Preprocessing for Network Reconstruction: Feasibility Test and Handling Infeasibility

Annegret K. Wagler and Jan-Thierry Wegener*

Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes
Université Blaise Pascal (Clermont-Ferrand II)
BP 10125, 63173 Aubière Cedex, France
Annegret.Wagler@univ-bpclermont.fr wegener@isima.fr

Abstract. The context of this work is the reconstruction of Petri net models for biological systems from experimental data. Such methods aim at generating all network alternatives fitting the given data. For a successful reconstruction, the data need to satisfy two properties: reproducibility and monotonicity. In this paper, we focus on a necessary preprocessing step for a recent reconstruction approach. We test the data for reproducibility, provide a feasibility test to detect cases where the reconstruction from the given data may fail, and provide a strategy to cope with the infeasible cases. After having performed the preprocessing step, it is guaranteed that the (given or modified) data are appropriate as input for the main reconstruction algorithm.

1 Introduction

The aim of systems biology is to analyze and understand different phenomena as, e.g., responses of cells to environmental changes, host-pathogen interactions, or effects of gene defects. To gain the required insight into the underlying biological systems, experiments are performed and the resulting experimental data have to be interpreted in terms of models that reflect the observed phenomena. Depending on the biological aim and the type and quality of the available data, different types of mathematical models are used and corresponding methods for their reconstruction have been developed. We focus on Petri nets, a framework which turned out to coherently model both static interactions in terms of networks and dynamic processes in terms of state changes [1–4].

In fact, a (standard) network $\mathcal{P} = (P, T, A, w)$ reflects the involved system components by places $p \in P$ and their interactions by transitions $t \in T$, the arcs in $A \subset (P \times T) \cup (T \times P)$ link places and transitions, and the arc weights $w : A \to \mathbb{N}$ reflect stoichiometric coefficients of the corresponding reactions. Moreover, each place $p \in P$ can be marked with an integral number x_p of tokens defining a system state $\boldsymbol{x} \in \mathbb{Z}_+^{|P|}$. If a capacity $\text{cap}(p)$ is given for the places,

* This work was founded by the French National Research Agency, the European Commission (Feder funds) and the Région Auvergne in the Framework of the LabEx IMobS^3.

then $x_p \leq \text{cap}(p)$ follows and we obtain $\mathcal{X} := \{\boldsymbol{x} \in \mathbb{N}^{|P|} : x_p \leq \text{cap}(p)\}$ as set of potential states. A transition $t \in T$ is *enabled* in a state $\boldsymbol{x}$ if $x_p \geq w(p,t)$ for all p with $(p,t) \in A$ (and $x_p + w(t,p) \leq \text{cap}(p)$ for all $(t,p) \in A$), switching or firing t yields a successor state $\text{succ}(\boldsymbol{x}) = \boldsymbol{x}'$ with $x'_p = x_p - w(p,t)$ for all $(p,t) \in A$ and $x'_p = x_p + w(t,p)$ for all $(t,p) \in A$. Dynamic processes are represented by sequences of such state changes.

Petri net models can be reconstructed from experimental time-series data by means of exact, exclusively data-driven reconstruction approaches [5–10]. These approaches take as input a set P of places and discrete time-series data $\mathcal{X}'$ given by sequences $(\boldsymbol{x}^0; \boldsymbol{x}^1, \ldots, \boldsymbol{x}^m)$ of experimentally observed system states. The goal is to determine all Petri nets (P,T,A,w) that are able to reproduce the data, i.e., that perform for each $\boldsymbol{x}^j \in \mathcal{X}'$ the experimentally observed state change to $\boldsymbol{x}^{j+1} \in \mathcal{X}'$ in a simulation.

In general, there can be more than one transition enabled at a state. The decision which transition switches is typically taken randomly (and the dynamic behavior is analyzed in terms of reachability, starting from a certain initial state). To properly predict the dynamic behavior, (standard) Petri nets have to be equipped with additional activation rules to force the switching or firing of special transitions, and to prevent all others from switching.

This can be done by using priority relations and control-arcs and leads to the notion of $\mathcal{X}'$-deterministic Petri nets [11], which show a prescribed behavior on the experimentally observed subset $\mathcal{X}'$ of states: the reconstructed Petri nets do not only contain enough transitions to reach the experimentally observed successors $\boldsymbol{x}^{j+1}$ from $\boldsymbol{x}^j$, but exactly this transition will be selected among all enabled ones in $\boldsymbol{x}^j$ which is necessary to reach $\boldsymbol{x}^{j+1}$ (see Section 2.2 for details).

For a successful reconstruction, the data $\mathcal{X}'$ need to satisfy two properties: reproducibility (for each $\boldsymbol{x}^j \in \mathcal{X}'$ there is a unique observed successor state $\text{succ}_{\mathcal{X}'}(\boldsymbol{x}^j) = \boldsymbol{x}^{j+1} \in \mathcal{X}'$) and monotonicity (meaning that all essential responses are indeed reported in the experiments), see Section 2.1. Having reproducible data is clearly evident for a successful reconstruction; the necessity of monotone data is shown in [12].

In this paper, we focus on a necessary preprocessing step for the reconstruction approach described in [8]. We test the data for reproducibility, provide a feasibility test (based on previous works in [7]) to detect cases where the reconstruction from the given data may fail (see Section 3.1), and provide a strategy (based on previous works in [7,9]) to cope with infeasible cases (see Section 3.2). We close with some concluding remarks.

2 Reconstructing Petri Nets from Experimental Data

In this section we describe the input and the desired output of the reconstruction method from [8], whereas we refer the reader for details on the reconstruction approach itself to [8].

2.1 Input: Experimental Time-Series Data

First, a set of components P (later represented by the set of places) is chosen which is expected to be crucial for the studied phenomenon and which can be treated in terms of measurements[1].

To perform an experiment, the system is stimulated in a state $\boldsymbol{x}^0$ (by external stimuli like the change of nutrient concentrations or the exposition to some pathogens) to generate an initial state $\boldsymbol{x}^1 \in \mathcal{X}$. Then the system's response to the stimulation is observed and the resulting state changes are measured at certain time points. This yields a sequence $(\boldsymbol{x}^1, \dots, \boldsymbol{x}^k)$ of states $\boldsymbol{x}^i \in \mathcal{X}$ reflecting the time-dependent response of the system to the stimulation, denoted by

$$\mathcal{X}'(\boldsymbol{x}^1, \boldsymbol{x}^k) = (\boldsymbol{x}^0; \boldsymbol{x}^1, \dots, \boldsymbol{x}^k).$$

Note that we also provide the state $\boldsymbol{x}^0$ as the starting point for the stimulation, which will be needed later (see Section 3.2). Every sequence has an observed *terminal state* $\boldsymbol{x}^k \in \mathcal{X}$, without further changes of the system. The set of all terminal states in $\mathcal{X}'$ is denoted by $\mathcal{X}'_{term}$.

For technical reasons, we interpret a terminal state $\boldsymbol{x}^k \in \mathcal{X}'_{term}$ as a state which has itself as observed successor state, i.e., $\boldsymbol{x}^k = \mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x}^k)$.

Typically, several experiments starting from different initial states in a set $\mathcal{X}'_{ini} \subseteq \mathcal{X}$ are necessary to describe the whole phenomenon, and we obtain *experimental time-series data* of the form

$$\mathcal{X}' = \{\mathcal{X}'(\boldsymbol{x}^1, \boldsymbol{x}^k) : \boldsymbol{x}^1 \in \mathcal{X}'_{ini}, \boldsymbol{x}^k \in \mathcal{X}'_{term}\}.$$

We write $\boldsymbol{x} \in \mathcal{X}'$ to indicate that $\boldsymbol{x}$ is an element of a sequence $\mathcal{X}'(\boldsymbol{x}^1, \boldsymbol{x}^k) \in \mathcal{X}'$.

Example 1. As running example, we consider the *light-induced sporulation of Physarum polycephalum* [10]. The developmental decision of *P. polycephalum* plasmodia to enter the sporulation pathway is controlled by environmental factors like visible light [13]. A phytochrome-like photoreversible photoreceptor protein is involved in the control of sporulation Spo which occurs in two stages P_{FR} and P_R. If the dark-adapted form P_{FR} absorbs far-red light FR, the receptor is converted into its red-absorbing form P_R, which causes sporulation [14]. If P_R is exposed to red light R, it is photo-converted back to the initial stage P_{FR}, which can prevent sporulation in an early stage, but does not prevent sporulation in a later stage. Figure 1 gives an example of experimental time-series data reflecting this behavior, containing three time-series: $\mathcal{X}(\boldsymbol{x}^1, \boldsymbol{x}^4) = (\boldsymbol{x}^0; \boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{x}^3, \boldsymbol{x}^4)$, $\mathcal{X}(\boldsymbol{x}^5, \boldsymbol{x}^0) = (\boldsymbol{x}^2; \boldsymbol{x}^5, \boldsymbol{x}^0)$ and $\mathcal{X}(\boldsymbol{x}^6, \boldsymbol{x}^8) = (\boldsymbol{x}^3; \boldsymbol{x}^6, \boldsymbol{x}^7, \boldsymbol{x}^8)$.

In the best case, two consecutively measured states $\boldsymbol{x}^j, \boldsymbol{x}^{j+1} \in \mathcal{X}'$ are also consecutive system states, i.e., $\boldsymbol{x}^{j+1}$ can be obtained from $\boldsymbol{x}^j$ by switching a single transition. This is, however, in general not the case (and depends on the chosen time points to measure the states in $\mathcal{X}'$), but $\boldsymbol{x}^{j+1}$ is obtained from $\boldsymbol{x}^j$

[1] Possibly, it is known that a certain component plays a crucial role, but it is not possible to measure the values of that component experimentally.

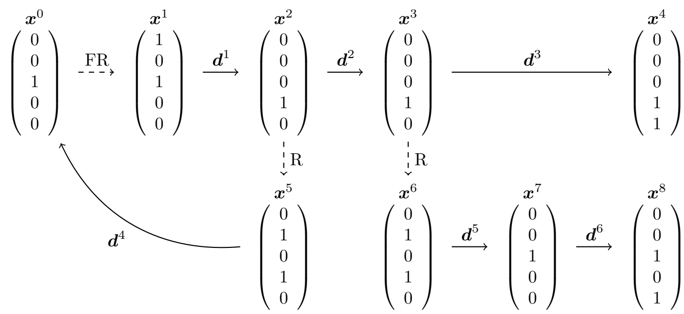

Fig. 1. This figure shows experimental time-series data $\mathcal{X}'$ for the light-induced sporulation of Physarum polycephalum. The experimental setting uses the set $P = \{FR, R, P_{fr}, P_r, Sp\}$ of studied components, observed states are represented by vectors of the form $\boldsymbol{x} = (x_{FR}, x_R, x_{P_{fr}}, x_{P_r}, x_{Sp})^T$ having 0/1-entries only. Dashed arrows represent stimulations to the system and solid arrows represent the observed responses.

by a switching sequence of some length, where the intermediate states are not reported in $\mathcal{X}'$.

For a successful reconstruction, the data $\mathcal{X}'$ need to satisfy two properties: reproducibility and monotonicity. The data $\mathcal{X}'$ are *reproducible* if for each $\boldsymbol{x}^j \in \mathcal{X}'$ there is a unique observed successor state $\mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x}^j) = \boldsymbol{x}^{j+1} \in \mathcal{X}'$. Moreover, the data $\mathcal{X}'$ are *monotone* if for each such pair $(\boldsymbol{x}^j, \boldsymbol{x}^{j+1}) \in \mathcal{X}'$, the possible intermediate states $\boldsymbol{x}^j = \boldsymbol{y}^1, \boldsymbol{y}^2, ..., \boldsymbol{y}^{m+1} = \boldsymbol{x}^{j+1}$ satisfy

$$y_p^1 \leq y_p^2 \leq \ldots \leq y_p^m \leq y_p^{m+1} \text{ for all } p \in P \text{ with } x_p^j \leq x_p^{j+1} \text{ and}$$
$$y_p^1 \geq y_p^2 \geq \ldots \geq y_p^m \geq y_p^{m+1} \text{ for all } p \in P \text{ with } x_p^j \geq x_p^{j+1}.$$

Whereas reproducibility is obviously necessary, it was shown in [12] that monotonicity has to be required or, equivalently, that all essential responses are indeed reported in the experiments.

Remark 1. When continuous data is discretized for the reconstruction approach, all local minima and maxima of the measured values have to be kept for each $p \in P$ to ensure monotonicity.

2.2 Output: $\mathcal{X}'$-Deterministic Extended Petri Nets

A standard Petri net $\mathcal{P} = (P, T, A, w)$ fits the given data $\mathcal{X}'$ when it is able to perform every observed state change from $\boldsymbol{x}^j \in \mathcal{X}'$ to $\mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x}^j) = \boldsymbol{x}^{j+1} \in \mathcal{X}'$. This can be interpreted as follows. With $\mathcal{P}$, an *incidence matrix* $M \in \mathbb{Z}^{|P| \times |T|}$ is associated, where each row corresponds to a place $p \in P$ of the network, and each column $M_{\cdot t}$ to the *update vector* $\boldsymbol{r}^t$ of a transition $t \in T$:

$$r_p^t = M_{pt} := \begin{cases} -w(p,t) & \text{if } (p,t) \in A, \\ +w(t,p) & \text{if } (t,p) \in A, \\ 0 & \text{otherwise.} \end{cases}$$

Reaching $\boldsymbol{x}^{j+1}$ from $\boldsymbol{x}^j$ by a switching sequence using the transitions from a subset $T' \subseteq T$ is equivalent to obtain the state vector $\boldsymbol{x}^{j+1}$ from $\boldsymbol{x}^j$ by adding the corresponding columns $M_{\cdot t}$ of M for all $t \in T'$:

$$\boldsymbol{x}^j + \sum_{t \in T'} M_{\cdot t} = \boldsymbol{x}^{j+1}. \tag{1}$$

Hence, T has to contain enough transitions to perform all experimentally observed switching sequences. The network $\mathcal{P} = (P, T, A, w)$ is *conformal* with $\mathcal{X}'$ if, for any two consecutive states $\boldsymbol{x}^j, \mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x}^j) = \boldsymbol{x}^{j+1} \in \mathcal{X}'$, the linear equation system $\boldsymbol{x}^{j+1} - \boldsymbol{x}^j = M\lambda$ has an integral solution $\lambda \in \mathbb{N}^{|T|}$ such that λ is the incidence vector of a sequence $(t^1, ..., t^m)$ of transition switches, i.e., there are intermediate states $\boldsymbol{x}^j = \boldsymbol{y}^1, \boldsymbol{y}^2, ..., \boldsymbol{y}^{m+1} = \boldsymbol{x}^{j+1}$ with $\boldsymbol{y}^l + M_{\cdot t^l} = \boldsymbol{y}^{l+1}$ for $1 \leq l \leq m$. Hereby, monotonicity avoids unnecessary solutions, since no homogeneous solutions of equation (1) have to be considered, see [10, 12].

To also force that the networks exhibit the experimentally observed dynamic behavior in a simulation, we equip standard networks with additional activation rules to further control the switching of enabled transitions, see [5, 6, 8, 11].

On the one hand, the concept of control-arcs can be used to represent catalytic or inhibitory dependencies. An *extended Petri net* $\mathcal{P} = (P, T, (A \cup A_R \cup A_I), w)$ is a Petri net which has, besides the (standard) arcs in A, two additional sets of so-called control-arcs: the set of read-arcs $A_R \subset P \times T$ and the set of inhibitor-arcs $A_I \subset P \times T$. We denote the set of all arcs by $\mathcal{A} = A \cup A_R \cup A_I$. Here, an enabled transition $t \in T$ coupled with a read-arc (resp. an inhibitor-arc) to a place $p \in P$ can switch in a state $\boldsymbol{x}$ only if a token (resp. no token) is present in p; we denote by $T_{\mathcal{A}}(\boldsymbol{x})$ the set of all such transitions.

On the other hand, in [9, 10, 15] the concept of priority relations among the transitions of a network was introduced in order to allow the modeling of deterministic systems. In Marwan et al. [9] it is proposed to model such priorities with the help of partial orders $\mathcal{O}$ on the transitions in order to reflect the rates of the corresponding reactions where the fastest reaction has highest priority and, thus, is taken. For each state $\boldsymbol{x}$, only a transition is allowed to switch if it is enabled and there is no other enabled transition with higher priority according to $\mathcal{O}$; we denote by $T_{\mathcal{A},\mathcal{O}}(\boldsymbol{x})$ the set of all such transitions. We call $(\mathcal{P}, \mathcal{O})$ a *Petri net with priorities* if $\mathcal{P} = (P, T, \mathcal{A}, w)$ is a (standard or extended) Petri net and $\mathcal{O}$ a priority relation on T.

For a deterministic behavior, $T_{\mathcal{A},\mathcal{O}}(\boldsymbol{x})$ must contain at most one element for each state $\boldsymbol{x}$ to enforce that $\boldsymbol{x}$ has a unique successor state $\mathrm{succ}_{\mathcal{X}}(\boldsymbol{x})$, see [15] for more details. For our purpose we consider a relaxed condition, namely that $T_{\mathcal{A},\mathcal{O}}(\boldsymbol{x})$ contains at most one element for each experimentally observed state $\boldsymbol{x} \in \mathcal{X}'$, but $T_{\mathcal{A},\mathcal{O}}(\boldsymbol{x})$ may contain several elements for non-observed states $\boldsymbol{x} \in \mathcal{X} \setminus \mathcal{X}'$. We call such Petri nets $\mathcal{X}'$*-deterministic* (see [11]).

The extended Petri net $\mathcal{P} = (P, T, \mathcal{A}, w)$ is *catalytically conformal* with $\mathcal{X}'$ if $t^l \in T_{\mathcal{A}}(\boldsymbol{y}^l)$ for each intermediate state $\boldsymbol{y}^l$ of any pair $(\boldsymbol{x}^j, \boldsymbol{x}^{j+1}) \in \mathcal{X}'$, and the extended Petri net with priorities $(\mathcal{P}, \mathcal{O})$ is $\mathcal{X}'$*-deterministic* if $\{t^l\} = T_{\mathcal{A},\mathcal{O}}(\boldsymbol{y}^l)$ holds for all $\boldsymbol{y}^l$.

The desired output of the reconstruction approach consists of the set of all $\mathcal{X}'$-deterministic extended Petri nets $(\mathcal{P}, \mathrm{cap}, \mathcal{O})$ (all having the same set P of places and the same capacities cap deduced from $\mathcal{X}'$ by $\mathrm{cap}(p) = \max\{x_p : \boldsymbol{x} \in \mathcal{X}'\}$).

Figure 2 shows an $\mathcal{X}'$-deterministic extended Petri net fitting the experimental data from Example 1.

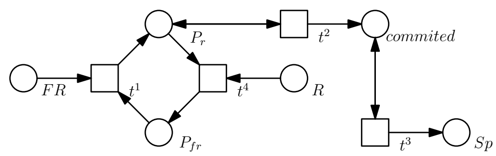

Fig. 2. This figure shows an $\mathcal{X}'$-deterministic extended Petri net fitting the experimental data from Example 1. The set of components is $P = \{FR, R, P_{fr}, P_r, Sp, \mathit{committed}\}$, where FR, R, P_{fr}, P_r and Sp have been measured directly in the experiment. The here added component *committed* cannot be measured directly, but only indirectly by the behavior of Physarum polycephalum observed in the experiment. The here shown network corresponds to solution (a) from Figure 4. In this $\mathcal{X}'$-deterministic extended Petri net there is a read-arc from P_r to t^2 and one from *committed* to t^3. Furthermore, we have the set of priorities $\mathcal{O} = \{t^2 < t^4, t^3 < t^4\}$. The control-arcs and priorities ensure $|T_{\mathcal{A},\mathcal{O}}(\boldsymbol{x})| = 1$ for every state $\boldsymbol{x} \in \mathcal{X}'$.

3 Feasibility Test and Handling Infeasibility

Before the reconstruction is started, a preprocessing step is necessary in order to verify or falsify whether the experimental time-series data $\mathcal{X}'$ is suitable for reconstructing $\mathcal{X}'$-deterministic extended Petri nets (see Section 3.1). If the test is successful, the reconstruction algorithm can be applied. For the case that the given data are not suitable for the reconstruction, we provide a method to handle the infeasible cases (see Section 3.2).

For that, we interpret (as in [7]) the experimental time-series data $\mathcal{X}'$ as a directed graph $\mathcal{D}(\mathcal{X}') = (V_{\mathcal{X}'}, A_D \cup A_S)$ having the measured states $\boldsymbol{x} \in \mathcal{X}'$ as nodes and two kinds of arcs:

- $A_D := \{(\boldsymbol{x}^j, \boldsymbol{x}^{j+1}) : \boldsymbol{x}^{j+1} = \mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x}^j)\}$ for the observed responses,
- $A_S := \{(\boldsymbol{x}^0, \boldsymbol{x}^1) : \mathcal{X}'(\boldsymbol{x}^1, \boldsymbol{x}^k) = (\boldsymbol{x}^0; \boldsymbol{x}^1, \ldots, \boldsymbol{x}^k)\}$ for the stimulations.

We call $\mathcal{D}(\mathcal{X}')$ the *experiment graph* of $\mathcal{X}'$. It can be interpreted as a minor of the reachability graph, where observed responses may correspond to directed paths with intermediate states.

Our main objective is to test the given experimental time-series data $\mathcal{X}'$ for reproducibility, i.e., whether each state $\boldsymbol{x} \in \mathcal{X}'$ has a unique successor state $\mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x}) \in \mathcal{X}'$. We provide a feasibility test to ensure this property (based on previous tests for standard Petri nets [7] and extended Petri nets [5], see

Section 3.1). If this test fails, we have a state $\boldsymbol{x} \in \mathcal{X}'$ with at least two successors in $\mathcal{X}'$, and it is not possible to reconstruct an $\mathcal{X}'$-deterministic extended Petri net from $\mathcal{X}'$ in its current form. As proposed in [7, 9, 10], this situation can be resolved by adding further components[2] to P with the goal to split any state $\boldsymbol{x} \in \mathcal{X}'$ with two successors into different states each having a unique successor. We present in Section 3.2 an approach for this step (based on previous works for standard Petri nets [7, 9]).

3.1 $\mathcal{X}'$-Determinism Conflicts and Feasibility Test

Definition 1. *Let $\mathcal{X}'$ be experimental time-series data. We say that two time-series $\mathcal{X}_i = \mathcal{X}'(\boldsymbol{x}^{i_0}, \boldsymbol{x}^{i_k})$ and $\mathcal{X}_\ell = \mathcal{X}'(\boldsymbol{x}^{\ell_0}, \boldsymbol{x}^{\ell_m})$ are in* $\mathcal{X}'$-determinism conflict, *when there exists a state $\boldsymbol{x} \in \mathcal{X}'$ with* $\mathrm{succ}_{\mathcal{X}_i}(\boldsymbol{x}) \neq \mathrm{succ}_{\mathcal{X}_\ell}(\boldsymbol{x})$ *and call $\boldsymbol{x}$ the corresponding* $\mathcal{X}'$-determinism conflict state. *We have*

- *a* strong *$\mathcal{X}'$-determinism conflict if $\boldsymbol{x}^{i_k} \neq \boldsymbol{x}^{\ell_m}$ or $\mathcal{X}_i = \mathcal{X}_\ell$;*
- *a* weak *$\mathcal{X}'$-determinism conflict if $\boldsymbol{x}^{i_k} = \boldsymbol{x}^{\ell_m}$ and $\mathcal{X}_i \neq \mathcal{X}_\ell$.*

The definition of strong $\mathcal{X}'$-determinism conflicts includes the case discussed in [5, 7] that there must not exist a terminal state $\boldsymbol{x}^j \in \mathcal{X}'_{term}$ that occurs as intermediate state in an experiment. Furthermore, it includes the case that a state $\boldsymbol{x}^j \in \mathcal{X}' \setminus \mathcal{X}'_{term}$ has itself as successor, i.e., $\mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x}^j) = \boldsymbol{x}^j$, which would result in $\boldsymbol{d}^j = 0$ (see Example 2).

Example 2. In the experimental time-series data $\mathcal{X}'$ shown in Figure 1 we have no weak but two strong $\mathcal{X}'$-determinism conflicts:

- in the sequence $\mathcal{X}'(\boldsymbol{x}^1, \boldsymbol{x}^4)$ the states $\boldsymbol{x}^2$ and $\boldsymbol{x}^3$ are equal but have different successor states,
- the sequences $\mathcal{X}'(\boldsymbol{x}^5, \boldsymbol{x}^0)$ and $\mathcal{X}'(\boldsymbol{x}^6, \boldsymbol{x}^8)$ have equal initial state $\boldsymbol{x}^5 = \boldsymbol{x}^6$, but different terminal states. Besides the initial states, the states $\boldsymbol{x}^0$ and $\boldsymbol{x}^7$ are $\mathcal{X}'$-determinism conflict states.

Obviously, every $\mathcal{X}'$-determinism conflict violates the condition of the data being reproducible, and the reconstruction of $\mathcal{X}'$-deterministic extended Petri nets from $\mathcal{X}'$ is not possible. However, the converse is true:

Lemma 1. *Let $\mathcal{X}'$ be experimental time-series data. If every state $\boldsymbol{x} \in \mathcal{X}'$ has a unique successor state* $\mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x}) \in \mathcal{X}'$ *then there exists an $\mathcal{X}'$-deterministic extended Petri net.*

Sketch of the proof. The pre-condition that every state has a unique successor in $\mathcal{X}'$ includes the cases that no non-terminal state has itself as successor and that no terminal state is an intermediate state of any experiment. This guarantees

[2] Since P is only a projection from the real world, it is possible that some components of the system, crucial for the studied phenomenon, were not taken into account or could not be experimentally measured.

the existence of a standard network $\mathcal{P} = (P, T, \cdot, \cdot)$ being conformal with $\mathcal{X}'$. Since every state has a unique successor state it follows for all states $\boldsymbol{x}^j, \boldsymbol{x}^l \in \mathcal{X}'$ with $\text{succ}(\boldsymbol{x}^j) \neq \text{succ}(\boldsymbol{x}^l)$ that there exists a non-empty subset $P' \subseteq P$ so that $\boldsymbol{x}_p^j \neq \boldsymbol{x}_p^l$ holds for every $p \in P'$. Therefore, $\mathcal{P}$ can be made $\mathcal{X}'$-deterministic by adding appropriate control-arcs (p, t), where $p \in P'$ and $t \in T$, in a way that exactly the transition is enabled which was observed in the experiments. □

Two time-series $\mathcal{X}'(\boldsymbol{x}^{i_0}, \boldsymbol{x}^{i_k})$ and $\mathcal{X}'(\boldsymbol{x}^{\ell_0}, \boldsymbol{x}^{\ell_m})$ with $\boldsymbol{x}^{i_k} = \boldsymbol{x}^{\ell_m}$ may be in weak $\mathcal{X}'$-determinism conflict, due to differently chosen time points of the measurements. We test the data for such a situation and try to resolve the conflict by linearizing these sequences, respecting monotonicity.

A *linear order* $\mathcal{L}$ (or *total order*) on a set S is a partial order where additionally $(a \leq b) \in \mathcal{L}$ or $(b \leq a) \in \mathcal{L}$ holds for all $a, b \in S$. In this case, we say that the set S is *totally ordered* (w.r.t. $\mathcal{L}$). A totally ordered subset $U \subseteq S$ of a partially ordered set S is called a *chain* of S.

On a time-series $\mathcal{X}'(\boldsymbol{x}^1, \boldsymbol{x}^k) = (\boldsymbol{x}^0; \boldsymbol{x}^1, \ldots, \boldsymbol{x}^k)$, a linear order is induced by the successor relation: $\boldsymbol{x}^j \leq \boldsymbol{x}^{j+1}$ iff $\boldsymbol{x}^{j+1} = \text{succ}_{\mathcal{X}'(\boldsymbol{x}^1, \boldsymbol{x}^k)}(\boldsymbol{x}^j)$, hence $\mathcal{X}'$ can be considered as a partially ordered set (ordered by the successor relation), where each time-series $\mathcal{X}'(\boldsymbol{x}^1, \boldsymbol{x}^k)$ is a chain of $\mathcal{X}'$. Let $\text{succ}_{\mathcal{X}'}(\boldsymbol{x}^j) = \boldsymbol{x}^{j+1}$ and

$$\text{Box}(\boldsymbol{x}^j, \boldsymbol{x}^{j+1}) := \left\{ \boldsymbol{y} \in \mathcal{X} : \begin{array}{ll} x_p^j \leq y_p \leq x_p^{j+1} & \text{if } x_p^j \leq x_p^{j+1} \\ x_p^j \geq y_p \geq x_p^{j+1} & \text{if } x_p^j \geq x_p^{j+1} \end{array} \right\}.$$

Note that due to monotonicity, all intermediate states $\boldsymbol{y}$ of any refined sequence from $\boldsymbol{x}^j$ to $\boldsymbol{x}^{j+1}$ lie in $\text{Box}(\boldsymbol{x}^j, \boldsymbol{x}^{j+1})$. Consequently, if two time-series $\mathcal{X}_i = \mathcal{X}'(\boldsymbol{x}^{i_0}, \boldsymbol{x}^{i_k})$ and $\mathcal{X}_\ell = \mathcal{X}'(\boldsymbol{x}^{\ell_0}, \boldsymbol{x}^{\ell_m})$ with $\boldsymbol{x}^{i_k} = \boldsymbol{x}^{\ell_m}$ are in weak $\mathcal{X}'$-determinism conflict, and $\boldsymbol{x}$ is a determinism conflict state then we have to test whether

(i) $\text{succ}_{\mathcal{X}_i}(\boldsymbol{x}) \in \text{Box}(\boldsymbol{x}, \text{succ}_{\mathcal{X}_\ell}(\boldsymbol{x}))$ or
(ii) $\text{succ}_{\mathcal{X}_\ell}(\boldsymbol{x}) \in \text{Box}(\boldsymbol{x}, \text{succ}_{\mathcal{X}_i}(\boldsymbol{x}))$,

see Figure 3 for an illustration. If one of the two conditions holds, we conclude $\text{succ}_{\mathcal{X}_i}(\boldsymbol{x}) < \text{succ}_{\mathcal{X}_\ell}(\boldsymbol{x})$ (resp. $\text{succ}_{\mathcal{X}_\ell}(\boldsymbol{x}) < \text{succ}_{\mathcal{X}_i}(\boldsymbol{x})$); otherwise we cannot find a $\mathcal{X}'$-deterministic linear order. Therefore, $\boldsymbol{x}$ is no longer a $\mathcal{X}'$-determinism conflict state, but a new $\mathcal{X}'$-determinism conflict state $\boldsymbol{x}'$ is detected since either

(i) $\boldsymbol{x}' = \text{succ}_{\mathcal{X}_i}(\boldsymbol{x})$ has two successor states: $\text{succ}_{\mathcal{X}_i}(\text{succ}_{\mathcal{X}_i}(\boldsymbol{x}))$, $\text{succ}_{\mathcal{X}_\ell}(\boldsymbol{x})$ or
(ii) $\boldsymbol{x}' = \text{succ}_{\mathcal{X}_\ell}(\boldsymbol{x})$ has two successor states: $\text{succ}_{\mathcal{X}_i}(\boldsymbol{x})$ and $\text{succ}_{\mathcal{X}_\ell}(\text{succ}_{\mathcal{X}_\ell}(\boldsymbol{x}))$.

Hence, the procedure has to be repeated for $\boldsymbol{x}'$ until $\text{succ}_{\mathcal{X}_i}(\boldsymbol{x}') = \text{succ}_{\mathcal{X}_\ell}(\boldsymbol{x}')$ holds or the test fails (see Algorithm 1). This works since in case of a weak $\mathcal{X}'$-determinism conflict at least the terminal states $\boldsymbol{x}^{i_k}$ and $\boldsymbol{x}^{\ell_m}$ are equal.

Whenever the test described above is successful for $\boldsymbol{x}$ and all subsequent $\mathcal{X}'$-determinism conflict states $\boldsymbol{x}'$, we say that it is *resolvable*, otherwise we say it is an *unresolvable* weak $\mathcal{X}'$-determinism conflict. We further obtain:

Theorem 1. *Let $\mathcal{X}'$ be experimental time-series data. There exists an $\mathcal{X}'$-deterministic extended Petri net if and only if there are neither strong $\mathcal{X}'$-determinism conflicts nor unresolvable weak $\mathcal{X}'$-determinism conflicts.*

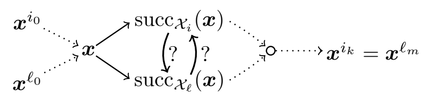

Fig. 3. This figure shows a weak $\mathcal{X}'$-determinism conflict. To resolve this conflict we can test if the two different successor states (resulting from two different experiments) of the $\mathcal{X}'$-determinism conflict state $\boldsymbol{x}$ can be ordered in such a way that the monotonicity constraint is not violated. In other words, we test if one of these successor states is an unmeasured intermediate state of $\boldsymbol{x}$ and the other successor state.

Sketch of the proof. If neither strong $\mathcal{X}'$-determinism conflicts nor unresolvable weak $\mathcal{X}'$-determinism conflicts exists, the statement follows from the procedure described above and from Lemma 1.

Conversely, suppose that an unresolvable weak (or strong) $\mathcal{X}'$-determinism conflict state exists. In the case that $\boldsymbol{x} = \mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x})$ holds for at least one strong $\mathcal{X}'$-determinism conflict state $\boldsymbol{x}$, then there does not exist any standard network being conformal with $\mathcal{X}'$. Otherwise, there exist conformal standard networks, but none of them can be made $\mathcal{X}'$-deterministic.

Let $\boldsymbol{x}$ be an unresolvable weak (or strong) $\mathcal{X}'$-determinism conflict state for two time-series $\mathcal{X}'_i = \mathcal{X}'(\boldsymbol{x}^{i_0}, \boldsymbol{x}^{i_k})$ and $\mathcal{X}'_\ell = \mathcal{X}'(\boldsymbol{x}^{\ell_0}, \boldsymbol{x}^{\ell_m})$. First note that $\boldsymbol{x}$ remains an unresolvable weak (or strong) $\mathcal{X}'$-determinism conflict state for every refined sequence (respecting the monotonicity constraint) of $\mathcal{X}'_i$ and $\mathcal{X}'_\ell$. Thus, w.l.o.g. we can assume that $\mathrm{succ}_{\mathcal{X}'_i}(\boldsymbol{x}) \neq \mathrm{succ}_{\mathcal{X}'_\ell}(\boldsymbol{x})$ and denote the respective transitions by t^i and t^ℓ. Since both transitions t^i and t^ℓ are (and need to stay) enabled at $\boldsymbol{x}$, there is no way to add priorities and/or control-arcs to force the network to deterministically show the observed behavior of $\mathcal{X}'_i$ and of $\mathcal{X}'_\ell$ simultaneously. □

3.2 Handling Infeasibility

Due to Theorem 1, it is impossible to reconstruct $\mathcal{X}'$-deterministic extended Petri nets from experimental time-series data $\mathcal{X}'$ containing a strong $\mathcal{X}'$-determinism conflict or an unresolvable weak $\mathcal{X}'$-determinism conflict. In this section we show how these conflicts can be resolved by using additional components.

For that we extend, as proposed in [7,9], all the n-dimensional state vectors $\boldsymbol{x} \in \mathcal{X}'$ to suitable $(n+a)$-dimensional vectors

$$\overline{\boldsymbol{x}}^j := \begin{pmatrix} \boldsymbol{x}^j \\ \boldsymbol{z}^j \end{pmatrix} \in \overline{\mathcal{X}'} = \left\{ \overline{\boldsymbol{x}} = \begin{pmatrix} \boldsymbol{x} \\ \boldsymbol{z} \end{pmatrix} \in \mathbb{Z}^{n+a} : \ \mathbf{0} \leq \boldsymbol{z} \leq \mathbf{1}, \ \boldsymbol{x} \in \mathcal{X}' \right\}.$$

The studied extensions $\overline{\boldsymbol{x}}^j \in \mathbb{N}^{n+a}$ of the states $\boldsymbol{x}^j \in \mathcal{X}'$ correspond to suitable *labelings* of the experiment graph $\mathcal{D}(\mathcal{X}')$:

Algorithm 1 Resolving weak $\mathcal{X}'$-determinism conflicts by linearization

Input: time-series $\mathcal{X}'(\boldsymbol{x}^{i_0}, \boldsymbol{x}^{i_k})$, $\mathcal{X}'(\boldsymbol{x}^{\ell_0}, \boldsymbol{x}^{\ell_m})$ in weak $\mathcal{X}'$-determinism conflict
Output: adjusted time-series if resolvable weak $\mathcal{X}'$-determinism conflict or *false* otherwise

for all conflict states $\boldsymbol{x}$ **do**
  $\boldsymbol{x}^i \leftarrow \mathrm{succ}_{\mathcal{X}'(\boldsymbol{x}^{i_0}, \boldsymbol{x}^{i_k})}(\boldsymbol{x})$, $\boldsymbol{x}^\ell \leftarrow \mathrm{succ}_{\mathcal{X}'(\boldsymbol{x}^{\ell_0}, \boldsymbol{x}^{\ell_m})}(\boldsymbol{x})$
  $\mathcal{L} \leftarrow \emptyset$ ▷ stores the linear order
  while $\boldsymbol{x}^i \neq \boldsymbol{x}^\ell$ **do**
    if $\boldsymbol{x}^i \in \mathrm{Box}(\boldsymbol{x}, \boldsymbol{x}^\ell)$ **then**
      $\mathcal{L} \leftarrow \mathcal{L} \cup \{\boldsymbol{x}^i < \boldsymbol{x}^\ell\}$
      $\boldsymbol{x} \leftarrow \boldsymbol{x}^i$
      $\boldsymbol{x}^i \leftarrow \mathrm{succ}_{\mathcal{X}'(\boldsymbol{x}^{i_0}, \boldsymbol{x}^{i_k})}(\boldsymbol{x}^i)$
    else if $\boldsymbol{x}^l \in \mathrm{Box}(\boldsymbol{x}, \boldsymbol{x}^i)$ **then**
      $\mathcal{L} \leftarrow \mathcal{L} \cup \{\boldsymbol{x}^\ell < \boldsymbol{x}^i\}$
      $\boldsymbol{x} \leftarrow \boldsymbol{x}^\ell$
      $\boldsymbol{x}^\ell \leftarrow \mathrm{succ}_{\mathcal{X}'(\boldsymbol{x}^{\ell_0}, \boldsymbol{x}^{\ell_m})}(\boldsymbol{x}^\ell)$
    else
      return *false*
return adjusted time-series according to $\mathcal{L}$

- if $a = 1$, to $(0,1)$-labelings, where label i is assigned to node $\boldsymbol{x}^j$ if $\overline{x}^j_{n+1} = \boldsymbol{z}^j = i$ is selected for $i \in \{0,1\}$;
- if $a = 2$, to $(0,1,2,3)$-labelings, where the labels are assigned to the four different states $(0,0)^T, (1,0)^T, (0,1)^T$ and $(1,1)^T$;
- if $a \geq 3$ we use similar encodings for all 2^a different 0/1-vectors.

By using appropriate additional components, states that appear equal in experimental time-series data $\mathcal{X}'$ become different in $\overline{\mathcal{X}'}$ (see Figure 4 for an illustration). It is already stressed in [7] that not every labeling for the experiment graph $\mathcal{D}(\mathcal{X}')$ is reasonable, as a state $\overline{\boldsymbol{x}}^k \in \overline{\mathcal{X}'}$ with $\boldsymbol{x}^k \in \mathcal{X}'_{term}$ might have a successor state, a state $\overline{\boldsymbol{x}}^j$ might have multiple successor states, or some stimulation changes more than the target input component(s). To obtain suitable labelings for $\mathcal{X}'$-deterministic extended Petri nets, we adjust Definition 15 from [7]:

Definition 2. *A labeling L of $\mathcal{X}'$ is* valid *if it satisfies the following conditions:*

(i) every state $\overline{\boldsymbol{x}}$ has a unique successor state $\mathrm{succ}(\overline{\boldsymbol{x}})$*,*
(ii) any stimulation preserves the values on the additional component(s),
(iii) for every $\boldsymbol{d} = \mathrm{succ}(\boldsymbol{x}) - \boldsymbol{x}$ and $\boldsymbol{d}' = \mathrm{succ}(\boldsymbol{x}') - \boldsymbol{x}'$ with $\boldsymbol{d} = \boldsymbol{d}'$ follows $\overline{\boldsymbol{d}} = \overline{\boldsymbol{d}}'$.

From Condition (i) we can conclude that we have $\boldsymbol{x} = \mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x})$ if and only if $\boldsymbol{x} \in \mathcal{X}'_{term}$. Condition (ii) ensures that a stimulation does not change more than the target input component(s), and finally, Condition (iii) ensures a minimal number of label switches, while keeping the data as close as possible to the original measurements. Furthermore, due to symmetry reasons, we can choose a label for one state, e.g., a conflict state.

Example 3. Besides symmetric solutions, there are two possible valid labelings with $a = 1$ for the experimental time-series data from Figure 1. These two solutions are shown in Figure 4. The solutions are obtained by applying the conditions of Definition 2 as follows. We start by selecting an $\mathcal{X}'$-determinism conflict state, here $\boldsymbol{x}^2$, and choose its label as $\boldsymbol{x}_z^2 = 0$. Due to Condition (ii), $\boldsymbol{x}_z^5 = 0$ follows. Condition (i) implies that $\boldsymbol{x}^3$ (resp. $\boldsymbol{x}^6$) must be different from $\boldsymbol{x}^2$ (resp. $\boldsymbol{x}^5$). Therefore, $\boldsymbol{x}_z^3 = 1$ and $\boldsymbol{x}_z^6 = 1$ follows. Since we have $\boldsymbol{d}^4 = \boldsymbol{d}^5$, Condition (iii) implies that the only valid labels for $\boldsymbol{x}^0$ and $\boldsymbol{x}^7$ are 0 and 1, respectively. Condition (ii) shows $\boldsymbol{x}_z^1 = 0$. Finally, we can choose a label for $\boldsymbol{x}^4$ and $\boldsymbol{x}^8$, respectively. However, since $\boldsymbol{d}^3 = \boldsymbol{d}^6$, if follows from (iii) that both labels must be equal.

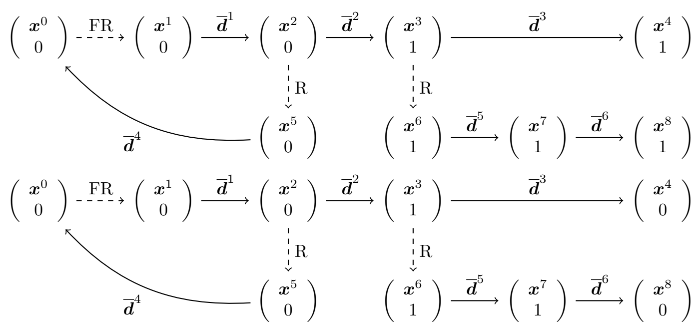

Fig. 4. This figure shows values for additional components resolving the strong $\mathcal{X}'$-determinism conflicts from Example 2 in Figure 1.

In order to find all valid labelings of a general experiment graph $\mathcal{D}(\mathcal{X}') = (V_{\mathcal{X}'}, A_D \cup A_S)$ we set up an optimization problem encoding the conditions for valid labelings and having as objective the minimization of the number a of additional components. For that we introduce decision variables y_{ji} to determine whether label i is assigned to $\boldsymbol{x}^j$.

We are interested in finding $\min\{a \in \mathbb{N} : \mathcal{P}(a) \neq \emptyset\}$, where $\mathcal{P}(a)$ is given by

$$\sum_{i=1}^{a} |y_{ji} - y_{li} - (y_{pi} - y_{qi})| \geq 1 \quad \text{for all}(\boldsymbol{x}^j, \boldsymbol{x}^l), (\boldsymbol{x}^p, \boldsymbol{x}^q) \in A_D, \text{ with } \boldsymbol{x}^j = \boldsymbol{x}^p, \boldsymbol{x}^l \neq \boldsymbol{x}^q \tag{2a}$$

$$y_{ji} - y_{li} = 0 \quad \text{for all } (\boldsymbol{x}^j, \boldsymbol{x}^l) \in A_S \tag{2b}$$

$$y_{ji} - y_{li} = y_{pi} - y_{qi} \quad \text{for all } (\boldsymbol{x}^j, \boldsymbol{x}^l), (\boldsymbol{x}^p, \boldsymbol{x}^q) \in A_D, \text{ with } \boldsymbol{x}^l - \boldsymbol{x}^j = \boldsymbol{x}^p - \boldsymbol{x}^q \tag{2c}$$

$$y_{j1}, \ldots, y_{j2^a} \in \{0, 1\} \quad \text{for all } (\boldsymbol{x}^j, \boldsymbol{x}^l) \in A_D, i = 1, \ldots, 2^a, \tag{2d}$$

where equations (2a) ensure that every state has a unique successor state (Condition (i) from Definition 2), equations (2b) that no stimulation changes the state of additional components (Condition (ii)), and equations (2c) preserve equal difference vectors (Condition (iii)). The conditions (2d) ensure that we have binary decision variables y_{ij}. Each valid labeling corresponds to a vector in $\mathcal{P}(a)$.

Note, due to inequalities (2a) the optimization problem is non-linear and has a non-convex set of feasible solutions. However, it is only necessary to find the minimal a so that $\mathcal{P}(a) \neq \emptyset$. We can consider the set $\mathcal{P}(a)$ as the union of 2^a convex sets (see Figure 5 for an illustration). Therefore, we can split the problem into 2^a linear subproblems, each having a convex (=polyhedral) feasible region. For that, we define two sets for each subproblem $1 \leq k \leq 2^a$, namely $P^+(k)$ and $P^-(k)$, so that $P^+(k) \cup P^-(k) = \{1, \ldots, a\}$ and $P^+(k) \cap P^-(k) = \emptyset$ and $P^+(p) \neq P^+(q)$, $P^-(p) \neq P^-(q)$ for all $p \neq q$. The sets induce the indices i so that $y_{ji} - y_{li} - (y_{pi} - y_{qi}) \geq 0$ and $y_{ji} - y_{li} - (y_{pi} - y_{qi}) \leq 0$, respectively. Hereby, we have all possible combinations. For the sake of readability let $z_{jlpqi} = y_{ji} - y_{li} - (y_{pi} - y_{qi})$. Then we replace inequalities (2a) by the following constraints

$$\sum_{i^+ \in P^+(k)} z_{jlpqi^+} - \sum_{i^- \in P^-(k)} z_{jlpqi^-} \geq 1 \quad \text{for all } (\boldsymbol{x}^j, \boldsymbol{x}^l), (\boldsymbol{x}^p, \boldsymbol{x}^q) \in \mathcal{A}_D, \tag{3a}$$

$$z_{jlpqi^+} \geq 0 \quad \begin{array}{l} \text{for all } i^+ \in P^+(k), \\ \text{for all } (\boldsymbol{x}^j, \boldsymbol{x}^l), (\boldsymbol{x}^p, \boldsymbol{x}^q) \in \mathcal{A}_D, \end{array} \tag{3b}$$

$$z_{jlpqi^-} \leq 0 \quad \begin{array}{l} \text{for all } i^+ \in P^+(k), \\ \text{for all } (\boldsymbol{x}^j, \boldsymbol{x}^l), (\boldsymbol{x}^p, \boldsymbol{x}^q) \in \mathcal{A}_D, \end{array} \tag{3c}$$

where $\mathcal{A}_D := \{(\boldsymbol{x}^j, \boldsymbol{x}^l), (\boldsymbol{x}^p, \boldsymbol{x}^q) \in A_D \text{ with } \boldsymbol{x}^j = \boldsymbol{x}^p, \boldsymbol{x}^l \neq \boldsymbol{x}^q\}$. These linear subproblems can be solved by standard solvers, and the optimal solution a of the original problem is obtained if one subproblem turns out to be feasible. All (minimal) valid labelings are then in $\mathcal{P}(a)$.

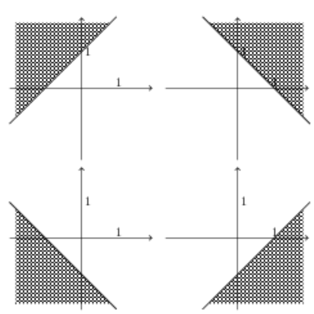

Fig. 5. In this figure the division of (2a) into 2^a subproblems is illustrated within the 2-dimensional space (i.e., $a = 2$). Each of the resulting 4 subproblems has a convex feasible region (highlighted by the dotted regions) whose union corresponds to the feasible region of the original problem.

4 Conclusion

In this work, we give a preprocessing step for a reconstruction algorithm from [8] that reconstructs extended Petri nets with priorities from experimental time-series data $\mathcal{X}'$, so-called $\mathcal{X}'$-deterministic extended Petri nets. For a successful reconstruction the data must be reproducible and monotone. While reproducibility is clearly evident, the necessity of monotone data is shown in [12]. In this paper we give a feasibility test for the data and a strategy for handling infeasible cases.

Firstly, the preprocessing step examines the given experimental time-series data for reproducibility, i.e., it tests if all measured states $\boldsymbol{x} \in \mathcal{X}'$ have a unique successor state (see Section 3.1). If this test is successful we can reconstruct an $\mathcal{X}'$-deterministic extended Petri net (Lemma 1).

Whenever two time-series $\mathcal{X}_i$ and $\mathcal{X}_\ell$ have a common state $\boldsymbol{x}$ but different successor states in each of these sequences (i.e., $\text{succ}_{\mathcal{X}_i}(\boldsymbol{x}) \neq \text{succ}_{\mathcal{X}_\ell}(\boldsymbol{x})$) we have an $\mathcal{X}'$-determinism conflict. Depending on whether the terminal states of these conflicts are equal or not, we have a weak or a strong $\mathcal{X}'$-determinism conflict.

When we encounter a weak $\mathcal{X}'$-determinism conflict we try to linearize the two sequences by the induced order of the successor relation. This is done in the second step of the preprocessing (see Section 3.1).

If linearizing the time-series is not possible or when there are strong $\mathcal{X}'$-determinism conflicts, we cannot reproduce $\mathcal{X}'$-deterministic extended Petri nets (Theorem 1). In this case we extend the data by adding additional components to every state of $\mathcal{X}'$ (see Section 3.2). Finally, in order to compute valid vectors of additional components, we solve an optimization problem.

After having performed the preprocessing step, the reproducibility of the (given or modified) data $\mathcal{X}'$ can be guaranteed such that $\mathcal{X}'$ can serve as appropriate input for the main reconstruction algorithm.

References

1. Chen, M., Hofestädt, W.: Quantitative Petri net model fo gene regulated metabolic networks in the cell. In Silico Biology **3** (2003) 347–365
2. Koch, I., Heiner, M.: Petri nets. In Junker, B.H., Schreiber, F., eds.: Biological Network Analysis. Wiley Book Series on Bioinformatics (2007) 139–179
3. Marwan, W., Wagler, A.K., Weismantel, R.: Petri nets as a framework for the reconstruction and analysis of signal transduction pathways and regulatory networks. Natural Computing **10** (2011) 639–654
4. Pinney, J.W., Westhead, R.D., McConkey, G.A.: Petri net representations in systems biology. Biochem. Soc. Tarns. **31** (2003) 1513–1515
5. Durzinsky, M., Marwan, W., Wagler, A.K.: Reconstruction of extended Petri nets from time-series data by using logical control functions. Journal of Mathematical Biology **66** (2013) 203–223
6. Durzinsky, M., Marwan, W., Wagler, A.K.: Reconstruction of extended Petri nets from time series data and its application to signal transduction and to gene regulatory networks. BMC Systems Biology **5** (2011)

7. Durzinsky, M., Wagler, A.K., Weismantel, R.: An algorithmic framework for network reconstruction. Journal of Theoretical Computer Science **412**(26) (2011) 2800–2815
8. Favre, M., Wagler, A.K.: Reconstructing $\mathcal{X}'$-deterministic extended Petri nets from experimental time-series data $\mathcal{X}'$. In: Preceedings of the 4th International Workshop on Biological Processes & Petri Nets. (2013) 45–59
9. Marwan, W., Wagler, A.K., Weismantel, R.: A mathematical approach to solve the network reconstruction problem. Math. Methods of Operations Research **67**(1) (2008) 117–132
10. Wagler, A.K.: Prediction of network structure. In Koch, I., Schreiber, F., Reisig, W., eds.: Modeling in Systems Biology. Volume 16 of Computational Biology., Springer London (2010) 309–338
11. Wagler, A.K., Wegener, J.T.: On minimality and equivalence of Petri nets. Proceedings of Concurrency, Specification and Programming CS&P'2012 Workshop **2** (2012) 382–393
12. Durzinsky, M., Wagler, A.K., Weismantel, R.: A combinatorial approach to reconstruct Petri nets from experimental data. In Heiner, M., Uhrmacher, A.M., eds.: CMSB. Volume 5307 of Lecture Notes in Computer Science., Springer (2008) 328–346
13. Starostzik, C., Marwan, W.: Functional mapping of the branched signal transduction pathway that controls sporulation in Physarum polycephalum. Photochem Photobiol **62**(5) (1995) 930–933
14. Lamparter, T., Marwan, W.: Spectroscopic detection of a phytochrome-like photoreceptor in the myxomycete Physarum polycephalum and the kinetic mechanism for the photocontrol of sporulation by pfr. Photochem Photobiol **73**(6) (2001) 697–702
15. Torres, L.M., Wagler, A.K.: Encoding the dynamics of deterministic systems. Math. Methods of Operations Research **73** (2011) 281–300