.2018.004

4.4.2 Weakly reversible linearly conjugate realization

The computation of a weakly reversible linearly conjugate realization begins with the characterization of the dense linearly conjugate realization (Y, A_{k2}) of the kinetic system $[M, Y]$, where A_{k2} is the Kirchhoff matrix and T^{-1} is the state transformation matrix corresponding to the realization. Using the previously introduced notation of linearly conjugate realizations it would be referred to as (T^{-1}, A_{b2}), where the transformed Kirchhoff matrix A_{b2} is equal to $A_{k2} \cdot \Psi_T^{-1}$

$$A_{k2} = \begin{bmatrix} -5.7e3 & 0 & 0 & 0 & 0 & 0 & 7.1e2 & 0 & 0 & 0 \\ 5.7e3 & -2.9e3 & 7.1e6 & 0 & 0 & 0 & 7.1e2 & 0 & 0 & 0 \\ 0 & 1.4e3 & -1.4e7 & 0 & 0 & 0 & 1.4e3 & 0 & 0 & 0 \\ 0 & 1.4e3 & 7.1e6 & -1.4e4 & 0 & 7.1e6 & 1.4e3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8.6e3 & 0 & 0 & 4.6e3 & 0 & 0 & 1.4e7 \\ 0 & 0 & 0 & 1.4e3 & 0 & -1.4e7 & 7.1e3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7.1e3 & 0 & 0 & -1.4e7 & 0 & 1.4e7 & 7.1e6 \\ 0 & 0 & 0 & 7.1e3 & 0 & 7.1e6 & 1.4e3 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7.1e3 & 0 & 0 & 1.4e7 & 0 & -1.4e7 & 7.1e6 \\ 0 & 0 & 0 & 1.4e3 & 0 & 0 & 2.9e3 & 0 & 0 & -2.9e7 \end{bmatrix}$$

$$T_2^{-1} = \begin{bmatrix} 5.7143e3 & 0 & 0 \\ 0 & 4.2857e3 & 0 \\ 0 & 0 & 7.1429e3 \end{bmatrix}$$

For this graph as well the strong components are defined, and the edges between different strong components are drawn with dashed lines.

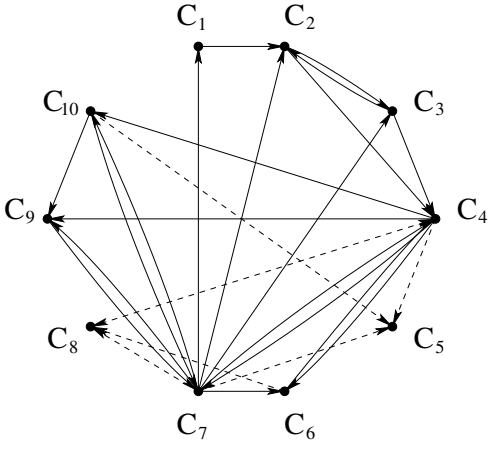


FIGURE 4.3: Reaction graph structure of the dense linearly conjugate realization (Y, A_{k2}).

Because of the superstructure property of the dense linearly conjugate realization the reactions between different strong components cannot be present in a weakly reversible

DOI:10.15774/PPKE.ITK.2018.004

realization. Therefore, in the second step a constrained dense linearly conjugate realization is computed, that does not contain these reactions. The computation returns the reaction network (Y, A_{k3}), where

$$A_{k3} = \begin{bmatrix} -2.1e3 & 0 & 0 & 0 & 0 & 0 & 2e3 & 0 & 0 & 0 \\ 2.1e3 & -3.6e3 & 7.1e6 & 0 & 0 & 0 & 7.1e2 & 0 & 0 & 0 \\ 0 & 1.4e3 & -1.4e7 & 0 & 0 & 0 & 3.6e2 & 0 & 0 & 0 \\ 0 & 2.1e3 & 7.1e6 & -5.7e3 & 0 & 0 & 7.1e2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1.4e3 & 0 & 0 & 3.6e2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7.1e2 & 0 & 0 & -1.4e7 & 0 & 1.4e7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7.1e2 & 0 & 0 & 1.4e7 & 0 & -1.4e7 & 0 \\ 0 & 0 & 0 & 2.9e3 & 0 & 0 & 1.4e4 & 0 & 0 & 0 \end{bmatrix}$$

$$T_3^{-1} = \begin{bmatrix} 2.1429e3 & 0 & 0 \\ 0 & 5.7149e3 & 0 \\ 0 & 0 & 7.1429e3 \end{bmatrix}$$

This realization is also not weakly reversible, there are reactions between complexes from different strong components,

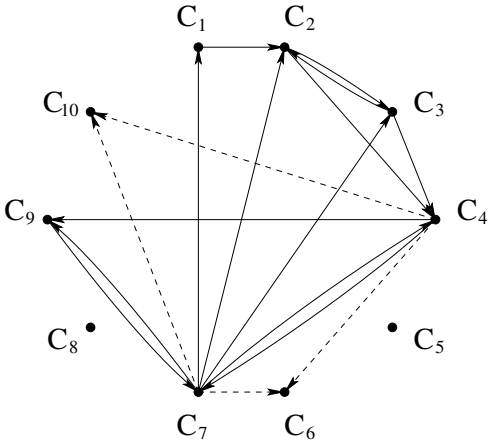


FIGURE 4.4: Reaction graph structure of the reaction network (Y, A_{k3}) computed in the second step of the algorithm.

consequently in the next step a constrained realization has to be computed, where the set of constraints is complemented by the restriction of the cross reactions of the previous result. The set of constraints is defined to omit the reactions $C_4 \to C_5$, $C_4 \to C_8$, $C_6 \to C_8$, $C_7 \to C_5$, $C_7 \to C_8$ and $C_{10} \to C_5$ based on the realization (Y, A_{k2}), and to omit the reactions $C_4 \to C_6$, $C_4 \to C_{10}$, $C_7 \to C_6$ and $C_7 \to C_{10}$ because of the realization (Y, A_{k3}).

Since in each step a (constrained) dense realization is computed, by the superstructure property of these realizations it follows that it is enough to restrict just the union of

DOI:10.15774/PPKE.ITK.2018.004

the reactions that have been declared as cross-component edges in this or in any of the previous steps.

The solution of the optimization problem is the realization (Y, A_{k4}) defined by the matrices

$$A_{k4} = \begin{bmatrix} -5.5e2 & 0 & 0 & 0 & 0 & 0 & 2.7e3 & 0 & 0 & 0 \\ 5.5e2 & -4e3 & 1e7 & 0 & 0 & 0 & 1.7e2 & 0 & 0 & 0 \\ 0 & 2e3 & -2e7 & 0 & 0 & 0 & 1.8e2 & 0 & 0 & 0 \\ 0 & 2e3 & 1e7 & -1.1e3 & 0 & 0 & 2e2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 7.3e2 & 0 & 0 & -2.0e7 & 0 & 2e7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 3.6e2 & 0 & 0 & 2.0e7 & 0 & -2e7 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$T_4^{-1} = \begin{bmatrix} 5.4848e2 & 0 & 0 \\ 0 & 6e3 & 0 \\ 0 & 0 & 2.1940e3 \end{bmatrix}$$

This realization has 5 strong components, four trivial components and a non-trivial one, while there are no edges connecting them, therefore it is a weakly reversible realization. Interestingly, there are exactly those reactions present in this realization that are in (Y, A_{k3}), only the restricted reactions are missing.

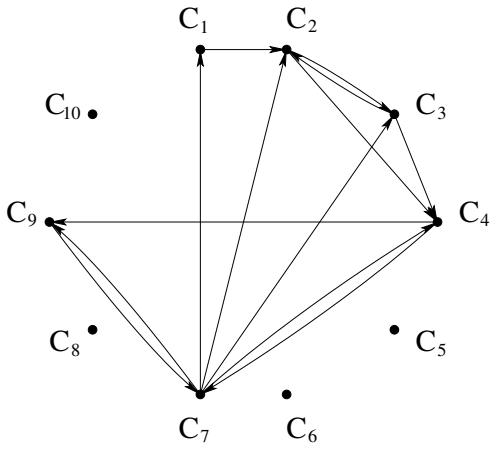


FIGURE 4.5: Reaction graph describing the realization computed in the second step of the algorithm.

It follows that there exists a linearly conjugate weakly reversible realization but no dynamically equivalent weakly reversible realization of the kinetic system $[M, Y]$.

DOI:10.15774/PPKE.ITK.2018.004

It can be seen from the example that linear conjugacy may significantly increase the number and extend certain important properties of reaction graph structures corresponding to a given kinetic system in comparison to dynamical equivalence.

4.5 Summary

I have proposed a new polynomial-time algorithm for computing a weakly reversible linearly conjugate realization of a kinetic system by extending the method introduced in [22]. I have proven that the CRN returned by the algorithm is a dense weakly reversible linearly conjugate realization of the kinetic system, if there exists such a realization. I have also shown that the returned realization defines a superstructure among all linearly conjugate weakly reversible realizations of the kinetic system. The results are summarized in Thesis II.

The related publications are [51], [57] and [58].

DOI:10.15774/PPKE.ITK.2018.004

Chapter 5

Computing all possible reaction graph structures

It is known that most kinetic systems have several different realizations. This rises the question: Is it possible to determine all these realizations? As it has been shown in Corollary 3.7 there might be a continuum of dense realizations of a kinetic system. Similarly, it is possible that there are other different but structurally identical realizations as well. Therefore, the more reasonable and better formulated problem discussed in this chapter is the computation of all possible reaction graph structures representing linearly conjugate realizations of a kinetic system $[M, Y]$ with a fixed set of complexes.

Two different algorithms are presented in Sections 5.1 and 5.2 to compute the possible structures. Since the aim of these methods is the same, some of the applied notations are identical. These notations are introduced below.

During the computations the reaction graph structures are represented by binary sequences, that decode the presence of the edges like a characteristic vector. For simplicity in the notations R refers to such a binary sequence and also the corresponding linearly conjugate realization determined during the computation. The reaction graph structure of the realization R is denoted by G_R, and $E(G_R)$ refers to the set of edges of this graph.

According to Proposition 3.3 a reaction can be present in any of the linearly conjugate realizations of the kinetic system $[M, Y]$ if and only if this reaction takes place also in the dense linearly conjugate realization of the same kinetic system. Consequently, the edges which are not present in the dense realization do not require representation. It was proven in Proposition 3.5 that the dense realization can be computed using a polynomial-time algorithm, and this is the initial step of both methods.

There might exist so-called **core reactions** as well, that are present in every linearly conjugate realization of the kinetic system. It is possible that there are no core reactions of a kinetic system in the case of a given set of complexes, but also there might be several such reactions. The core reactions of the kinetic system $[M, Y]$ are represented by the **core edges** that form the set $E_c[M, Y]$. There are several methods for determining the set of core edges, see [39] or for a faster method [47]. A generalized version of the

DOI:10.15774/PPKE.ITK.2018.004

latter one is presented also in this work as Algorithm 7. The computation of the core reactions is not necessary for the correct operation of the algorithm, but performing this step might save computational time and space. Therefore this is the second step in both methods.

Based on the above, every reaction graph structure representing a linearly conjugate realization is uniquely determined if it is known which of the non-core reactions of the dense realization are present in the realization. Let the dense realization and its reaction graph structure be denoted by D and G_D, respectively. Then the required length of the binary sequences applied for the representation is $q = |E(G_D) \setminus E_c|$. For the definition of the sequences an ordering of the non-core edges needs to be fixed. Let e_i be the ith edge and let $R[i]$ denote the ith coordinate of the sequence R corresponding to this edge. The formal definition of the coordinates of a sequence is

$$e \in E(G_R) \iff \begin{cases} \exists i \; e = e_i, \; R[i] = 1 \\ \quad \text{or} \\ e \in E_c \end{cases} \tag{5.1}$$

From now on the term 'sequence' will refer to such a binary sequence of length q. It is easy to see that using the notation of the sequences the dense realization is characterized by the sequence $\mathbf{1}_q$ with all coordinates equal to 1.

During both algorithms the sequences are temporarily stored in indexed stacks. The rules that determine which sequences might be stored in a particular stack are different but the data structures are defined identically. The stack with index k is referred to as $S(k)$, and the number of elements in this stack is denoted as $|S(k)|$. At the beginning all stacks are empty, but during the running of the algorithm sequences are pushed in and popped out from them. The command 'push R into $S(k)$' pushes the sequence R into the stack $S(k)$, and the command 'pop from $S(k)$' pops a sequence out from $S(k)$ and returns it. (By the definition of the data structure the sequence pushed in last will be popped out first.)

The discovered graph structures are stored in a binary array of size 2^q called $Exist$, where the indices of the fields are the sequences as binary numbers. At the beginning the value in each field is zero, and after the computation the value of the field $Exist[R]$ is 1 if and only if there is a linearly conjugate realization that can be represented by the sequence R.

5.1 Stacking algorithm for computing all reaction graph structures

The algorithm presented in this section is the first method in the literature to compute all reaction graph structures, presented in [52]. The basic idea for the dynamically equivalent case comes from Prof. Zsolt Tuza, but the detailed formulation and extension

DOI:10.15774/PPKE.ITK.2018.004

to the linearly conjugate case as well as the proof of correctness is the work of the author of this thesis.

During the computation the binary sequence representation and indexed stacks are applied, while the results are stored in a binary array, as defined earlier. In this method the rule of storing a sequence R in a stack depends on the number of edges in the reaction graph G_R, which is referred to as $e(R)$. The sequence R is stored in the stack $S(k)$ at some point during the computation if and only if $e(R) = k$ holds, consequently the stacks are indexed from 1 to q.

Within the algorithm the following procedure is used repeatedly:

FindLinConjWithoutEdge$([M, Y], R, i)$ computes a constrained dense linearly conjugate realization of the kinetic system $[M, Y]$ with a fixed set of complexes. It is an application of Algorithm 1. The additional inputs R and i are a sequence encoding the input reaction graph structure, and an integer index, respectively. The procedure returns a sequence U encoding the computed linearly conjugate realization for which the reaction graph G_U is a subgraph of G_R and $U[i] = 0$ holds. If there exist no such realization then the sequence $\mathbf{0}_q$ with all coordinates equal to zero is returned. This computation can be performed in polynomial time as it was proven in Proposition 3.5.

Algorithm 3 Stacking algorithm to determine all reaction graph structures
input: $[M, Y]$, q
output: $Exist$

```
1: push 1_q into S(q)
2: Exist[1_q]:=1
3: for k = q to 1 do
4:     while |S(k)| > 0 do
5:         R:= pop S(k)
6:         for i = 1 to q do
7:             if R[i] = 1 then
8:                 U := FindLinConjWithoutEdge([M, Y], R, i)
9:                 if U ≠ 0_q and Exist[U] = 0 then
10:                    Exist[U]:= 1
11:                    push U into S(e(U))
12:                end if
13:            end if
14:        end for
15:        Print R
16:    end while
17: end for
```

The algorithm starts with the computation of the dense realization and its binary sequence $\mathbf{1}_q$ is put into stack $S(q)$. Then in a for loop the stacks are taken according to decreasing order of indices from q to 1. In each step a sequence R is popped out from

DOI:10.15774/PPKE.ITK.2018.004

the actual stack $S(k)$, until there is any, and for all of its reactions it the computation of a realization U is attempted where the given reaction is not present and G_U is a subgraph of G_R. If there exists such a realization then the binary sequence representing it is pushed into the stack $S(l)$ for which $l = e(U)$ holds. For every determined reaction graph structure its existence is saved in the array *Exist*, which is the output of the algorithm, and it is applied also to avoid doing the same computation multiple times.

It can be proven that the presented method is correct and the search is indeed exhaustive.

Proposition 5.1. *For any kinetic system $[M, Y]$ with any fixed set of complexes all the possible reaction graph structures representing linearly conjugate realizations can be computed after finitely many steps by* Algorithm 3. *The whole computation might require exponential time depending on the large number of possible directed graphs, but the time elapsed between the displaying of two linearly conjugate realizations is always polynomial.*

Proof. Let us assume indirectly that there is a sequence W which is not returned by the algorithm, but it corresponds to a linearly conjugate realization of the kinetic system $[M, Y]$. Let R represent another realization, which was computed by the algorithm and G_W is a subgraph of G_R, i.e. if $W[i] = 1$ then $R[i] = 1$ for all $i \in \{1, \ldots, q\}$. There must be such a realization since the dense realization D fulfils the requirements for each W, and if there is more than one such realization, then let R be the one with minimum number of reactions. Since the sequences W and R cannot be identical there exists an index j so that $W[j] = 0$ and $R[j] = 1$ hold. During the computation there is a point (line 8) when the procedure **FindLinConjWithoutEdge**$([M, Y], R, j)$ is applied. Since the realization W fulfils the constraints, the computed sequence U is not $\mathbf{0}_q$ and it represents the dense realization with the prescribed properties. Consequently, G_W is a subgraph of G_U. Since it was assumed that W is not returned by the algorithm the sequences U and W cannot be identical. From this it follows that R is not minimal, which is a contradiction.

The computation will eventually come to an end since every sequence R is put only once and into only one of the stacks, that is $S(e(R))$. The number of optimization steps considering the sequence R as input is exactly $e(R)$. The number of sequences with the same number of edges as R is at most $\binom{q}{e(R)}$. Consequently, the number of required optimization steps is at most $\sum_{i=1}^{q} i \cdot \binom{q}{i} = q \cdot 2^{q-1}$.

The computation between the printing of two consecutive sequences requires the application of the linear optimization step **FindLinConjWithoutEdge**$([M, Y], R, i)$ at most $e(R) \leq q$ times, with some additional minor computation. Therefore the time elapsed is always constant times polynomial. □

5.1.1 Computing dynamically equivalent realizations using smaller parallel steps

As it has been presented in [52], in the case of dynamically equivalent realizations it is possible to split the computation steps in smaller units that can be done simultaneously.

DOI:10.15774/PPKE.ITK.2018.004

These realizations are special linearly conjugate realizations where the transformation matrix T is the unit matrix. The variables of the model are are the entries of matrix A_k, and Equation (2.14) characterizing the desired dynamical properties can be written in a more simple form:

$$Y \cdot A_k = M \tag{5.2}$$

It is easy to see that the values of the variables in the jth column of matrix A_k depend only on the parameters in the jth column of matrix M and on the entries of matrix Y. Therefore the columns of A_k might be computed simultaneously, and any possible reaction graph structure corresponding to a dynamically equivalent realization can be determined by choosing a possible value in the case of each column and building the Kirchhoff matrix of the realization from them. It follows that the number of dynamically equivalent realizations, that define different reaction graph structures, is the product of the numbers of the possible column structures.

The super-structure property of dense realizations is inherited by the columns of the matrix A_k, therefore the same algorithm can be applied for the computation of columns as was used for determining linearly conjugate realizations.

However, in this version of the algorithm it is better to determine the ordering of non-core edges according to columns. Let N_j denote the number of non-core edges in column j, and let D_j refer to the sequence describing the jth column of the dense dynamically equivalent realization. The stacks are also needed to be defined separately for each column. The sequence R_j representing a jth column gets stored in stack $S_j(k)$ if and only if the number of coordinates equal to 1, denoted by $e(R_j)$, is exactly k.

The computed sequences are stored during the algorithm in a two-dimensional binary array called $ExistColumn[j, R_j]$. The first index refers to the column and the second index is the sequence as a binary number. At the beginning all coordinates are equal to zero.

The applied procedures are as follows:

DyneqColumnWithoutEdge$([M, Y], j, R_j, i)$ computes the jth column of the Kirchhoff matrix describing a constrained dense dynamically equivalent realization of the kinetic system $[M, Y]$. The constraints are determined by the two last inputs, a sequence R_j and an integer index i. The procedure returns a sequence U_j representing a jth column so that $U_j[l] = 0$ if $R_j[l] = 0$ for all $l \in \{1, \dots N_j\}$ and also $U_j[i] = 0$ hold. If there is no such column, then -1 is returned. This computation can be performed in polynomial time.

Build$A_k(ExistColumn)$ builds all possible dynamically equivalent realizations from the sequence parts in $ExistColumn$ and saves them in the array $Exist$.

DOI:10.15774/PPKE.ITK.2018.004

Algorithm 4 Computes all dynamically equivalent realizations
input: $[M, Y],\ N_1, \ldots, N_m$
output: $Exist$

```
1: for j = 1 to m do
2:     push D_j into S_j(N_j)
3:     for k = N_j to 1 do
4:         while size.S_j(k) > 0 do
5:             R_j:= pop S_j(k)
6:             for i = 1 to N_j do
7:                 if R_j[i] = 1 then
8:                     U_j := DyneqColumnWithoutEdge([M, Y], j, R_j, i)
9:                     if U_j ≥ 0 and ExistColumn[j, U_j] = 0 then
10:                        ExistColumn[j, U_j]:=1
11:                        push U_j into S_j(e(U_j))
12:                    end if
13:                end if
14:            end for
15:        end while
16:    end for
17: end for
18: BuildA_k(ExistColumn)
```

5.1.2 Parallel implementation of Algorithm 3

It is possible to apply parallel implementation of Algorithm 3 for computing linearly conjugate realizations as well, since the results of the optimization steps **FindLinConjWithoutEdge**$([M, Y], R, i)$ considering a fixed sequence R have no effect on each other. It holds for the newly computed realizations that the number of reactions is less than in the input realization, therefore when the procedure **FindLinConjWithoutEdge**$([M, Y], R, i)$ is applied there would be no more sequences put into the stack $S(e(R))$. Consequently it is also possible to do the optimization steps for different sequences at the same time. However, the recording of the computed reaction graph structures has to be done sequentially.

The details of the implementation were presented in [55]. The computations were carried out on a Lenovo D60 workstation with two 2.60GHz Xeon (E5-2650 v2) processors and with 32 Gb RAM (DDR3 1600 MHz, 0.6ns). The software was written in Python (ver. 2.7.6). Additionally, Python packages such as pyzmq (ver. 14.7.0), cyLP (0.7.2), Cython (ver. 0.23.4) and CBC (ver. 2.8.5) were used. The linear programs were solved with the CLP solver, which is part of CBC.

The efficiency of the parallel implementation has been tested in the case of two kinetic system models. Example 4 was introduced previously in Section 3.3.1, while Example 6 is taken from [48] and models a switch-like behaviour in yeast cell cycle regulations. The details of the model can be found in the original paper. The computation has returned

DOI:10.15774/PPKE.ITK.2018.004

that Example 4 has 17 160 structurally different linearly conjugate realizations on a fixed set of 6 complexes, while in the case of Example 6 there are 721 possible realizations that are defined on a set of 19 complexes.

The implementation was tested on both examples with different numbers of parallel processes (workers). Figure 5.1 depicts the total computation times calculated as the average of the execution times of the individual workers. The numbers in the brackets are the standard deviations which indicate that the work load is evenly distributed among the workers.

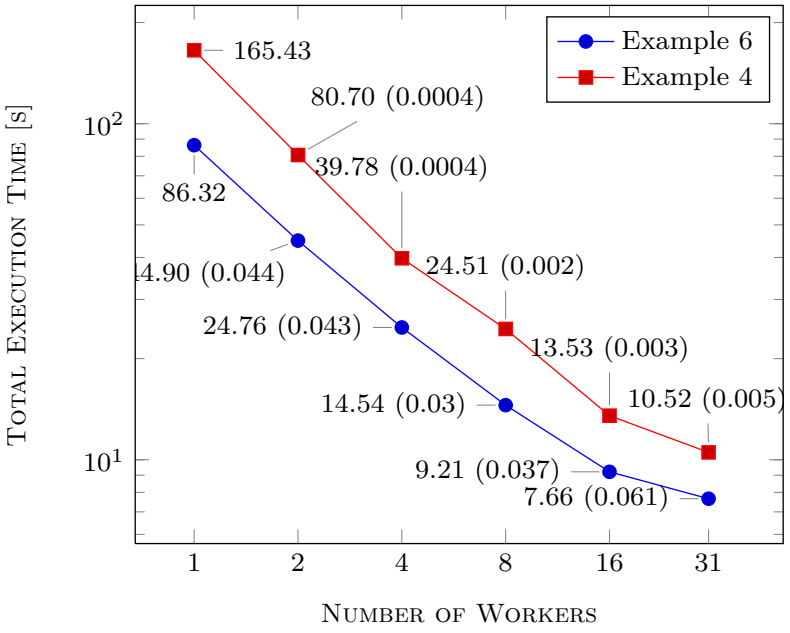


Figure 5.1: Average execution times of the workers in the cases of Examples 6 and 4.

5.2 Sequencing algorithm for computing all reaction graph structures

The algorithm presented in this section is another method for computing all reaction graph structures representing linearly conjugate realizations of a kinetic system $[M, Y]$ with a fixed set of complexes. Its development as well as the proof of correctness and efficiency is the work of the author of this thesis. The results were presented in [53].

In this method the starting slices of the sequences have a distinguished role. For every index $k \in \{1, \dots q\}$ a special equivalence relation is defined on the set of sequences, so that $R =_k R'$ holds if and only if for all indices $i \in \{1, \dots k\}$ the coordinates $R[i]$ and $R'[i]$ are equal, i.e. the first slices of length k in the sequences are the same. The equivalence class of the relation $=_k$ containing the sequence R is referred to as $C_k(R)$. It has to be mentioned that for every equivalence class any of its elements can be assigned as representative element.

DOI:10.15774/PPKE.ITK.2018.004

During the algorithm $q+2$ pieces of stacks are applied, that are indexed from 0 to $q+1$. The definition and notations of the stacks is the same as before, but the rule of storing the sequences is different. The sequence R might be stored in stack $S(k)$ at some point during the computation only if R represents the dense realization in $C_k(R)$. According to the definition the stacks $S(0)$, $S(q)$ and $S(q+1)$ have no use, but these have a technical role in the algorithm, as it will be shown later.

The property of having some set of reactions and have another set forbidden can be ensured by linear constraints. It can be given as some elements of the matrix A_k are strictly positive and others are zero. Because of the non-strict inequalities it cannot properly be included in the LP model, but the proof of Proposition 3.3 can still be applied, since it was not supposed there that all the halfplanes should be closed. Consequently, if a sequence R is in the stack $S(k)$ then R defines a superstructure among the sequences in $C_k(R)$. More formally

$$[R \in S(k),\ W \in C_k(R)\ ] \Longrightarrow [\ \forall j \in \{1,\dots,q\}\quad W[j] \leq R[j]\ ] \tag{5.3}$$

The binary array *Exist* defined as before is also applied during the computation. As it will be shown in Proposition 5.3 the application of this array is not necessary for the running of the algorithm, since none of the possible sequences is returned multiple times. Furthermore, at the end of the computation all the computed sequences are contained by the stacks $S(q)$ and $S(q+1)$, but it might be more convenient to sum up the results of the algorithm in a single data structure that allows easy access to the individual structures.

It will be shown in Section 6.2.3 that the application of the indexed stacks can also be avoided and one stack is sufficient. This approach makes the application of parallel implementations possible.

Within the algorithm two procedures are applied:

FindLinConjWithZeros$([M,Y],R,k,i)$ computes a dense linearly conjugate realization of the kinetic system $[M,Y]$ where all those coordinates are zero, that have indices at most k and are zero in the sequence R, or are indexed from $k+1$ to i. If there exists such a realization of the kinetic system, then the sequence W representing it is generated and it is compared to R. If $W =_k R$ holds then the procedure returns W. If there is no realization fulfilling the given constraints or the computed realization does not belong to $C_k(R)$ then the procedure returns -1.
(The comparison step is necessary since there might be more zero coordinates corresponding to the indices 1 to k of W than required, and in this case the computed realization will turn up as the result of the procedure with a different input sequence R' as well.)

FindNextOne(R,k) returns the smallest index i for which $k<i$ and $R[i]=1$ hold. If there is no such index, i.e. $R[j]$ is zero for all $k<j$, then $q+1$ is returned.

DOI:10.15774/PPKE.ITK.2018.004

Algorithm 5 Sequencing algorithm to determine all reaction graph structures
input: $[M, Y]$, q
output: $Exist$

```
1:  push 1_q into S(0)
2:  Exist[1_q] := 1
3:  for k = 0 to q − 1 do
4:      while |S(k)| > 0 do
5:          R := pop from S(k)
6:          i := FindNextOne(R, k)
7:          push R into S(i)
8:          while i ≤ q do
9:              W := FindLinConjWithZeros([M, Y], R, k, i)
10:             if W < 0 then
11:                 BREAK
12:             else
13:                 i := FindNextOne(W, i)
14:                 push W into S(i)
15:                 Exist[W] := 1
16:             end if
17:         end while
18:     end while
19: end for
```

The computation starts by putting the sequence $\mathbf{1}_q$ representing the dense realization into the stack $S(0)$ and also saving it in the array $Exist$. Then the stacks are taken in increasing order of indices from zero to $q-1$. In the general step a sequence R is popped out from the actual stack $S(k)$ as long as it is not empty. If the index $i = \mathbf{FindNextOne}(R, k)$ is smaller than $q+1$, then by the definitions it follows that R defines a superstructure not only in $C_k(R)$ but in $C_i(R)$ as well. Therefore R is put into stack $S(i)$ in order to save it for further examination. If there is no index i greater than k for which $R[i] = 1$ holds, then i is equal to $\mathbf{FindNextOne}(R, k) = q+1$. In this case R defines a superstructure in $C_q(R)$, but in order to avoid further complications it is put into $S(i) = S(q+1)$ and the examination of sequence R finished. This step is the reason why all the sequences with last coordinate equal to zero are all stored in the stack $S(q+1)$ after the computation. All the other computed sequences, with last coordinate equal to one, are saved in the stack $S(q)$.

If i is smaller than $q+1$, in the while loop in lines $8-17$ the possible zero gaps starting from index $k+1$ in the sequences of $C_k(R)$ are examined. Since the values corresponding to the indices between k and i are all known to be zero, at the first attempt the largest index of the gap should be i. If there is no suitable realization, i.e. the procedure $\mathbf{FindLinConjWithZeros}(R, k, i)$ returns -1, then there cannot be any realization with $i-k$ or more consecutive zeros following the coordinate k in $C_k(R)$. In this case the examination of R is finished and another sequence is taken into account, if there is any.

DOI:10.15774/PPKE.ITK.2018.004

If a real sequence W is returned by the procedure, index $i = \mathbf{FindNextOne}(W, k)$ is looked up, that is equal to $\mathbf{FindNextOne}(W, i)$ in the case of the actual value of the index i. Since W is a dense realization in $C_i(W)$, it is put into $S(i)$ and saved in array *Exist* as well. Then, if i is smaller than $q+1$, it is attempted to find a realization with at least one more consecutive zeros following the coordinate k than W has.

The computation stops when there are no more sequences in stacks with indices smaller than q. At this point all possible sequences representing linearly conjugate realizations of the kinetic system are stored in the stacks $S(q)$ and $S(q+1)$, and it can be proven that this computation is accurate.

Proposition 5.2. *For any kinetic system $[M, Y]$ with a fixed set of complexes all the possible reaction graph structures representing linearly conjugate realizations can be determined by* Algorithm 5 *after finitely many steps.*

Proof. Let us assume indirectly that there is a linearly conjugate realization represented by the sequence V which is not returned by the algorithm. Let R be another sequence, which was returned by the algorithm, it was in stack $S(p)$ at some point during the computation, $V =_p R$ holds and p is the greatest such number. There must be such a sequence, since the dense realization D and $p = 0$ meet the conditions. If $p = q$ or $q+1$, then V is equal to R, consequently it is returned by the algorithm, so it can be assumed that $p \leq q - 1$ holds.

There is a point during the computation when the sequence R is popped out from the stack $S(p)$. Let us assume, that $\mathbf{FindNextOne}(R, p) = i$ and $\mathbf{FindNextOne}(V, p) = j$ hold. It follows from the superstructure property of R that i can be at most j.

If j is equal to i then $V =_i R$ holds. But in this case at some point of the computation sequence R is in stack $S(i)$, which means that p is not maximal, and it is a contradiction.
If $i < j$ holds then the procedure $\mathbf{FindLinConjWithZeros}(R, p, i)$ is applied first during the examination of the sequence R. Since the realization V fulfils the constraints, the procedure must return a sequence W_1. W_1 represents the dense realization in $C_i(W_1)$, but $V \in C_i(W_1)$ and $V[j] = 1$ hold, therefore $W_1[j]$ is also equal to 1 and $\mathbf{FindNextOne}(W_1, p) = j_1 \leq j$ must be true.

If $j_1 = j$ holds, then $W_1 =_j V$ and W_1 is in stack $S(j)$ at some point of the computation. This means that p is not maximal, which is again a contradiction.
If $j_1 < j$ holds, then the procedure $\mathbf{FindLinConjWithZeros}(R, p, j_1)$ is applied. Since the realization represented by sequence V fulfils the constraints, it returns a sequence W_2 for which $\mathbf{FindNextOne}(W_2, p) = j_2 \leq j$ holds. If j_2 is equal to j, then it is a contradiction, otherwise the computation can be continued similarly as before.

These steps either lead to a contradiction for p not being maximal or result an infinite increasing sequence of integers $j_1, j_2, \ldots$ which has an upper bound q, and it is again a contradiction. This means that there cannot be any sequence V which represents a linearly conjugate realization of the kinetic system that is not returned by the algorithm.

DOI:10.15774/PPKE.ITK.2018.004

The total computation time can be well described by the number of optimization steps, i.e. the application of the procedure **FindLinConjWithZeros**. An optimization step concerning a realization is performed only when it is in a stack,at most as many times as the number of not fixed coordinates. In stack $S(k)$ there might be at most 2^k different sequences, therefore a very rough upper bound on the number of optimization steps is $\sum_{k=0}^{q}(q-k)\cdot 2^k = 2^{q+1}-q-2$. □

Proposition 5.3. *Within the computation according to* Algorithm 5 *no realization is returned twice by the procedure* **FindLinConjWithZeros**.

Proof. Assume by contradiction that there is a sequence W which is computed twice during the algorithm, i.e. there are sequences R_1 and R_2, and integers k_1, k_2, i_1 and i_2 so that the objects in similar positions are not all identical, and the following holds:

$$W = \textbf{FindLinConjWithZeros}(R_1, k_1, i_1) = \textbf{FindLinConjWithZeros}(R_2, k_2, i_2) \tag{5.4}$$

It can be assumed that $k_1 \leq k_2$ holds, and according to this relation two cases can be distinguished.

First let us assume that k_1 and k_2 are equal. It comes from the working of the algorithm that $R_1 =_{k_1} W$ holds and since R_1 is in stack $S(k_1)$ at some point of the computation, it defines a superstructure in $C_{k_1}(W)$. Similarly it follows that R_2 defines a superstructure in $C_{k_2}(W) = C_{k_1}(W)$, and on account of the uniqueness of the superstructure the sequences R_1 and R_2 must be identical. There must be some difference among the inputs, therefore it can be assumed that $i_1 < i_2$ holds. Both optimization steps are done when the sequence R_1 is popped out from the stack $S(k_1)$, but the smaller index i_1 is applied first. The result of the optimization is the sequence $W = \textbf{FindLinConjWithZeros}(R_1, k_1, i_1)$ for which the next nonzero coordinate after k_1 is at the index $j = \textbf{FindNextOne}(W, k_1)$. If j equal to q or $q+1$ then the the examination of the sequence R_1 from the coordinate k_1 is finished, and the procedure $\textbf{FindLinConjWithZeros}(R_1, k_1, i_2)$ is not applied. In the case of a smaller j the procedure $\textbf{FindLinConjWithZeros}(R_1, k_1, j)$ is applied and it returns either -1 or a proper sequence V. In the first case the examination is finished. Otherwise, the returned sequence V cannot be equal to W, since $W[j]$ is one but $V[j]$ is zero. This property is fulfilled in the case of every larger value of the index j, therefore $i_1 < i_2 < j$ must hold, which implies that in this case during the algorithm the procedure $\textbf{FindLinConjWithZeros}(R_1, k_1, i_2)$ is not called. Consequently, if $k_1 = k_2$ holds, then R_1 is identical to R_2 and assuming $i_1 < i_2$ in all cases the procedure $\textbf{FindLinConjWithZeros}(R_1, k_1, i_2)$ is not called, which is a contradiction.

Now it can be assumed that k_1 is smaller than k_2. From the definitions it follows that $R_1 =_{k_1} W$ holds, at some point of the computation R_1 is in the stack $S(k_1)$ and therefore $R_1[k_1]$ is 1. Similarly it follows that $R_2 =_{k_2} W$ and $R_2[k_2] = 1$ hold. Consequently, R_1 and R_2 are in the equivalence class $C_{k_1}(R_1) = C_{k_1}(W)$, where R_1 represents the dense realization. From this it follows that $R_1[k_1] = R_2[k_1] = R_1[k_2] = R_2[k_2] = 1$ holds, and $W[k_1] = W[k_2] = 1$ is also true. As it was assumed the procedure

DOI:10.15774/PPKE.ITK.2018.004

FindLinConjWithZeros(R_1, k_1, i_1) returns W. But since there are consecutive zero coordinates from index k_1+1 to i_1 in the sequence W and $W[k_2] = 1$ holds, the inequality $k_1 < i_1 < k_2$ must be true and $j =$ **FindNextOne**(W, k_1) can be at most k_2. Since $W =_{k_2} R_2$ holds, $W =_j R_2$ is also true. According to the algorithm the sequence W is in the stack $S(j)$ at some point of the computation, therefore the reaction graph G_{R_2} must be a subgraph of graph G_W. It leads to contradiction, since as results the procedure returns a realization with less coordinates equal to 1, therefore from the result **FindLinConjWithZeros**$(R_2, k_2, i_2) = W$ it should follow that G_W is a subgraph of G_{R_2}, and the sequences R and W are not identical. □

Remark 5.4. It is possible that a realisation of the given kinetic system is computed multiple times by the procedure **FindLinConjWithZeros**, however it is returned only once, when it is in the required equivalence class, in all other cases the procedure returns -1. This is the property stated and proved as Proposition 5.3.

5.2.1 Parallel implementation of Algorithm 5

This algorithm is also suitable for parallel implementation. The sequences can be stored in a single stack and the algorithm can work with them simultaneously, since the actual result depends only on the properties of the actually computed realization. This version of the algorithm was introduced in [54] and in Section 6.2.3 as well for computing all realizations of an uncertain kinetic system. Since a kinetic system can be considered as a special type of uncertain kinetic system, and linear conjugacy does not change the linearity of the model, this algorithm can be applied for the efficient computation of all possible linearly conjugate realizations of a kinetic system.

5.3 Examples

In this section the operation of both algorithms is demonstrated and compared to each other on four kinetic systems. For all the examples the results of the two algorithms were identical, differences were only in the running times.

It will be shown that the number of possible reaction graphs describing linearly conjugate realizations can grow very fast depending on the number of complexes, but can also be very small in a big network.

5.3.1 Example 1 (continued)

This example was published in [19], and it was also introduced in Section 2.5.1. The kinetic system $[M, Y]$ is defined by the matrices

$$M = \begin{bmatrix} 3k_1 & -k_2 & 0 \\ -3k_1 & k_2 & 0 \end{bmatrix} \qquad Y = \begin{bmatrix} 0 & 3 & 2 \\ 3 & 0 & 1 \end{bmatrix}$$

DOI:10.15774/PPKE.ITK.2018.004

For the numerical computations, the parameter values $k_1 = 1$ and $k_2 = 2$ were used. As the result of the algorithm 18 different sequences/reaction graph structures were returned. This small example is special in the sense that the sets of different reaction graph structures corresponding to dynamically equivalent and linearly conjugate realizations are identical, since the computed transformation matrix T was the unit matrix in every case. The reaction graphs denoted by $G_1, \ldots, G_{18}$ are presented in Figure 5.2.

Using the numerical results, it was easy to solve the equations for dynamical equivalence symbolically as well. With the application of this the reaction rate coefficients are given as functions of the parameters k_1 and k_2. From the computation it follows that there are two reaction rate coefficients k_{31} and k_{32} which do not depend on the input parameters, just on each other, and the reactions determined by these might together be present or non-present in the reaction network. Therefore, a nonnegative parameter p is applied to determine the values of these coefficients. In the reaction graphs $G_1, \ldots, G_9$ the parameter p is positive, and it is zero in the reaction graphs $G_{10}, \ldots, G_{18}$. Furthermore, the reaction graph G_i for $i \in \{1, \ldots, 9\}$ has exactly the structure of the graph G_{i+9} extended with the reactions $C_3 \to C_1$ and $C_3 \to C_2$.

The reaction graph G_1 (the complete directed graph) represents the dense realization, and consequently all other reaction graphs are subgraphs of it (not considering the edge weights).

DOI:10.15774/PPKE.ITK.2018.004

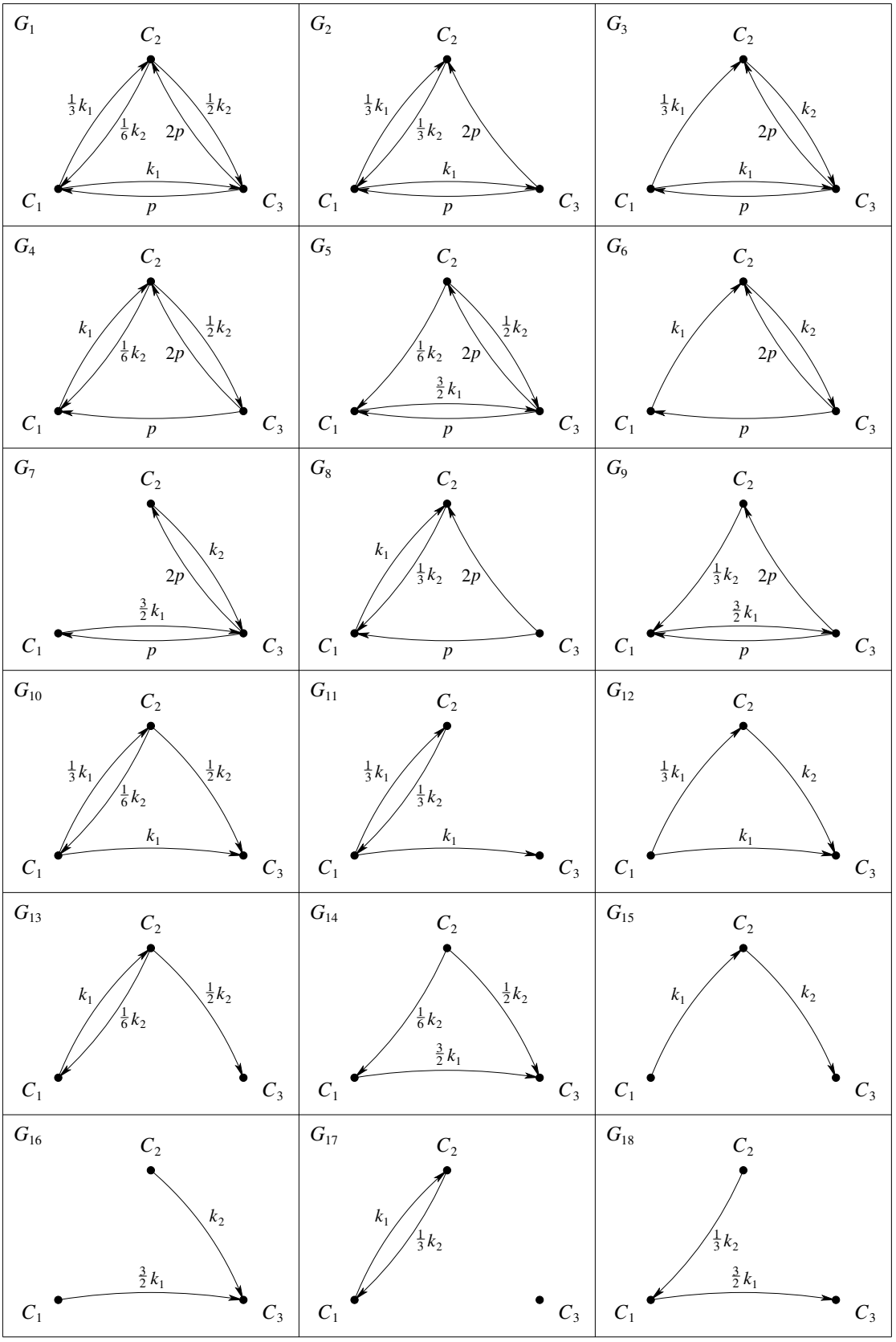


FIGURE 5.2: All reaction graph structures of Example 1 with possible reaction rate coefficients.

DOI:10.15774/PPKE.ITK.2018.004

5.3.2 Example 4 (continued)

The purpose of this example is to show the possible large number of structurally different linearly conjugate realizations even in the case of a relatively small kinetic system. The reaction network examined in this section was published in [40] as example $A1$, and it was also examined in Section 3.3.1 in this thesis as well.

The kinetic system $[M, Y]$ is defined by the matrices

$$M = \begin{bmatrix} 0 & -k_2 & k_3 & -2k_4 & k_5 & 0 \\ k_1 & 0 & -k_3 & k_4 & -k_5 & 0 \end{bmatrix} \qquad Y = \begin{bmatrix} 0 & 1 & 0 & 2 & 2 & 3 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

The reaction rate coefficients used in the computations are the same as in [40], namely: $k_1 = 1,\ k_2 = 1,\ k_3 = 0.05,\ k_4 = 0.1,\ k_5 = 0.1$, in which case the system shows oscillatory behaviour. The dense linearly conjugate realization (T_{ld}^{-1}, A_b^{ld}) has 19 reactions, as it can be seen in Figure 5.3.

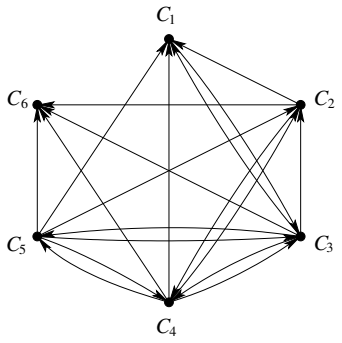


FIGURE 5.3: Reaction graph structure of the dense linearly conjugate realization (T_{ld}^{-1}, A_b^{ld}).

$$A_b^{ld} = \begin{bmatrix} -80 & 1.167e7 & 3.083 & 3.333e6 & 0.333 & 0 \\ 0 & -2e7 & 0.5 & 5e6 & 0.5 & 0 \\ 80 & 0 & -4.25 & 4 & 0.5 & 0 \\ 0 & 5e6 & 0.25 & -2e7 & 1 & 0 \\ 0 & 0 & 0.25 & 4 & -8.5 & 0 \\ 0 & 3.333e6 & 0.167 & 1.167e7 & 6.167 & 0 \end{bmatrix} \qquad T_{ld}^{-1} = \begin{bmatrix} 40 & 0 \\ 0 & 80 \end{bmatrix}$$

Algorithms 3 and 5 both returned as many as 17160 different reaction graph structures corresponding to linearly conjugate realizations of this kinetic system, all of which can be found in the electronic supplement available at:

`http://daedalus.scl.sztaki.hu/PCRG/works/publications/Ex2_AllRealSuppl.pdf`

Out of these, 17154 can be described by a weakly connected reaction graph, while 6 have disconnected reaction graphs, with the same linkage classes. Since this property can be ensured by linear constraints (the edges between the linkage classes are

DOI:10.15774/PPKE.ITK.2018.004

excluded), according to Proposition 3.3 the realization having the maximum number of edges determines a superstructure among realizations obeying the same constraints. This constrained dense realization can be described by the matrices A_b^{cld} and T_{cld}^{-1}, and its reaction graph is shown in Figure 5.4.

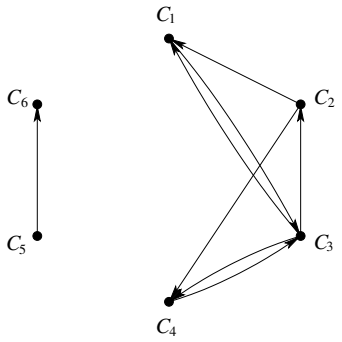


FIGURE 5.4: The reaction graph structure of the dense linearly conjugate realization $(T_{cld}^{-1}, A_b^{cld})$ having two linkage classes.

$$A_b^{cld} = \begin{bmatrix} -50 & 1.25e7 & 0.625 & 0 & 0 & 0 \\ 0 & -2.5e7 & 1.25 & 0 & 0 & 0 \\ 50 & 0 & -2.5 & 5 & 0 & 0 \\ 0 & 1.25e7 & 0.625 & -5 & 0 & 0 \\ 0 & 0 & 0 & 0 & -5 & 0 \\ 0 & 0 & 0 & 0 & 5 & 0 \end{bmatrix} \qquad T_{cld}^{-1} = \begin{bmatrix} 50 & 0 \\ 0 & 50 \end{bmatrix}$$

It also turned out from the computations that in this case the sparse realization is unique, and it is the initial network, that has been given in the original article to characterize the kinetic system. It is shown in Figure 5.5.

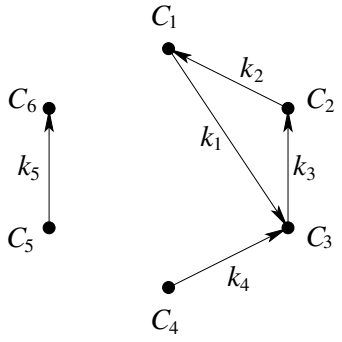


FIGURE 5.5: The reaction graph structure of the initial CRN (Y, A_k), that is the sole sparse realization of this kinetic system.

DOI:10.15774/PPKE.ITK.2018.004

$$A_k = \begin{bmatrix} -k_1 & k_2 & 0 & 0 & 0 & 0 \\ 0 & -k_2 & k_3 & 0 & 0 & 0 \\ k_1 & 0 & -k_3 & k_4 & 0 & 0 \\ 0 & 0 & 0 & -k_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & -k_5 & 0 \\ 0 & 0 & 0 & 0 & k_5 & 0 \end{bmatrix}$$

The distribution of the computed reaction graph structures over the number of reactions is shown in Figure 5.6.

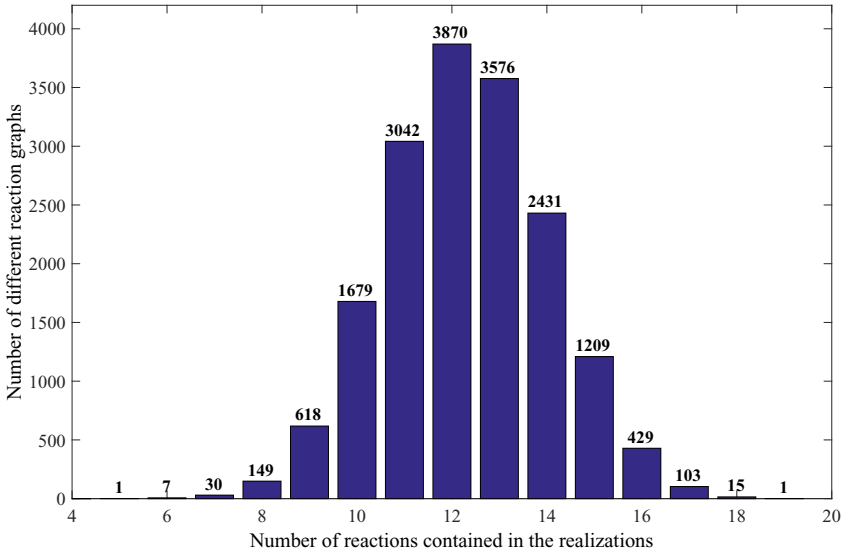


FIGURE 5.6: Number of different reaction graph structures with given numbers of reactions in the case of Example 4.

5.3.3 Example 5 (continued)

The kinetic system examined in this section was published in [41], and it has also been examined in Section 3.3.2 of this thesis. It can be originated from a reaction network modelling the operation of the glyoxylate bypass. The kinetic system $[M, Y]$ characterizing its dynamics is defined by the matrices:

$$M = \begin{bmatrix} 0.3 & 0.12 & 1.25297 & -1.6 & 0 & 0 & 0 & 0 & -33 \\ 1.06 & 0.94 & 3.9 & 0 & -4.62 & 0 & 0 & -0.6 & 0 \\ 1.36 & 1.06 & 0 & -1.6 & -4.62 & 0 & 0 & 0 & 0 \\ -1.36 & 0 & 3.48297 & 1.6 & 0 & 0 & 0 & -0.6 & 0 \\ 0 & -1.06 & 1.67 & 0 & 4.62 & 0 & 0 & 0 & -33 \\ 0 & 0 & -5.15297 & 0 & 0 & 0 & 0 & 0.6 & 33 \end{bmatrix}$$

DOI:10.15774/PPKE.ITK.2018.004

$$Y = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

By running Algorithm 3 or 5 it can be seen that this kinetic system has only 3 structurally different dynamically equivalent realizations. One is the dense realization presented in Section 3.3.2, the other two are both one reaction short compared to the dense realization. The reaction graph structures of these other realizations can be seen in Figures 5.7 and 5.8.

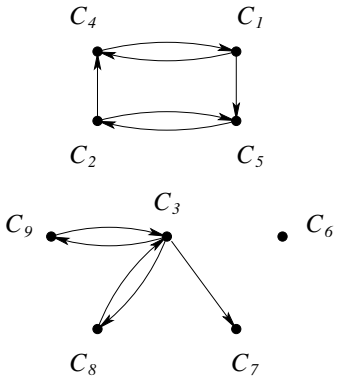


FIGURE 5.7

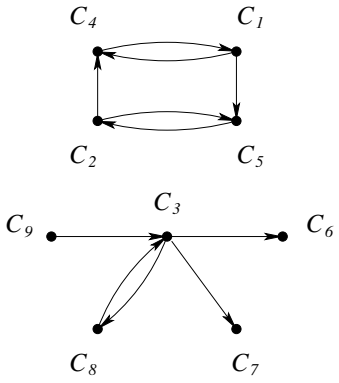


FIGURE 5.8

5.4 Computation results and efficiency analysis

In this section the performances of Algorithms 3 and 5 are compared to each other by the application of two examples. In the case of both algorithms single threaded implementations were applied in order to make the comparison fair. It was checked and confirmed that the two algorithms computed exactly the same reaction graph structures for both examples, but in performance they have shown a considerable difference.

All the computations were performed using single thread implementations on a Lenovo D60 workstation with two 2.60GHz Xeon (E5-2650 v2) processors and 32 Gb RAM (DDR3 1600 MHz, 0.6ns). The algorithms were implemented in MATLAB [49] using the YALMIP modelling language [50].

5.4.1 Example 7

The kinetic system examined in this section was originally published in [7], where it was represented by Equations (7-3) and (7-4) characterizing a dynamically equivalent realization (Y, A_k) of it, where

DOI:10.15774/PPKE.ITK.2018.004

$$Y = \begin{bmatrix} 2 & 3 & 1 & 0 \\ 1 & 0 & 2 & 3 \end{bmatrix} \qquad A_k = \begin{bmatrix} -1 & 0 & 0 & \varepsilon \\ 1 & -\varepsilon & 0 & 0 \\ 0 & \varepsilon & -1 & 0 \\ 0 & 0 & 1 & -\varepsilon \end{bmatrix}$$

The reaction graph is depicted in Figure 5.9.

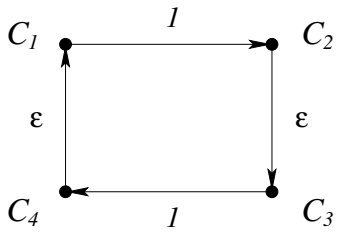


FIGURE 5.9: The reaction graph representing the CRN.

The coefficient matrix M of the kinetic system $[M, Y]$ is defined as

$$M = Y \cdot A_k = \begin{bmatrix} 1 & -2\varepsilon & -1 & 2\varepsilon \\ -1 & 2\varepsilon & 1 & -2\varepsilon \end{bmatrix}$$

In the case of the parameter value $\varepsilon = 1/7$ both algorithms have found 784 different reaction graph structures representing linearly conjugate realizations of this simple kinetic system. The distribution of possible different graph structures with given numbers of reactions is depicted in Fig. 5.10. As it is visible, the number of sparse structures, that have 4 directed edges is 9 in this case.

It can also be obtained that the dense realization (A_b^d, T_d^{-1}) contains all the 12 possible reactions, i.e. it can be represented by the complete directed graph. The matrices characterizing the dense realization are the following:

$$A_b^d = \begin{bmatrix} -5e7 & 7.143e2 & 1.25e7 & 0.00036 \\ 2.917e7 & -9.524e2 & 8.33e6 & 2.381e2 \\ 1.25e7 & 0.00036 & -5e7 & 7.143e2 \\ 8.33e6 & 2.381e2 & 2.917e7 & -9.524e2 \end{bmatrix} \qquad T_d^{-1} = \begin{bmatrix} 5000.005 & 0 \\ 0 & 5000.005 \end{bmatrix}$$

DOI:10.15774/PPKE.ITK.2018.004

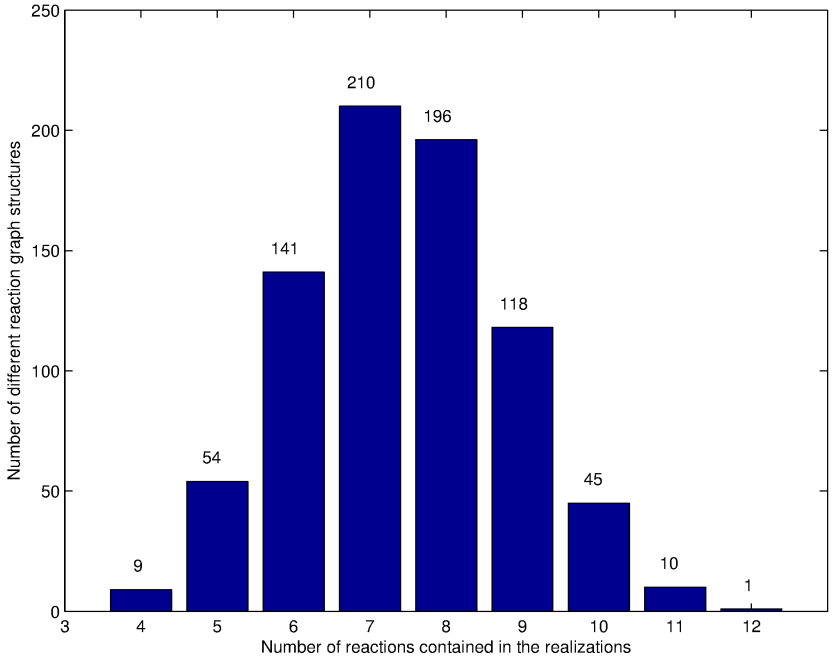


FIGURE 5.10: Number of different reaction graph structures with given numbers of reactions of the kinetic system $[M, Y]$

It is interesting to mention that in the case of this kinetic system there is only one realization (T_{s2}^{-1}, A_b^{s2}) where the reaction graph is not connected. In this case there are two linkage classes and the realization is a sparse one. Its characterizing matrices are as follows:

$$A_b^s = \begin{bmatrix} -5000.005 & 1428.573 & 0 & 0 \\ 5000.005 & -1428.573 & 0 & 0 \\ 0 & 0 & -5000.005 & 1428.573 \\ 0 & 0 & 5000.005 & -1428.573 \end{bmatrix} \qquad (T^s)^{-1} = \begin{bmatrix} 5000.005 & 0 \\ 0 & 5000.005 \end{bmatrix}$$

Table 5.1 shows the comparison results for Algorithms 3 and 5. The explanation of the compared features in Tables 5.1 and 5.2 is the following:

1. The total running time of the algorithm from start to end.
2. The computation time spent for solving optimization (i.e. linear programming) problems, including the setup of constraints.
3. The number of different reaction graph structures corresponding to linearly conjugate realizations of the examined kinetic system found by the applied methods.
4. The total number of function calls for computing constrained dense realizations (**FindLinConjWithZeros** in Algorithm 5, and **FindLinConjWithoutEdge** in Algorithm 3).

DOI:10.15774/PPKE.ITK.2018.004

5. The number of infeasible function calls for computing constrained dense realizations (these are computed in item 4 as well).

6. The number of computed valid reaction graph structures that had been computed previously at least once.

7. The average time that is elapsed between displaying/storing two distinct consecutive reaction graph structures.

8. The maximal time elapsed between displaying/storing two distinct consecutive reaction graph structures.

9. The variance of time intervals elapsed between displaying/storing two distinct consecutive reaction graph structures.

The computation times are measured and presented in every corresponding property in seconds.

TABLE 5.1: Comparison of the properties of the proposed algorithms in the case of Example 7

feat. no.	**feature description**	**Algorithm 5**	**Algorithm 3**
1.	total running time [s]	617.5386	2933.4868
2.	total optimization time [s]	616.1069	2923.1713
3.	no. of distinct valid structures found	784	784
4.	no. of constrained dense real. comps.	1096	5825
5.	no. of infeasible comp. steps	312	2240
6.	no. of structures found again	0	2801
7.	avg. computation interval [s]	0.788	3.7416
8.	max. computation interval [s]	14.97	12.0999
9.	variance of computation intervals [s^2]	0.30339	2.3894

5.4.2 Example 4 (continued)

In this section as well the example first introduced in [40] as example $A1$ is examined. The kinetic system $[M, Y]$ is characterized by the matrices

In the original article [40] the kinetic system is given by the following realization of it, described by the matrices

$$M = \begin{bmatrix} 0 & -k_2 & k_3 & -2k_4 & k_5 & 0 \\ k_1 & 0 & -k_3 & k_4 & -k_5 & 0 \end{bmatrix} \qquad Y = \begin{bmatrix} 0 & 1 & 0 & 2 & 2 & 3 \\ 0 & 0 & 1 & 0 & 1 & 0 \end{bmatrix}$$

The reaction rate coefficients are the same as in [40], namely: $k_1 = 1$, $k_2 = 1$, $k_3 = 0.05$, $k_4 = 0.1$, $k_5 = 0.1$.

Table 5.2 shows the comparison results for this example.

DOI:10.15774/PPKE.ITK.2018.004

It can be seen that the advantage of Algorithm 5 over Algorithm 3 increased slightly in the case of the studied larger example considering overall computation time. This fact mainly comes from Proposition 5.3 that results in a significantly lower number of constrained dense realization computation steps than [52]. However, as features 8 and 9 show, the solution in [52] may guarantee a 'more even' run with smaller variance in the computation intervals. The reason for this is the important property of Algorithm 3 that polynomial time is elapsed between displaying any two consecutive realizations.

TABLE 5.2: Comparison of the properties of the proposed algorithms in the case of Example 4

feat. no.	**feature description**	**Algorithm 5**	**Algorithm 3**
1.	total running time [s]	23 359.4359	139 456.3050
2.	total optimization time [s]	23 126.6604	135 217.6015
3.	no. of distinct valid structures found	17 160	17 160
4.	no. of constrained dense real. comps.	39 662	211 265
5.	no. of infeasible comp. steps	22 502	79 304
6.	no. of structures found again	3 820	114 801
7.	avg. computation interval [s]	1.1698	8.1268
8.	max. computation interval [s]	813.7367	22.2412
9.	variance of computation intervals [s^2]	38.6325	7.8097

5.5 Summary

I have proven the correctness of a new algorithm for computing all possible reaction graph structures representing linearly conjugate realizations of a kinetic system on a fixed set of complexes. This algorithm is the first method in the literature for determining all the reaction graph structures realizing a given kinetic dynamics. The computation might require exponential time because of the large number of possible structures, but it can be shown that between the determination of two different structures always polynomial time is elapsed. Furthermore, the algorithm is suitable for parallel implementation. The results are described in detail in Section 5.1 and are summarized in Thesis III.a.
The related publications are [52], [55] and [59].

I have designed another new and efficient algorithm as well for the computation of all structurally different linearly conjugate realizations of a kinetic system. I have shown that the algorithm returns every realization only once, furthermore, it is also suitable for parallel implementation. The performance of the new algorithm has been compared to that of the algorithm in Section 5.1, and considering all the examples the number of required optimization steps decreased by more than 80% in the case of this algorithm. The results are described in detail in Section 5.2 and are summarized in Thesis III.b.
The related publications are [53] and [60].

DOI:10.15774/PPKE.ITK.2018.004

Chapter 6

Uncertain kinetic systems

There are cases when the parameters characterizing the kinetic model are not precisely known, for example if these are determined from noisy measurements. In this case the polynomial system describing the dynamics requires to be written using uncertain parameters. In this chapter a generalized version of kinetic systems is defined that is able to represent uncertain parameters and additional linear constraints as well. The results summarized in this section were presented in [54].

6.1 Introduction of uncertain kinetic systems

Let us consider a kinetic system $[M, Y]$ where both the coefficient matrix M and the complex composition matrix Y are in $\mathbb{R}^{n\times m}$. The entries of the matrix M characterize the column vector $vec(M) = [[M]_{.1}^\top, \ldots, [M]_{.m}^\top]^\top$, which is the concatenation of the columns of the matrix M and it represents a point in the Euclidean space $\mathbb{R}^{nm}$. In the uncertain model it is assumed that the possible points $vec(M)$ are all the points of a closed convex polyhedron $\mathcal{P}$, which is defined as the intersection of q halfspaces. The boundaries of the halfspaces are hyperplanes with normal vectors $n_1, \ldots, n_q \in \mathbb{R}^{nm}$ and constants $b_1, \ldots, b_q \in \mathbb{R}$. Applying these notations, the polyhedron $\mathcal{P}$ can be characterized as the set of solutions of a linear inequality system.

$$\mathcal{P} = \{v \in \mathbb{R}^{nm} \mid v^\top \cdot n_i \leq b_i,\ 1 \leq i \leq q\}\} \tag{6.1}$$

In the characterization of the polyhedron $\mathcal{P}$ not only the estimated values of the parameters but also the kinetic property of the polynomial system should be considered. This can be ensured by prescribing the sign pattern of the coefficient matrix M as defined in Proposition 2.4. However, in the case of uncertain kinetic models the complex composition matrix Y is known and the coefficients are unknown, while for a given dynamics the set of complexes can be defined in various ways, therefore the constraints characterizing the kinetic condition should be given differently:

$$[Y]_{ij} = 0 \implies [M]_{ij} \geq 0 \qquad i \in \{1, \ldots, n\},\ j \in \{1, \ldots, m\} \tag{6.2}$$

DOI:10.15774/PPKE.ITK.2018.004

These constraints are of the same form as the inequalities in Equation (6.1), for example the constraint $vec(M)_k = [M]_{ij} \geq 0$ (where $k = (i-1)m+j$) can be written by choosing the normal vector n_i to be the unit vector $-e_k^{nm}$ and b_i to be the null vector $\mathbf{0}^{nm}$.

It is possible to define a set L of finitely many additional linear constraints on the variables to characterize certain additional properties, and these constraints can affect not only the entries of the coefficient matrix M but the Kirchhoff matrix of the realizations as well. Such constraint can be for example the exclusion of a set of reactions, or the restriction of mass conservation on a given level, see e.g. [52]. Let the column vector $vec(A_k) \in \mathbb{R}^{m^2-m}$ be defined as the column extension of the Kirchhoff matrix having only its off-diagonal entries as coordinates, and let r be the number of constraints in the set L. Then the equations can be written in the form

$$vec(M)^\top \cdot \alpha_i + vec(A_k)^\top \cdot \beta_i \leq d_i \tag{6.3}$$

where $\alpha_i \in \mathbb{R}^{nm}$, $\beta_i \in \mathbb{R}^{m^2-m}$ and $d_i \in \mathbb{R}$ hold for all $i \in \{1, \ldots, r\}$. These constraints do not change the general properties of the model, and it can be described by a linear programming problem, as it will be shown in Section 6.1.1. It is a more general model, within which uncertain kinetic systems without additional linear constraints form special case where L is $\emptyset$, and non-uncertain kinetic systems are a special case as well, where $L = \emptyset$ holds and $\mathcal{P}$ is a point in $\mathbb{R}^{nm}$.

In the uncertain case, too, it is assumed that the set of complexes is fixed, therefore the parameters of the model are the polyhedron $\mathcal{P}$ of possible coefficients, the set L of constraints and the complex composition matrix Y. Hence a **constrained uncertain kinetic system** is referred to as the triple $[\mathcal{P}, L, Y]$, but it will be called in this thesis uncertain kinetic system for brevity.

Definition 6.1. *A reaction network (Y, A_k) is called a **realization of the uncertain kinetic system** $[\mathcal{P}, L, Y]$ if there exists a coefficient matrix $M \in \mathbb{R}^{n\times m}$ so that the equation $M = Y \cdot A_k$ holds, the point $vec(M)$ is in the polyhedron $\mathcal{P}$ and the entries of the matrices M and A_k fulfil the set L of constraints. Since the matrix Y is fixed but the coefficients of the polynomial system can vary, this realization is referred to as the matrix pair (M, A_k).*

In other words if there is a coefficient matrix M defining suitable parameters and a reaction network (Y, A_k) that is a dynamically equivalent realization of the kinetic system $[M, Y]$ then this CRN is called a realization of the uncertain model.

6.1.1 Computational model for uncertain kinetic systems

A realization (M, A_k) of an uncertain kinetic system $[\mathcal{P}, L, Y]$ in the case of a fixed set of complexes similarly to realizations of a non-uncertain kinetic system can be computed using the linear optimization framework.

In the optimization model the variables are all the entries of the matrix M and the off-diagonal entries of matrix A_k. The constraints regarding the realizations of the uncertain

DOI:10.15774/PPKE.ITK.2018.004

model can be written as follows:

$$vec(M)^\top \cdot n_i \leq b_i \qquad i \in \{1,\dots,q\} \tag{6.4}$$

$$M = Y \cdot A_k \tag{6.5}$$

$$[A_k]_{ij} \geq 0 \qquad i \neq j,\ i,j \in \{1,\dots,m\} \tag{6.6}$$

$$\sum_{i=1}^{n} [A_k]_{ij} = 0 \qquad j \in \{1,\dots,m\} \tag{6.7}$$

$$[M]_{ij} \geq 0 \qquad [Y]_{ij} = 0,\ i \in \{1,\dots,n\},\ j \in \{1,\dots,m\} \tag{6.8}$$

Equations (6.4) ensure that the parameters of the dynamics correspond to a point of the polyhedron $\mathcal{P}$. Dynamical equivalence is defined by Equation (6.5), while Equations (6.6) and (6.7) are required for the Kirchhoff property of matrix A_k to be fulfilled. The constraints in the set L can be written in the form of Equation (6.3) and also added to the computation model.

The objective function of the optimization model can be defined according to the desired properties of the realization, for example in order to examine if the reaction $C_i \to C_j$ can be present in the reaction network or not the objective can be defined as $max[A_k]_{ji}$.

Since a realization of an uncertain kinetic system is defined by the corresponding coefficient matrix $M \in \mathbb{R}^{n\times m}$ and Kirchhoff matrix $A_k \in \mathbb{R}^{m\times m}$, it can be represented as a vector $[vec(A_k)^\top, vec(M)^\top]^\top$ in the Euclidean space $\mathbb{R}^{m^2-m+nm}$. Due to the linearity of the constraints in the computational model, the set of possible realizations of an uncertain kinetic system $[\mathcal{P}, L, Y]$ is represented by all the points of a convex polyhedron $\mathcal{Q}$ defined by Equations (6.4)–(6.8).

6.1.2 Properties

It has been proven in Proposition 3.3 that the dense linearly conjugate realization of a kinetic system defines a superstructure among all realizations. This property can be extended to the case of uncertain kinetic systems.

Definition 6.2. *A reaction network (Y, A_k) is called the **dense/sparse** realizations of the uncertain kinetic system $[\mathcal{P}, L, Y]$ if it is realization where the maximum/minimum number of reactions are present.*

The proof of the superstructure property of the dense realization is based on the same idea as the proof of Proposition 3.3. Since the representing set $\mathcal{Q}$ of all solution of an uncertain kinetic system is convex, it contains the intervals connecting any two of its points, therefore there cannot be any realization containing a reaction that is not present in the dense realization.

Proposition 6.3. *A dense realization (M, A_k) of an uncertain kinetic system $[\mathcal{P}, L, Y]$ determines a superstructure among all realizations of the uncertain model.*

DOI:10.15774/PPKE.ITK.2018.004

It follows from Proposition 6.3 that the structure of the dense realization is unique. If there were two different dense realizations, then the reaction graphs representing them would contain each other as subgraphs, which implies that these realizations are structurally identical.

In some cases it is possible to prove the structural uniqueness of the uncertain model.

Proposition 6.4. *The dense and sparse realizations of an uncertain kinetic system* $[\mathcal{P}, L, Y]$ *have the same number of reactions if and only if all realizations of the model are structurally identical.*

Proof. According to the definitions if in the dense and sparse realizations there is the same number of reactions, then in all realizations there must be the same number of reactions. Since the structure of the dense realization is unique, there cannot be two realizations with the maximal number of reactions but having different structures, therefore all realizations must be structurally identical to the dense realization.

The converse statement is trivial. If all the realizations of the model are structurally identical, then the dense and sparse realizations must also have identical structures. □

6.2 Algorithms to compute realizations and properties of an uncertain kinetic system

Due to the similar model structure, certain algorithms designed for computing realizations of a non-uncertain kinetic system can be modified in order to work for determining similar properties in the case of uncertain kinetic systems. In fact, since the uncertain model is a generalized version of a kinetic system that allows parameter uncertainty and additional linear constraints, the methods can directly be applied for computing the given properties in connection with dynamically equivalent realizations of a kinetic system if it is defined as an uncertain model.

6.2.1 Polynomial-time algorithm to determine dense realizations

The dense realization of the uncertain kinetic system can be computed by the application of a recursive polynomial-time algorithm. The basic principle of the method is similar to the one presented in Section 3.2. To every off-diagonal entry of the matrix A_k a realization is assigned where this value is positive, if this is possible. If a realization is assigned to the entry $[A_k]_{ij}$ then the reaction $C_j \to C_i$ must be present in this realization. In general, the same realization can be assigned to several reactions, therefore there is no need to perform a computation step for each reaction separately. The convex combination of the assigned realizations is also a realization of the uncertain model, although they are in general realizations of different kinetic systems that fulfil the constraints regarding the coefficients. If all the coefficients of the convex combination are positive, none of them is zero, then all reactions that take place in any of the

DOI:10.15774/PPKE.ITK.2018.004

assigned realizations are present in the convex combination as well. Consequently, the obtained realization represents a dense realization, where all reactions are present that are possible.

The computation can be performed in polynomial time since it requires at most $m^2 - m$ LP optimization steps, but in most cases much fewer steps are sufficient.

In the algorithm the assigned realizations are represented as points in $\mathbb{R}^{m^2-m+nm}$ and form a convex polyhedron $\mathcal{Q}$ as defined earlier. The realizations are determined using the following procedure:

FindPositive$([\mathcal{P}, L, Y], H)$ returns a pair (R, B). The point $R \in \mathcal{Q}$ represents the realization of the uncertain model $[\mathcal{P}, L, Y]$ for which the value of the objective function $\sum_{j\in H} R_j$ considering a set $H \subseteq \{1, \ldots, m^2 - m\}$ of indices is maximal. The other returned object is a set B of indices where $k \in B$ if and only if $R_k > 0$. If there is no realization fulfilling the constraints then the pair $(\mathbf{0}^{m^2-m+nm}, \emptyset)$ is returned.

At the construction of the dense realization the arithmetic mean is applied as convex combination, i.e. if the number of the assigned realizations is k then all the coefficients of the convex combination are equal to $\frac{1}{k}$.

Algorithm 6 Computes a dense realization
Input: $[\mathcal{P}, L, Y]$
Output: *Result*

$H := \{1, \ldots, m^2 - m\}$
$B := H$
$Result := \mathbf{0} \in \mathbb{R}^{m^2-m+nm}$
$loops := 0$
while $B \neq \emptyset$ **do**
    $(R, B) :=$ **FindPositive**$([\mathcal{P}, L, Y], H)$
    $Result := Result + R$
    $H := H \setminus B$
    $loops := loops + 1$
end while
$Result := Result/loops$
if $Result = \mathbf{0}$ **then**
    There is no realization with the given properties.
else
    Result is a dense realization.
end if

Proposition 6.5. *The realization returned by* **Algorithm 6** *is a dense realization of the uncertain kinetic system* $[\mathcal{P}, L, Y]$.

Proof. Since the set of solutions is represented as a convex set $\mathcal{Q}$, the the point *Result* computed as the convex combination of realizations is indeed a realization of the uncertain kinetic system $[\mathcal{P}, L, Y]$. If the returned point *Result* does not represent the dense

DOI:10.15774/PPKE.ITK.2018.004

realization, then there is a reaction $C_i \to C_j$ which is present in the dense realization but it does not take place in *Result*. By the operation of the algorithm it follows that there must be a realization assigned to the reaction $C_i \to C_j$, consequently this reaction takes place in the realization computed as the convex combination of the assigned realizations as well. This is a contradiction. □

6.2.2 Core reactions of the uncertain model

A reaction is called **core reaction** of a kinetic system if it is present in every realization of the kinetic system [39]. It is possible that there are no core reactions, but there might be several of them as well. Furthermore, if all the realizations are structurally identical, then by Proposition 6.4 it follows that every reaction is a core reaction.

The notion of core reactions can be extended to the case of uncertain models as well, in order to further analyse the structural properties of the realizations.

Definition 6.6. *A reaction $C_i \to C_j$ is called a **core reaction** of the uncertain kinetic system $[\mathcal{P}, Y, L]$ if it is present in every realization of the model. In the realizations all possible coefficient matrices M has to be considered for which $vec(M) \in \mathcal{P}$ holds.*

Let $[\mathcal{P}, L, Y]$ and $[\mathcal{P}', L, Y]$ be two uncertain kinetic systems defined with identical sets of complexes and additional linear constraints so that the polyhedron $\mathcal{P}'$ is a subset of $\mathcal{P}$. If the sets of core reactions in the models are denoted as $C_{\mathcal{P}}$ and $C_{\mathcal{P}'}$, respectively, then it is easy to see that $C_{\mathcal{P}} \subseteq C_{\mathcal{P}'}$ must hold. This property holds also in the special case, when the polyhedron $\mathcal{P}'$ is a single point in $\mathbb{R}^{nm}$ and $[\mathcal{P}', L, Y]$ is a kinetic system defined as an uncertain kinetic system.

The set of core reactions of an uncertain kinetic system can be computed using a polynomial-time algorithm. This method has been first published in [47] for a special case, where the coefficients of the polynomial system have to be in predefined intervals and the polyhedron $\mathcal{P}$ is a cuboid. Since the model applies only the property that all the constraints characterizing the model are linear, it can be applied without any modification to uncertain kinetic systems as well.

The question whether a certain reaction is a core reaction of a kinetic model or not, can be answered by solving a linear optimization problem. If this question has to be decided for all the possible reactions, the computation can be done more efficiently than doing separate optimization steps for every reaction. The idea is to minimize the sum of variables representing the entries of the Kirchhoff matrix. Generally, several variables in the minimized sum are zero in the computed realization, which means that the reactions corresponding to these variables are not core reactions. This step is repeated with the remaining set of variables until the computation does not return any non-core reactions. Finally, the remaining variables need to be checked one-by-one.

In the algorithm the sets of indices corresponding to the off-diagonal entries of the Kirchhoff matrix A_k are referred to by their characteristic vectors. The set $B \subseteq \{1, \ldots, m^2 - m\}$ is represented by the vector $b \in \{0, 1\}^{m^2-m}$, which is defined as

DOI:10.15774/PPKE.ITK.2018.004

$$b_i = \begin{cases} 1 & \text{if } i \in B \\ 0 & \text{if } i \notin B \end{cases} \tag{6.9}$$

It necessary to utilize some operations on the sets, that is written using the notation of the characteristic vectors:

$b * c$ represents the set $B \cap C$, i.e. it is an element-wise 'logical and'

$\overline{c}$ represents the complement of set C, i.e. it is an element-wise negation.

More formally the procedure applied during the computation is the following:

FindNonCore$([\mathcal{P}, L, Y], b)$ computes a realization of the uncertain kinetic system $[\mathcal{P}, L, Y]$ represented as a point $R \in \mathbb{R}^{m^2-m+nm}$, for which the sum of the coordinates with indices in the set $B \in \{1, \ldots, m^2 - m\}$ is minimal. The procedure does not return this realization but the vector c, the characteristic vector of set C which contains the indices corresponding to zero entries of the Kirchhoff matrix of the realization R, i.e. $C \subseteq \{1, \ldots, m^2 - m\}$ and $[i \in C \iff R_i = 0]$.

Algorithm 7 Computes the set of core reactions
Inputs: $[\mathcal{P}, L, Y]$
Output: b

$b := \mathbf{1}$
$c := b$
while $c \neq \mathbf{0}$ **do**
    $c :=$ **FindNonCore**$([\mathcal{P}, L, Y], b)$
    $c := c * b$
    $b := b * \overline{c}$
end while
for $i = 1$ to $m^2 - m$ **do**
    if $b_i \neq 0$ **then**
        $c :=$ **FindNonCore**$([\mathcal{P}, L, Y], e_i^{m^2-m})$
        $b := b * \overline{c}$
    end if
end for
if $b = \mathbf{0}$ **then**
    There are no core reactions of the model $[\mathcal{P}, L, Y]$.
else
    The vector b characterizes the core reactions of the model $[\mathcal{P}, L, Y]$.
end if

Proposition 6.7. Algorithm 7 *computes the set of core reactions of the uncertain kinetic system* $[\mathcal{P}, L, Y]$ *in polynomial time.*

Proof. Let us assume by contradiction that the algorithm does not return the proper set of core reactions. There can be two different types of error:

DOI:10.15774/PPKE.ITK.2018.004

a) Let us assume that there is an index i for which the corresponding reaction is a core reaction, but according to the algorithm it is not. In this case there must be a realization R computed by the algorithm so that R_i is zero. This is a contradiction.

b) Let us assume that there is an index j for which the corresponding reaction is not a core reaction but the algorithm returns the opposite answer. Consequently, after the while loop of the computation (from line 8) the coordinate b_j must be equal to 1. Then there is a point when this singular reaction is exmined by the application of the procedure **FindNonCore**$([\mathcal{P}, L, Y], e_j^{m^2-m})$. According to the assumption in the returned realization R the coordinate R_j must be zero, which also yields a contradiction.

The computation according to the algorithm can be performed in polynomial time, since it requires the solution of at most m^2-m LP optimization problems and some additional minor computation. □

6.2.3 Algorithm to determine all possible reaction graph structures of uncertain models

Especially in the case when the dynamical model is not precisely known, it is useful to be able to enumerate all the possible realization structures. The principles of methods presented earlier in Sections 5.1 and 5.2 for computing all structurally different linearly conjugate realizations of a given kinetic system can be applied with appropriate modifications regarding the description of the optimization model.

In this section an efficient algorithm is introduced for computing all possible reaction graph structures of an uncertain kinetic system $[\mathcal{P}, L, Y]$ defined with a fixed set of complexes. The proposed method is an improved version of Algorithm 5 presented in Section 5.2, where all the optimization steps can be done simultaneously. The correctness of the presented method is also proven here.

The data representation and notations are very similar to the ones in Section 5.2.

The reaction graph structures are represented as binary sequences, and due to the superstructure property it is sufficient to denote the non-core reactions of the dense realization. Let the number of these reactions be denoted by z. The dense realization is referred to as the sequence $\mathbf{1}_z$ with all coordinates equal to 1.

In the notations the equivalence relations $=_k$ and the corresponding equivalence classes $C_k(R)$ for $k \in \{1, \ldots, z\}$ are also applied. The difference compared to Algorithm 5 is that the computed sequences get stored during the computation in a single stack S instead of several ones. For further examinations it is necessary to know the largest index of the fixed part. Therefore, the realizations are represented as a pair (R, k), where R is the binary sequence and k is an integer. The realization (R, k) can be put into the stack S if R represents the dense realization in $C_k(R)$. The command 'push (R, k) into S' puts the sequence R into the stack and 'pop from S' takes a sequence out of the stack and returns it. The number of sequences in the stack S is denoted by $size(S)$.

DOI:10.15774/PPKE.ITK.2018.004

The results of the whole computation are stored in a binary array called $Exist$, defined as before. Since in this method the sequences are stored in just one stack, the application of the array $Exists$ is necessary.

Within the algorithm two procedures are applied, that are similar to the subroutines of Algorithm 5:

FindRealization$((R,k),i)$ computes a dense realization of the uncertain kinetic system $[\mathcal{P},L,Y]$, for which the representing binary sequence W is in $C_k(R)$ and for every index $j \in \{k+1,\dots,i\}$ the coordinate W_j is zero, if it exists. Since in case of the strict inequalities the calibration of the small number ε might be a difficult task, to make the computation more accurate a dense realization is computed that fulfils only the constraints corresponding to the specified coordinates of a sequence that need to be zero. It is possible that among the first k coordinates there are more zeros than required, therefore the computed sequence W is compared to the sequence R. The procedure returns the sequence W only if $W =_k R$ holds, otherwise -1 is returned. If the optimization model is infeasible then the returned object is also -1.

FindNextOne$((R,k))$ returns the smallest index i for which $k < i$ and $R_i = 1$ hold. If there is no such index, i.e. R_j is zero for all $k < j$, then it returns $z+1$.

Algorithm 8 Computes all reaction graph structures
Inputs: $[\mathcal{P},L,Y],z$
Output: $Exist$

```
1: push (1_z, 0) into S
2: Exist[1_z := 1
3: while size(S) > 0 do
4:     (R, k) := pop from S
5:     i := FindNextOne((R, k))
6:     if i < z then
7:         push (R, i) into S
8:     end if
9:     while i < z do
10:        W := FindRealization((R, k), i)
11:        if W < 0 then
12:            BREAK
13:        else
14:            i := FindNextOne(W, i)
15:            Exist[W] := 1
16:            if i < z then
17:                push (W, i) into S
18:            end if
19:        end if
20:    end while
21: end while
```

DOI:10.15774/PPKE.ITK.2018.004

Since the Algorithm 8 is a generalized version Algorithm 5 applying the advantages of parallel implementations, the proof of its accuracy is similar to that in Proposition 5.2.

Proposition 6.8. Algorithm 8 *computes all possible reaction graph structures representing realizations of an uncertain kinetic system* $[\mathcal{P}, L, Y]$.

Proof. Let us assume by contradiction that there is a realization of the uncertain kinetic system $[\mathcal{P}, L, Y]$ represented by the sequence V which is not returned by Algorithm 8. Let R be another sequence that was stored in the stack S as (R, p) at some point during the computation, for which $V =_p R$ holds and p is the greatest such number. If $p = 0$ then D is suitable to be R, and by the operation of the algorithm it follows that $p < z$ holds. (If p were equal to z, then V would be equivalent to R and it is a contradiction.)

There is a point during the computation when (R, p) is popped out from the stack S. Let us assume that **FindNextOne**(R, p) returns i and **FindNextOne**(V, p) returns j. In this case $i \leq j$ must hold since R represents the superstructure in $C_p(R)$ and if i were equal to j then p would not be maximal.

At the examination of sequence R the procedure **FindRealization** $((R, p), i)$ is applied first, and it must return a valid sequence W_1 since its constrains are fulfilled by the realization V as well. If **FindNextOne**(W_1, p) is j_1 then $j_1 < j$ must hold, since W_1 represents the dense realization in $C_i(W_1)$, V is also in $C_i(W_1)$, and if j_1 was equal to j then p would not be maximal.

The computation can be continued with the procedure **FindRealization**$((R, p), j_1)$. It must return a valid sequence W_2 for which **FindNextOne**$(W_2, p) = j_2 \leq j$ holds, that can be shown by applying similar reasoning as earlier.

These steps must lead to contradiction either by p not being maximal or by creating an infinite increasing sequence of integers that has an upper bound. Consequently, every possible reaction graph structure that represents a realization of the uncertain kinetic system $[\mathcal{P}, L, Y]$ is returned by the algorithm. □

A similar proof to that in the case of Proposition 5.3 can be applied to show that during the computation every reaction graph structure is returned only once. Since the calculations of procedure **FindRealization**$((R, k), i)$ are independent of the results of every other call of the same procedure, therefore the order of the calls is irrelevant regarding the result of the entire computation.

6.3 Illustrative examples

In this section the operation of Algorithms 6, 7 and 8 is demonstrated on two examples in the case of different degrees and types of uncertainties, and even with additional linear constraints.

DOI:10.15774/PPKE.ITK.2018.004

6.3.1 Example 1 (continued)

The model that serves as a basis for this example was presented previously in [19], [52] and it was examined in Sections 2.5.1 and 5.3.1. The uncertain model is generated using the kinetic system $[M, Y]$, where the characterizing matrices are

$$M = \begin{bmatrix} 3c_1 & -c_2 & 0 \\ -3c_1 & c_2 & 0 \end{bmatrix} \qquad Y = \begin{bmatrix} 0 & 3 & 2 \\ 3 & 0 & 1 \end{bmatrix}$$

During the computations the parameter values $c_1 = 1$ and $c_2 = 2$ were applied.

In the case of this example two types of parameter uncertainty with the corresponding two different polyhedrons are examined.

6.3.1.1 Uncertainty defined by independent intervals

This model represents a special case in the class of uncertain kinetic systems, since the possible values of every coefficient of the kinetic system are determined by independent upper and lower bounds that are defined as relative distances. If the entry $[M]_{ij}$ of the coefficient matrix M is represented by the coordinate $vec(M)_l$ of the point $vec(M) \in \mathbb{R}^6$ and the relative distances of the upper and lower bounds are given by the real constants γ_l and ρ_l from the interval $[0, 1]$, respectively, then the equations defining the polyhedron $\mathcal{P}_A \subset \mathbb{R}^6$ of the uncertain parameters can be written in terms of the coordinates $\widetilde{M}_l$ as

$$vec(M)^\top \cdot e_l^6 \leq (1 + \gamma_l) \cdot [M]_{ij}$$
$$vec(M)^\top \cdot (-e_l^6) \leq (\rho_l - 1) \cdot [M]_{ij}$$

In the examined uncertain kinetic system $[\mathcal{P}_A, L, Y]$ no additional linear constraints are considered, i.e. $L = \emptyset$.

In Section 5.3.1 all possible reaction graph structures representing dynamically equivalent realizations of the kinetic system $[M, Y]$ have been presented. Obviously, these structures must appear among the realizations of the uncertain kinetic model $[\mathcal{P}_A, \emptyset, Y]$ as well, but there might be more possible structures among the realizations of the uncertain kinetic system.

Interestingly, the result of the computation was that in the case of any degree of the uncertainty coefficients $\gamma_l, \rho_l \in [0, 1)$ for all $l \in \{1, \dots 6\}$ the sets of possible reaction graph structures of the uncertain model $[\mathcal{P}_A, \emptyset, Y]$ and of the original kinetic system $[M, Y]$ are identical. This result might be contrary to expectations, but for this small example it is easy to prove that the obtained graph structures are correct for all positive values of the parameters c_1 and c_2. To do this, the computation has to be divided into smaller steps.

It has been shown in [53] that in the case of dynamically equivalent realizations the computation can be done column-wise. By the definition of matrix multiplication it

DOI:10.15774/PPKE.ITK.2018.004

follows that $Y \cdot A_k = M$ holds if and only if $Y \cdot [A_k]_{.j} = [M]_{.j}$ holds for all $j \in \{1, \dots, m\}$. Consequently, the jth column of matrix A_k depends only on the jth column of matrix M. These computations can be done separately, and all the possible reaction graph structures can be constructed by choosing a column structure for every index $j \in \{1, \dots, m\}$ and building the Kirchhoff matrix A_k of the realization from them. Consequently, if in the case of the jth column the number of different structures is p_j, then the number of structurally different realizations is $\prod_{j=1}^{m} p_j$.

In order to make the notations less complicated the entries of the Kirchhoff matrix are denoted as the reaction rate coefficients, i.e. $[A_k]_{ij} = k_{ji}$ for all $i, j \in \{1, 2, 3\}$, $i \neq j$.

The results in the case of the first column:

$$Y \cdot \begin{bmatrix} -k_{12} - k_{13} \\ k_{12} \\ k_{13} \end{bmatrix} = \begin{bmatrix} 3c_1 \\ -3c_1 \end{bmatrix} \quad k_{12}, k_{13} \in \mathbb{R}^+ \quad \Longrightarrow \quad k_{12} \in [0, c_1], \; k_{13} = \frac{3}{2}c_1 - \frac{3}{2}k_{12} \tag{6.10}$$

It can be seen that for every positive value of the parameter c_1 the two corresponding reaction rates can realize 3 of the $2^2 = 4$ possible structurally different solutions. Both can be positive, or either one can be positive while the other one is zero. (Possible outcomes are for example: $k_{12} = \frac{1}{2}c_1, k_{13} = \frac{3}{4}c_1$ or $k_{12} = 0, k_{13} = \frac{3}{2}c_1$ or $k_{12} = c_1, k_{13} = 0$.) The fourth case, when both k_{12} and k_{13} are zero is possible only when $[M]_{.1} = [0 \;\; 0]^\top$, which requires the corresponding parameters of uncertainty ρ_i to be at least one.

In the case of the second column, 3 of the 4 possible outcomes can be realized and a similar reasoning can be applied.

$$Y \cdot \begin{bmatrix} k_{21} \\ -k_{21} - k_{23} \\ k_{23} \end{bmatrix} = \begin{bmatrix} -c_2 \\ c_2 \end{bmatrix} \quad k_{21}, k_{23} \in \mathbb{R}^+ \quad \Longrightarrow \quad k_{21} \in (0, \frac{c_2}{3}), \quad k_{23} = c_2 - 3k_{21} \tag{6.11}$$

In the third column there is no uncertainty because there are only zero entries in $[M]_{.3}$. Consequently, in the case of $[A_k]_{.3}$ only 2 solutions are possible. The two corresponding reactions can either be both present or both missing.

$$Y \cdot \begin{bmatrix} k_{31} \\ k_{32} \\ -k_{31} - k_{32} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad k_{31}, k_{32} \in \mathbb{R}^+ \quad \Longrightarrow \quad k_{31} \in \mathbb{R}^+, \quad k_{32} = 2k_{31} \tag{6.12}$$

It follows from the above computations that the number of possible reaction graph structures is $3 \cdot 3 \cdot 2 = 18$, and the generated structures are identical to the ones presented in Section 5.3.1. This number could be larger only if all the reaction rates in the first or second column of A_k can be zero, but this requires the entries in the corresponding column $[M]_{.1}$ or $[M]_{.2}$ to be zero.

DOI:10.15774/PPKE.ITK.2018.004

6.3.1.2 Uncertainty defined as a general polyhedron

It is interesting to examine another uncertain kinetic system generated from the kinetic system $[M, Y]$, where the set $\mathcal{P}_B$ of possible coefficients is defined as a polyhedron. If the matrix M of coefficients is represented by the vector $vec(M) \in \mathbb{R}^6$, where $vec(M)^\top = [[M]_{11}, [M]_{21}, [M]_{12}, [M]_{22}, [M]_{13}, [M]_{23}]$, then let the equations determining the polyhedron $\mathcal{P}_B$ be the following:

$$\begin{aligned} vec(M)^\top \cdot (-e_1^6) &\leq 0 \\ vec(M)^\top \cdot (-e_4^6) &\leq 0 \\ vec(M)^\top \cdot e_5^6 &= 0 \\ vec(M)^\top \cdot e_6^6 &= 0 \\ vec(M)^\top \cdot [1,1,1,1,0,0]^\top &= 0 \\ vec(M)^\top \cdot [0,-1,-1,0,0,0]^\top &\leq 7 \\ vec(M)^\top \cdot [-1,0,0,1,0,0]^\top &\leq -1 \end{aligned} \tag{6.13}$$

In this case again, no additional linear constraints are considered in the uncertain model. The computation of all possible reaction graph structures shows that in addition to the structures realizing the original kinetic system $[M, Y]$ depicted in Figure 5.2 there are 6 more structures, presented in Figure 6.1.

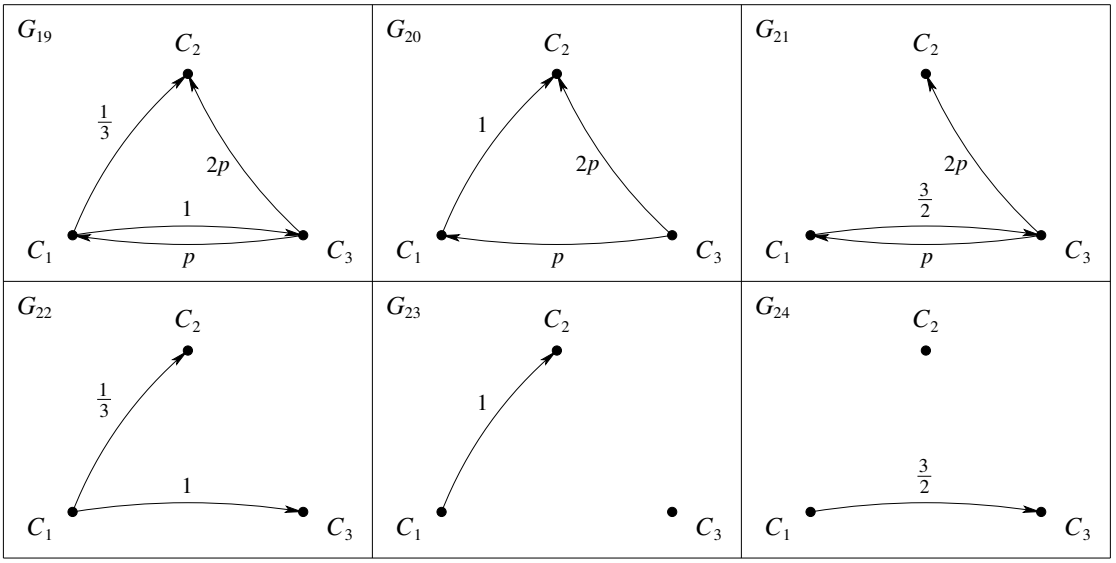


FIGURE 6.1: The additional possible reaction graph structures of the kinetic system $[\mathcal{P}, L, Y]$ compared to the original kinetic system $[M, Y]$.

It can be seen that the point $vec(M_1)^\top = [3, -3, -2, 2, 0, 0]$ corresponding to the original non-uncertain kinetic system is in the polyhedron $\mathcal{P}_B$, therefore the 18 structures determined by its realizations must be among the realizations of the uncertain kinetic system.

DOI:10.15774/PPKE.ITK.2018.004

Then similarly to that in Section 6.3.1.1, the columns of the matrix A_k can be considered independently. Since the entries in column $[M]_{.3}$ are all zero in every point in the polyhedron $\mathcal{P}_B$, only the previously mentioned two outcomes are possible in the case of this column. There can be more realizations only if all the reaction rates in at least one of the columns $[A_k]_{.1}$ or $[A_k]_{.2}$ can be zero, which is possible only if all the entries in $[M]_{.1}$ or $[M]_{.2}$ are zero. By the constraints of the polyhedron $\mathcal{P}_B$ it follows that $[M]_{11} \geq 1$. Consequently, the column $[M]_{.1}$ cannot be zero, but $[M]_{.2}$ can, for example the point $vec(M_2) = [3, -3, 0, 0, 0, 0]^\top \in \mathcal{P}_B$ fulfils this property. For the columns of the matrices M and M_2, the following hold: $[M_2]_{.1} = [M]_{.1}$ and $[M_2]_{.3} = [M]_{.3}$. Therefore, for the first and third columns of A_k there are 3 and 2 possible outcomes, respectively. This means that the number of further reaction graph structures (compared to the original kinetic system $[M, Y]$) is $3 \cdot 2 = 6$. It is easy to see that these are exactly the ones presented in Figure 6.1 with possible reaction rates, where $p \in \mathbb{R}_+ \setminus \{0\}$.

6.3.2 Example 5 (continued)

The other example examined here is the glyoxylate bypass model presented in [41] and also in Section 3.3.2. The kinetic system $[M, Y]$ characterizing its dynamics is defined by the matrices:

$$M = \begin{bmatrix} 0.3 & 0.12 & 1.25297 & -1.6 & 0 & 0 & 0 & 0 & -33 \\ 1.06 & 0.94 & 3.9 & 0 & -4.62 & 0 & 0 & -0.6 & 0 \\ 1.36 & 1.06 & 0 & -1.6 & -4.62 & 0 & 0 & 0 & 0 \\ -1.36 & 0 & 3.48297 & 1.6 & 0 & 0 & 0 & -0.6 & 0 \\ 0 & -1.06 & 1.67 & 0 & 4.62 & 0 & 0 & 0 & -33 \\ 0 & 0 & -5.15297 & 0 & 0 & 0 & 0 & 0.6 & 33 \end{bmatrix}$$

$$Y = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

It has been shown in Section 5.3.3 that there are 3 structurally different realizations of the kinetic system and 10 of the 12 reactions are core reactions. Let us examine the uncertain models $[\mathcal{P}_\alpha, \emptyset, Y]$ where the uncertainty of the parameters is defined with independent intervals characterized as relative distances. The polyhedron $\mathcal{P}_\alpha$ is a cuboid defined similarly to $\mathcal{P}_A$, the upper and lower bounds are determined by the rate coefficients $\gamma_l = \rho_l = \alpha \in [0, 1]$ for all indices $l \in \{1, \ldots, 54\}$ of $vec(M)$.

$$vec(M)^\top \cdot e_l^{54} \leq (1 + \gamma_l) \cdot vec(M)_l \qquad l \in \{1, \ldots, 54\} \tag{6.14}$$

$$vec(M)^\top \cdot (-e_l^{54}) \leq (\rho_l - 1) \cdot vec(M)_l \qquad l \in \{1, \ldots, 54\} \tag{6.15}$$

DOI:10.15774/PPKE.ITK.2018.004

It is easy to see that the number of realizations increases as α grows, while the number of core reactions decreases, since the polyhedrons $\mathcal{P}_\alpha$ define an ascending system of sets. In Figure 6.2 the numbers of possible realizations and core reactions can be seen depending on α. It has to be mentioned however, that in the case of this particular kinetic system for any rate $0 < \alpha < 1$ of uncertainty the number of reactions present in the dense realization is the same, and based on the superstructure property its structure as well is unchanged.

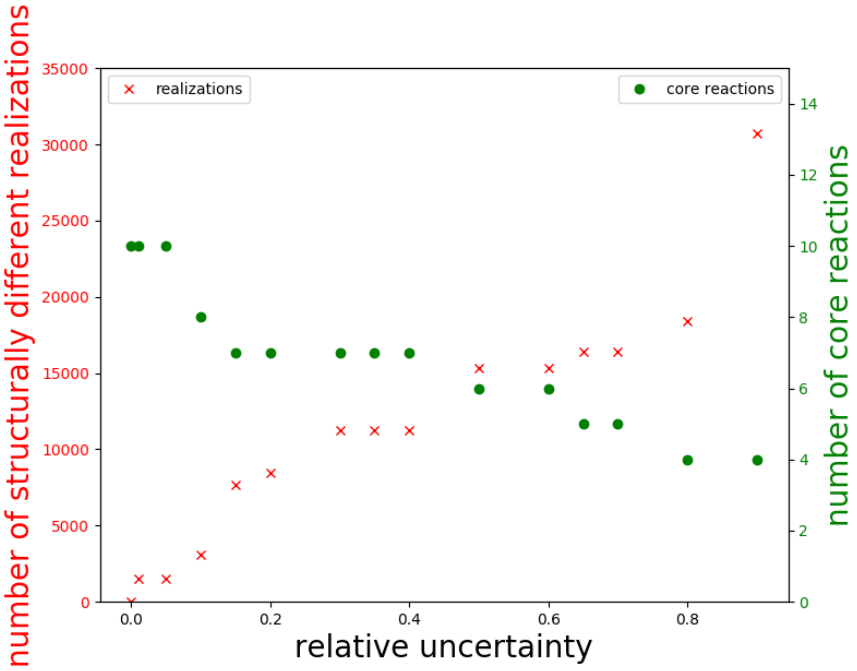


FIGURE 6.2: The number of structurally different realizations and core reactions depending on the rate of relative parametric uncertainty.

6.4 Summary

I have described an uncertain kinetic system model which is a generalization of kinetic models and it can include a finite set of additional linear constraints as well. I have proven that the superstructure property of dense realizations holds also in the case of the uncertain model, and the algorithms designed for computing the dense realization, the set of core reactions and all realization structures of a given kinetic system can be extended for the case of uncertain models. The results are described in detail in Sections 6.1.2 and 6.2, and summarized in Thesis IV.

The related publications are [54] and [56].

DOI:10.15774/PPKE.ITK.2018.004

Chapter 7

Conclusions

In this thesis the structural properties of mass action kinetic systems were examined using the different approaches of linear algebra, graph theory and convex geometry, as well as by introducing accurate and efficient novel algorithms for the computational analysis of kinetic models applying the linear programming framework.

7.1 New scientific results

Thesis I. ***I have proven new results regarding to dense realizations of kinetic systems, using a geometric approach.***

The realizations represented as points in the Euclidean space form a convex polyhedron, and this property can be utilized efficiently from a computational point of view.

Thesis I.a ***I have proven that a dense linearly conjugate realization of a kinetic system with a fixed set of complexes and an additional finite set of linear constraints determines a superstructure considering all realizations of the constrained model.***

The superstructure property is essential for the correct operation of all the algorithms presented in this dissertation.

The results are described in detail in Section 3.1.
Related publications: [51], [57], [58]

Thesis I.b ***I have developed a novel polynomial-time algorithm to compute a dense linearly conjugate realization of a kinetic system with a fixed set of complexes and fulfilling an additional finite set of linear constraints.***

The advantage of the method is that it applies linear optimization methods, it avoids the use of integer variables, and it works for every kinetic system without restrictions on the variables. I have proven that the algorithm returns the dense linearly conjugate realization, or as special case the dense dynamically equivalent realization of any kinetic

DOI:10.15774/PPKE.ITK.2018.004

system. This algorithm is applied as a subroutine in the algorithms presented in Theses II, III.a, III.b and IV.b.

I have shown that even if there are arbitrarily predefined upper bounds considering the variables the set of possible reaction graph structures representing linearly conjugate realizations is the same as in the unbounded case, therefore the computer implementations of the algorithms presented in this dissertation can work accurately.

The results are described in detail in Section 3.2.
Related publications: [51], [57]

Thesis II. ***I have proposed a new algorithm for computing a weakly reversible linearly conjugate realization of a kinetic system by extending the method introduced in*** [22]**.**

I have proven that the algorithm runs in polynomial time, and it returns a dense weakly reversible linearly conjugate realization of the kinetic system, if it exists.

I have also shown that the computed dense realization defines a superstructure among all linearly conjugate weakly reversible realizations of the kinetic system.

The results are described in detail in Chapter 4.
Related publications: [51], [57], [58]

Thesis III. ***I have achieved new results on computing all possible reaction graph structures representing linearly conjugate realizations of a kinetic system.***

Thesis III.a ***I have proven the correctness of a new algorithm for computing all possible reaction graph structures representing linearly conjugate realizations of a kinetic system on a fixed set of complexes.***

The algorithm is the first method in the literature for computing all the reaction graph structures realizing a given kinetic dynamics.

The computation might require exponential time because of the large number of possible structures, however, between the determination of two different structures polynomial time is elapsed. Furthermore, it is possible to apply parallel implementation of the algorithm using e.g. many core architectures.

The results are described in detail in Section 5.1.
Related publications: [52], [55], [59]

Thesis III.b ***I have designed a new efficient algorithm for computing all structurally different linearly conjugate realizations of a kinetic system.***

I have proven that this algorithm also returns all possible reaction graph structures representing linearly conjugate realizations of a kinetic system.

I have also shown that the algorithm returns every realization only once, furthermore, it is also suitable for parallel implementation.

DOI:10.15774/PPKE.ITK.2018.004

The performance of the new algorithm has been compared to that of the algorithm in Thesis III.a, and considering all the examples the number of required optimization steps decreased by more than 80% in the case of the new algorithm.

The results are described in detail in Section 5.2.
Related publications: [53], [60]

Thesis IV. ***I have proven new results regarding to special uncertain kinetic system models, where the parameters are in a convex polyhedron.***

The introduced model is a generalization of the original kinetic model that can include a finite set of additional linear constraints as well.

Thesis IV.a ***I have shown that the superstructure property of dense realizations holds also in the case of uncertain kinetic systems.***

This property depends on the fact that the set of solutions of an uncertain kinetic model is a convex polyhedron. The results are described in detail in Section 6.1.2.

Thesis IV.b ***I have proven that the algorithms designed for computing the dense realization, the set of core reactions and all realization structures of a given kinetic system can be extended for the case of uncertain kinetic systems.***

I have also shown that the algorithm developed for computing all structurally different realizations and presented in Thesis III.b is suitable for parallel implementation.

The results are described in detail in Section 6.2.

Related publications: [54], [56]

7.2 Application possibilities

The possibilities of applying existing algorithms is wide, since these can often be used as parts of other computational methods. The algorithm developed for computing dense realizations has already been applied as subroutine in all the other algorithms introduced in this thesis.

The algorithm extended for the computation of weakly reversible linearly conjugate realizations is the first method for solving this problem, which will possibly generate some new ideas and interesting structure based results considering these special realizations. For example one can take advantage of the fact that the algorithm returns the dense weakly reversible realization which defines a superstructure among weakly reversible realizations of the kinetic system. Furthermore, the algorithm can be generalized to the case of constrained kinetic systems, and by using this kind of computation steps it is possible to design an algorithm for determining every weakly reversible realization corresponding to a given kinetic system.

DOI:10.15774/PPKE.ITK.2018.004

The algorithms designed for determining the set of possible reaction graph structures can be applied for the accurate computation of realizations which are more difficult to characterize, for example sparse realizations. An other application of this computational method might be the CRN design based on dynamics.

It is clear that the kinetic model defined with uncertain parameters has the biggest potential in practical applications. For example in the case of a system model identified by the application of noisy measurements the kinetic model with polyhedric uncertainty introduced in this thesis can be defined using the estimated values of the parameters.

7.3 Plans for future work

I intend to further examine the structural properties of kinetic systems and to develop new computational methods by analysing the properties of chemical reaction networks and the results of the known algorithms. The following are the topics of interest:

- Analysis of the large number of possible reaction graph structures characterizing realizations of a kinetic system returned by Algorithm 5 using structural and algebraic properties.
- Symbolic implementation of Algorithm 5 at least in the dynamically equivalent case in order to be able to characterize the possible reaction rates in the computed structures.
- Development of more advanced distinguishability analysis methods using the structural properties of dense and sparse realizations.

DOI:10.15774/PPKE.ITK.2018.004

References

[1] W. M. Haddad, VS. Chellaboina, and Q. Hui. *Nonnegative and Compartmental Dynamical Systems*. Princeton University Press, 2010.

[2] N. Samardzija, L. D. Greller, and E. Wassermann. Nonlinear chemical kinetic schemes derived from mechanical and electrical dynamical systems. *Journal of Chemical Physics*, 90 (4):2296–2304, 1989.

[3] L. Brenig. Complete factorisation and analytic solutions of generalized Lotka-Volterra equations. *Physics Letters A*, 133:378–382, 1988.

[4] A. Figueiredo, T. M. Rocha Filho, and L. Brenig. Necessary conditions for the existence of quasi-polynomial invariants: the quasi-polynomial and Lotka-Volterra systems. *Physica A*, 262:158–180, 1999.

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

[6] V. Hárs and J. Tóth. On the inverse problem of reaction kinetics. In M. Farkas and L. Hatvani, editors, *Qualitative Theory of Differential Equations*, volume 30 of *Coll. Math. Soc. J. Bolyai*, pages 363–379. North-Holland, Amsterdam, 1981.

[7] F. Horn and R. Jackson. General mass action kinetics. *Archive for Rational Mechanics and Analysis*, 47:81–116, 1972.

[8] G. Craciun and C. Pantea. Identifiability of chemical reaction networks. *Journal of Mathematical Chemistry*, 44:244–259, 2008.

[9] M. D. Johnston and D. Siegel. Linear conjugacy of chemical reaction networks. *Journal of Mathematical Chemistry*, 49:1263–1282, 2011.

[10] M. Feinberg. Chemical reaction network structure and the stability of complex isothermal reactors - I. The deficiency zero and deficiency one theorems. *Chemical Engineering Science*, 42 (10):2229–2268, 1987.

[11] G. Craciun, Y. Tang, and M. Feinberg. Understanding bistability in complex enzyme-driven reaction networks. *Proc. of the National Academy of Sciences of the USA*, 103 (23):8697–8702, 2006.

[12] G. Craciun. Toric differential inclusions and a proof of the global attractor conjecture. `https://arxiv.org/abs/1501.02860`, 2015.

DOI:10.15774/PPKE.ITK.2018.004

[13] D. Angeli. A tutorial on chemical network dynamics. *European Journal of Control*, 15:398–406, 2009.

[14] E. Sontag. Structure and stability of certain chemical networks and applications to the kinetic proofreading model of T-cell receptor signal transduction. *IEEE Transactions on Automatic Control*, 46:1028–1047, 2001.

[15] M. D. Johnston, D. Siegel, and G. Szederkényi. Dynamical equivalence and linear conjugacy of chemical reaction networks: new results and methods. *MATCH Commun. Math. Comput. Chem.*, 68:443–468, 2012.

[16] J. Rudan, G. Szederkényi, and K. M. Hangos. Efficiently computing alternative structures of large biochemical reaction networks using linear programming. *MATCH Commun. Math. Comput. Chem.*, 71:71–92, 2014.

[17] G. Szederkényi. Computing sparse and dense realizations of reaction kinetic systems. *Journal of Mathematical Chemistry*, 47:551–568, 2010.

[18] Z. A. Tuza, G. Szederkényi, K. M. Hangos, and J. R. Banga A. A. Alonso. Computing all sparse kinetic structures for a Lorenz system using optimization methods. *International Journal of Bifurcation and Chaos*, accepted:to appear, 2013.

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

[20] G. Szederkényi and K. M. Hangos. Finding complex balanced and detailed balanced realizations of chemical reaction networks. *Journal of Mathematical Chemistry*, 49:1163–1179, 2011.

[21] I. Nagy, B. Kovács, and J. Tóth. Detailed balance in ion channels: application of Feinberg's theorem. *React. Kinet. Catal. Lett.*, 96:263–267, 2009.

[22] G. Szederkényi, K. M. Hangos, and Zs. Tuza. Finding weakly reversible realizations of chemical reaction networks using optimization. *MATCH Commun. Math. Comput. Chem.*, 67:193–212, 2012.

[23] M. D. Johnston, D. Siegel, and G. Szederkényi. Computing weakly reversible linearly conjugate chemical reaction networks with minimal deficiency. *Mathematical Biosciences*, 241:88–98, 2013.

[24] G. Lipták, G. Szederkényi, and K. M. Hangos. Computing zero deficiency realizations of kinetic systems. *Systems & Control Letters*, 81:24–30, 2015.

[25] G. B. Dantzig. *Linear Programming and Extensions.* Princeton University Press, 1963.

[26] G. B. Dantzig. Origins of the simplex method. *Technical Report Sol*, 5, 1987.

[27] V. Klee and G. J. Minty. How good is the simplex algorithm? In *Proceedings of the Third Symposium on Inequalities, September 1–9, 1969, University of California, Los Angeles, USA*, page 159–175, 1969.

DOI:10.15774/PPKE.ITK.2018.004

[28] A. Schrijver. *Theory of Linear and Integer Programming.* Wiley & Sons, 1998.

[29] H. Marchand, A. Martin, and R. Weismanteland L. Wolsey. Cutting planes in integer and mixed integer programming. *Discrete Applied Mathematics*, 123:387–446, 2002.

[30] A. H. Land and A. G. Doig. An automatic method of solving discrete programming problems. *Econometrica*, 28 (3):497–520, 1960. DOI: 10.2307/1910129.

[31] M. Feinberg. *Lectures on chemical reaction networks.* Notes of lectures given at the Mathematics Research Center, University of Wisconsin, 1979.

[32] F. Horn. Necessary and sufficient conditions for complex balancing in chemical kinetics. *Archive for Rational Mechanics and Analysis*, 49:172–186, 1972.

[33] M. Feinberg. Complex balancing in general kinetic systems. *Archive for Rational Mechanics and Analysis*, 49:187–194, 1972.

[34] D. F. Anderson. A proof of the Global Attractor Conjecture in the single linkage class case. *SIAM Journal on Applied Mathematics*, 71:1487–1508, 2011. http://arxiv.org/abs/1101.0761,.

[35] M. D. Johnston, D. Siegel, and G. Szederkényi. A linear programming approach to weak reversibility and linear conjugacy of chemical reaction networks. *Journal of Mathematical Chemistry*, 50:274–288, 2012.

[36] J. Rudan, G. Szederkényi, and K. M. Hangos. Polynomial time algorithms to determine weakly reversible realizations of chemical reaction networks. *Journal of Mathematical Chemistry*, 52(5):1386–1404, 2014.

[37] V. Chellaboina, S. P. Bhat, W. M. Haddad, and D. S. Bernstein. Modeling and analysis of mass-action kinetics – nonnegativity, realizability, reducibility, and semistability. *IEEE Control Systems Magazine*, 29:60–78, 2009.

[38] G. Farkas. Kinetic lumping schemes. *Chemical Engineering Science*, 54:3909–3915, 1999.

[39] G. Szederkényi, J. R. Banga, and A. A. Alonso. Inference of complex biological networks: distinguishability issues and optimization-based solutions. *BMC Systems Biology*, 5:177, 2011.

[40] A. Császár, L. Jicsinszky, and T. Turányi. Generation of model reactions leading to limit cycle behaviour. *Reaction Kinetics and Catalysis Letters*, 18:65–71, 1981.

[41] G. Shinar, J. D. Rabinowitz, and U. Alon. Robustness in glyoxylate bypass regulation. *PLOS Computational Biology*, 5(3):e1000297, 2009.

[42] M. Feinberg. The existence and uniqueness of steady states for a class of chemical reaction networks. *Archive for Rational Mechanics and Analysis*, 132:311–370, 1995.

[43] D. F. Anderson. Boundedness of trajectories for weakly reversible, single linkage class reaction systems. *Journal of Mathematical Chemistry*, 49:1–16, 2011. DOI: 10.1007/s10910-011-9886-4.

DOI:10.15774/PPKE.ITK.2018.004

[44] A. V. Aho, J. E. Hopcroft, and J. D. Ullman. *Data Structures and Algorithms.* Addison-Wesley, 1983.

[45] M. Sharir. A strong-connectivity algorithm and its application in data flow analysis. *Computers and Mathematics with Applications*, 7:67–72, 1981.

[46] R. E. Tarjan. Depth-first search and linear graph algorithms. *SIAM Journal of Computation*, **1** (2):146–160, 1972.

[47] Z. A. Tuza and G. Szederkényi. Computing core reactions of uncertain polynomial kinetic systems. In *23rd Mediterranean Conference on Control and Automation (MED), June 16-19, 2015. Torremolinos, Spain*, pages 1187–1194, 2015.

[48] Carsten Conradi, Dietrich Flockerzi, Jörg Raisch, and Jörg Stelling. Subnetwork analysis reveals dynamic features of complex (bio)chemical networks. *Proceedings of the National Academy of Sciences*, 104(49):19175–19180, 2007.

[49] The Math Works, Inc., Natick, MA. *Matlab User's Guide*, 2000.

[50] J. Löfberg. YALMIP : A toolbox for modeling and optimization in MATLAB. In *Proceedings of the CACSD Conference*, Taipei, Taiwan, 2004.

DOI:10.15774/PPKE.ITK.2018.004

The author's journal publications

[51] B. Ács, G. Szederkényi, Z. A. Tuza, and Zs. Tuza. Computing linearly conjugate weakly reversible kinetic structures using optimization and graph theory. *MATCH Commun. Math. Comput. Chem.*, 74:481–504, 2015. IF: 3.858, Q1.

[52] B. Ács, G. Szederkényi, Zs. Tuza, and Z. A. Tuza. Computing all possible graph structures describing linearly conjugate realizations of kinetic systems. *Computer Physics Communications*, 204:11–20, 2016. IF: 3.936, D1.

[53] B. Ács, G. Szederkényi, and D. Csercsik. A new efficient algorithm for determining all structurally different realizations of kinetic systems. *MATCH Commun. Math. Comput. Chem.*, 77:299–320, 2017. IF: 3.139, Q1.

[54] B. Ács, G. Szlobodnyik, and G. Szederkényi. A computational approach to the structural analysis of uncertain kinetic systems. *submitted*, 2017. `https://arxiv.org/abs/1704.08633`.

DOI:10.15774/PPKE.ITK.2018.004

The author's conference publications

[55] Z. A. Tuza, B. Ács, G. Szederkényi, and F. Allgöwer. Efficient computation of all distinct realization structures of kinetic systems. In *6th IFAC Conference on Foundations of Systems Biology in Engineering, October 9-12, 2016 Magdeburg, Germany, IFAC-PAPERSONLINE 49:(26)*, pages 194–200, 2016.

[56] G. Szederkényi, B. Ács, and G. Szlobodnyik. Structural analysis of kinetic systems with uncertain parameters. In *2nd IFAC Workshop on Thermodynamic Foundations for a Mathematical Systems Theory - TFMSTII, September 28-30, 2016, Vigo, Spain, IFAC-PAPERSONLINE 49:(24)*, pages 24–27, 2016.

[57] B. Ács. A new method for computing linearly conjugate weakly reversible structures of kinetic polynomial systems. In C. Wiuf and E. Feliu, editors, *Workshop on Mathematical Trends in Reaction Network Theory, July 1-3, 2015, Copenhagen, Denmark*, page 21, 2015.

[58] B. Ács. Computing linearly conjugate weakly reversible kinetic structures using graph theory and optimization. In *PhD Proceedings Annual Issues of the Doctoral School, Faculty of Information Technology and Bionics*, pages 69–71, 2014.

[59] B. Ács. Computational analysis of kinetic systems. In *PhD Proceedings Annual Issues of the Doctoral School, Faculty of Information Technology and Bionics*, pages 107–110, 2015.

[60] B. Ács. Case studies for the structural analysis of biochemically motivated nonlinear systems. In *PhD Proceedings Annual Issues of the Doctoral School, Faculty of Information Technology and Bionics*, pages 117–120, 2016.