



HAL
open science

Preprocessing for Network Reconstruction: Feasibility Test and Handling Infeasibility

Annegret K Wagler, Jan-Thierry Wegener

► **To cite this version:**

Annegret K Wagler, Jan-Thierry Wegener. Preprocessing for Network Reconstruction: Feasibility Test and Handling Infeasibility. *Fundamenta Informaticae*, 2014, 10.3233/FI-2014-1138 . hal-02047309

HAL Id: hal-02047309

<https://hal.science/hal-02047309v1>

Submitted on 24 Feb 2019

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Preprocessing for Network Reconstruction: Feasibility Test and Handling Infeasibility

Annegret K. Wagler, 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 [1,7,9,10].

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 $w : A \rightarrow \mathbb{N}$ 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 $\mathbf{x} \in \mathbb{Z}_+^{|P|}$. If a capacity $\text{cap}(p)$ is given for the places,

* This work was funded by the French National Research Agency, the European Commission (Feder funds) and the Région Auvergne in the Framework of the LabEx IMobS³.

then $x_p \leq \text{cap}(p)$ follows and we obtain $\mathcal{X} := \{\mathbf{x} \in \mathbb{N}^{|P|} : x_p \leq \text{cap}(p)\}$ as set of potential states. A transition $t \in T$ is *enabled* in a state \mathbf{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 $\text{succ}(\mathbf{x}) = \mathbf{x}'$ 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 nets can be reconstructed from experimental data by exact, exclusively data-driven approaches [2,3,5,6,8,13]. These approaches take as input a set P of places and discrete time-series data \mathcal{X}' given by sequences $(\mathbf{x}^0; \mathbf{x}^1, \dots, \mathbf{x}^m)$ of experimentally observed system states. The goal is to determine all Petri nets (P, T, A, w) that are able to reproduce the data 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 [14,15], 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 \mathbf{x}^{j+1} from \mathbf{x}^j , but exactly this transition will be selected among all enabled ones in \mathbf{x}^j which is necessary to reach \mathbf{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 $\mathbf{x}^j \in \mathcal{X}'$ there is a unique observed successor state $\text{succ}_{\mathcal{X}'}(\mathbf{x}^j) = \mathbf{x}^{j+1} \in \mathcal{X}'$) 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 [4].

In this paper, we focus on a necessary preprocessing step for the reconstruction approach described in [6]. We test the data for reproducibility, provide a feasibility test (based on previous works in [5]) 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 [5,8]) to cope with infeasible cases (see Section 3.2). We close with some concluding remarks.

Note that the here presented results appeared without proofs in [16].

2 Reconstructing Petri Nets from Experimental Data

In this section we describe the input and the desired output of the reconstruction method from [6]. Moreover, we briefly sketch the reconstruction procedure; for a detailed description, we refer the reader to [6].

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 measurements¹.

To perform an experiment, the system is stimulated in a state \mathbf{x}^0 (by external stimuli like the change of nutrient concentrations or the exposition to some pathogens) to generate an initial state $\mathbf{x}^1 \in \mathcal{X}$. 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 $\mathcal{X}'(\mathbf{x}^1, \mathbf{x}^k) = (\mathbf{x}^0; \mathbf{x}^1, \dots, \mathbf{x}^k)$ of states $\mathbf{x}^i \in \mathcal{X}$ reflecting the time-dependent response of the system to the stimulation. Note that we also provide the state \mathbf{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* $\mathbf{x}^k \in \mathcal{X}$, 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 $\mathbf{x}^k \in \mathcal{X}'_{term}$ as a state which has itself as observed successor state, i.e., $\mathbf{x}^k = \text{succ}_{\mathcal{X}'}(\mathbf{x}^k)$.

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

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

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

Example 1. As running example, we consider the *light-induced sporulation of Physarum polycephalum*. The developmental decision of *P. polycephalum* plasmodia to enter the sporulation pathway is controlled by environmental factors like visible light [11]. 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. 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: $\mathcal{X}(\mathbf{x}^1, \mathbf{x}^4) = (\mathbf{x}^0; \mathbf{x}^1, \mathbf{x}^2, \mathbf{x}^3, \mathbf{x}^4)$, $\mathcal{X}(\mathbf{x}^5, \mathbf{x}^0) = (\mathbf{x}^2; \mathbf{x}^5, \mathbf{x}^0)$ and $\mathcal{X}(\mathbf{x}^6, \mathbf{x}^8) = (\mathbf{x}^3; \mathbf{x}^6, \mathbf{x}^7, \mathbf{x}^8)$.

In the best case, two consecutively measured states $\mathbf{x}^j, \mathbf{x}^{j+1} \in \mathcal{X}'$ are also consecutive system states, i.e., \mathbf{x}^{j+1} can be obtained from \mathbf{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 \mathbf{x}^{j+1} is obtained from \mathbf{x}^j by a switching sequence of some length, where the intermediate states are not reported in \mathcal{X}' .

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.

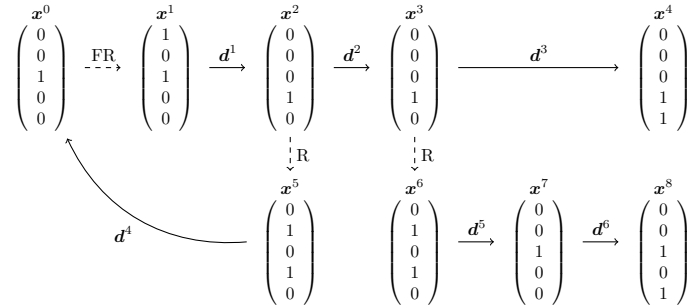


Fig. 1. This figure shows experimental time-series data \mathcal{X}' for the light-induced sporulation of *Physarum polycephalum*. The experimental setting uses the set $P = \{FR, R, P_{fr}, P_r, S_p\}$ of studied components, observed states are represented by vectors of the form $\mathbf{x} = (x_{FR},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

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 $\mathbf{x}^j \in \mathcal{X}'$ there is a unique observed successor state $\text{succ}_{\mathcal{X}'}(\mathbf{x}^j) = \mathbf{x}^{j+1} \in \mathcal{X}'$. Moreover, the data \mathcal{X}' are *monotone* if for each such pair $(\mathbf{x}^j, \mathbf{x}^{j+1}) \in \mathcal{X}'$, the possible intermediate states $\mathbf{x}^j = \mathbf{y}^1, \mathbf{y}^2, \dots, \mathbf{y}^{m+1} = \mathbf{x}^{j+1}$ satisfy

$$\begin{aligned} y_p^1 &\leq y_p^2 \leq \dots \leq y_p^m \leq y_p^{m+1} \text{ for all } p \in P \text{ with } x_p^j \leq x_p^{j+1} \text{ and} \\ y_p^1 &\geq y_p^2 \geq \dots \geq y_p^m \geq y_p^{m+1} \text{ for all } p \in P \text{ with } x_p^j \geq x_p^{j+1}. \end{aligned}$$

Whereas reproducibility is obviously necessary, it was shown in [4] that monotonicity has to be required or, equivalently, that all essential responses are indeed reported in the experiments ².

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 $\mathbf{x}^j \in \mathcal{X}'$ to $\text{succ}_{\mathcal{X}'}(\mathbf{x}^j) = \mathbf{x}^{j+1} \in \mathcal{X}'$. 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_t to the *update vector* \mathbf{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 \mathbf{x}^{j+1} from \mathbf{x}^j by a switching sequence using the transitions from a subset $T' \subseteq T$ is equivalent to obtain the state vector \mathbf{x}^{j+1} from \mathbf{x}^j by adding

² When continuous data is discretized, all local minima and maxima of the measured values have to be kept for each $p \in P$ to ensure monotonicity.

the corresponding columns $M_{.t}$ of M for all $t \in T'$:

$$\mathbf{x}^j + \sum_{t \in T'} M_{.t} = \mathbf{x}^{j+1}. \quad (1)$$

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 $\mathbf{x}^j, \text{succ}_{\mathcal{X}'}(\mathbf{x}^j) = \mathbf{x}^{j+1} \in \mathcal{X}'$, the linear equation system $\mathbf{x}^{j+1} - \mathbf{x}^j = M\lambda$ has an integral solution $\lambda \in \mathbb{N}^{|T|}$ such that λ is the incidence vector of a sequence (t^1, \dots, t^m) of transition switches, i.e., there are intermediate states $\mathbf{x}^j = \mathbf{y}^1, \mathbf{y}^2, \dots, \mathbf{y}^{m+1} = \mathbf{x}^{j+1}$ with $\mathbf{y}^l + M_{.t^l} = \mathbf{y}^{l+1}$ for $1 \leq l \leq m$. Hereby, monotonicity avoids unnecessary solutions, since no homogeneous solutions of equation (1) have to be considered, see [4,13].

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 [2,3,6,14,15].

On the one hand, 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 \mathbf{x} only if a token (resp. no token) is present in p ; we denote by $T_{\mathcal{A}}(\mathbf{x})$ the set of all such transitions.

On the other hand, in [8,12,13] the concept of priority relations among the transitions of a network was introduced in order to allow the modeling of deterministic systems. In [8] it is proposed to model priorities by partial orders \mathcal{O} on the transitions to reflect the rates of the corresponding reactions where the fastest reaction has highest priority and, thus, is taken. For each state \mathbf{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 $T_{\mathcal{A}, \mathcal{O}}(\mathbf{x})$ 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 .

The extended Petri net with priorities $(\mathcal{P}, \mathcal{O})$ is *\mathcal{X}' -deterministic* if $\{t^l\} = T_{\mathcal{A}, \mathcal{O}}(\mathbf{y}^l)$ holds for all \mathbf{y}^l . The desired output of the reconstruction approach consists of the set of all \mathcal{X}' -deterministic extended Petri nets $(\mathcal{P}, \text{cap}, \mathcal{O})$ (all having the same set P of places and the same capacities cap deduced from \mathcal{X}' by $\text{cap}(p) = \max\{x_p : \mathbf{x} \in \mathcal{X}'\}$).

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

2.3 Steps of the Reconstruction Approach

To reconstruct \mathcal{X}' -deterministic extended Petri nets from experimental time-series data \mathcal{X}' , the following approach is proposed by [6], based on previous works in [2,3,4,5,8].

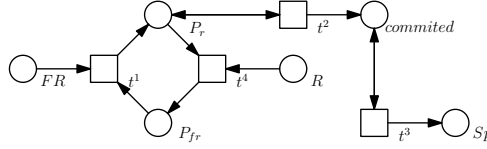


Fig. 2. This figure shows an \mathcal{X}' -deterministic extended Petri net fitting the experimental data from Example 1. The set P of components has been extended by a component *committed* which cannot be measured directly, but only indirectly deduced by the behavior of *Physarum polycephalum* observed in the experiment. The here shown network corresponds to solution (a) from Figure 4. It has 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 $|T_{\mathcal{A}, \mathcal{O}}(\mathbf{x})| = 1$ for every state $\mathbf{x} \in \mathcal{X}'$.

As initial step, extract the observed changes of states from the experimental data. For that, define the set $\mathcal{D} := \{\mathbf{d}^j = \mathbf{x}^{j+1} - \mathbf{x}^j : \mathbf{x}^{j+1} = \text{succ}_{\mathcal{X}'}(\mathbf{x}^j) \in \mathcal{X}'\}$. Generating the complete list of all \mathcal{X}' -deterministic extended Petri nets $\mathcal{P} = (P, T, \mathcal{A}, w)$ includes finding the corresponding standard networks and their incidence matrices $M \in \mathbb{Z}^{|P| \times |T|}$. The first step is to describe the potential update vectors which might constitute the columns of M . Due to monotonicity, it suffices to represent any $\mathbf{d}^j \in \mathcal{D}$ using update vectors from the following set only:

$$\text{Box}(\mathbf{d}^j) = \left\{ \mathbf{r} \in \mathbb{Z}^{|P|} : \begin{array}{l} 0 \leq r_p \leq d_p \text{ if } d_p^j > 0 \\ d_p \leq r_p \leq 0 \text{ if } d_p^j < 0 \\ r_p = 0 \text{ if } d_p^j = 0 \end{array} \right\} \setminus \{\mathbf{0}\}.$$

Next, we determine for any $\mathbf{d}^j \in \mathcal{D}$, the set $\Lambda(\mathbf{d}^j)$ of all integral solutions of

$$\mathbf{d}^j = \sum_{\mathbf{r}^t \in \text{Box}(\mathbf{d}^j)} \lambda_t \mathbf{r}^t, \quad \lambda_t \in \mathbb{Z}_+,$$

and for each $\lambda \in \Lambda(\mathbf{d}^j)$, the (multi-)set $\mathcal{R}(\mathbf{d}^j, \lambda) = \{\mathbf{r}^t \in \text{Box}(\mathbf{d}^j) : \lambda_t \neq 0\}$ of update vectors used for this solution λ . Every permutation $\pi = (\mathbf{r}^{t_1}, \dots, \mathbf{r}^{t_m})$ of the elements of a set $\mathcal{R}(\mathbf{d}^j, \lambda)$ gives rise to a sequence of intermediate states $\mathbf{x}^j = \mathbf{y}^1, \mathbf{y}^2, \dots, \mathbf{y}^m, \mathbf{y}^{m+1} = \mathbf{x}^{j+1}$ with

$$\sigma = \sigma_{\pi, \lambda}(\mathbf{x}^j, \mathbf{d}^j) = ((\mathbf{y}^1, \mathbf{r}^{t_1}), (\mathbf{y}^2, \mathbf{r}^{t_2}), \dots, (\mathbf{y}^m, \mathbf{r}^{t_m}))$$

which induces a priority relation \mathcal{O}_σ since transition t_i resulting from \mathbf{r}^{t_i} is supposed to have highest priority in \mathbf{y}^i for $1 \leq i \leq m$. Two sequences σ and σ' are in *priority conflict* if there are update vectors $\mathbf{r}^t \neq \mathbf{r}^{t'}$ and intermediate states \mathbf{y}, \mathbf{y}' such that $t, t' \in T(\mathbf{y}) \cap T(\mathbf{y}')$ and $(\mathbf{y}, \mathbf{r}^t) \in \sigma$ but $(\mathbf{y}', \mathbf{r}^{t'}) \in \sigma'$ (since this implies $t > t'$ in \mathcal{O}_σ but $t' > t$ in $\mathcal{O}_{\sigma'}$). We have a weak (resp. strong) priority conflict if $\mathbf{y} \neq \mathbf{y}'$ (resp. $\mathbf{y} = \mathbf{y}'$) which can (resp. cannot) be resolved by adding appropriate control-arcs. In [6], it is proposed to construct a *priority conflict graph* \mathcal{G} whose nodes correspond to all possible sequences $\sigma_{\pi, \lambda}(\mathbf{x}^j, \mathbf{d}^j)$ and whose edges reflect weak and strong priority conflicts. In \mathcal{G} , all

node subsets S are generated that select exactly one sequence $\sigma_{\pi,\lambda}(\mathbf{x}^j, \mathbf{d}^j)$ per difference vector $\mathbf{d}^j \in \mathcal{D}$ such that no strong priority conflicts occur between the selected sequences. Each such subset S gives rise to a standard network $\mathcal{P}_S = (P, T_S, A_S, w)$ which is conformal with \mathcal{X}' and can be made \mathcal{X}' -deterministic by inserting control-arcs and combining the priority relations \mathcal{O}_σ for all $\sigma \in S$:

- we obtain the columns of the incidence matrix M_S of the network by taking the union of all sets $\mathcal{R}(\mathbf{d}^j, \lambda)$ corresponding to the sequences $\sigma = \sigma_{\pi,\lambda}(\mathbf{x}^j, \mathbf{d}^j)$ selected by $\sigma \in S$;
- for each weak priority conflict between $\sigma, \sigma' \in S$ involving update vectors $\mathbf{r}^t \neq \mathbf{r}^{t'}$ and intermediate states $\mathbf{y} \neq \mathbf{y}'$, include either a read-arc $(p, t) \in A_R$ with weight $w(p, t) > y'_p$ for some p with $y_p > y'_p$ or an inhibitor-arc $(p, t) \in A_I$ with weight $w(p, t) < y_p$ for some p with $y_p < y'_p$ to disable transition t resulting from \mathbf{r}^t at \mathbf{y}' ,
- for each $\sigma \in S$, define \mathcal{O}_σ by $\mathcal{O}_\sigma = \{t_i > t : t \in T_{A_S \cup A_R \cup A_I}(\mathbf{y}^i) \setminus \{t_i\}, 1 \leq i \leq m\}$ and let $\mathcal{O}_S = \bigcup_{\sigma \in S} \mathcal{O}_\sigma$ be the studied partial order.

This implies finally that every extended network $\mathcal{P}_S = (P, T_S, A_S \cup A_R \cup A_I, w)$ together with the partial order \mathcal{O}_S is \mathcal{X}' -deterministic, see [6] for details.

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

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

$\mathcal{D}(\mathcal{X}')$ 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 $\mathbf{x} \in \mathcal{X}'$ has a unique successor state $\text{succ}_{\mathcal{X}'}(\mathbf{x}) \in \mathcal{X}'$. We provide a feasibility test to ensure this property (based on previous tests for standard Petri nets [5] and extended Petri nets [3], see Section 3.1). If this test fails, we have a state $\mathbf{x} \in \mathcal{X}'$ 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 [5,8,13], this situation can be resolved by adding further components³ to P with the goal to split any state

³ 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.

$\mathbf{x} \in \mathcal{X}'$ 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 [5,8]).

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 $\mathcal{X}_i = \mathcal{X}'(\mathbf{x}^{i_0}, \mathbf{x}^{i_k})$ and $\mathcal{X}_\ell = \mathcal{X}'(\mathbf{x}^{\ell_0}, \mathbf{x}^{\ell_m})$ are in \mathcal{X}' -determinism conflict, when there exists a state $\mathbf{x} \in \mathcal{X}'$ with $\text{succ}_{\mathcal{X}_i}(\mathbf{x}) \neq \text{succ}_{\mathcal{X}_\ell}(\mathbf{x})$ and call \mathbf{x} the corresponding \mathcal{X}' -determinism conflict state. We have

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

The definition of strong \mathcal{X}' -determinism conflicts includes the case discussed in [3,5] that there must not exist a terminal state $\mathbf{x}^j \in \mathcal{X}'_{term}$ that occurs as intermediate state in an experiment and the case that a state $\mathbf{x}^j \in \mathcal{X}' \setminus \mathcal{X}'_{term}$ has itself as successor, which would result in $\mathbf{d}^j = 0$ (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 $\mathcal{X}'(\mathbf{x}^1, \mathbf{x}^4)$ the states \mathbf{x}^2 and \mathbf{x}^3 are equal but have different successor states,
- the sequences $\mathcal{X}'(\mathbf{x}^5, \mathbf{x}^0)$ and $\mathcal{X}'(\mathbf{x}^6, \mathbf{x}^8)$ have equal initial state $\mathbf{x}^5 = \mathbf{x}^6$, but different terminal states. Besides the initial states, the states \mathbf{x}^0 and \mathbf{x}^7 are \mathcal{X}' -determinism conflict states.

Obviously, every \mathcal{X}' -determinism conflict violates the condition of the data being reproducible. Conversely, if no \mathcal{X}' -determinism conflict occurs, the data are reproducible and we have:

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

Proof. The pre-condition that every state $\mathbf{x} \in \mathcal{X}'$ has a unique successor in \mathcal{X}' includes the cases that no state $\mathbf{x}^j \in \mathcal{X}' \setminus \mathcal{X}'_{term}$ has itself as successor (and, thus, $\mathbf{d}^j \neq 0$ follows for all $\mathbf{d}^j \in \mathcal{D}$) and that no terminal state $\mathbf{x}^k \in \mathcal{X}'_{term}$ is an intermediate state of any experiment.

Having $\mathbf{d}^j \neq 0$ for all $\mathbf{d}^j \in \mathcal{D}$ guarantees the existence of a *standard network* being conformal with \mathcal{X}' : By construction, $\text{Box}(\mathbf{d}^j)$ is non-empty due to $\mathbf{d}^j \neq 0$ for all $\mathbf{d}^j \in \mathcal{D}$. Hence, \mathbf{d}^j is a trivial representation λ^0 for itself with $\mathcal{R}(\mathbf{d}^j, \lambda^0) = \{\mathbf{d}^j\}$, and the standard network \mathcal{P} whose incidence matrix has all vectors $\mathbf{d}^j \in \mathcal{D}$ as columns is conformal with \mathcal{X}' .

\mathcal{P} can be made \mathcal{X}' -deterministic by adding appropriate control-arcs: Suppose that there are $\mathbf{x}^j, \mathbf{x}^l \in \mathcal{X}'$ such that \mathbf{d}^j and \mathbf{d}^l with $\mathbf{d}^j \neq \mathbf{d}^l$ are enabled at both states \mathbf{x}^j and \mathbf{x}^l . By pre-condition, one of them is not a terminal state, say

$\mathbf{x}^j \notin \mathcal{X}'_{term}$. Then \mathbf{d}^j has to be turned into a transition t^j disabled at \mathbf{x}^l . For that, include either a read-arc $(p, t^j) \in A_R$ with weight $w(p, t^j) > x_p^l$ for some p with $x_p^j > x_p^l$, or an inhibitor-arc $(p, t^j) \in A_R$ with weight $w(p, t^j) < x_p^l$ for some p with $x_p^j < x_p^l$. This can be done since $\mathbf{x}^j \neq \mathbf{x}^l$ by pre-condition (otherwise, $\mathbf{x}^j = \mathbf{x}^l$ would be a state having two different successors $\mathbf{x}^j + \mathbf{d}^j$ and $\mathbf{x}^l + \mathbf{d}^j$).

Therefore, the existence of an extended Petri net (where on each $\mathbf{x}^j \in \mathcal{X}'$ the transition t^j resulting from \mathbf{d}^j has highest priority, including priority $t^l < t^j \in \mathcal{O}$ for each of the above described conflicts) being \mathcal{X}' -deterministic is ensured. \square

Two time-series $\mathcal{X}'(\mathbf{x}^{i_0}, \mathbf{x}^{i_k})$ and $\mathcal{X}'(\mathbf{x}^{\ell_0}, \mathbf{x}^{\ell_m})$ with $\mathbf{x}^{i_k} = \mathbf{x}^{\ell_m}$ 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 $\mathcal{X}'(\mathbf{x}^1, \mathbf{x}^k) = (\mathbf{x}^0; \mathbf{x}^1, \dots, \mathbf{x}^k)$, a linear order is induced by the successor relation: $\mathbf{x}^j \leq \mathbf{x}^{j+1}$ iff $\mathbf{x}^{j+1} = \text{succ}_{\mathcal{X}'(\mathbf{x}^1, \mathbf{x}^k)}(\mathbf{x}^j)$, hence \mathcal{X}' can be considered as a partially ordered set (ordered by the successor relation), where each time-series $\mathcal{X}'(\mathbf{x}^1, \mathbf{x}^k)$ is a chain of \mathcal{X}' . Let $\text{succ}_{\mathcal{X}'}(\mathbf{x}^j) = \mathbf{x}^{j+1}$ and

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

Note that due to monotonicity, all intermediate states \mathbf{y} of any refined sequence from \mathbf{x}^j to \mathbf{x}^{j+1} lie in $\text{Box}(\mathbf{x}^j, \mathbf{x}^{j+1})$. Consequently, if two time-series $\mathcal{X}_i = \mathcal{X}'(\mathbf{x}^{i_0}, \mathbf{x}^{i_k})$ and $\mathcal{X}_\ell = \mathcal{X}'(\mathbf{x}^{\ell_0}, \mathbf{x}^{\ell_m})$ with $\mathbf{x}^{i_k} = \mathbf{x}^{\ell_m}$ are in weak \mathcal{X}' -determinism conflict, and \mathbf{x} is a determinism conflict state then we have to test whether $\text{succ}_{\mathcal{X}_i}(\mathbf{x}) \in \text{Box}(\mathbf{x}, \text{succ}_{\mathcal{X}_\ell}(\mathbf{x}))$ or $\text{succ}_{\mathcal{X}_\ell}(\mathbf{x}) \in \text{Box}(\mathbf{x}, \text{succ}_{\mathcal{X}_i}(\mathbf{x}))$, see Figure 3. If the test fails, we cannot find a \mathcal{X}' -deterministic linear order. Otherwise, $\mathbf{x}' = \text{succ}_{\mathcal{X}_i}(\mathbf{x})$ or $\mathbf{x}' = \text{succ}_{\mathcal{X}_\ell}(\mathbf{x})$ is a new \mathcal{X}' -determinism conflict state, and the test has to be repeated for \mathbf{x}' (see Algorithm 1). This works since at least the terminal states \mathbf{x}^{i_k} and \mathbf{x}^{ℓ_m} are equal.

Whenever this test is successful for \mathbf{x} and all subsequent \mathcal{X}' -determinism conflict states \mathbf{x}' , we say that it is *resolvable*, otherwise we say it is an *unresolvable* weak \mathcal{X}' -determinism conflict.

We next prove the correctness of Algorithm 1.

Lemma 2. *Let \mathcal{X}' be experimental time-series data and let $\mathcal{X}'_i = \mathcal{X}'(\mathbf{x}^{i_0}, \mathbf{x}^{i_k})$ and $\mathcal{X}'_\ell = \mathcal{X}'(\mathbf{x}^{\ell_0}, \mathbf{x}^{\ell_m})$ be two time-series in a weak \mathcal{X}' -determinism conflict. Algorithm 1 returns linearized times-series for \mathcal{X}'_i and \mathcal{X}'_ℓ if and only if the weak \mathcal{X}' -determinism conflict is resolvable.*

Proof. Let $\mathbf{x} \in \mathcal{X}'$ be a conflict state for \mathcal{X}'_i and \mathcal{X}'_ℓ . We show that “false” is returned if and only if the conflict in \mathbf{x} is not resolvable.

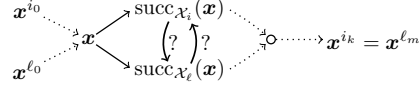


Fig. 3. This figure shows a weak \mathcal{X}' -determinism conflict. To resolve this conflict we can test if the two different successor states (resulting from two different experiments) of the \mathcal{X}' -determinism conflict state \mathbf{x} can be ordered respecting monotonicity. In other words, we test if one of these successor states is an unmeasured intermediate state of \mathbf{x} and the other successor state.

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

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

```

1: for all conflict states  $\mathbf{x}$  do
2:    $\mathbf{x}^i \leftarrow \text{succ}_{\mathcal{X}'(\mathbf{x}^{i_0}, \mathbf{x}^{i_k})}(\mathbf{x})$ ,  $\mathbf{x}^\ell \leftarrow \text{succ}_{\mathcal{X}'(\mathbf{x}^{\ell_0}, \mathbf{x}^{\ell_m})}(\mathbf{x})$ 
3:    $\mathcal{L} \leftarrow \emptyset$  ▷ stores the linear order
4:   while  $\mathbf{x}^i \neq \mathbf{x}^\ell$  do
5:     if  $\mathbf{x}^i \in \text{Box}(\mathbf{x}, \mathbf{x}^\ell)$  then
6:        $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathbf{x}^i < \mathbf{x}^\ell\}$ 
7:        $\mathbf{x} \leftarrow \mathbf{x}^i$ 
8:        $\mathbf{x}^i \leftarrow \text{succ}_{\mathcal{X}'(\mathbf{x}^{i_0}, \mathbf{x}^{i_k})}(\mathbf{x}^i)$ 
9:     else if  $\mathbf{x}^\ell \in \text{Box}(\mathbf{x}, \mathbf{x}^i)$  then
10:       $\mathcal{L} \leftarrow \mathcal{L} \cup \{\mathbf{x}^\ell < \mathbf{x}^i\}$ 
11:       $\mathbf{x} \leftarrow \mathbf{x}^\ell$ 
12:       $\mathbf{x}^\ell \leftarrow \text{succ}_{\mathcal{X}'(\mathbf{x}^{\ell_0}, \mathbf{x}^{\ell_m})}(\mathbf{x}^\ell)$ 
13:     else
14:       return false
15: return adjusted time-series according to  $\mathcal{L}$ 

```

In line 2 the variables \mathbf{x}^i and \mathbf{x}^ℓ are initialized as successor state of \mathbf{x} in the corresponding time-series. The set \mathcal{L} , which stores the linear order, is initialized with the empty set in line 3.

If we have $\mathbf{x}^i = \mathbf{x}^\ell$ then one of the following is true:

- (i) $\mathbf{x}^i = \mathbf{x}^\ell = \mathbf{x}^{i_k} = \mathbf{x}^{\ell_m}$, i.e., both states are the terminal states of the time-series,
- (ii) all successor states after \mathbf{x} are equal in both time-series, and thus further linearization is not necessary,
- (iii) both time-series have equal successor states after \mathbf{x} , but there is another weak conflict state in the time-series.

While in the first two cases the algorithm stops, in (iii) the algorithm continues due to the **for** loop in line 1.

In line 5 (resp. 9) it is tested if $\mathbf{x}^i \in \text{Box}(\mathbf{x}, \mathbf{x}^\ell)$ (resp. $\mathbf{x}^\ell \in \text{Box}(\mathbf{x}, \mathbf{x}^i)$) (see Figure 3 for an illustration). Note that this is possibly true for several successor states of \mathbf{x} . However, the intermediate states of a decomposition must be monotone and therefore, the tested states must respect the monotonicity constraint as well. This is ensured by lines 7 and 8 (resp. 11 and 12). If some states are not monotone intermediate states of the other time-series, then it follows that there exist a state $\mathbf{x}^{i+1} = \text{succ}_{\mathcal{X}'_i}(\mathbf{x}^i)$ so that $\mathbf{x}^{i+1} \notin \text{Box}(\mathbf{x}^i, \mathbf{x}^\ell)$ but $\mathbf{x}^{i+1} \in \text{Box}(\mathbf{x}^{i-1}, \mathbf{x}^\ell)$, with $\mathbf{x}^i = \text{succ}_{\mathcal{X}'_i}(\mathbf{x}^i)$.

Depending on which state comes next in the linear order, the set \mathcal{L} is updated accordingly in line 6 (resp. 10). Since the case $\mathbf{x}^\ell \in \text{Box}(\mathbf{x}, \mathbf{x}^i)$ is tested analogously, we ensure that the successor states of \mathbf{x} in one time-series is within the box of \mathbf{x} and the successor state of the other time-series; and, in the case of several successor states within that box, that they are monotone. If neither is true, then it follows that the conflict is not resolvable. Hereby, line 14 returns “false”. When the algorithm stops in line 15, two time-series are returned based on the computed linear order.

The algorithm always stops after a finite number of steps since we have $\mathbf{x}^{i_k} = \mathbf{x}^{\ell_m}$, ensuring that line 4 is called finitely often (time-series are by definition finite) before being “true” or line 14 is called before. \square

This enables us to formulate the following feasibility test:

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.*

Proof. “ \Leftarrow ” If every state $\mathbf{x} \in \mathcal{X}'$ has a unique successor in \mathcal{X}' , the assertion follows from Lemma 1.

Let there be two time-series $\mathcal{X}'(\mathbf{x}^{i_0}, \mathbf{x}^{i_k})$ and $\mathcal{X}'(\mathbf{x}^{\ell_0}, \mathbf{x}^{\ell_m})$ in a resolvable weak \mathcal{X}' -determinism conflict in state $\mathbf{x} \in \mathcal{X}'$. Due to monotonicity of the data and by definition of a resolvable weak \mathcal{X}' -determinism conflict state, there exists a linear order on the subsequences of the time-series starting with \mathbf{x} , so that every successor after \mathbf{x} is unique, i.e., we have a refined sequence $(\mathbf{x}, \tilde{\mathbf{x}}^1, \dots, \tilde{\mathbf{x}}^{\tilde{k}})$, with $\tilde{\mathbf{x}}^{\tilde{k}} = \mathbf{x}^{i_k} = \mathbf{x}^{\ell_m}$, and for all $1 \leq j \leq \tilde{k}$ we have $\tilde{\mathbf{x}}^j \in \mathcal{X}'(\mathbf{x}^{i_0}, \mathbf{x}^{i_k})$ or $\tilde{\mathbf{x}}^j \in \mathcal{X}'(\mathbf{x}^{\ell_0}, \mathbf{x}^{\ell_m})$. Now we consider the following time-series $\tilde{\mathcal{X}}'(\mathbf{x}^{i_0}, \mathbf{x}^{i_k}) = (\mathbf{x}^{i_1}, \mathbf{x}^{i_2}, \dots, \mathbf{x}, \tilde{\mathbf{x}}^1, \dots, \tilde{\mathbf{x}}^{\tilde{k}})$ and $\tilde{\mathcal{X}}'(\mathbf{x}^{\ell_0}, \mathbf{x}^{\ell_m}) = (\mathbf{x}^{\ell_1}, \mathbf{x}^{\ell_2}, \dots, \mathbf{x}, \tilde{\mathbf{x}}^1, \dots, \tilde{\mathbf{x}}^{\tilde{k}})$.

Let $\tilde{\mathcal{X}}'$ be now the experimental time-series data containing all (if necessary linearized) time-series from \mathcal{X}' . Then every state in $\tilde{\mathcal{X}}'$ has a unique successor state, and thus, Lemma 1 can be applied to $\tilde{\mathcal{X}}'$, proving the statement.

“ \Rightarrow ” We show, if there exists a strong \mathcal{X}' -determinism conflict or an unresolvable weak \mathcal{X}' -determinism conflicts then there does not exist an \mathcal{X}' -deterministic extended Petri net.

Firstly, let $\mathbf{x}^j \in \mathcal{X}'$ be a strong \mathcal{X}' -determinism conflict state so that \mathbf{x}^j has itself as successor state. Then $\mathbf{d}^j = 0$ follows and, thus, $\text{Box}(\mathbf{d}^j) = \emptyset$. Therefore, there does not exist a standard Petri net conformal with \mathcal{X}' and, thus, no \mathcal{X}' -deterministic extended Petri net.

Secondly, let $\mathbf{x}^j \in \mathcal{X}'$ be a strong \mathcal{X}' -determinism conflict state for two time-series $\mathcal{X}_i = \mathcal{X}'(\mathbf{x}^{i_0}, \mathbf{x}^{i_k})$ and $\mathcal{X}_\ell = \mathcal{X}'(\mathbf{x}^{\ell_0}, \mathbf{x}^{\ell_m})$. Then, by definition, $\text{succ}_{\mathcal{X}_i}(\mathbf{x}^j) \neq \text{succ}_{\mathcal{X}_\ell}(\mathbf{x}^j)$. We show that this \mathcal{X}' -determinism conflict can neither be resolved by priorities nor by control-arcs.

Let $\mathbf{d}^i, \mathbf{d}^\ell$ such that $\text{succ}_{\mathcal{X}_i}(\mathbf{x}^j) = \mathbf{x}^j + \mathbf{d}^i$ and $\text{succ}_{\mathcal{X}_\ell}(\mathbf{x}^j) = \mathbf{x}^j + \mathbf{d}^\ell$. For the trivial decompositions $\sigma_{\pi, \lambda}(\mathbf{x}^j, \mathbf{d}^i) = ((\mathbf{x}^j, \mathbf{r}^{t^i}))$ and $\sigma_{\pi, \lambda}(\mathbf{x}^j, \mathbf{d}^\ell) = ((\mathbf{x}^j, \mathbf{r}^{t^\ell}))$, the time-series \mathcal{X}_i implies $t^\ell < t^i \in \mathcal{O}$ while \mathcal{X}_ℓ implies $t^i < t^\ell \in \mathcal{O}$. This conflict can only be resolved by adding control arcs. Let $t^i < t^\ell \in \mathcal{O}$, then control-arcs must be added to disable t^ℓ in \mathbf{x}^j . But then t^i can never fire in \mathbf{x}^j . Analogously, if $t^i < t^\ell \in \mathcal{O}$ then t^ℓ can never fire in \mathbf{x}^j . In the case that $t^\ell < t^i \notin \mathcal{O}$ and $t^i < t^\ell \notin \mathcal{O}$ then again either \mathcal{X}_i and/or \mathcal{X}_ℓ are no longer valid using the same arguments. Thus, there is no extended Petri net with priorities being \mathcal{X}' -deterministic. Now we consider non-trivial decompositions $\sigma_{\pi, \lambda}(\mathbf{x}^j, \mathbf{d}^i) = ((\mathbf{y}^{j_1}, \mathbf{r}^{t^{i_1}}), \dots, (\mathbf{y}^{i_{m_i}}, \mathbf{r}^{t^{i_{m_i}}}))$ and $\sigma_{\pi, \lambda}(\mathbf{x}^j, \mathbf{d}^\ell) = ((\mathbf{y}^{\ell_1}, \mathbf{r}^{t^{\ell_1}}), \dots, (\mathbf{y}^{\ell_{m_\ell}}, \mathbf{r}^{t^{\ell_{m_\ell}}}))$. Since the successor of \mathbf{x}^j is not equal in both time-series, i.e., $\text{succ}_{\mathcal{X}_i}(\mathbf{x}^j) \neq \text{succ}_{\mathcal{X}_\ell}(\mathbf{x}^j)$ it follows that there exist \mathbf{y}^{j_i} and $\mathbf{y}^{\ell_{j_\ell}}$ with $\mathbf{y}^{j_i} = \mathbf{y}^{\ell_{j_\ell}}$ but $\mathbf{y}^{i(j_i+1)} \neq \mathbf{y}^{\ell(j_\ell+1)}$. Then the same argument as above can be applied to \mathbf{y}^{j_i} (resp. $\mathbf{y}^{\ell_{j_\ell}}$).

Finally, let $\mathbf{x}^j \in \mathcal{X}'$ be an unresolvable weak \mathcal{X}' -determinism conflict state for two time-series $\mathcal{X}_i = \mathcal{X}'(\mathbf{x}^{i_0}, \mathbf{x}^{i_k})$ and $\mathcal{X}_\ell = \mathcal{X}'(\mathbf{x}^{\ell_0}, \mathbf{x}^{\ell_m})$. We consider the two decompositions $\sigma_{\pi, \lambda}(\mathbf{x}^j, \mathbf{d}^i) = ((\mathbf{y}^{i_1}, \mathbf{r}^{t^{i_1}}), \dots, (\mathbf{y}^{i_{m_i}}, \mathbf{r}^{t^{i_{m_i}}}))$ and $\sigma_{\pi, \lambda}(\mathbf{x}^j, \mathbf{d}^\ell) = ((\mathbf{y}^{\ell_1}, \mathbf{r}^{t^{\ell_1}}), \dots, (\mathbf{y}^{\ell_{m_\ell}}, \mathbf{r}^{t^{\ell_{m_\ell}}}))$ so that $\mathbf{y}^{i_1} = \mathbf{y}^{\ell_1}, \mathbf{y}^{i_2} = \mathbf{y}^{\ell_2}, \dots, \mathbf{y}^{i_j} = \mathbf{y}^{\ell_j}$ but $\mathbf{y}^{i(j+1)} \neq \mathbf{y}^{\ell(j+1)}$. By definition of an unresolvable weak \mathcal{X}' -determinism conflict, such a decomposition always exists. Now again, the same arguments from above can be applied to \mathbf{y}^{i_j} (resp. \mathbf{y}^{ℓ_j}). Thus, there does not exist an \mathcal{X}' -deterministic extended Petri net, which proves this theorem. \square

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 [5,8], all the n -dimensional state vectors $\mathbf{x} \in \mathcal{X}'$ to suitable $(n+a)$ -dimensional vectors

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

The studied extensions $\bar{\mathbf{x}}^j \in \mathbb{N}^{n+a}$ of the states $\mathbf{x}^j \in \mathcal{X}'$ correspond to suitable *labelings* of the experiment graph $\mathcal{D}(\mathcal{X}')$: if $a = 1$, to $(0, 1)$ -labelings, where label i is assigned to node \mathbf{x}^j if $\bar{\mathbf{x}}_{n+1}^j = \mathbf{z}^j = i$ 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). It is already stressed in [5] that not every labeling for the experiment graph $\mathcal{D}(\mathcal{X}')$ is reasonable, as a state $\overline{\mathbf{x}}^k \in \overline{\mathcal{X}'}$ with $\mathbf{x}^k \in \mathcal{X}'_{term}$ might have a successor state, a state $\overline{\mathbf{x}}^j$ 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 [5]:

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

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

>From Condition (i) we can conclude that we have $\mathbf{x} = \text{succ}_{\mathcal{X}'}(\mathbf{x})$ if and only if $\mathbf{x} \in \mathcal{X}'_{term}$. 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 \mathbf{x}^2 , and choose its label as $\mathbf{x}_z^2 = 0$. Due to Condition (ii), $\mathbf{x}_z^5 = 0$ follows. Condition (i) implies that \mathbf{x}^3 (resp. \mathbf{x}^6) must be different from \mathbf{x}^2 (resp. \mathbf{x}^5). Therefore, $\mathbf{x}_z^3 = 1$ and $\mathbf{x}_z^6 = 1$ follows. Since we have $\mathbf{d}^4 = \mathbf{d}^5$, Condition (iii) implies that the only valid labels for \mathbf{x}^0 and \mathbf{x}^7 are 0 and 1, respectively. Condition (ii) shows $\mathbf{x}_z^1 = 0$. Finally, we can choose a label for \mathbf{x}^4 and \mathbf{x}^8 , respectively. However, since $\mathbf{d}^3 = \mathbf{d}^6$, it follows from (iii) that both labels must be equal.

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 \mathbf{x}^j .

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

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

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

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

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

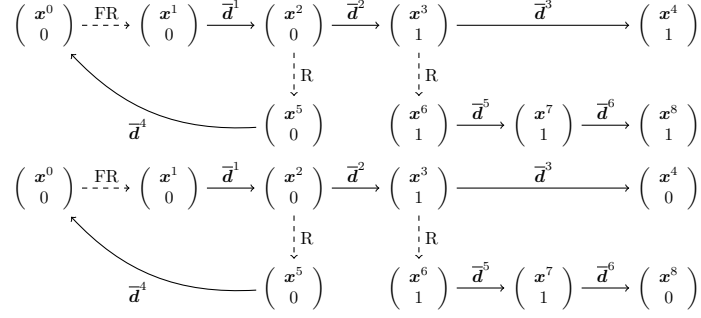


Fig. 4. This figure shows values for additional components resolving the strong \mathcal{X}' -determinism conflicts from Example 2 in Figure 1.

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 $P^+(k) \cup P^-(k) = \{1, \dots, a\}$ 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

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

$$z_{jlpqi^+} \geq 0 \quad \text{for all } i^+ \in P^+(k), \quad (3b)$$

$$z_{jlpqi^-} \leq 0 \quad \text{for all } i^- \in P^-(k), \quad (3c)$$

where $\mathcal{A}_D := \{(\mathbf{x}^j, \mathbf{x}^l), (\mathbf{x}^p, \mathbf{x}^q) \in \mathcal{A}_D \text{ with } \mathbf{x}^j = \mathbf{x}^p, \mathbf{x}^l \neq \mathbf{x}^q\}$. 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 $\mathcal{P}(a)$.

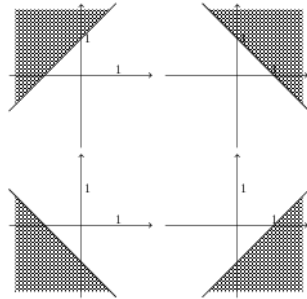


Fig. 5. In this figure the division of (2a) into 2^a subproblems is illustrated within the 2-dimensional space (i.e., $a = 2$). Each of the resulting 4 subproblems has a convex feasible region (highlighted by the dotted regions) whose union corresponds to the feasible region of the original problem.

4 Conclusion

In this work, we give a preprocessing step for a reconstruction algorithm from [6] 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 [4]. 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 $\mathbf{x} \in \mathcal{X}'$ 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}_ℓ have a common state \mathbf{x} but different successor states in each of these sequences (i.e., $\text{succ}_{\mathcal{X}_i}(\mathbf{x}) \neq \text{succ}_{\mathcal{X}_\ell}(\mathbf{x})$) 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. M. Chen and W. Hofestädt. Quantitative Petri net model for gene regulated metabolic networks in the cell. *In Silico Biology*, 3:347–365, 2003.
2. M. Durzinsky, W. Marwan, and A. K. Wagler. 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.
3. M. Durzinsky, W. Marwan, and A. K. Wagler. Reconstruction of extended Petri nets from time-series data by using logical control functions. *Journal of Mathematical Biology*, 66:203–223, 2013.
4. M. Durzinsky, A. K. Wagler, and R. Weismantel. A combinatorial approach to reconstruct Petri nets from experimental data. In Monika Heiner and Adelinde M. Uhrmacher, editors, *CMSB*, volume 5307 of *Lecture Notes in Computer Science*, pages 328–346. Springer, 2008.
5. M. Durzinsky, A. K. Wagler, and R. Weismantel. An algorithmic framework for network reconstruction. *Journal of Theoretical Computer Science*, 412(26):2800–2815, 2011.
6. M. Favre and A. K. Wagler. Reconstructing \mathcal{X}' -deterministic extended Petri nets from experimental time-series data \mathcal{X}' . In *Proceedings of the 4th International Workshop on Biological Processes & Petri Nets*, pages 45–59, 2013.
7. I. Koch and M. Heiner. Petri nets. In B. H. Junker and F. Schreiber, editors, *Biological Network Analysis*, Wiley Book Series on Bioinformatics, pages 139–179, 2007.
8. W. Marwan, A. K. Wagler, and R. Weismantel. A mathematical approach to solve the network reconstruction problem. *Math. Methods of Operations Research*, 67(1):117–132, 2008.
9. W. Marwan, A. K. Wagler, and R. Weismantel. Petri nets as a framework for the reconstruction and analysis of signal transduction pathways and regulatory networks. *Natural Computing*, 10:639–654, 2011.
10. J. W. Pinney, R. D. Westhead, and G. A. McConkey. Petri net representations in systems biology. *Biochem. Soc. Trans.*, 31:1513–1515, 2003.
11. C. Starostzik and W. Marwan. Functional mapping of the branched signal transduction pathway that controls sporulation in *Physarum polycephalum*. *Photochem Photobiol*, 62(5):930–933, 1995.
12. L. M. Torres and A. K. Wagler. Encoding the dynamics of deterministic systems. *Math. Methods of Operations Research*, 73:281–300, 2011.
13. A. K. Wagler. Prediction of network structure. In I. Koch, F. Schreiber, and W. Reisig, editors, *Modeling in Systems Biology*, volume 16 of *Computational Biology*, pages 309–338. Springer London, 2010.
14. A. K. Wagler and J.-T. Wegener. On minimality and equivalence of Petri nets. *Proceedings of Concurrency, Specification and Programming CS&P'2012 Workshop*, 2:382–393, 2012.
15. A. K. Wagler and J.-T. Wegener. On minimality and equivalence of Petri nets. *Fundamenta Informaticae*, 128(1-27), 2013.
16. A. K. Wagler and J.-T. Wegener. Preprocessing for network reconstruction: Feasibility test and handling infeasibility. In M. S. Szczuka, L. Czaja, and M. Kacprzak, editors, *CS&P*, volume 1032 of *CEUR Workshop Proceedings*, pages 434–447. CEUR-WS.org, 2013.