

# On the Complexity of Reconstructing Chemical Reaction Networks

Rolf Fagerberg<sup>1</sup>, Christoph Flamm<sup>2</sup>,  
Daniel Merkle<sup>1</sup>, Philipp Peters<sup>1</sup>, and Peter F. Stadler<sup>2-7</sup>

<sup>1</sup> Department of Mathematics and Computer Science  
University of Southern Denmark, Denmark

<sup>2</sup> Institute for Theoretical Chemistry, University of Vienna, Austria.

<sup>3</sup> Bioinformatics Group, Department of Computer Science, and  
Interdisciplinary Center for Bioinformatics, University of Leipzig, Germany.

<sup>4</sup> Max Planck Institute for Mathematics in the Sciences, Leipzig, Germany.

<sup>5</sup> Fraunhofer Institute for Cell Therapy and Immunology, Leipzig, Germany.

<sup>6</sup> Center for non-coding RNA in Technology and Health  
University of Copenhagen, Denmark.

<sup>7</sup> Santa Fe Institute, 1399 Hyde Park Rd, Santa Fe, NM 87501, USA

September 25, 2013

## Abstract

The analysis of the structure of chemical reaction networks is crucial for a better understanding of chemical processes. Such networks are well described as hypergraphs. However, due to the available methods, analyses regarding network properties are typically made on standard graphs derived from the full hypergraph description, e.g. on the so-called species and reaction graphs. However, a reconstruction of the underlying hypergraph from these graphs is not necessarily unique. In this paper, we address the problem of reconstructing a hypergraph from its species and reaction graph and show NP-completeness of the problem in its Boolean formulation. Furthermore we study the problem empirically on random and real world instances in order to investigate its computational limits in practice.

## 1 Introduction

The use of graph models of chemical reaction networks has a long history in physical chemistry as a means of connecting structural properties of chemical reaction systems with the system's dynamical behavior. The aim of this line of research to determine constraints on the dynamics in the absence of detailed quantitative information on the reaction kinetics, as reviewed e.g. in [13].

Graph models are derived either directly from a combinatorial view on the reaction networks, i.e., from the collection of chemical reaction equations, or from the system of ordinary differential equations that describe the reaction kinetics. In the S-graph (also called species graph, compound graph) for instance, we have a (directed) edge from species  $v_i$  to species  $v_j$  iff a chemical reaction produces output molecule  $v_j$  from input molecule  $v_i$ . In the closely related interaction graph, we have an edge from

species  $v_i$  to species  $v_j$  iff  $\partial[v_i]/\partial[v_j] \neq 0$ , where  $[v_i]$  denotes the concentration of species  $v_i$  as usual in the chemical literature. Here, edges are in addition endowed with a sign given by the partial derivative. Strong constraints on the dynamics can be obtained if the interaction graph has no directed cycle with an odd number of negative signs; for instance multistability can be ruled out [33, 35].

The bipartite SR-graph (species-reaction graph) [11] treats species and reactions as two types of vertices. In its directed version, there is a directed edge from a species  $v_i$  to a reaction node  $e$  if  $v_i$  is a reactant (input molecule) in  $e$ , and a directed edge from  $e$  to a species  $v_j$  if  $v_j$  is a product (output molecule) in  $e$ . This model is equivalent to the hypergraph formulation we use in this paper (see Sec. 2). The closely related plain SR-graph is undirected and instead uses an elaborate labeling scheme; it can be viewed as a special type of Petri-net graphs. SR-graphs have a close relationship to classical deficiency theory [16, 20], the historically first approach towards a qualitative theory of kinetics of reaction networks (see also [30]). It can be seen as an undirected version of the (directed) incidence graph of the directed hypergraph advocated e.g. in [5, 41]. The directed version of the SR-Graph has been explored e.g. in [21, 29, 38]. These constructions have many uses, including providing necessary conditions for Turing instabilities in reaction-diffusion systems [28] and providing conditions to rule out multistability or oscillations. A direct relation between the directed SR-graphs and interaction graphs that simplifies many proofs in described is [23].

A different motivation for graph models for chemical reaction systems arose a decade ago from attempts to identify universality classes of networks that appear in nature or have been produced by technological or social processes [40]. In this context, which focuses on connectivity properties and path statistics as means of capturing global features, species (S-)graphs and reaction (R-)graphs were used in particular to compare metabolic networks [39]. Being complementary to the S-graph, the R-graph has reactions as vertices and has an edge  $(a, b)$  iff an output molecule of reaction  $a$  is an input molecule of reaction  $b$ .

S- and R-graph are also used for the analysis of networks, where the directed hypergraphs, or equivalently, a directed variant of the bipartite SR-graphs, are reformulated to S- and R- graphs. This is motivated mostly by the available statistical toolkit employed for most other networks models, which works on simple graphs and not hypergraphs, and the desire to place chemical reaction networks within the scheme of small world and scale free networks [39]. S-graphs furthermore have proved useful as means of exploring large chemical networks [24]. They are of practical use e.g. in approximation algorithms for the minimal seed set problem [8], i.e., for finding the smallest set of substrates that can generate all metabolites.

Methods for a systematic sampling of chemical networks that share non-trivial, chemically motivated characteristic features would be very valuable, as this would allow for statistically significant statements for networks that appear in nature. Having an identical S- and R-graph is one very natural characteristic feature that can be used.

Both the S-graph and the R-graph obviously capture only partial information on the chemical network from which they are derived. It is less obvious, however, to what extent S-graph and R-graph together determine the SR-graph and, equivalently, the underlying hypergraph structure. Conversely, is there, for any given *pair* of S- and R-graphs, a chemical network from which they derive?

In this paper we address the latter question in detail by phrasing it as a combinatorial decision problem, called the Compound-Reaction-Reconstruction (CRR) Problem. We prove that it is NP-complete and investigate its computational limits, using a reformulation of the CRR problem as a Boolean satisfiability problem.

This paper is organized as follows: In Section 2 we introduce the necessary graph theoretic formalism. The definition of the CRR problem and the NP-completeness proof are presented in Section 3. In Sec. 4 and 5, we reformulate the CRR problem as a Boolean satisfiability problem and investigate the practical applicability and performance of different declarative solving methods. We furthermore consider in some detail the differences between random instances and instances derived from known metabolic networks.

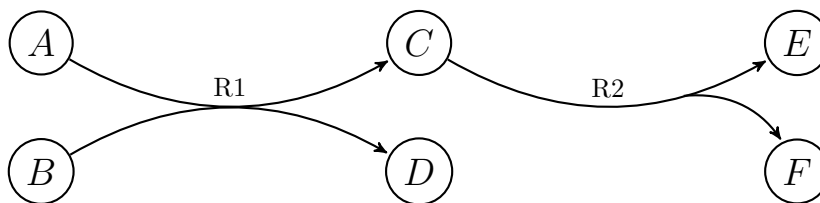


Figure 1: A directed hypergraph, consisting of two reactions  $\{R1, R2\}$  and six chemical species  $\{A, B, \dots, F\}$ . Hyperarc R1 has  $t(R1) = \{A, B\}$  and  $h(R1) = \{C, D\}$ , hyperarc R2 has  $t(R2) = \{C\}$  and  $h(R2) = \{E, F\}$ .

## 2 Formalization of Chemical Reaction Networks

### 2.1 Chemical Reaction Networks as Directed Hypergraphs

In this paper, a *chemical reaction network* consists of set  $V = \{v_1, v_2, \dots\}$  of molecule types (usually termed *compounds* or *species* [26]) and a collection of reactions  $A$ . Each reaction  $a$  transforms a set  $a^- \subseteq V$  of reactants (possibly with multiplicities) into a set  $a^+ \subseteq V$  of products (possibly with multiplicities). In chemical terms this is written as *reaction equations* of the form

$$\sum_{v_i \in a^-} \alpha_{ia} v_i \rightarrow \sum_{v_i \in a^+} \beta_{ia} v_i \quad (1)$$

where the *stoichiometric coefficients*  $\alpha_{ia}$  and  $\beta_{ia}$  are non-negative integers giving the multiplicity with which a compound  $v_i \in V$  appears as reactant or product in the given reaction  $a$ . We note that in general  $a^- \cap a^+$  does not need to be empty. The molecules in the intersection of products and reactants are the so-called catalysts of the reaction. Each reaction  $(a^-, a^+)$  can be interpreted as a directed edge in a directed hypergraph.

More formally, a *directed hypergraph*  $H$  is a pair  $H = (V, A)$ , with a set of vertices  $V$  and a set of hyperarcs  $A$ , where each hyperarc  $a \in A$  is an ordered pair  $(t(a), h(a))$ . The tail  $t(a) = a^-$  of the hyperarc in our setting refers to the reactants, while its head  $h(a) = a^+$  identifies the products [17, 3]. Fig. 1 illustrates the hypergraph of a small chemical reaction network with two reactions.

### 2.2 Hypergraphs as Matrices

The stoichiometric matrix  $\Sigma$  with entries

$$\sigma_{ia} = \beta_{ia} - \alpha_{ia} \quad (2)$$

provides a complete description of the mass balance of the each reaction in the chemical reaction network. Each row of the matrix  $\Sigma$  corresponds to a species, while each column is identified with a reaction. The stoichiometric matrix is a complete encoding of the chemical reaction network (hence of the directed hypergraph), provided  $t(a) \cap h(a) = \emptyset$  for all reactions  $a \in A$ . Note that  $\sigma_{ia} < 0$  if  $v_i$  is consumed by reaction  $a$  while  $\sigma_{ia} > 0$  if  $v_i$  is produced.

In the following, we focus on the topological structure of the chemical reaction network and will ignore the multiplicities of reactants and products. This corresponds to replacing  $\sigma_{ia}$  by its sign in  $\Sigma$ . Instead of using this reduced version of  $\Sigma$ , however, it will be more convenient to use two *binary* incidence matrices,  $\mathbf{E}$  and  $\mathbf{P}$ , defined by

$$e_{ia} = \begin{cases} 1 & \text{iff } v_i \in t(a) \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$p_{ai} = \begin{cases} 1 & \text{iff } v_i \in h(a) \\ 0 & \text{otherwise} \end{cases}$$

Here,  $e_{ia}$  is  $\text{sgn } \alpha_{ia}$  and  $p_{ai}$  is  $\text{sgn } \beta_{ia}$ . The rows of the  $n \times m$  matrix  $E$  correspond to the reactants of each reaction, while the columns of the  $m \times n$  matrix  $P$  corresponds to the products. The matrices  $E$  and  $P$  corresponding to the hypergraph of Fig. 1 are shown as part of Fig. 4.

In the S-graph (species graph), two species  $v_i, v_j \in V$  are adjacent iff there is a reaction that has species  $v_i$  as reactant and species  $v_j$  as product. Correspondingly, the R-graph (reaction graph) two reactions  $a$  and  $b$  are adjacent iff there is a species  $v$  that is a product of  $a$  and a reactant of  $b$ . In other words, the adjacency matrices  $S = (s_{ij})$  and  $R = (r_{ab})$  are given by:

$$s_{ij} = \begin{cases} 1 & \text{iff } \exists a \in A : v_i \in t(a) \wedge v_j \in h(a) \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

$$r_{ab} = \begin{cases} 1 & \text{iff } \exists v \in V : v \in h(a) \wedge v \in t(b) \\ 0 & \text{else} \end{cases}$$

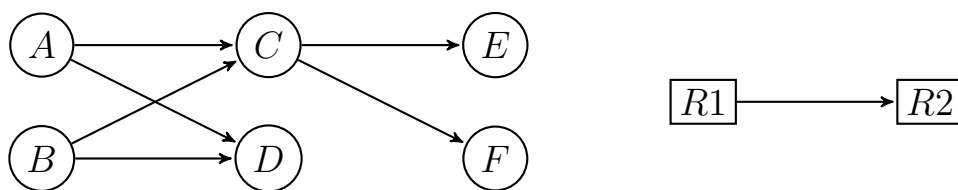


Figure 2: The directed hypergraph from Fig. 1 has the  $S$ -graph shown on the left and the  $R$ -graph, shown on the right.

### 2.3 Relationships Between $S$ , $R$ , $E$ , and $P$

The two incidence matrices  $E$  and  $P$  together contain all necessary information to uniquely define the corresponding hypergraph  $H$ . In contrast, the adjacency matrices  $S$  and  $R$  do not determine  $H$  uniquely, in the sense that two different hypergraphs can lead to the same  $S$  and  $R$ , as illustrated in Fig. 3. Also, for a given pair of matrices  $S'$  and  $R'$ , there may exist no hypergraph having  $S'$  and  $R'$  as its species and reaction graph. We provide an example of this in the appendix. These observations lead to the question we study in the remaining sections of this paper.

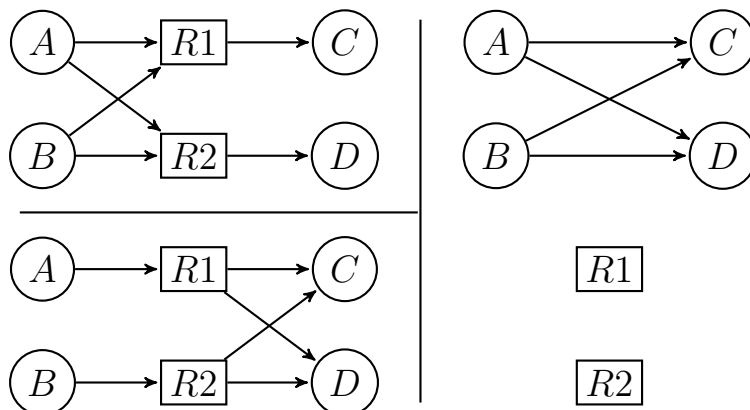


Figure 3: Ambiguous  $S$ - and  $R$ -graphs: On the left, two different hypergraphs are shown, both giving rise to the same  $S$  and  $R$  shown on the right.

The four matrices  $S$ ,  $R$ ,  $E$ , and  $P$  can be combined in a natural way to the *total graph*  $T$  [9, 31]. It

corresponds to a  $(n + m) \times (n + m)$  binary matrix of which  $S$ ,  $R$ ,  $E$ , and  $P$  form four blocks:

$$T = \begin{pmatrix} S & E \\ P & R \end{pmatrix} \quad (5)$$

Formally, the total graph has vertex set  $V \cup A$  and has edges defined as follows:  $(i, a)$  is an edge in the total graph iff  $i \in t(a)$  (i.e., iff  $e_{ia} = 1$ ). Correspondingly,  $(b, j)$  is an edge iff  $j \in h(b)$  (i.e., iff  $p_{bj} = 1$ ). For pairs of vertices of  $H$  we have  $(i, j) \in E(T)$  iff  $i, j$  are adjacent in the species graph, i.e.,  $s_{ij} = 1$ . For pairs of arcs of  $H$ , finally,  $(a, b) \in E(T)$  iff  $a, b$  are adjacent in the reaction graph, i.e.,  $r_{ab} = 1$ . An illustration of the total graph of the hypergraph of Fig. 1 is displayed in Fig. 4.

Figure 4: The total graph of the hypergraph from Fig. 1. All non-zero entries in column  $k$  in  $E$  are reactants of reaction  $k$ . All non-zero entries in row  $k$  in  $P$  are products of reaction  $k$ . An entry  $s_{ij} = 1$  (or  $r_{ab} = 1$ , resp.) requires at least one witness, which is R1 for the framed entry in the table, as indicated by the gray marked entries in  $E$  and  $P$ .

$$T = \begin{pmatrix} S & E \\ P & R \end{pmatrix} = \begin{array}{cccccc|cc} & & & & & & \text{R1} & \\ & & & & & & \downarrow & \\ & 0 & 0 & 1 & 1 & 0 & 0 & 1 & 0 \\ & 0 & 0 & \boxed{1} & 1 & 0 & 0 & \text{1} & 0 \\ & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline \text{R1} \rightarrow & 0 & 0 & \text{1} & 1 & 0 & 0 & 0 & 1 \\ & 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \end{array}$$

The definitions of the S- and R-matrices can be recast in terms of  $E$  and  $P$ . For  $S$  we have  $s_{ij} = 1 \iff \exists a : (e_{ia} = 1 \wedge p_{aj} = 1)$ . The reaction  $a$  for which the conjunction  $(e_{ia} \wedge p_{aj})$  is true is called a *witness*  $w_{ija}^S$  [2]. For every non-zero entry in  $S$  there must be at least one witness, while no witness exists for the zero entries. For illustration, a witness for the framed entry in  $S$  in Fig. 4 is reaction R1, and the conjunction-fulfilling entries of  $E$  and  $P$  are marked gray. For  $R$ , respectively, we have  $r_{ab} = 1 \iff \exists i : (e_{ib} = 1 \wedge p_{ai} = 1)$  and the species  $v_i$  called a witness  $w_{abi}^R$ .

Reinterpreting the matrix elements over the standard Boolean Algebra  $(\{0, 1\}, \vee, \wedge)$  allows us to rewrite the definition of  $s_{ij}$  in the follow form:

$$s_{ij} = \bigvee_{a \in A} (e_{ia} \wedge p_{aj}) \quad (6)$$

Analogously, we have

$$r_{ab} = \bigvee_{i \in V} (p_{ai} \wedge e_{ib}) \quad (7)$$

for the entries of matrix  $R$ . Using matrix notation, we can write  $S$  and  $R$  as Boolean matrix products

$$S = E \cdot P \quad R = P \cdot E \quad (8)$$

where the matrix multiplication  $\cdot$  uses the Boolean operations  $\vee$  and  $\wedge$  as addition and multiplication operations.

### 3 The Compound-Reaction-Reconstruction (CRR( $S, R$ )) Problem

Given  $H$ , or more precisely,  $E$  and  $P$ , we have seen above that  $S$  and  $R$  are uniquely determined. Here we ask the converse question. Given  $S$  and  $R$ , is there a hypergraph  $H$  that has  $S$  and  $R$  as its S- and

R-matrices, respectively. We call this problem the *Compound-Reaction-Reconstruction (CRR(S,R)) problem*.

### 3.1 Problem Definition

**Definition 3.1. CRR(S, R) problem:** *Given a Boolean  $n \times n$  matrix  $S$  and a Boolean  $k \times k$  matrix  $R$ , is there a Boolean  $n \times k$  matrix  $E$  and a Boolean  $k \times n$  matrix  $P$  so that  $S = E \cdot P$  and  $R = P \cdot E$  ?*

The CRR( $S,R$ ) problem can be seen as a special kind of matrix decomposition problem. Matrix decomposition methods have been studied extensively, but mostly on real- and integer-valued matrices, as Singular value, QR- or LU-factorization. In 2008, Miettinen et al. showed that simple Boolean matrix decomposition (i.e., the first half of Eqn. (8)) is NP-hard [27].

### 3.2 NP-Completeness

In this section, we study the complexity of the CRR( $S,R$ ) problem. A problem is NP-complete if it is in NP and it is NP-hard, see [18]. We will prove NP-completeness of the CRR( $S,R$ ) problem by a reduction from the NP-complete Set Basis (SB) Problem [18, problem SP7], defined in Def. 3.2 below.

**Definition 3.2. SB(S, k) Problem:** *Input: A collection  $\mathcal{S}$  of subsets of a finite set  $\mathcal{E}$  and a positive integer  $k \leq |\mathcal{S}|$ . Is there a collection  $\mathcal{P}$  of subsets of  $\mathcal{E}$  with  $|\mathcal{P}| = k$  such that, for each  $s \in \mathcal{S}$ , there is a subcollection of  $\mathcal{P}$  whose union is  $s$ ?*

Note that in contrast to the SB Problem, which has to satisfy one equation, the CRR( $S,R$ ) problem has to fulfill the two equations of (8), i.e., entries of  $E$  and  $P$  are dependent on  $S$  and  $R$ , making it a twofold Boolean matrix decomposition. The SB( $\mathcal{S}, k$ ) problem can be formulated as Boolean matrix decomposition as it was noted in [27], where the subset collection  $\mathcal{S}$  is represented as a matrix  $S$  (i.e., each Boolean entry  $s_{ij} = 1$  denotes that the  $i$ -th subset contains the  $j$ -th element). Hence,  $P$  is the basis vector matrix and  $E$  is the usage matrix. Thus, the matrix decomposition formulation (SB( $S,k$ ) Problem) can be given as follows: Given an  $n \times m$  matrix  $S$  and a  $k \in \mathbb{N}^+$ , do an  $n \times k$  matrix  $E$  and a  $k \times m$  matrix  $P$  exist for which the following hold:

$$S = E \cdot P$$

We will now show how to transform any instance of the SB( $S,k$ ) problem into an instance of the CRR( $S,R$ ) problem. The transformation will be via an intermediate problem called SBMOD. We first modify matrix  $S$  to an  $\bar{n} \times \bar{m}$  matrix  $\bar{S}$ , with  $\bar{n} = n + 2$ ,  $\bar{m} = m + 2$ , and  $\bar{k} = k + 2$ , as follows:

$$\bar{S} = \begin{pmatrix} 1 & \cdots & \cdots & 1 \\ \vdots & 0 & \cdots & 0 \\ \vdots & \vdots & \boxed{S} & \\ 1 & 0 & & \end{pmatrix}$$

With the extended matrix  $\bar{S}$ , we define a modified version of the SB( $S,k$ ) problem:

**Definition 3.3. SBMOD( $\bar{S}, \bar{k}$ ) Problem:** *Given an  $\bar{n} \times \bar{m}$  matrix  $\bar{S}$  in the form above and a  $\bar{k} \in \mathbb{N}^+$ , do an  $\bar{n} \times \bar{k}$  matrix  $\bar{E}$  and a  $\bar{k} \times \bar{m}$  matrix  $\bar{P}$  exist for which  $\bar{S} = \bar{E} \cdot \bar{P}$ ?*

**Lemma 3.4.** *If  $\bar{E}$  and  $\bar{P}$  exist for an instance of SBMOD( $\bar{S}, \bar{k}$ ), then we can assume they have the following structure:*

$$\bar{E} = \left( \begin{array}{cccc|cccc} 1 & \cdots & \cdots & 1 & & & & \\ 1 & 0 & \cdots & 0 & & & & \\ \bar{e}_{31} & 0 & & & & & & \\ \vdots & \vdots & & & & & & \\ \bar{e}_{\bar{n}1} & 0 & & & & & & \end{array} \right), \bar{P} = \left( \begin{array}{cccc|cccc} 1 & 0 & \cdots & \cdots & 0 & & & \\ \vdots & 1 & \bar{p}_{23} & \cdots & \bar{p}_{2\bar{m}} & & & \\ \vdots & \bar{p}_{32} & & & & & & \\ \vdots & \vdots & & & & & & \\ \vdots & \vdots & & & & & & \\ 1 & \bar{p}_{\bar{k}2} & & & & & & \end{array} \right)$$

where the entries  $\bar{e}_{ij}$  and  $\bar{p}_{kl}$  are not specified.

*Proof.* Each row of  $\bar{S}$  is a binary linear combination of rows in  $\bar{P}$ . Therefore, to obtain the second row of  $\bar{S}$ , there must exist the row  $(1, 0, \dots, 0)$  in  $\bar{P}$ . Note that permuting columns of matrix  $\bar{E}$  concurrently with rows of  $\bar{P}$  will not change the result of  $\bar{E} \cdot \bar{P}$ . We can therefore permute the columns of  $\bar{E}$  and the rows of  $\bar{P}$  concurrently, such that  $(1, 0, \dots, 0)$  becomes the first row of  $\bar{P}$ , without changing  $\bar{E} \cdot \bar{P}$ . The second row of  $\bar{E}$  are the coefficients for the linear combination for the second row of  $\bar{S}$ . We can choose it to be  $(1, 0, \dots, 0)$ , again without changing  $\bar{E} \cdot \bar{P}$ .

The first row of  $\bar{E}$  are the coefficients for the linear combination of rows in  $\bar{P}$  giving the first row of  $\bar{S}$  (which are all 1's). Converting any zeros to ones does not change  $\bar{S}$ , therefore we choose the first row of  $\bar{E}$  to be  $(1, \dots, 1)$ .

As the first row of  $\bar{S}$  is 1's only, we can choose the first column of  $\bar{P}$  to be  $(1, \dots, 1)$ .

Due to the first row of  $\bar{S}$ , there must exist a row of  $\bar{P}$ , where  $\bar{p}_{i2} = 1$ . It is not in the first row by the choice of the first row of  $\bar{P}$  just before. We permute rows 2 to  $\bar{k}$  in  $\bar{P}$  concurrently with columns 2 to  $\bar{k}$  in  $\bar{E}$  such that entry  $\bar{p}_{22} = 1$ .

From  $\bar{s}_{i2} = 0$  and  $\bar{p}_{22} = 1$  follows, that  $\bar{e}_{i2} = 0$  for  $i \geq 3$ .  $\square$

**Lemma 3.5.** *The SBMOD( $\bar{S}, \bar{k}$ ) problem has a solution iff the SB( $S, k$ ) problem has a solution.*

*Proof.* Forward direction: Let  $S, k$  be given, and assume the SBMOD( $\bar{S}, \bar{k}$ ) problem has a solution  $\bar{E}, \bar{P}$ . By Lemma 3.4, we can assume the solution  $\bar{E}, \bar{P}$  to SBMOD( $\bar{S}, \bar{k}$ ) has the following form:

$$\left( \begin{array}{cccc|cccc} 1 & \cdots & \cdots & 1 & & & & \\ 1 & 0 & \cdots & 0 & & & & \\ \bar{e}_{31} & 0 & & & & & & \\ \vdots & \vdots & & & & & & \\ \bar{e}_{\bar{n}1} & 0 & & & & & & \end{array} \right) \cdot \left( \begin{array}{cccc|cccc} 1 & 0 & \cdots & \cdots & 0 & & & \\ \vdots & 1 & \bar{p}_{23} & \cdots & \bar{p}_{2\bar{m}} & & & \\ \vdots & \bar{p}_{32} & & & & & & \\ \vdots & \vdots & & & & & & \\ \vdots & \vdots & & & & & & \\ 1 & \bar{p}_{\bar{k}2} & & & & & & \end{array} \right) = \left( \begin{array}{cccc|cccc} 1 & \cdots & \cdots & 1 & & & & \\ \vdots & 0 & \cdots & 0 & & & & \\ \vdots & \vdots & & & & & & \\ \vdots & \vdots & & & & & & \\ 1 & 0 & & & & & & \end{array} \right)$$

$\bar{E} \quad \cdot \quad \bar{P} \quad = \quad \bar{S}$

For  $\bar{s}_{ij}$  with  $i, j \geq 3$  it holds:

$$\bar{s}_{ij} = \bar{e}_{i1} \cdot 0 + 0 \cdot \bar{p}_{2j} + \sum_{l \geq 3}^{\bar{k}} \bar{e}_{il} \cdot \bar{p}_{lj} = s_{i-2, j-2}$$

Hence  $\bar{E} \cdot \bar{P} = \bar{S}$ , implying that the SB( $S, k$ ) problem has a solution  $E, P$ .

Backward direction: Let  $S, k$  be given, and assume that the SB( $S, k$ ) problem has a solution  $E, P$ . A trivial solution to the SBMOD( $\bar{S}, \bar{k}$ ) problem arises from specifying as follows the entries which were left undetermined in Lemma 3.4:  $\bar{e}_{31} = \cdots = \bar{e}_{\bar{n}1} = 1$ ,  $\bar{p}_{32} = \cdots = \bar{p}_{\bar{k}2} = 0$ ,  $\bar{p}_{23} = \cdots = \bar{p}_{2\bar{m}} = 0$ . This results in:

$$\left( \begin{array}{cccc|c} 1 & \cdots & \cdots & 1 & \\ 1 & 0 & \cdots & 0 & \\ \vdots & \vdots & & & \\ 1 & 0 & & & \end{array} \right) \cdot \left( \begin{array}{cccc|c} 1 & 0 & \cdots & 0 & \\ \vdots & 1 & \cdots & 1 & \\ \vdots & 0 & & & \\ \vdots & \vdots & & & \\ \vdots & \vdots & & & \\ 1 & 0 & & & \end{array} \right) = \left( \begin{array}{cccc|c} 1 & \cdots & \cdots & 1 & \\ \vdots & 0 & \cdots & 0 & \\ \vdots & \vdots & & & \\ 1 & 0 & & & \end{array} \right)$$

□

**Lemma 3.6.** *The SB(S,k) problem is NP-complete even if S is a square matrix. The same applies to the SBMOD(S,k) problem.*

*Proof.* Showing that the two problems are in NP is done by a simple guess-and-check argument, verifying the solution in polynomial time. We now prove the NP-hardness. Assuming  $n < m$ ,  $S$  can easily be extended by  $m - n$  0-rows to gain a square matrix  $S'$ . A solution for SB( $S',k$ ) consists of a  $(n + (m - n)) \times k = m \times k$  matrix  $E'$  and a  $k \times m$  matrix  $P$  for which  $E' \cdot P = S'$ . It is easy to see that such  $E', P$  exist iff there exist an  $n \times k$  matrix  $E$  and a  $k \times m$  matrix  $P$  such that  $E \cdot P = S$  (in one direction, let  $E'$  be  $E$  extended with  $m - n$  0-rows, in the other let  $E$  be  $E'$  with last  $m - n$  rows removed). Following from Lemma 3.5, the same holds for the SBMOD( $S,k$ ) problem. The same holds for  $n > m$  by extending  $S$  by  $n - m$  0-columns and an analogous construction of  $P'$ . □

**Theorem 3.7.** *The CRR(S,R) problem is in NP-complete.*

*Proof.* To show that the CRR( $S,R$ ) problem is in NP is done by the simple guess-and-check argument (given  $E$  and  $P$ , it can be checked in polynomial time if they fulfill Eq. 8). For the NP-hardness we give a polynomial time reduction from the SBMOD( $S,k$ ) problem to the CRR( $S,R$ ) problem. Let SBMOD( $S,k$ ) be an instance with a squared matrix  $S$ . Convert it into an instance of the CRR( $S,R$ ) problem where  $R = (1)^{k \times k}$ , i.e.,  $R$  is the  $k \times k$  matrix with all entries equal to 1.

Forward direction: If SBMOD( $S,k$ ) has a solution, then CRR( $S,(1)^{k \times k}$ ) has a solution.

To see this, let  $E, P$  be a solution for SBMOD( $S,k$ ), i.e.  $S = E \cdot P$ . By Lemma 3.4 we can assume that all entries in the first row of  $E$  are equal to 1. The same applies to the structure of  $P$ , where all entries of the first column are equal to 1. Therefore  $P \cdot E = (1)^{k \times k}$ .

Backward direction: If CRR( $S,(1)^{k \times k}$ ) has a solution, then SBMOD( $S,k$ ) has a solution.

To see this, let CRR( $S,(1)^{k \times k}$ ) have a solution  $E, P$  with  $S = E \cdot P$  and  $(1)^{k \times k} = P \cdot E$ . The second equation of CRR( $S,(1)^{k \times k}$ ) ( $R = P \cdot E$ ) forces  $P$  to have  $k$  rows and  $E$  to have  $k$  columns. By the first equation of CRR( $S,(1)^{k \times k}$ ) ( $S = E \cdot P$ ) it follows that  $E, P$  are a solution to SBMOD( $S,k$ ). □

## 4 Declarative Formulations

Despite its NP-completeness, the CRR problem is relevant to chemistry. We therefore want to empirically investigate the CRR problem and examine for how large networks the CRR problem can be solved in a reasonable time. To this end, we formulate it as a Boolean satisfiability problem, allowing us to use established declarative approaches, such as Satisfiability- (SAT-) solvers, Satisfiability-Modulo-Theory- (SMT-) solvers and solvers for Integer Linear Programming (ILP). In this section, we describe these formulations.

### 4.1 Satisfiability Modulo Theories

SMT can be seen as a generalized approach to Boolean satisfiability problems, increasing expressiveness by using e.g. first order logic instead of propositional logic. Another approach is to ask for satisfiability with respect to some background theory. This theory can fix interpretations of predicate and function



symbols to be e.g. integers, reals, arithmetic, quantifiers or arrays [7]. Here, we use the core theory, which is defining the basic Boolean operators, and we have quantifiers available, which allows the following straight forward formulation of the  $CRR(S,R)$  problem:

$$\begin{aligned}
s_{ij} = 1 &\Leftrightarrow \exists a : (e_{ia} \wedge p_{aj}) &= \text{TRUE}, & \forall i, j = 1, \dots, n \\
s_{ij} = 0 &\Leftrightarrow \forall a : (\neg e_{ia} \vee \neg p_{aj}) &= \text{TRUE}, & \forall i, j = 1, \dots, n \\
r_{ab} = 1 &\Leftrightarrow \exists i : (e_{ib} \wedge p_{ai}) &= \text{TRUE}, & \forall a, b = 1, \dots, m \\
r_{ab} = 0 &\Leftrightarrow \forall i : (\neg e_{ib} \vee \neg p_{ai}) &= \text{TRUE}, & \forall a, b = 1, \dots, m
\end{aligned} \tag{9}$$

A formulation in propositional logic is also possible, but Eq. (9) can be expressed directly in the SMT-LIB language [4], which is the standard description language for input for SMT-solvers. We formalized  $S$ ,  $R$ ,  $E$  and  $P$  as uninterpreted functions, which take as arguments species and reactions, and give as result a Boolean value. Thus, in the SMT formulation we have 4 uninterpreted functions (one for each matrix), we declare two data types (species and reactions) and we have  $n + m$  variables of the mentioned data types.

## 4.2 Boolean Logic

SAT-solvers for Boolean satisfiability problems do not allow first order logic, thus Eq. (9) is translated into propositional order formulae. Furthermore, the standard input format is the DIMACS format, which comprises the Boolean formula in Conjunctive Normal Form (CNF). The Boolean formula derived from Eq. (9) is not in CNF and must therefore be converted. A plain conversion leads in worst case to a formula with exponentially many clauses, but methods like Tseitin encoding create an equisatisfiable formula in CNF, i.e., new variables are introduced and the new formula is satisfiable iff the original input formula is satisfiable [36, 37]. This new formula grows only linearly in terms of variables and clauses relative to the input formula. For this conversion, we use the tool *limboole* [1]. The number of variables for the original SAT formulation is in  $O(nm)$ , since only entries of  $E$  and  $P$  appear in the Boolean formula. By transforming the problem into CNF-form, new variables are introduced. An overview of the numbers of variables in the empirical study is shown in Table 1 and discussed in Sec. 5.1.3.

## 4.3 Integer Linear Programming

Our third approach to solve the  $CRR(S,R)$  problem is to use Integer Linear Programming (ILP). Here, the quadratic program of Boolean matrix multiplication has to be linearized over the domain  $\{0,1\}$ . The reformulation of the constraints of matrix  $S$  is shown in Eqns. (10-15), the reformulation for matrix  $R$  is carried out analogously. The reformulation follows standard techniques for linearizing quadratic programs to ILP. As described in Sec. 2.3, helper variables  $w_{ija}^S$  were additionally introduced to describe the existence of witness reactions  $a$  for entries  $s_{ij} = 1$  (resp. variables  $w_{abi}^R$  for witness species for entries  $r_{ab} = 1$ ). The objective function of the ILP is not of importance, since we only want to check for satisfiability. The number of variables for an ILP formulation is  $O(nm + n^2m + m^2n)$ , where the first term gives the number of entries in  $E$  and  $P$ , and the second and third term give the variables  $w_{ija}^S$  and  $w_{abi}^R$ . Empirical numbers of variables from the experiments are presented in Table 1 and discussed in Sec. 5.1.3.

$$\begin{aligned}
& \min && 0 \\
& \text{such that} && \\
\forall s_{ij} = 0 : & & e_{ia} + p_{aj} \leq 1 & \quad \forall a = 1, \dots, m & (10) \\
\forall s_{ij} = 1 : & & \sum_{a=1}^m e_{ia} \cdot p_{aj} \geq 1 & & (11) \\
& & \Leftrightarrow & & \\
& & \sum_{a=1}^m w_{ija}^S \geq 1 & & (12) \\
& & w_{ija}^S \leq e_{ia} & & (13) \\
& & w_{ija}^S \leq p_{aj} & & (14) \\
& & w_{ija}^S \geq e_{ia} + p_{aj} - 1 & \quad \forall a = 1, \dots, m & (15)
\end{aligned}$$

## 5 Empirical Section

In this section, we first apply different solvers on randomly created instances to get a better understanding of the hardness of the problem. We then compare instances of  $S$  and  $R$  from real-world reaction networks with random instances of comparable size.

### 5.1 Computational Limits with Random Test Cases

To examine the solvers' computational limits on the CRR( $S, R$ ) problem, we apply them on randomly generated instances of  $S$  and  $R$ , which we constructed in the following way.

#### 5.1.1 Test Data Set

The test set was created with two pairs of parameters: The pair  $(n, m)$  of sizes of  $S$  ( $n \times n$ ) and  $R$  ( $m \times m$ ), which were chosen as  $(n, m) \in \{(10, 10), (20, 10), (20, 20), (40, 20), (40, 40)\}$ .

The second pair of parameters is  $(p, q)$ , which defines the proportion  $p$  (resp.  $q$ ) of zeros out of all entries in  $S$  (resp.  $R$ ),  $p, q \in [0, 1]$ . For each pair of matrices  $S$  and  $R$ ,  $p$  and  $q$  were chosen uniformly at random in  $[0, 1]$ . According to the parameters  $(p, q)$ , the respective number of zero entries were determined, and their positions in the matrices  $S$  and  $R$  were chosen uniformly at random.

#### 5.1.2 Experiments

The experiments were run on Intel(R) Core(TM) i5 CPU 650 @ 3.20GHz machines. We used state-of-the-art solvers for all declarative approaches, namely the SMT-solver *Z3* [12], the SAT-solvers *MiniSAT* [14, 32] and *lingeling* [6], and the ILP-solver IBM ILOG *CPLEX* Optimization Studio 12.5.

We ran the different solvers each on 1000 instances of each of the 5 different matrix sizes. In case of a successful reconstruction (i.e., a pair  $E$  and  $P$  can be found from the given  $S$  and  $R$ ), an instance is called "satisfiable". It is called "unsatisfiable" if no reconstruction exists. In both cases the instances are called "solvable". If the solving time exceeded 3600 seconds, the process was aborted for that instance, which is then called "indetermined".

#### 5.1.3 Number of Variables

By converting a Boolean formula into CNF using, e.g., the Tseitin method [36, 37], the number of variables in CNF formulation grows only linearly. The number of variables in the original formula is  $O(nm)$ , where we observed approximately  $(n + m) \cdot 2nm$  variables in the CNF formula of the test

cases (Table 1). Striking is the low variance in the number of variables, which shows that the number of zeros and ones have only a small impact on the size of the CNF.

As mentioned earlier, the number of variables for an ILP formulation is  $O(nm + n^2m + m^2n)$ . Empirical numbers of variables from the experiments reflect the influence of the number of ones in the matrices  $S$  and  $R$  on the number of used variables. The minimal value in number of variables is seen for a low number of ones in the matrices, since a low number of witness variables  $w_{ijk}^S$  and  $w_{jik}^R$  are needed to be added. Vice versa, the maximal value is seen for a high number of ones in the matrices.

Table 1: Numbers of variables from the empirical studies, using test data as described in 5.1.3. As expected, the number of variables in CNF formulation grows only linearly in terms of  $n$  and  $m$ . The number of variables in the ILP formulation highly depends on the number of zeros in  $S$  and  $R$ , which determines the number of introduced witness variables.

Size	SAT			ILP		
	in CNF			‡variables		
	min	max	median	min	max	median
(10,10)	4159	4399	4349	310	2170	1205
(20,10)	12303	12799	12699	560	6340	3435
(20,20)	32885	33599	33399	1180	16440	8920
(40,20)	97424	99199	98799	2600	49240	25010
(40,40)	260033	262399	261599	3600	128840	67840

#### 5.1.4 Computational Limits on Random Instances

Fig. 5 shows the proportion of different outcomes among the 1000 given instances for each solver and each instance size: The number of satisfiable instances are colored green, and the unsatisfiable are colored gray. The number of indetermined instances, which could not be solved in the given time border of 3600 seconds, are marked in red. The SMT- and SAT-solvers behave similar w.r.t. to the number of solvable instances, whereas the ILP-solver CPLEX had more time-outs than the other two and hence appear to be the slowest of the methods. For the largest instance size  $(n, m) = (40, 40)$ , the ILP-solver timed out in around 60 % of all cases, instead of around 40 % time-outs for the SMT- and SAT-solvers.

To compare the solving methods in more detail and additionally to investigate on which type of the instances the solving methods succeed or fail, the solving time of the instances is investigated. For each solver and instance size, Fig. 6 shows for time  $t$  how many instances have a solving time below  $t$ . The ILP-solver has markedly fewer solved instances for each  $t$  than the SMT- and SAT-solvers. Additionally, the ILP-solver as well as the SAT-solver *lingeling* show a different solving behavior than the other two solvers, in terms of a delay in providing solutions. This different behavior occurs especially on the larger instances. That the SMT- and SAT-solver MiniSAT behave similarly could indicate that Z3 is using a propositional SAT-solver, as the input contains only Boolean variables.

The increase of indetermined instances correlated to the instance size (Fig. 5) and the increasing number of solvable instances over time (Fig. 6) brings up the question, which types of instances can be solved and which can not. In Fig. 7 we show whether an instance of a certain size with certain  $(p, q)$ -values was solvable (green for satisfiable, red for unsatisfiable) or was indetermined (blue). Especially from the case of  $(n, m) = (10, 10)$  it becomes apparent, that a phase transition from satisfiable to unsatisfiable instances takes place, when the value of  $p$  and  $q$  crosses a certain level. Additionally, another observation is striking: If the value of  $(p, q)$ -values of the matrices is in a certain range, the instances are hard to solve. The solvers timed out on instances with an amount of zeros in the vicinity of the phase transition, but did solve very dense and sparse instances. This observation shows that the hardest instances to solve appear close to the phase transition line from satisfiable to unsatisfiable instances. This intuitively makes good sense as it is harder for the solvers to find evidence of the

outcome. Note, that in the top right corner there exist satisfiable instances among the unsatisfiable as well (cmp. Fig. 8A). Especially, if an instance of  $S$  and  $R$  contains only zeros, then it is apparent from Eq. 9, that such an instance is an 2-SAT formula and additionally the  $\text{CRR}(S, R)$  problem is trivially satisfiable.

## 5.2 Real World Instances

In this section, we compare random instances with real world networks. Reaction networks are usually large and sparse, thus they would belong in the upper right corner of a figure like Fig. 7. To compare real world networks with random instances and to compare the solvers’ behavior on them, we selected test data in the following way.

### 5.2.1 Test Data Sets

We chose instances of reaction networks to be pathways of the yeast *Saccharomyces cerevisiae* (strain S288C), which are derived from the database *MetExplore* [10]. Table 2 shows this selection of small real world networks, where networks of size larger than  $(n, m) = (130, 130)$  had to be discarded, because the translating tool to CNF or the solvers themselves ran out of memory.

Note, that in case of the real networks, the hypergraph (defined by  $E$  and  $P$ ) is given and  $S$  and  $R$  are derived from that. Thus, the  $\text{CRR}(S, R)$  problem is satisfiable for all of these instances.

Table 2 also provides the  $(p, q)$ -values of the real networks, where  $p$  and  $q$  were always greater than 0.96. Additionally, we created three sets of each 10000 random instances of comparable size, which we chose to be  $(n, m) \in \{(40, 40), (100, 100), (120, 120)\}$ . To achieve as well a comparable sparsity to the real chemical networks, the values of  $p$  and  $q$  are uniformly at random chosen in  $[0.9, 1]$ . According to the  $p$ - and  $q$ -value, entries in the matrices  $S$  and  $R$  were chosen to be zero.

### 5.2.2 Comparison of Random Instances with Real-world Networks

As one can see in Table 2, all sparse instances tested, real instances as well as random instances, are solvable with the SAT-solvers within seconds, in contrast to hard instances of smaller sizes. The fact that no solver timed out on any instance indicates a high impact of the sparsity of the matrices  $S$  and  $R$  has on the solvers’ performances.

The solvability of these instances is depicted in Fig. 8, where only the the upper right corner (i.e., instances with  $(p, q)$ -values of  $(\geq 0.9, \geq 0.9)$ ) is shown. For comparison, the real networks are included in blue in Fig. 8C. The fact that just a small fraction of randomly created instances of same sparseness as our real world instances is satisfiably solvable, whereas all  $S$  and  $R$  instances derived from real networks of course are, indicates that another structural property than sparsity alone characterizes real networks, e.g. clustering or connectedness.

## 6 Conclusion

In this paper we addressed the problem of reconstructing a hypergraph from two given simple graphs. The problem, called the Compound-Reaction-Reconstruction ( $\text{CRR}(S, R)$ ) Problem, is motivated by methods in chemistry, and models the reconstruction of a reaction network from the chemical  $S$ -graph and  $R$ -graph. Since these simple graphs are objects of analysis in chemistry, but contain only partial information about the underlying chemical network, this problem is of significant interest. As our first contribution, we proved this problem to be NP-complete.

As our second contribution, we empirically investigated the solvability of the problem using standard declarative approaches. The results in Sec. 5.1 show that for random instances we quickly get to the computational limits with exact solving methods. It must be noted that the size of the largest random instances can still be regarded as quite small, since naturally occurring reaction networks normally contain a larger number of species and reactions (cmp. Table 2). However, the computational

Table 2: Some characteristic parameters of four metabolic pathways of *Saccharomyces cerevisiae*, and additionally its full reaction network are shown. A dash indicates an aborted run, due to memory issues. The networks are obtained from the MetExplore database [10]. Additionally, the parameters of random networks of different sizes are depicted. The numbers on variables and time in the random networks are the median over 10000 instances. The solver used is lingeling.

Pathway	Size	CNF var	time	(p,q)
4-Hydroxybenzoate	(110, 102)	4790939	29.69	(0.979, 0.964)
TCA	(117, 111)	5961032	33.77	(0.980, 0.961)
Sphingo Lipids	(125, 116)	7032499	43.35	(0.982, 0.967)
Chorismate	(148, 143)	-	-	(0.984, 0.972)
<i>Saccharomyces cerevisiae</i>	(441, 504)	-	-	(0.993, 0.989)
Random Networks	(40,40)	260951	0.9	( $\geq 0.9, \geq 0.9$ )
Random Networks	(100,100)	4030951	16.44	( $\geq 0.9, \geq 0.9$ )
Random Networks	(120,120)	6956569	28.52	( $\geq 0.9, \geq 0.9$ )

limits proved to be quite different for real world instances than for random instances. This should not be very surprising, since e.g. the degree distribution of nodes in real networks does not follow a uniform random distribution, but tend to have certain structural properties. There is an ongoing debate on suitable measures for similarity between random graph models and natural reaction networks. There is agreement on the modularity of the networks [19], but besides this, the different modeling approaches focus on different measures. Popular approaches to simulate natural reaction networks are Erdős-Rényi networks [15], small-world graphs [39], and scale-free structures [22] (which has also been subject of criticism [34]). We refer to [25] for an overview.

By comparing the solvability of random and real world instances, our results indicate that sparsity seems to be a key property of natural networks which allows them to be solved for larger instances, although our data also indicate that this property alone is not sufficient for an adequate characterization of natural networks.

A natural object of further research is the characterization of network properties, in order to be able to sample real-world-like networks. Properties to investigate include the above mentioned scale-freeness of the directed hypergraph and whether it is transferable to the graphs of  $S$  and  $R$ , as well as parameters like clustering coefficient, depth, diameter, and connectedness.

Additionally, future work could include validation of how robust our statements in 5.2 are, regarding other simulated classes of graphs and further real world instances of reaction networks.

## References

- [1] Limboole. <http://fmv.jku.at/limboole/index.html>, May 2013. Institute for Formal Models and Verification.
- [2] N. Alon, Z. Galil, O. Margalit, and M. Naor. Witnesses for boolean matrix multiplication and for shortest paths. In *Foundations of Computer Science, 1992. Proceedings., 33rd Annual Symposium on*, pages 417–426. IEEE, 1992.
- [3] J. Andersen, C. Flamm, D. Merkle, and P. Stadler. Maximizing output and recognizing autocatalysis in chemical reaction networks is NP-complete. *Journal of Systems Chemistry* 2012, 3(1), 2012.

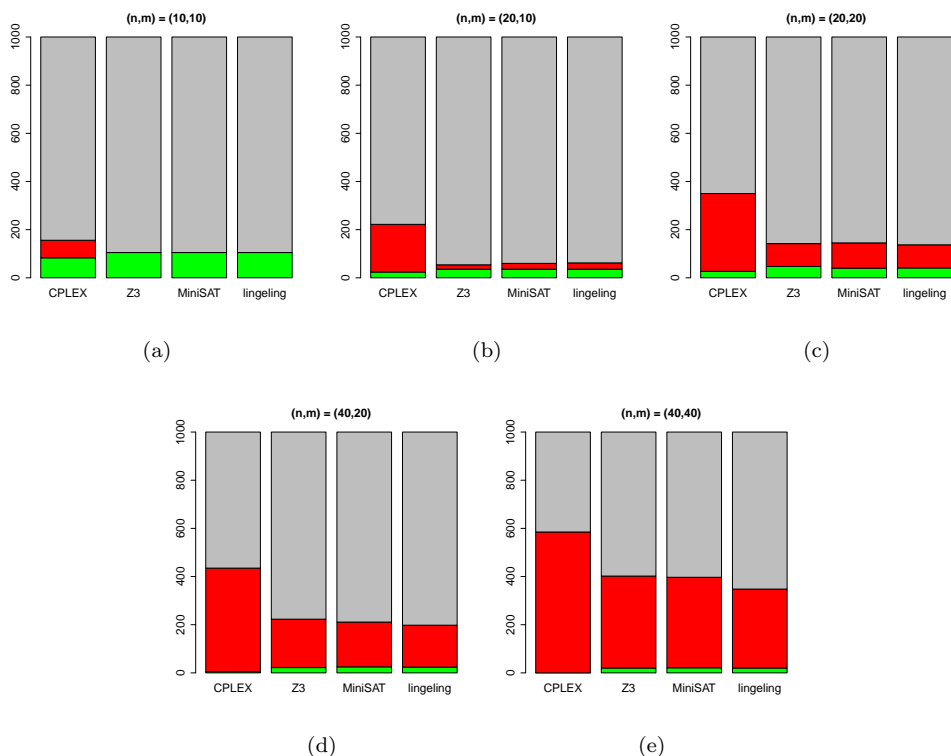


Figure 5: The bars show the portion of satisfiable (green), indetermined (red), and unsatisfiable (gray) instances among the 1000 given instances. They are arranged by the solving method.

- [4] C. Barrett, A. Stump, and C. Tinelli. The SMT-LIB Standard: Version 2.0. In A. Gupta and D. Kroening, editors, *Proceedings of the 8th International Workshop on Satisfiability Modulo Theories (Edinburgh, UK)*, 2010.
- [5] G. Benkő, F. Centler, P. Dittrich, C. Flamm, B. M. R. Stadler, and P. F. Stadler. A topological approach to chemical organizations. *Alife*, 15:71–88, 2009.
- [6] A. Biere. Lingeling, plingeling, picosat and precosat at sat race 2010. *FMV Report Series Technical Report*, 10(1), 2010.
- [7] A. Biere, M. Heule, H. van Maaren, and T. Walsh. Handbook of satisfiability, frontiers in artificial intelligence and applications, vol. 185, 2009.
- [8] E. Borenstein, M. Kupiec, M. W. Feldman, and E. Ruppín. Large-scale reconstruction and phylogenetic analysis of metabolic environments. *Proc. Natl. Acad. Sci. USA*, 105:14482–14487, 2008.
- [9] M. Capobianco and J. C. Molluzzo. Examples and counterexamples in graph theory. *Journal of Graph Theory*, 2(3):274–274, 1978.
- [10] L. Cottret, D. Wildridge, F. Vinson, M. P. Barrett, H. Charles, M.-F. Sagot, and F. Jourdan. Metexplore: a web server to link metabolomic experiments and genome-scale metabolic networks. *Nucleic acids research*, 38(suppl 2):W132–W137, 2010.
- [11] G. Craciun and M. Feinberg. Multiple equilibria in complex chemical reaction networks: II: The species-reaction graph. *SIAM J Appl Math*, 66:1321–1338, 2006.

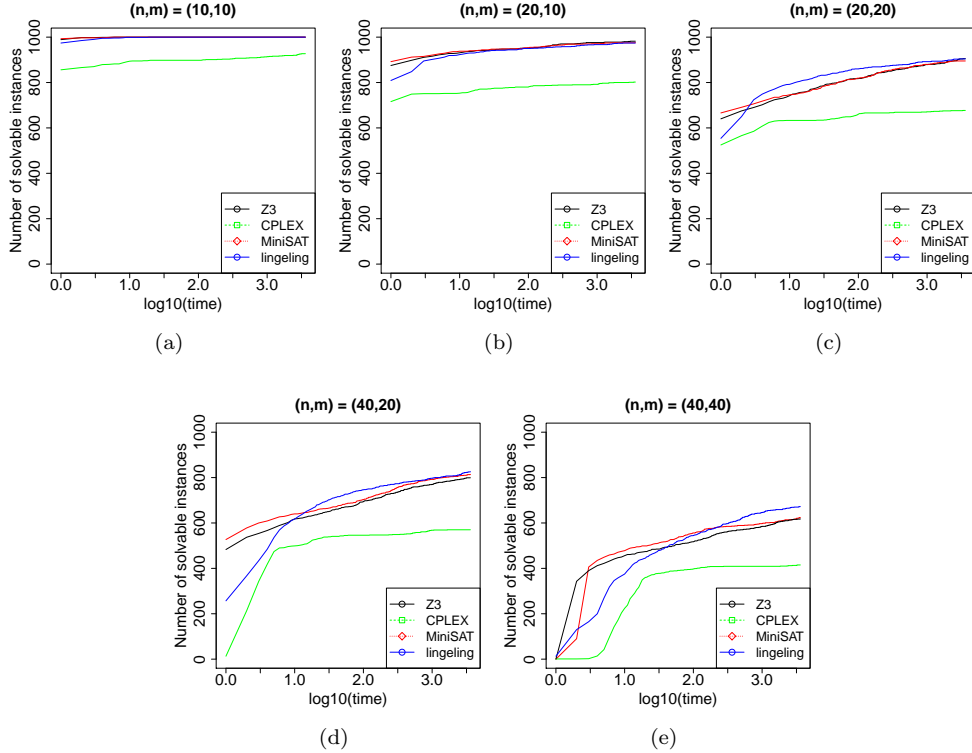


Figure 6: Plots for the various instance sizes of how many of the 1000 random instances each could be solved (as satisfiable or unsatisfiable) within in a certain amount of time. Depicted for each solver and over logarithmically scaled time axis. Time is measured in seconds.

- [12] L. De Moura and N. Bjørner. Z3: An efficient SMT solver. *Tools and Algorithms for the Construction and Analysis of Systems (TACAS'08)*, 4963:337–340, 2008.
- [13] M. Domijan and M. Kirkilionis. Graph theory and qualitative analysis of reaction networks. *Networks Heterog. Media*, 3:295–322, 2008.
- [14] N. Eén and N. Sörensson. An extensible sat-solver. In *Theory and Applications of Satisfiability Testing*, pages 333–336. Springer, 2004.
- [15] P. Erdős and A. Rényi. On the evolution of random graphs. *Magyar Tud. Akad. Mat. Kutató Int. Közl.*, 5:17–61, 1960.
- [16] M. Feinberg. Complex balancing in general kinetic systems. *Arch. Rational Mech. Anal.*, 49:187–194, 1972.
- [17] G. Gallo, G. Longo, S. Pallottino, and S. Nguyen. Directed hypergraphs and applications. *Discrete applied mathematics*, 42(2):177–201, 1993.
- [18] M. Garey and D. Johnson. *Computers and intractability: A guide to the theory of NP-completeness*, 1979.
- [19] J.-D. J. Han, N. Bertin, T. Hao, D. S. Goldberg, G. F. Berriz, L. V. Zhang, D. Dupuy, A. J. Walhout, M. E. Cusick, F. P. Roth, et al. Evidence for dynamically organized modularity in the yeast protein–protein interaction network. *Nature*, 430(6995):88–93, 2004.

- [20] F. J. M. Horn and R. Jackson. General mass action kinetics. *Arch. Rational Mech. Anal.*, 47:81–116, 1972.
- [21] A. N. Ivanova. Conditions for uniqueness of stationary state of kinetic systems, related, to structural scheme of reactions. *Kinet. Katal.*, 20:10191023, 1979.
- [22] H. Jeong, B. Tombor, R. Albert, Z. Oltvai, and A. Barabási. The large-scale organization of metabolic networks. *Nature*, 407(6804):651–654, 2000.
- [23] H.-M. Kaltenbach. A unified view on bipartite species-reaction and interaction graphs for chemical reaction networks. Technical Report 1210.0320, arXiv, 2012.
- [24] S. Klamt, U.-U. Haus, , and F. Theis. Hypergraphs and cellular networks. *PLoS Comput Biol*, 5:e1000385, 2009.
- [25] V. Lacroix, L. Cottret, P. Thébault, and M. Sagot. An introduction to metabolic networks and their structural analysis. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 5(4):594–617, 2008.
- [26] A. McNaught and A. Wilkinson. *IUPAC. Compendium of Chemical Terminology*, volume 2. Blackwell Scientific Publications, Oxford, 1997.
- [27] P. Miettinen, T. Mielikainen, A. Gionis, G. Das, and H. Mannila. The discrete basis problem. *Knowledge and Data Engineering, IEEE Transactions on*, 20(10):1348–1362, 2008.
- [28] M. Mincheva and M. R. Roussel. A graph-theoretic method for detecting potential Turing bifurcations. *J. Chem. Phys.*, 125:204102, 2006.
- [29] M. Mincheva and M. R. Roussel. Graph-theoretic methods for the analysis of chemical and biochemical networks. I. Multistability and oscillations in ordinary differential equation models. *J. Math. Biol.*, 55:61–68, 2007.
- [30] G. Shinar and M. Feinberg. Concordant chemical reaction networks and the Species-Reaction graph. *Math Biosci*, 241:1–23, 2013.
- [31] S. Skiena. Implementing discrete mathematics: Combinatorics and graph theory with mathematics. 1990.
- [32] N. Sörensson and N. Eén. Minisat v1. 13-a sat solver with conflict-clause minimization. *SAT*, 2005:53, 2005.
- [33] C. Soulé. Graphic requirements for multistationarity. *ComplexUs*, 1:123–133, 2003.
- [34] R. Tanaka. Scale-rich metabolic networks. *Physical review letters*, 94(16):168101, 2005.
- [35] R. Thomas. On the relation between the logical structure of systems and their ability to generate multiple steady states or sustained oscillations. In J. Della Dora, J. Demongeot, and B. Lacombe, editors, *Numerical Methods in the Study of Critical Phenomena*, volume 9, pages 180–193. Springer, 1981.
- [36] G. Tseitin. On the complexity of derivation in propositional calculus. *Slisenko, A.O. (ed.) Structures in Constructive Mathematics and Mathematical Logic, Part II, Seminars in Mathematics (translated from Russian), Steklov Mathematical Institute*, page 115125, 1968.
- [37] G. Tseitin. On the complexity of derivation in propositional calculus. *Automation of Reasoning: Classical Papers in Computational Logic*, pages 2:466–483, 1983.
- [38] A. Volpert and A. N. Ivanova. Mathematical models in chemical kinetics. In *Mathematical modeling*, pages 57–102. Nauka, Moscow, USSR, 1987. (Russian).



- [39] A. Wagner and D. A. Fell. The small world inside large metabolic networks. *Proc. R. Soc. Lond. B*, 268:1803–1810, 2001.
- [40] D. J. Watts and S. H. Strogatz. Collective dynamics of 'small-world' networks. *Nature*, 393:409–410, 1998.
- [41] A. V. Zeigarnik. On hypercycles and hypercircuits in hypergraphs. In P. Hansen, P. W. Fowler, and M. Zheng, editors, *Discrete Mathematical Chemistry*, volume 51 of *DIMACS series in discrete mathematics and theoretical computer science*, pages 377–383. American Mathematical Society, Providence, RI, 2000.

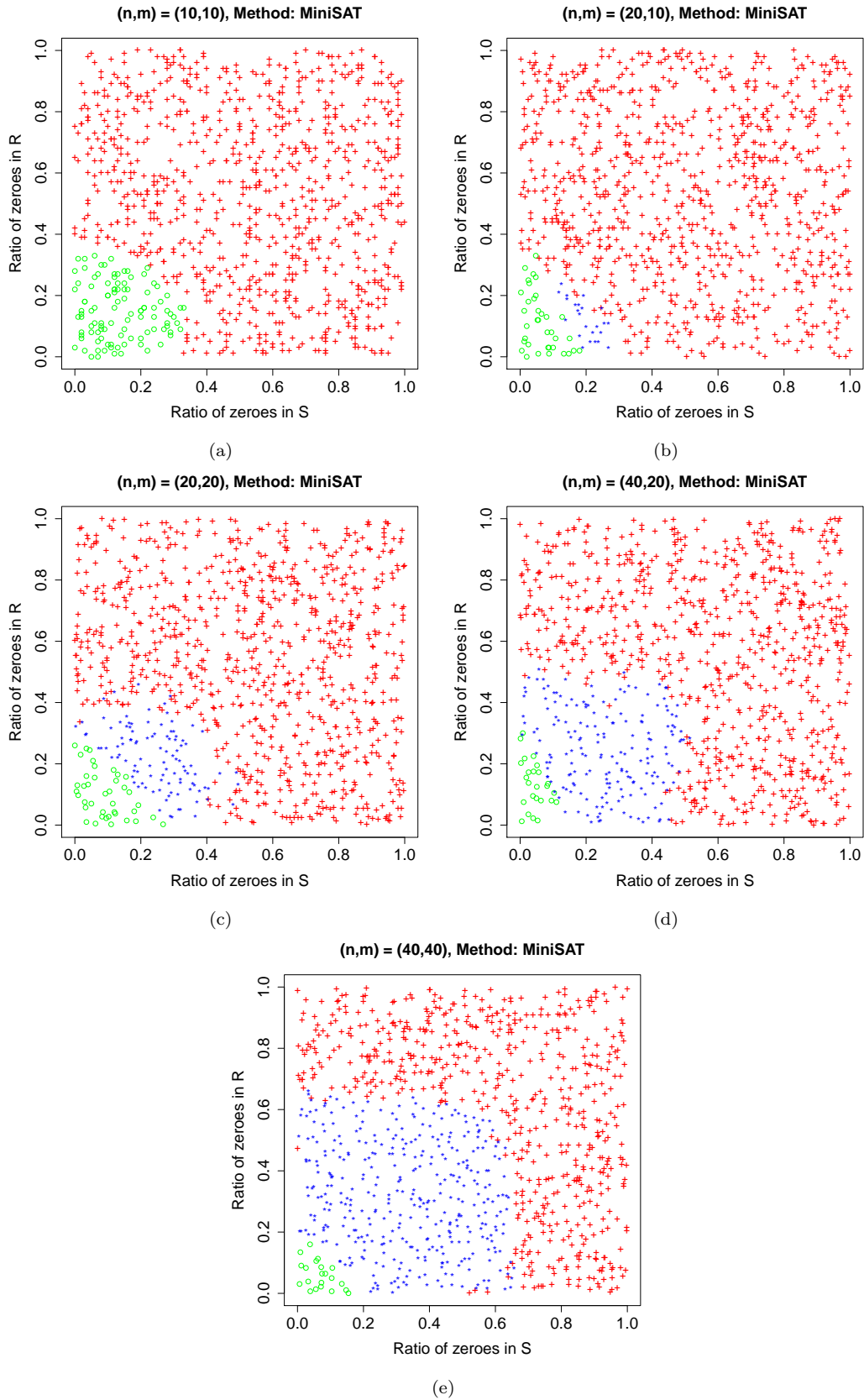
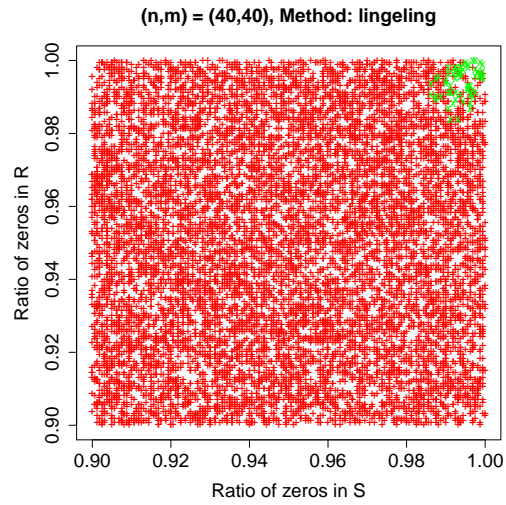
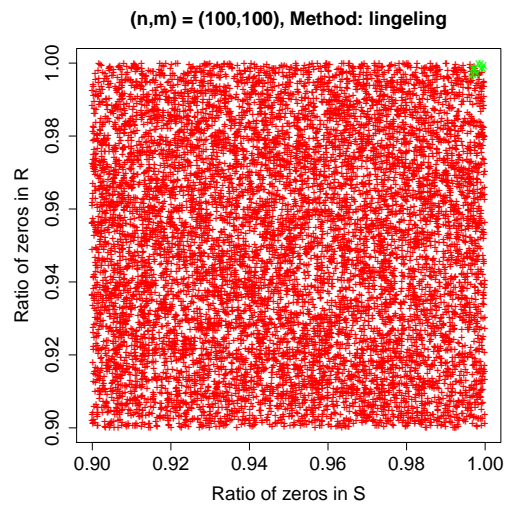


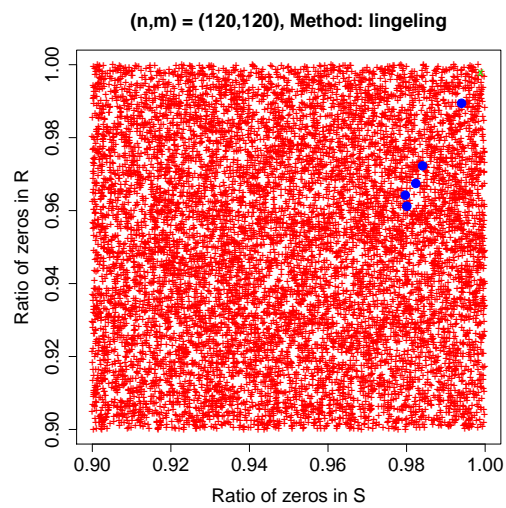
Figure 7: Each point marks the  $(p, q)$ -value of a random instance of a certain size. If the instance was satisfiable solvable, it is marked by a green circle, otherwise by a red cross. If the instance was not solvable in the given time of 3600 seconds, the instance is marked by a blue asterisk. A phase transition of easy and hard instance classes seems apparent.



(a)



(b)



(c)

Figure 8: The upper right corner of Fig. 7 for certain sizes. The red crosses mark unsatisfiable instances, the green crosses satisfiable instances. Note that there only appear solvable random instances for these  $(p, q)$ -values greater than  $(0.9, 0.9)$ , the solver lingeling did not time out on any instance. The big blue dots mark the  $(p, q)$ -values of the real world instances from Table 2. The grid-like structure of Fig. 8A follows from the instance's size and not from a lack of randomness.

## Appendix

In this appendix, we provide an example of a pair of graphs  $S$  and  $R$  for which there exists no hypergraph having  $S$  and  $R$  as its species and reaction graphs.

For the graphs  $S$  and  $R$  of Fig. 9, it is easily checked that the hypergraph in Fig. 10 has these as its species and reaction graphs. However, we now argue that if we change the  $S$  and  $R$  pair by leaving out the red arc in Fig. 9, such a hypergraph no longer exists.

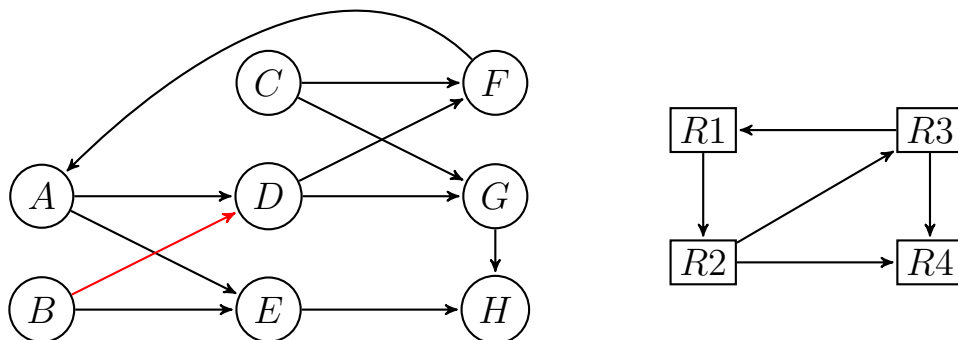


Figure 9: These graphs are the  $S$ -graph and  $R$ -graph of the hypergraph of Fig. 10. However, removing the red arc  $BD$  from  $S$  gives a pair of graphs which is not the  $S$ -graph and  $R$ -graph of any hypergraph.

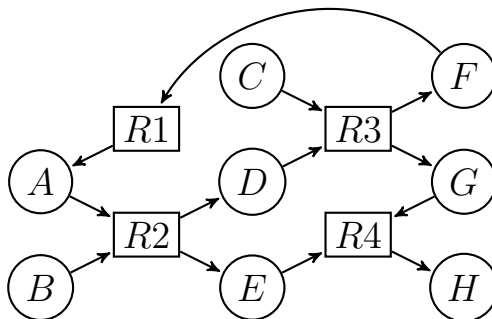


Figure 10: A hypergraph having the graphs of Fig. 9 as its  $S$ -graph and  $R$ -graph.

So, assume a hypergraph  $H$  exists having the graphs  $S$  and  $R$  of Fig. 9 with the red arc  $BD$  removed as its species and reaction graphs. Recall that in the species graph, two species  $v_i$  and  $v_j$  are adjacent iff there is a reaction  $a$  that has species  $v_i$  as reactant and species  $v_j$  as product. Consequently, a hyperarc  $a = (t(a), h(a))$  in  $H$  induces in the species graph  $S$  of  $H$  all edges of the complete directed bipartite graph between the vertex sets  $t(a)$  and  $h(a)$ , and every edge of  $S$  is induced in this way by at least one hyperarc.

Since no vertex in  $S$  has in- or out-degree greater than two, it follows that for all hyperarcs  $a$  we have  $|t(a)| \leq 2$  and  $|h(a)| \leq 2$ . In particular, no hyperarc can induce more than four edges of  $S$ . Also, since in  $S$  the out-degree of vertex  $F$  is one, it can only be in  $t(a)$  of hyperarcs  $a$  for which  $|h(a)|$  is one. Similarly, since the in-degree of vertex  $A$  is one, it can only be in  $h(a)$  of hyperarcs  $a$  for which  $|t(a)|$  is one. It follows that the arc  $FA$  can only be induced by the hyperarc  $(\{F\}, \{A\})$ , hence  $H$  must contain this.

Similarly, by the in- and out-degrees of vertices  $E$ ,  $G$ , and  $H$ , the arcs  $EH$  and  $GH$  must be

induced by one or more of the hyperarcs  $(\{E\}, \{H\})$ ,  $(\{G\}, \{H\})$  and  $(\{E, G\}, \{H\})$ . Using two or more hyperarcs for inducing  $EH$  and  $GH$  leaves at most one hyperarc to induce the remaining seven arcs of  $S$ , as  $R$  shows that there are exactly four hyperarcs in  $H$ . One hyperarc can induce at most four arcs, so this is impossible. Hence,  $H$  must contain the hyperarc  $(\{E, G\}, \{H\})$ . Since seven arcs of  $S$  remains to be induced, we must have  $2 = |t(a)| = |h(a)| = |t(b)| = |h(b)|$  for the last two hyperarcs  $a$  and  $b$ . But this contradicts the outdegree of vertex  $B$  being only one.