Integrating prior knowledge in automatic network reconstruction

Marie C.F. Favre[1]*, Wolfgang Marwan[2], Annegret K. Wagler[1]
{marie.favre,wagler}@isima.fr, wolfgang.marwan@ovgu.de

[1] Laboratoire d'Informatique, de Modélisation et d'Optimisation des Systèmes (LIMOS, UMR CNRS 6158), Université Blaise Pascal, Clermont-Ferrand, France
[2] Magdeburg Center for Systems Biology (MaCS), Otto-von-Guericke-Universität, Magdeburg, Germany

Abstract. The reconstruction of models from experimental data is a challenging problem due to the inherited complexity of biological systems. We developed an exact, exclusively data-driven approach to reconstruct Petri nets from experimental time-series data. Our approach aims at reconstructing all such netwoks that fit the given experimental data, to provide all possible alternatives of mechanisms behind the experimental observations, which typically results in a large set of solution alternatives. To keep this solution set reasonably small while still guaranteeing its completeness, we firstly generate only Petri nets being minimal in the sense that all other networks fitting the data contain the reconstructed ones. We further aim at avoiding the generation of minimal solutions which are "technically correct" but would be ruled out later during a subsequent verification process to check whether the returned solutions are "biological meaningful" or even contradict well-established biological knowledge. For that, we propose to extent the considered input (beyond the information given with the experimental time-series data) for the reconstruction process and demonstrate with the help of a running example the influence on the generated solution set.

1 Introduction

Systems biology aims at the integrated experimental and theoretical analysis of molecular or cellular networks to achieve a holistic understanding of biological systems and processes. To gain the required insight into the underlying biological processes, experiments are performed and experimental data are interpreted in terms of models. Depending on the biological aim, the type and quality of the available data, different types of mathematical models are used and corresponding reconstruction methods have been developed. Our work is dedicated to Petri nets which turned out to coherently model both static interactions in terms of networks and dynamic processes in terms of state changes, see *e.g.* [7,10].

* This work was funded by the French National Research Agency, the European Commission (Feder funds) and the Région Auvergne within the LabEx IMobS[3].

M. Heiner (Ed.): BioPPN 2014, a satellite event of PETRI NETS 2014,
CEUR Workshop Proceedings Vol. 1159, 2014.

In fact, a network $\mathcal{P} = (P, T, A, w)$ reflects the involved components by places $p \in P$ and their interactions by transitions $t \in T$, linked by weighted directed arcs $(p,t), (t,p) \in A$. Each place $p \in P$ can be marked with an integral number x_p of tokens defining a system state $\boldsymbol{x} \in \mathbb{Z}_+^{|P|}$, *i.e.*, we obtain $\mathcal{X} := \{\boldsymbol{x} \in \mathbb{Z}^{|P|} : x_p \geq 0\}$ 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 $(p,t) \in A$, we denote the set of all such transitions by $T(\boldsymbol{x})$. Switching $t \in T(\boldsymbol{x})$ yields a successor state $\mathrm{succ}(\boldsymbol{x}) = \boldsymbol{x}'$ 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.

Our central question is to reconstruct models of this type from experimental time-series data by means of an exact, exclusively data-driven approach. This approach takes as input a set P of places and discrete time-series data $\mathcal{X}' \subseteq \mathcal{X}$ given by sequences $(\boldsymbol{x}^0; \boldsymbol{x}^1, \ldots, \boldsymbol{x}^k)$ 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. Hence, in contrast to the normally used stochastic simulation, we require that for states where at least two transitions are enabled, the decision between the alternatives is not taken randomly, but a specific transition is selected. Thus, (standard) Petri nets have to be equipped with additional activation rules to force the switching of specific transitions (to reach $\boldsymbol{x}^{j+1}$ from $\boldsymbol{x}^j$), and to prevent all others from switching. For that, different types of additional activation rules are possible.

On the one hand, control-arcs are 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, a transition $t \in T(\boldsymbol{x})$ coupled with a read-arc (resp. an inhibitor-arc) to a place $p \in P$ can switch only if at least $w(p,t)$ tokens (resp. less than $w(p,t)$ tokens) are present in p; we denote by $T_{\mathcal{A}}(\boldsymbol{x})$ the set of all such transitions.

On the other hand, in [9,12,13] priority relations among the transitions of a network are employed to reflect the rate of the corresponding reactions, where the fastest reaction has highest priority and, thus, is taken. In Marwan et al. [9] it is proposed to model such priorities with the help of partial orders $\mathcal{O}$ on the transitions. We call $(\mathcal{P}, \mathcal{O})$ an *(extended) Petri net with priorities*, if $\mathcal{P} = (P, T, \mathcal{A}, w)$ is an (extended) Petri net and $\mathcal{O}$ a priority relation on T. Priorities can prevent enabled transitions from switching: For each state $\boldsymbol{x}$, a transition $t \in T_{\mathcal{A}}(\boldsymbol{x})$ is allowed to switch only if there is no other enabled transition in $T_{\mathcal{A}}(\boldsymbol{x})$ with higher priority; we denote by $T_{\mathcal{A},\mathcal{O}}(\boldsymbol{x})$ the set of all such transitions.

We call $(\mathcal{P}, \mathcal{O})$ $\mathcal{X}'$*-deterministic* if $T_{\mathcal{A},\mathcal{O}}(x)$ contains at most one element for each experimentally observed state $\boldsymbol{x} \in \mathcal{X}'$. Based on earlier results in [3,4,5,9,13], an integrative method to reconstruct all $\mathcal{X}'$-deterministic extended Petri nets with priorities fitting given experimental time-series data $\mathcal{X}'$ was proposed in [6] (see Section 2).

Our approach aims at reconstructing all netwoks of the studied type that fit the given experimental data, to provide all possible alternatives of mechanisms behind the experimentally observed phenomena. Typically, this results in a large set of solution alternatives. To keep this solution set reasonably small while still guaranteeing its completeness, we generate only Petri nets being minimal in the sense that all other networks fitting the data contain the reconstructed ones. Here, we propose a method to insert only minimal sets of control-arcs during the reconstruction process (see Section 2). We further aim at avoiding the generation of minimal solutions which are "technically correct" but would be ruled out later during a subsequent verification process to check whether the returned solutions are "biological meaningful" or even contradict well-established biological knowledge. For that, we extent the considered input by integrating biological prior knowledge (beyond the information given with the experimental time-series data) into the reconstruction process and demonstrate with the help of a running example the influence on the generated solution set (see Section 3). We close with some concluding remarks and lines of future work.

2 Reconstructing extended Petri nets with priorities

We describe the input, the main ideas, and the output of our approach from [6].

Input. A set of components P (standing for proteins, enzymes, genes, receptors or their conformational states, later represented by the set of places) is chosen which is expected to be crucial for the studied phenomenon. If P contains known P-invariants (subsets $P' \subseteq P$ of places where the sum of the number of all tokens on all the places in P' is constant, e.g., different functional states of a cell or conformational states of a molecular complex), they are collected in a set $\mathcal{I}_P$.

To perform an experiment, one first triggers the system in some state $\boldsymbol{x}^0$ (by external stimuli like the exposition to a pathogen), to generate an initial state $\boldsymbol{x}^1$. Then the system's response to the stimulation is observed and the resulting state changes are measured for all considered components at certain time points. This yields a sequence of (discrete or discretized) states $\boldsymbol{x}^j \in \mathbb{Z}^{|P|}$ reflecting the time-dependent response of the system to the stimulation in $\boldsymbol{x}^1$, which typically terminates in a terminal state $\boldsymbol{x}^k$ where no further changes are observed. The corresponding experiment is $\mathcal{X}'(\boldsymbol{x}^1, \boldsymbol{x}^k) = (\boldsymbol{x}^0; \boldsymbol{x}^1, \ldots, \boldsymbol{x}^k)$. Several experiments starting from different initial states in a set $\mathcal{X}'_{ini} \subseteq \mathcal{X}'$, reporting the observed state changes, and ending at different terminal states in a set $\mathcal{X}'_{term} \subseteq \mathcal{X}'$ describe the studied phenomenon, and yield experimental time-series data of the form $\mathcal{X}' = \{\mathcal{X}'(\boldsymbol{x}^1, \boldsymbol{x}^k) : \boldsymbol{x}^1 \in \mathcal{X}'_{ini}, \boldsymbol{x}^k \in \mathcal{X}'_{term}\}$. Thus, the input of the reconstruction approach is given by $(P, \mathcal{I}_P, \mathcal{X}')$.

Example 1. As running example, we will consider experimental biological data from the *light-induced sporulation of Physarum polycephalum* as in [6,13]. In *P. polycephalum* plasmodia, the photoreceptor involved in the control of sporulation *Spo* is a protein which occurs in two stages P_{FR} and P_R. The developmental decision of starving *P. polycephalum* plasmodia to enter the sporulation

pathway is controlled by environmental factors like visible light [11]. 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 after several hours [8]. If P_R is exposed to red light R, it is photo-converted back to the initial state P_{FR} (photoreversion), which prevents sporulation if the red light pulse is given shortly after the far-red pulse, but not if the red pulse is delivered after more than an hour when the phytochrome photoreceptor has had sufficient time to cause the formation of a biochemical downstream signal G that subsequently causes the sporulation of the cell. The changes between the stages P_{FR} and P_R only require fractions of seconds and can be experimentally observed due to a change of color of the phytochrome protein. The experimental setting consists of

$$P = \{FR, R, P_{FR}, P_R, G, Spo\},\ \mathcal{X}'(\boldsymbol{x}^1, \boldsymbol{x}^4) = (\boldsymbol{x}^0; \boldsymbol{x}^1, \boldsymbol{x}^2, \boldsymbol{x}^3, \boldsymbol{x}^4),\ \mathcal{X}'_{ini} = \{\boldsymbol{x}^1, \boldsymbol{x}^5, \boldsymbol{x}^6\},$$
$$\mathcal{I}_P = \{P_{FR}, P_R\},\quad \mathcal{X}'(\boldsymbol{x}^5, \boldsymbol{x}^0) = (\boldsymbol{x}^2; \boldsymbol{x}^5, \boldsymbol{x}^0),\quad \mathcal{X}'_{term} = \{\boldsymbol{x}^4, \boldsymbol{x}^0, \boldsymbol{x}^8\}$$
$$\mathcal{X}'(\boldsymbol{x}^6, \boldsymbol{x}^8) = (\boldsymbol{x}^3; \boldsymbol{x}^6, \boldsymbol{x}^7, \boldsymbol{x}^8),$$

as input for the algorithm, we present all observed states schematically in Fig 1.

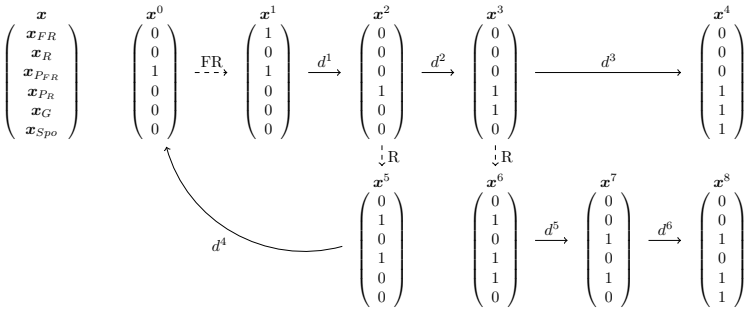


Fig. 1. Experimental time-series data $\mathcal{X}'$ for the light-induced sporulation of *Physarum polycephalum*. The experimental setting uses the set $P = \{FR, R, P_{fr}, P_r, G, Spo\}$ of studied components, observed states are represented by vectors of the form $\boldsymbol{x} = (x_{FR}, x_R, x_{P_{FR}}, x_{P_R}, x_G, x_{Spo})^T$ having 0/1-entries only. Dashed arrows represent a stimulation or perturbation of the system, solid arrows 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 $\boldsymbol{x}^j \in \mathcal{X}'$ there is a unique observed successor state $\mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x}^j) = \boldsymbol{x}^{j+1} \in \mathcal{X}'$. Reproducibility is obviously necessary and can be ensured by preprocessing [15]. Whether a state $\boldsymbol{x}^j \in \mathcal{X}'$ and its observed successor $\mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x}^j) = \boldsymbol{x}^{j+1} \in \mathcal{X}'$ are also consecutive system states depends on the chosen time points to measure the states in $\mathcal{X}'$. In fact, $\boldsymbol{x}^{j+1}$ may be obtained from $\boldsymbol{x}^j$ by a switching sequence of some length, where the intermediate states are not reported in $\mathcal{X}'$. The data $\mathcal{X}'$ are *monotone* if for each pair $(\boldsymbol{x}^j, \boldsymbol{x}^{j+1}) \in \mathcal{X}'$, the values of the elements do not oscillate in the possible intermediate states between $\boldsymbol{x}^j$ and $\boldsymbol{x}^{j+1}$. It was shown in [4] that monotonicity has to be required, too (which is equivalent to demand that all essential responses are indeed reported in $\mathcal{X}'$).

Output. An extended Petri net with priorities $(\mathcal{P}, \mathcal{O})$ with $\mathcal{P} = (P, T, \mathcal{A}, w)$ fits the given data $\mathcal{X}'$ when it is able to perform every observed state change from $\boldsymbol{x}^j \in \mathcal{X}'$ to $\boldsymbol{x}^{j+1} \in \mathcal{X}'$. For that, associate with $\mathcal{P}$ an incidence matrix $M \in \mathbb{Z}^{|P|\times|T|}$ whose rows correspond to the places $p \in P$ and whose columns $M_{\cdot t}$ to the *update vector* $\boldsymbol{r}^t$ of the transitions $t \in T$:

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

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 $\boldsymbol{x}^{j+1}$ from $\boldsymbol{x}^j$ by adding the corresponding columns $M_{\cdot t}$ of M for all $t \in T'$, i.e., $\boldsymbol{x}^j + \sum_{t\in T'} M_{\cdot t} = \boldsymbol{x}^{j+1}$.

T has to contain enough transitions to perform all experimentally observed switching sequences. The underlying standard network $\mathcal{P} = (P, T, A, w)$ is *conformal* with $\mathcal{X}'$ if, for any two consecutive states $\boldsymbol{x}^j, \boldsymbol{x}^{j+1} \in \mathcal{X}'$, the linear equation system $\boldsymbol{x}^{j+1} - \boldsymbol{x}^j = M\boldsymbol{\lambda}$ has an integral solution $\boldsymbol{\lambda} \in \mathbb{N}^{|T|}$ such that $\boldsymbol{\lambda}$ represents 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 $1 \le l \le m$. The extended Petri net $\mathcal{P} = (P, T, \mathcal{A}, w)$ is *catalytic conformal* with $\mathcal{X}'$ if $t^l \in T_{\mathcal{A}}(\boldsymbol{y}^l)$ for each intermediate state $\boldsymbol{y}^l$, 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 consists of all minimal $\mathcal{X}'$-deterministic extended Petri nets $(\mathcal{P}, \mathcal{O})$ (all having the same set P of places as part of the input).

Example 2. We represent in Fig. 3 (page 54) several alternative $\mathcal{X}'$-deterministic extended Petri nets fitting the experimental data $\mathcal{X}'$ from our running example.

We now briefly sketch the reconstruction procedure.

Representing the observed responses. As initial step, extract the observed changes of states from the experimental data. For that, define the set

$$\mathcal{D} := \left\{ \boldsymbol{d}^j = \boldsymbol{x}^{j+1} - \boldsymbol{x}^j \ : \ \boldsymbol{x}^{j+1} = \mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x}^j) \in \mathcal{X}' \right\}.$$

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 $M \in \mathbb{Z}^{|P|\times|T|}$. Due to monotonicity [4], it suffices to represent any $\boldsymbol{d}^j \in \mathcal{D}$ using sign-compatible update vectors from the following set only:

$$\mathrm{Box}(\boldsymbol{d}^j) = \left\{ \boldsymbol{r} \in \mathbb{Z}^{|P|} : \begin{array}{rl} 0 \le r_p \le d_p & \text{if } d_p^j > 0 \\ d_p \le r_p \le 0 & \text{if } d_p^j < 0 \\ r_p = 0 & \text{if } d_p^j = 0 \\ \sum_{p\in P'} r_p = 0 & \forall P' \in \mathcal{I}_P \end{array} \right\} \setminus \{\boldsymbol{0}\}.$$

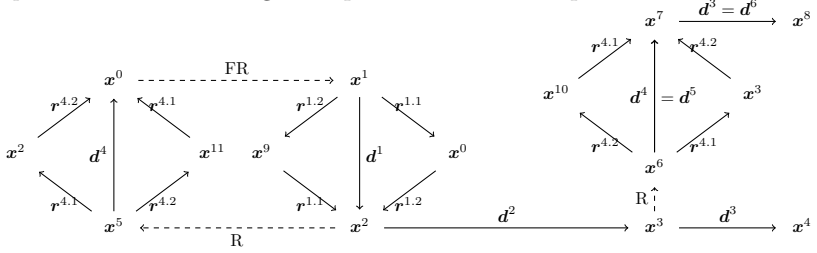
Next, we determine for any $\boldsymbol{d}^j \in \mathcal{D}$, the set $\Lambda(\boldsymbol{d}^j)$ of all integral solutions of

$$\boldsymbol{d}^j = \sum_{\boldsymbol{r}^t \in \mathrm{Box}(\boldsymbol{d}^j)} \lambda_t \boldsymbol{r}^t, \ \lambda_t \in \mathbb{Z}_+, \tag{1}$$

and for each $\lambda \in \Lambda(\boldsymbol{d}^j)$, the (multi-)set $\mathcal{R}(\boldsymbol{d}^j, \lambda) = \{\boldsymbol{r}^t \in \mathrm{Box}(\boldsymbol{d}^j) : \lambda_t \neq 0\}$ of update vectors used for this solution λ. By construction, $\mathrm{Box}(\boldsymbol{d}^j)$ and $\Lambda(\boldsymbol{d}^j)$ are always non-empty since $\boldsymbol{d}^j$ itself is always a solution due to reproducibility [6]. Every permutation $\pi = (\boldsymbol{r}^{t_1}, \ldots, \boldsymbol{r}^{t_m})$ of the elements of a set $\mathcal{R}(\boldsymbol{d}^j, \lambda)$ gives rise to a sequence of intermediate states $\boldsymbol{x}^j = \boldsymbol{y}^1, \boldsymbol{y}^2, ..., \boldsymbol{y}^m, \boldsymbol{y}^{m+1} = \boldsymbol{x}^{j+1}$ with

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

Example 3. For the running example we obtain as sequences



with $\boldsymbol{x}^9 = (1,0,0,1,0,0)^T$, $\boldsymbol{x}^{10} = (0,1,1,0,1,0)^T$ and $\boldsymbol{x}^{11} = (0,1,1,0,0,0)^T$.

To compose all possible standard networks, we have to select exactly one solution $\boldsymbol{\lambda} \in \Lambda(\boldsymbol{d}^j)$ for each $\boldsymbol{d}^j \in \mathcal{D}$ and to take the union of the corresponding sets $\mathcal{R}(\boldsymbol{d}^j, \boldsymbol{\lambda})$ in order to yield the columns $M_{\cdot t} = \boldsymbol{r}^t$ of an incidence matrix M of a conformal network. To ensure that the generated conformal networks can be made $\mathcal{X}'$-deterministic, we proceed as follows.

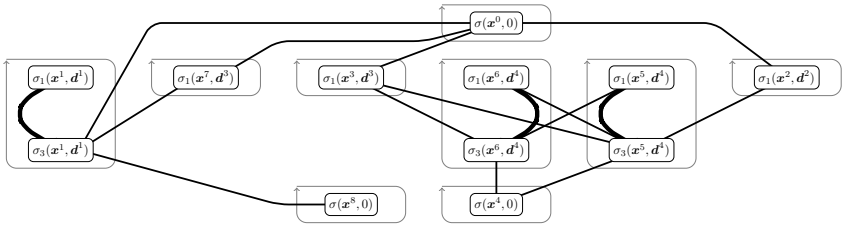
Detecting priority conflicts between sequences. By construction, every sequence $\sigma_{\pi,\boldsymbol{\lambda}}(\boldsymbol{x}^j, \boldsymbol{d}^j)$ induces a priority relation $\mathcal{O}_\sigma$, since it implies which transition t^i is supposed to have highest priority (and thus switches) for every intermediate state $\boldsymbol{y}^i$. To impose valid priority relations $\mathcal{O}$ among all transitions of the reconstructed networks, we have to take conflicts between priority relations $\mathcal{O}_\sigma$ induced by different sequences σ into account. Two sequences σ and σ' are in *priority conflict* if there are update vectors $\boldsymbol{r}^t \neq \boldsymbol{r}^{t'}$ and intermediate states $\boldsymbol{y}, \boldsymbol{y}'$ such that $t, t' \in T(\boldsymbol{y}) \cap T(\boldsymbol{y}')$ and $(\boldsymbol{y}, \boldsymbol{r}^t) \in \sigma$ but $(\boldsymbol{y}', \boldsymbol{r}^{t'}) \in \sigma'$ (since this implies $t > t'$ in $\mathcal{O}_\sigma$ but $t' > t$ in $\mathcal{O}_{\sigma'}$). We have a weak (resp. strong) priority conflict if $\boldsymbol{y} \neq \boldsymbol{y}'$ (resp. $\boldsymbol{y} = \boldsymbol{y}'$) which can (resp. cannot) be resolved by adding control-arcs.

We construct a *priority conflict graph* $\mathcal{G} = (V_D \cup V_{term}, E_D \cup E_W \cup E_S)$ whose nodes correspond to all possible sequences $\sigma_{\pi,\lambda}(\boldsymbol{x}^j, \boldsymbol{d}^j)$ and whose edges reflect weak and strong priority conflicts (WPC and SPC for short):

- V_D contains the sequences $\sigma_{\pi,\boldsymbol{\lambda}}(\boldsymbol{x}^j, \boldsymbol{d}^j)$ for all $\boldsymbol{x}^j \in \mathcal{X}' \setminus \mathcal{X}'_{term}$ and $\boldsymbol{d}^j = \mathrm{succ}_{\mathcal{X}'}(\boldsymbol{x}^j) - \boldsymbol{x}^j$, for all $\boldsymbol{\lambda} \in \Lambda(\boldsymbol{d}^j)$ and all permutations π of $\mathcal{R}(\boldsymbol{d}^j, \boldsymbol{\lambda})$.
- V_{term} contains for all $\boldsymbol{x}^k \in \mathcal{X}'_{term}$ the trivial sequence $\sigma(\boldsymbol{x}^k, \boldsymbol{0})$.
- E_D contains all edges between two sequences σ, σ' stemming from the same difference vector.
- E_S (resp. E_W) reflects all SPCs (resp. WPCs) between sequences σ, σ' stemming from distinct difference vectors.

In $\mathcal{G}$, we generate all node subsets S selecting exactly one sequence $\sigma_{\pi,\lambda}(\boldsymbol{x}^j, \boldsymbol{d}^j)$ per difference vector $\boldsymbol{d}^j \in \mathcal{D}$ such that no SPCs occur between the selected sequences and the nodes in V_{term}. Clearly, a node $\sigma \in V_D$ can never be selected for any solution S if it is in SPC with a terminal state sequence or with all sequences σ' stemming from one difference vector $\boldsymbol{d}^j \in \mathcal{D}$. Removing all such nodes and their incident edges from $\mathcal{G}$ yields the *reduced priority conflict graph* $\mathcal{G}'$. We can show that the sets of suitable selections S obtained in $\mathcal{G}$ and $\mathcal{G}'$ are equal and that there is always at least one feasible selection.

Example 4. We obtain the following reduced priority conflict graph $\mathcal{G}'$ for the running example, where bold edges indicate SPCs and thin edges WPCs.



From $\mathcal{G}'$, we obtain as feasible subsets $S_i = S' \cup S'_i$ with

$$S' = \{\sigma_1(\boldsymbol{x}^2, \boldsymbol{d}^2), \sigma_1(\boldsymbol{x}^3, \boldsymbol{d}^3), \sigma_1(\boldsymbol{x}^7, \boldsymbol{d}^3)\}$$

$$\begin{aligned}
S'_1 &= \{\sigma_1(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma_1(\boldsymbol{x}^5, \boldsymbol{d}^4), \sigma_1(\boldsymbol{x}^6, \boldsymbol{d}^4)\}, & S'_5 &= \{\sigma_3(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma_1(\boldsymbol{x}^5, \boldsymbol{d}^4), \sigma_1(\boldsymbol{x}^6, \boldsymbol{d}^4)\},\\
S'_2 &= \{\sigma_1(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma_1(\boldsymbol{x}^5, \boldsymbol{d}^4), \sigma_3(\boldsymbol{x}^6, \boldsymbol{d}^4)\}, & S'_6 &= \{\sigma_3(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma_1(\boldsymbol{x}^5, \boldsymbol{d}^4), \sigma_3(\boldsymbol{x}^6, \boldsymbol{d}^4)\},\\
S'_3 &= \{\sigma_1(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma_3(\boldsymbol{x}^5, \boldsymbol{d}^4), \sigma_1(\boldsymbol{x}^6, \boldsymbol{d}^4)\}, & S'_7 &= \{\sigma_3(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma_3(\boldsymbol{x}^5, \boldsymbol{d}^4), \sigma_1(\boldsymbol{x}^6, \boldsymbol{d}^4)\},\\
S'_4 &= \{\sigma_1(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma_3(\boldsymbol{x}^5, \boldsymbol{d}^4), \sigma_3(\boldsymbol{x}^6, \boldsymbol{d}^4)\}, & S'_8 &= \{\sigma_3(\boldsymbol{x}^1, \boldsymbol{d}^1), \sigma_3(\boldsymbol{x}^5, \boldsymbol{d}^4), \sigma_3(\boldsymbol{x}^6, \boldsymbol{d}^4)\}.
\end{aligned}$$

Constructing $\mathcal{X}'$-deterministic Petri nets. Every selected subset S corresponds to a conformal standard network $\mathcal{P}_S = (P, T_S, A_S, w)$: we obtain the columns of the incidence matrix M_S of the network by taking the union of all sets $\mathcal{R}(\boldsymbol{d}^j, \boldsymbol{\lambda})$ corresponding to the sequences $\sigma = \sigma_{\pi,\boldsymbol{\lambda}}(\boldsymbol{x}^j, \boldsymbol{d}^j)$ selected by $\sigma \in S$.

Example 5. We apply the method to the feasible sets S_1 and S_4 from Exp. 4 and obtain the standard networks presented in Fig. 2 with $T_{S_1} = \mathcal{D}$ and in Fig. 3 with $T_{S_4} = \{\boldsymbol{d}^1, \boldsymbol{d}^2, \boldsymbol{d}^3, \boldsymbol{r}^{4.1}, \boldsymbol{r}^{4.2}\}$, respectively.

If there are weak priority conflicts $\sigma\sigma' \in E_W$ for nodes $\sigma, \sigma' \in S \cup V_{term}$, denoted by $\mathrm{WPC}(\sigma, \sigma')$, the constructed standard network $\mathcal{P}_S$ needs to be made $\mathcal{X}'$-deterministic by inserting appropriate control-arcs. By [6], a $\mathrm{WPC}(\sigma, \sigma')$ between two sequences σ and σ' involving update vectors $\boldsymbol{r}^t \neq \boldsymbol{r}^{t'}$ and intermediate states $\boldsymbol{y} \neq \boldsymbol{y}'$ with $t, t' \in T(\boldsymbol{y}) \cap T(\boldsymbol{y}')$ such that $(\boldsymbol{y}, \boldsymbol{r}^t) \in \sigma$ but $(\boldsymbol{y}', \boldsymbol{r}^{t'}) \in \sigma'$ has to be resolved by adding appropriate control-arcs that either turn $\boldsymbol{r}^t$ into a transition t which is disabled at $\boldsymbol{y}'$ (then $t > t'$ forces t to switch in $\boldsymbol{y}$ whereas only t' is enabled at $\boldsymbol{y}'$), or vice versa. Let $P(\boldsymbol{y}, \boldsymbol{y}')$ be the set of places where $\boldsymbol{y}$ and $\boldsymbol{y}'$ differ and $\mathrm{CA}(\sigma, \sigma')$ the set of all possible control-arcs that can resolve $\mathrm{WPC}(\sigma, \sigma')$. Then $\mathrm{CA}(\sigma, \sigma')$ partitions into two subsets $\mathrm{CA}_{t>t'}(\sigma, \sigma')$ disabling t at $\boldsymbol{y}'$ containing

- a read-arc $(p,t) \in A_R$ with weight $w(p,t) > y'_p$ $\forall p \in P(\boldsymbol{y}, \boldsymbol{y}')$ with $y_p > y'_p$,
- an inhibitor-arc $(p,t) \in A_I$ with $w(p,t) < y_p$ $\forall p \in P(\boldsymbol{y}, \boldsymbol{y}')$ with $y_p < y'_p$,

and $\mathrm{CA}_{t<t'}(\sigma, \sigma')$ disabling t' at $\boldsymbol{y}$ containing

- a read-arc $(p,t') \in A_R$ with weight $w(p,t') > y_p$ $\forall p \in P(\boldsymbol{y}, \boldsymbol{y}')$ with $y'_p > y_p$,
- an inhibitor-arc $(p,t') \in A_I$ with $w(p,t') < y'_p$ $\forall p \in P(\boldsymbol{y}, \boldsymbol{y}')$ with $y'_p < y_p$.

Remark 1. If one of $\boldsymbol{y}, \boldsymbol{y}'$ is a terminal state, say $\boldsymbol{y}'$, then $\mathrm{CA}_{t<t'}(\sigma, \sigma') = \emptyset$ follows since t has to be disabled at $\boldsymbol{y}'$ and $t > t' = \mathbf{0}$ holds automatically. Moreover, if $\boldsymbol{y} = \boldsymbol{y}'$ then $P(\boldsymbol{y}, \boldsymbol{y}') = \emptyset$ follows which is the reason why SPCs cannot be resolved by adding control-arcs.

Due to [6], inserting one control-arc from $\mathrm{CA}(\sigma, \sigma')$ resolves the $\mathrm{WPC}(\sigma, \sigma')$ in $\mathcal{P}_S$. Here, we further discuss mutual influences of control-arcs in the resulting extended Petri nets as well as the issue of only constructing minimal catalytic conformal networks.

On the one hand, inserting a control-arc $(p,t) \in \mathrm{CA}(\sigma, \sigma')$ in $\mathcal{P}_S$ might disable t at a state in another sequence $\sigma'' \in S \setminus \sigma, \sigma'$ where t is supposed to switch. In this case, (p,t) has to be removed from $\mathrm{CA}(\sigma, \sigma')$, resulting in a reduced set $\mathrm{CA}_S(\sigma, \sigma')$. On the other hand, one control-arc may resolve several WPCs in $\mathcal{P}_S$ if the corresponding sets $\mathrm{CA}_S(\sigma, \sigma')$ intersect.

Therefore, we propose the following consideration: Introduce one variable $z_{(p,t)} \in \{0,1\}$ for each possible control-arc $(p,t) \in \mathrm{CA}_S(\sigma, \sigma')$ for all WPCs in $\mathcal{P}_S$. Construct a 0/1-matrix A_S whose columns correspond to all those variables (resp. control-arcs) and whose rows encode the incidence vectors of the sets $\mathrm{CA}_S(\sigma, \sigma')$ for all WPCs in $\mathcal{P}_S$. Then any 0/1-solution $\boldsymbol{z}$ of the system $A_S \boldsymbol{z} \geq \mathbf{1}$ encodes a suitable set of control-arcs resolving all WPCs in $\mathcal{P}_S$ and, thus, a *hitting set* or *cover* of A_S. By [14], we are only interested in finding minimal models fitting $\mathcal{X}'$, where minimality is interpreted in the sense that all non-minimal models contain another one also fitting the data. We can show that using non-minimal covers of A_S yields non-minimal extended Peri nets but that we need all of them for the sake of completeness, which corresponds to determining the so-called blocker $b(A_S)$ of A_S.

Example 6. We list all WPCs between sequences in our running example:

WPC1 between	$\sigma_1(\boldsymbol{x}^2, \boldsymbol{d}^2)$	and	$\sigma(\boldsymbol{x}^0, \mathbf{0})$	due to	$\boldsymbol{d}^2, \mathbf{0}$	$\in$	$T(\boldsymbol{x}^0) \cap T(\boldsymbol{x}^2)$
WPC2 between	$\sigma_1(\boldsymbol{x}^3, \boldsymbol{d}^3)$	and	$\sigma(\boldsymbol{x}^0, \mathbf{0})$	due to	$\boldsymbol{d}^3, \mathbf{0}$	$\in$	$T(\boldsymbol{x}^0) \cap T(\boldsymbol{x}^3)$
WPC3 between	$\sigma_1(\boldsymbol{x}^7, \boldsymbol{d}^3)$	and	$\sigma(\boldsymbol{x}^0, \mathbf{0})$	due to	$\boldsymbol{d}^3, \mathbf{0}$	$\in$	$T(\boldsymbol{x}^0) \cap T(\boldsymbol{x}^7)$
WPC4 between	$\sigma_3(\boldsymbol{x}^1, \boldsymbol{d}^1)$	and	$\sigma_1(\boldsymbol{x}^7, \boldsymbol{d}^3)$	due to	$\boldsymbol{d}^3, \boldsymbol{r}^{1.2}$	$\in$	$T(\boldsymbol{x}^1) \cap T(\boldsymbol{x}^7)$
WPC5 between	$\sigma_3(\boldsymbol{x}^1, \boldsymbol{d}^1)$	and	$\sigma(\boldsymbol{x}^0, \mathbf{0})$	due to	$\boldsymbol{r}^{1.2}, \mathbf{0}$	$\in$	$T(\boldsymbol{x}^0) \cap T(\boldsymbol{x}^1)$
WPC6 between	$\sigma_3(\boldsymbol{x}^1, \boldsymbol{d}^1)$	and	$\sigma(\boldsymbol{x}^8, \mathbf{0})$	due to	$\boldsymbol{r}^{1.2}, \mathbf{0}$	$\in$	$T(\boldsymbol{x}^8) \cap T(\boldsymbol{x}^1)$
WPC7 between	$\sigma_1(\boldsymbol{x}^2, \boldsymbol{d}^2)$	and	$\sigma_3(\boldsymbol{x}^5, \boldsymbol{d}^4)$	due to	$\boldsymbol{d}^2, \boldsymbol{r}^{4.2}$	$\in$	$T(\boldsymbol{x}^2) \cap T(\boldsymbol{x}^5)$
WPC8 between	$\sigma_3(\boldsymbol{x}^5, \boldsymbol{d}^4)$	and	$\sigma_1(\boldsymbol{x}^3, \boldsymbol{d}^3)$	due to	$\boldsymbol{d}^3, \boldsymbol{r}^{4.2}$	$\in$	$T(\boldsymbol{x}^3) \cap T(\boldsymbol{x}^5)$
WPC9 between	$\sigma_3(\boldsymbol{x}^5, \boldsymbol{d}^4)$	and	$\sigma(\boldsymbol{x}^4, \mathbf{0})$	due to	$\boldsymbol{r}^{4.2}, \mathbf{0}$	$\in$	$T(\boldsymbol{x}^4) \cap T(\boldsymbol{x}^5)$
WPC10 between	$\sigma_3(\boldsymbol{x}^6, \boldsymbol{d}^4)$	and	$\sigma_1(\boldsymbol{x}^3, \boldsymbol{d}^3)$	due to	$\boldsymbol{d}^3, \boldsymbol{r}^{4.2}$	$\in$	$T(\boldsymbol{x}^3) \cap T(\boldsymbol{x}^6)$
WPC11 between	$\sigma_3(\boldsymbol{x}^6, \boldsymbol{d}^4)$	and	$\sigma(\boldsymbol{x}^4, \mathbf{0})$	due to	$\boldsymbol{r}^{4.2}, \mathbf{0}$	$\in$	$T(\boldsymbol{x}^4) \cap T(\boldsymbol{x}^6)$
WPC12 between	$\sigma_1(\boldsymbol{x}^5, \boldsymbol{d}^4)$	and	$\sigma_3(\boldsymbol{x}^6, \boldsymbol{d}^4)$	due to	$\boldsymbol{d}^4, \boldsymbol{r}^{4.2}$	$\in$	$T(\boldsymbol{x}^5) \cap T(\boldsymbol{x}^6)$
WPC13 between	$\sigma_3(\boldsymbol{x}^5, \boldsymbol{d}^4)$	and	$\sigma_1(\boldsymbol{x}^6, \boldsymbol{d}^4)$	due to	$\boldsymbol{d}^4, \boldsymbol{r}^{4.2}$	$\in$	$T(\boldsymbol{x}^5) \cap T(\boldsymbol{x}^6)$

We obtain the following control-arcs to resolve WPCs between sequences:

$$\begin{aligned}
\mathrm{CA(WPC1)} &= \{(P_{FR},\boldsymbol{d}^2)\in A_I,(P_R,\boldsymbol{d}^2)\in A_R\},\\
\mathrm{CA(WPC2)} &= \{(P_{FR},\boldsymbol{d}^3)\in A_I,(P_R,\boldsymbol{d}^3)\in A_R,(G,\boldsymbol{d}^3)\in A_R\},\\
\mathrm{CA(WPC3)} &= \{(G,\boldsymbol{d}^3)\in A_R\},\\
\mathrm{CA(WPC4)} &= \{(FR,\boldsymbol{d}^3)\in A_I,(G,\boldsymbol{d}^3)\in A_R,(FR,\boldsymbol{r}^{1.2})\in A_R,(G,\boldsymbol{r}^{1.2})\in A_I\},\\
\mathrm{CA(WPC5)} &= \{(FR,\boldsymbol{r}^{1.2})\in A_R\},\\
\mathrm{CA(WPC6)} &= \{(Spo,\boldsymbol{r}^{1.2})\in A_I,(G,\boldsymbol{r}^{1.2})\in A_I\},\\
\mathrm{CA(WPC7)} &= \{(R,\boldsymbol{r}^{4.2})\in A_R,(R,\boldsymbol{d}^2)\in A_I\},\\
\mathrm{CA(WPC8)} &= \{(R,\boldsymbol{r}^{4.2})\in A_R,(G,\boldsymbol{r}^{4.2})\in A_I,(R,\boldsymbol{d}^3)\in A_I,(G,\boldsymbol{d}^3)\in A_R\},\\
\mathrm{CA(WPC9)} &= \{(R,\boldsymbol{r}^{4.2})\in A_R,(Spo,\boldsymbol{r}^{4.2})\in A_I,(G,\boldsymbol{r}^{4.2})\in A_I\},\\
\mathrm{CA(WPC10)} &= \{(R,\boldsymbol{d}^3)\in A_I,(R,\boldsymbol{r}^{4.2})\in A_R\},\\
\mathrm{CA(WPC11)} &= \{(R,\boldsymbol{r}^{4.2})\in A_R,(Spo,\boldsymbol{r}^{4.2})\in A_I\},\\
\mathrm{CA(WPC12)} &= \{(G,\boldsymbol{d}^4)\in A_I,(G,\boldsymbol{r}^{4.2})\in A_R\},\\
\mathrm{CA(WPC13)} &= \{(G,\boldsymbol{d}^4)\in A_R,(G,\boldsymbol{r}^{4.2})\in A_I\}.
\end{aligned}$$

The following reductions of sets of possible control-arcs are necessary: for all standard networks $\mathcal{P}_{S_i}$, we have $\sigma_1(\boldsymbol{x}^3,\boldsymbol{d}^3)$ and $\sigma_1(\boldsymbol{x}^7,\boldsymbol{d}^3)$ selected simultaneously, which both are in WPC with $\sigma(\boldsymbol{x}^0,\boldsymbol{0})$, see WPC2 and WPC3. To resolve WPC2, we have $\mathrm{CA(WPC2)}=\{(P_{FR},\boldsymbol{d}^3)\in\mathcal{A}_I,(P_R,\boldsymbol{d}^3)\in\mathcal{A}_R,(G,\boldsymbol{d}^3)\in\mathcal{A}_R\}$.

However, $(P_{FR},\boldsymbol{d}^3)\in\mathcal{A}_I$ and $(P_R,\boldsymbol{d}^3)\in\mathcal{A}_R$ do not only disable $\boldsymbol{d}^3$ at $\boldsymbol{x}^0$, but also $\boldsymbol{d}^3$ at $\boldsymbol{x}^7$ (due to $\boldsymbol{x}^0_{P_{FR}}=\boldsymbol{x}^7_{P_{FR}}=1$ and $\boldsymbol{x}^0_{P_R}=\boldsymbol{x}^7_{P_R}=0$). Since $\boldsymbol{d}^3$ is supposed to switch at $\boldsymbol{x}^7$, we obtain $\mathrm{CA}_{S_i}\mathrm{(WPC2)}=\{(G,\boldsymbol{d}^3)\in\mathcal{A}_R\}$ as reduced set of possible control-arcs to resolve WPC2 in all networks $\mathcal{P}_{(S_i)}$. Similarly, $(G,\boldsymbol{r}^{4.2})\in\mathcal{A}_I$ has to be excluded from CA(WPC8) and CA(WPC9) in $\mathcal{P}_{S_4}$ and $\mathcal{P}_{S_8}$ as otherwise $\boldsymbol{r}^{4.2}$ would be disabled at $\boldsymbol{x}^6$ where it is supposed to switch by $\sigma_3(\boldsymbol{x}^6,\boldsymbol{d}^4)\in S_4,S_8$.

For the feasible set S_4, we obtain as matrix A_{S_4}:

	$(\mathrm{P}_{FR},\boldsymbol{d}^2)\in A_I$	$(\mathrm{P}_R,\boldsymbol{d}^3)\in A_R$	$(\mathrm{G},\boldsymbol{d}^3)\in A_R$	$(\mathrm{R},\boldsymbol{r}^{4.2})\in A_R$	$(\mathrm{R},\boldsymbol{d}^2)\in A_I$	$(\mathrm{R},\boldsymbol{d}^3)\in A_I$	$(\mathrm{Spo},\boldsymbol{r}^{4.2})\in A_I$
WPC1	X	X					
WPC2			X				
WPC3			X				
WPC7				X	X		
WPC8			X	X		X	
WPC9				X			X
WPC10				X		X	
WPC11				X			X

The blocker $b(A_{S_4})$ contains four minimal covers of A_{S_4} which correspond to the different sets of control-arcs in the four extended Petri nets depicted in Fig. 3, all arising from the standard network $\mathcal{P}_{S_4}$.

For S_1, the matrix A_{S_1} contains the first 3 rows and columns of A_{S_4}, the blocker $b(A_{S_1})$ contains two minimal covers of A_{S_1} which correspond to the control-arcs in the two extended Petri nets in Fig. 2 arising from $\mathcal{P}_{S_1}$.

Note that $b(A_S)$ is non-empty if and only if none of the sets $\mathrm{CA}_S(\sigma,\sigma')$ is empty. We can show that there is at least one catalytic conformal network for any given $\mathcal{X}'$. All catalytic conformal extended Petri nets $\mathcal{P}_{S,P'}=(P,T_S,\mathcal{A}_{S,P'},w)$ based on $\mathcal{P}_S$ can be made $\mathcal{X}'$-deterministic by taking all the priorities $\mathcal{O}_\sigma$ for all $\sigma\in S$, where $\mathcal{O}_\sigma$ is defined by $\mathcal{O}_\sigma=\left\{t_i>t:t\in T_{\mathcal{A}_{S,P'}}(\boldsymbol{y}^i)\setminus t_i,1\le i\le m\right\}$. By construction, there are no priority conflicts in the extended network $\mathcal{P}_{S,P'}$ between $\mathcal{O}_\sigma$ and $\mathcal{O}_{\sigma'}$ for any $\sigma,\sigma'\in S$, thus we obtain the studied partial order

$$\mathcal{O}_{S,P'}=\bigcup_{\sigma\in S}\mathcal{O}_\sigma.$$

This finally implies:

Theorem 1. *Every extended network $\mathcal{P}_{S,P'} = (P, T_S, \mathcal{A}_{S,P'}, w)$ together with the partial order $\mathcal{O}_{S,P'}$ is an $\mathcal{X}'$-deterministic extended Petri net, and there are no other minimal extended Petri nets with priorities fitting the given data $\mathcal{X}'$.*

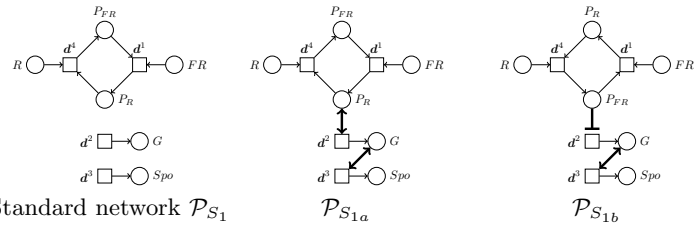


Fig. 2. Standard network $\mathcal{P}_{S_1} = (P, T_{S_1}, A_{S_1}, w)$ from solution S_1 and the two catalytic conformal extended Petri nets resulting from $\mathcal{P}_{S_1}$.

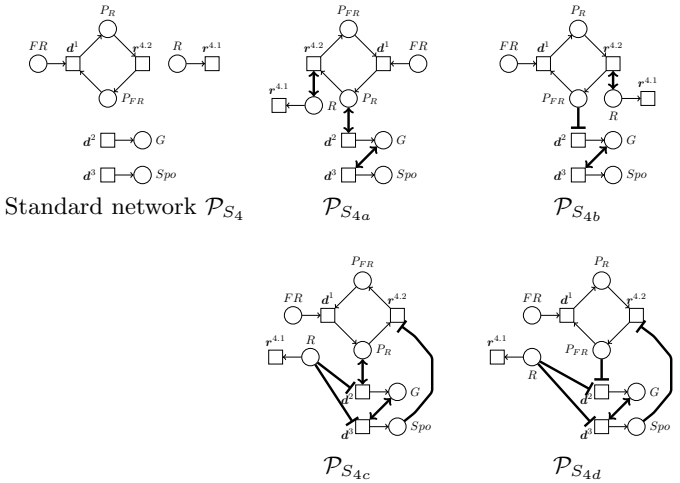


Fig. 3. Standard network $\mathcal{P}_{S_4} = (P, T_{S_4}, A_{S_4}, w)$ from solution S_4 and the four catalytic conformal extended Petri nets resulting from $\mathcal{P}_{S_4}$.

Example 7. For the two extended Petri nets based on $\mathcal{P}_{S_1}$ in Fig. 2, no priorities are needed to obtain $\mathcal{X}'$-deterministic extended Petri nets $(\mathcal{P}_{S_{1a}}, \emptyset)$ and $(\mathcal{P}_{S_{1b}}, \emptyset)$.

For the four extended Petri nets in Fig. 3 based on $\mathcal{P}_{S_4}$, the priority relation $\mathcal{O}_4 = \{(\boldsymbol{r}^{4.2} > \boldsymbol{d}^2)\}$ is required for $\mathcal{P}_{S_{4a}}$ and $\mathcal{P}_{S_{4b}}$, whereas $\mathcal{O}_4 = \{(\boldsymbol{d}^2 > \boldsymbol{r}^{4.2}), (\boldsymbol{d}^3 > \boldsymbol{r}^{4.2})\}$ is required for $\mathcal{P}_{S_{4c}}$ and $\mathcal{P}_{S_{4d}}$, to obtain $\mathcal{X}'$-deterministic extended Petri nets.

In total, the complete solution set contains 66 minimal $\mathcal{X}'$-deterministic extended Petri nets with priorities, see Table 1.

3 Integrating prior biological knowledge

3.1 Indecomposability of difference vectors

In the step *Representing the observed responses*, the set of potential update vectors which might constitute the incidence matrices of the networks are considered. Hereby, for each $\boldsymbol{d}^j \in \mathcal{D}$, the set $\mathrm{Box}(\boldsymbol{d}^j)$ contains only sign-compatible vectors due to monotonicity (which avoids homogeneous solutions in (1) according to minimality) and takes P-invariants into account (which avoids infeasible intermediate states according to prior biological knowledge). In some cases, one could restrict $\mathrm{Box}(\boldsymbol{d}^j)$ further, e.g., if

- $\boldsymbol{d}^j$ exactly corresponds to a well-known biochemical reaction (including the correct stoichiometry) or to a well-known mechanism (that a certain trigger is detected by a suitable receptor),
- experiments have shown that subsets of the input components of $\boldsymbol{d}^j$ do not lead to the observed response,
- $\boldsymbol{d}^j$ is treated as black box-like reaction where only input and output matter, but not the intermediate mechanism due to the chosen level of detail.

In such cases, we propose to exclude the corresponding response $\boldsymbol{d}^j \in \mathcal{D}$ from decomposition and, instead, just define $\mathrm{Box}(\boldsymbol{d}^j) := \{\boldsymbol{d}^j\}$ in accordance with the existing knowledge.

Example 8. Since the light-dependent reactions of the photoreceptor are so much faster than the subsequent processes that are considered in the reconstruction process, the difference vector describing the phytochrome photoconversion will not be decomposed into different reaction vectors.

Thus, for the difference vectors $\boldsymbol{d}^1$, $\boldsymbol{d}^4$ and $\boldsymbol{d}^5$ only the canonical sequences $\sigma_1(\boldsymbol{x}^1, \boldsymbol{d}^1)$, $\sigma_1(\boldsymbol{x}^5, \boldsymbol{d}^4)$ and $\sigma_1(\boldsymbol{x}^6, \boldsymbol{d}^4)$ remain in the priority conflict graph, and S_1 remains as only possible selection. Accordingly, the total number of solutions reduces from 66 to the 2 presented in Fig. 2, see Table 1.

3.2 Treating equal difference vectors in the same way

In the step *Detecting priority conflicts between sequences*, all observed responses $\boldsymbol{d}^j \in \mathcal{D}$ are treated independently from each other, so far. If, however, two difference vectors $\boldsymbol{d}^i, \boldsymbol{d}^j \in \mathcal{D}$ are equal, we clearly have $\mathrm{Box}(\boldsymbol{d}^i) = \mathrm{Box}(\boldsymbol{d}^j)$ and, thus, $\Lambda(\boldsymbol{d}^i) = \Lambda(\boldsymbol{d}^j)$. Here, it is natural to require that both $\boldsymbol{d}^i, \boldsymbol{d}^j$ are decomposed in the same way (i.e., by the same $\lambda \in \Lambda(\boldsymbol{d}^i) = \Lambda(\boldsymbol{d}^j)$) and that the involved reactions are applied in the same order (i.e., by the same permutation π of the elements in the sets $\mathcal{R}(\boldsymbol{d}^i, \lambda) = \mathcal{R}(\boldsymbol{d}^j, \lambda)$) to obtain the same transitions (with equal control-arcs and priorities) in both cases. Indeed, using the same

- $\lambda \in \Lambda(\boldsymbol{d}^i) = \Lambda(\boldsymbol{d}^j)$ corresponds to the fact that the same subset of molecules involved in a reaction will not interact according to different mechanisms,

– permutation π of the elements in $\mathcal{R}(\boldsymbol{d}^i, \lambda) = \mathcal{R}(\boldsymbol{d}^j, \lambda)$ corresponds to the fact that the order in which the reactions are applied reflects the relative rates of the reactions in $\mathcal{R}(\boldsymbol{d}^i, \lambda) = \mathcal{R}(\boldsymbol{d}^j, \lambda)$, so the same relation between reaction rates shall lead to the same priorities within the resulting sequences.

We call two sequences $\sigma_{\pi,\boldsymbol{\lambda}}(\boldsymbol{x}^i, \boldsymbol{d}^i)$ and $\sigma'_{\pi,\boldsymbol{\lambda}}(\boldsymbol{x}^j, \boldsymbol{d}^j)$ *twin sequences* if $\boldsymbol{d}^i = \boldsymbol{d}^j$ and the same $\boldsymbol{\lambda}$ and π has been used. To force that twin sequences are always selected together, we propose to add strong priority conflicts between all other sequences stemming from a pair $\boldsymbol{d}^i, \boldsymbol{d}^j$ of equal difference vectors while creating the priority conflict graph, since no two sequences in strong priority conflict are selected for the same network.

Example 9. In our running example, we have $\boldsymbol{d}^3 = \boldsymbol{d}^6$ and $\boldsymbol{d}^4 = \boldsymbol{d}^5$. The latter vectors can be decomposed in different ways, among the resulting sequences we have $\sigma_1(\boldsymbol{x}^5, \boldsymbol{d}^4)$, $\sigma_1(\boldsymbol{x}^6, \boldsymbol{d}^5)$ and $\sigma_3(\boldsymbol{x}^5, \boldsymbol{d}^4)$, $\sigma_3(\boldsymbol{x}^6, \boldsymbol{d}^5)$ as pairs of twin sequences. Forcing to select these pairs together rules out the four selections S_2, S_3, S_6, S_7 so that only S_1, S_4, S_5, S_8 remain as possible selections. Accordingly, the total number of solutions reduces from 66 to 18, as reported in Table 1.

3.3 Knowledge on relative reaction rates

In the step *Constructing $\mathcal{X}'$-deterministic Petri nets*, to resolve a $\mathrm{WPC}(\sigma, \sigma')$ between σ, σ' involving update vectors $\boldsymbol{r}^t \neq \boldsymbol{r}^{t'}$ and intermediate states $\boldsymbol{y} \neq \boldsymbol{y}'$, either transition t has to be disabled at $\boldsymbol{y}'$ or transition t' at $\boldsymbol{y}$, while the decision between t and t' on the other state can be handled by a priority. Here, prior knowledge about the relative reaction rates of t and t' (e.g. gained from the time-scales during the experiments) could help to decide whether $t > t'$ or $t < t'$ better reflects the reality, and than choose between control-arcs either from $\mathrm{CA}_{t>t'}(\sigma, \sigma')$ or from $\mathrm{CA}_{t'>t}(\sigma, \sigma')$.

So far, this idea is already applied for WPCs involving a terminal state: if $\boldsymbol{y}'$ is a terminal state and $\sigma' = \sigma(\boldsymbol{y}', \boldsymbol{0})$ its trivial sequence, then $t > \boldsymbol{0}$ holds automatically at $\boldsymbol{y}$, but t has to be disabled at $\boldsymbol{y}'$ using control-arcs from $\mathrm{CA}_{t>\boldsymbol{0}}(\sigma, \sigma')$ whereas $\mathrm{CA}_{t<\boldsymbol{0}}(\sigma, \sigma')$ is empty.

This idea can be generalized to any $\mathrm{WPC}(\sigma, \sigma')$ where the time-scale of the corresponding experimental observations clearly differs in order to deduce the correct priority $t > t'$ or $t < t'$. In such cases, we propose to reduce $\mathrm{CA}(\sigma, \sigma')$ accordingly either to $\mathrm{CA}_{t>t'}(\sigma, \sigma')$ or to $\mathrm{CA}_{t'>t}(\sigma, \sigma')$.

Example 10. In our running example, we have WPC1, WPC2, WPC3, WPC5, WPC6, WPC9, WPC11 involving a terminal state; the according reductions of the sets $\mathrm{CA}(\sigma, \sigma')$ to $\mathrm{CA}_{t>\boldsymbol{0}}(\sigma, \sigma')$ are already applied in Exp. 6.

Moreover, WPC4, WPC7, WPC8, WPC10 involve reactions with clearly different time-scales during the experimental observations:

– $\boldsymbol{d}^1$ and $\boldsymbol{d}^4 = \boldsymbol{d}^5$ need only milliseconds to occur,
– $\boldsymbol{d}^2$ needs about 1 hour to occur, and
– $\boldsymbol{d}^3 = \boldsymbol{d}^6$ need at least 10 hours.

Accordingly, we can reduce the sets $\mathrm{CA}(\sigma, \sigma')$ as follows:

- due to $\boldsymbol{r}^{1.2} > \boldsymbol{d}^3$, for WPC4 only $(FR, \boldsymbol{r}^{1.2}) \in A_R$ and $(G, \boldsymbol{r}^{1.2}) \in A_I$ remain;
- due to $\boldsymbol{r}^{4.2} > \boldsymbol{d}^2$, for WPC7 only $(R, \boldsymbol{r}^{4.2}) \in A_R$ remains;
- due to $\boldsymbol{r}^{4.2} > \boldsymbol{d}^3$, for WPC8 only $(R, \boldsymbol{r}^{4.2}) \in A_R$ and $(G, \boldsymbol{r}^{4.2}) \in A_I$ remain, while for WPC10 only $(R, \boldsymbol{r}^{4.2}) \in A_R$ is left.

At least one of these WPCs occurs in the standard networks coming from the selected sets $S_2 - S_4, S_6 - S_8$. Note that in the two remaining WPCs the time-scale of the involved responses is equal (see Exp. 6). Consequently, the number of extended Petri nets decreases from 66 to 36 as reported in Table 1.

4 Discussion

The subject of this paper was an approach from [6] that aims at reconstructing all $\mathcal{X}'$-deterministic extended Petri nets that fit given experimental data $\mathcal{X}'$, to provide all possible alternatives of mechanisms behind the experimentally observed phenomena. This typically results in a large set of solution alternatives. To keep this solution set reasonably small while still guaranteeing its completeness, we firstly generate only Petri nets being minimal in the sense that all other networks fitting the data contain the reconstructed ones. In the presented approach, the minimality concept is applied twice:

- *monotone data*: using only sign-compatible vectors in $\mathrm{Box}(\boldsymbol{d}^j)$ avoids homogeneous solutions during the decomposition of $(\boldsymbol{d}^j)$ (and superflous transitions in the networks), see [4].
- *minimal hitting sets*: we here proposed to use only minimal sets of control-arcs to resolve all weak priority conflicts in a standard network which avoids unnecessary control-arcs (and artificial dependencies), see Section 2.

This ensures that the presented approach exactly generates all minimal extended Petri nets with priorities (Theorem 1). We further avoid generating minimal solutions which are "technically correct" but would be ruled out later during a subsequent verification process to check whether the returned solutions are "biological meaningful" or even contradict well-established biological knowledge as in [2]. For that, we extent the considered input by integrating prior biological knowledge beyond the information given with the experimental time-series data into the reconstruction process. We propose to integrate prior knowledge in the following way:

- *P-invariants*: helps to obtain feasible intermediate states in all sequences which avoids the generation of solutions contradicting known facts, see [13].
- *indecomposable difference vectors*: help to keep already known subnetworks or mechanisms, see Section 3.1.
- *treating equal difference vectors in the same way*: helps to keep consistency in the interpretation of experimental observations, see Section 3.2.
- *obeying terminal states and reaction rates*: helps to chose meaningful priorities and to avoid artificial control-arcs, see Section 3.3.

So far, the algorithmic procedure due to [6] takes as input $(P, I_P, \mathcal{X}')$ where the list of P-invariants and the monotonicity of $\mathcal{X}'$ are already used in the first step *Decomposing difference vectors* to determine $\mathrm{Box}(\boldsymbol{d}^j)$. To integrate further knowledge in the reconstruction procedure to force the algorithm to make "the right decisions" in some intermediate steps, we suggest, based on the previous discussions, to extend the input to $(P, I_P, \mathcal{X}', \mathcal{D}_{in}, \mathcal{D}_{eq}, \mathcal{O}_D)$ where

- $\mathcal{D}_{in}$ contains all indecomposable difference vectors (for that, carefully select them according to the before mentioned critera, e.g., to preserve known subnetworks or mechanisms or to take a certain level of detail into account);
- $\mathcal{D}_{eq}$ contains all pairs of equal difference vectors that shall be treated in the same way (for that, only chose equal difference vectors where also the time elapsed during the experimental observation was equal);
- $\mathcal{O}_D$ contains information about the reaction rates between difference vectors (for that, only impose priorities for sufficiently different rates according to clearly different time scales during the experiments).

Example 11. The three examples from Section 3 can be interpreted as follows:

- Exp. 8 shows the result taking $(P, I_P, \mathcal{X}', \mathcal{D}_{in} = \{\boldsymbol{d}^1, \boldsymbol{d}^4, \boldsymbol{d}^5\}, \emptyset, \emptyset)$ as input;
- Exp. 9 shows the result with $(P, I_P, \mathcal{X}', \emptyset, \mathcal{D}_{eq} = \{\boldsymbol{d}^3 = \boldsymbol{d}^6, \boldsymbol{d}^4 = \boldsymbol{d}^5\}, \emptyset)$;
- Exp. 10 the result with $(P, I_P, \mathcal{X}', \emptyset, \emptyset, \mathcal{O}_D = \{\boldsymbol{d}^1, \boldsymbol{d}^4, \boldsymbol{d}^5 > \boldsymbol{d}^2 > \boldsymbol{d}^3, \boldsymbol{d}^6\})$.

Whereas the first setting reduces the number of solution alternatives to 2, the combination of the two latter scenarios reduces the number of solution alternatives to 12, see Table 1 below.

	$S1$	$S2$	$S3$	$S4$	$S5$	$S6$	$S7$	$S8$	TOTAL
minimal	2	8	8	4	4	16	16	8	66
$\mathcal{D}_{in} \neq \emptyset$	2	0	0	0	0	0	0	0	2
$\mathcal{D}_{eq} \neq \emptyset$	2	0	0	4	4	0	0	8	18
$\mathcal{O}_D \neq \emptyset$	2	4	4	2	4	8	8	4	36
$\mathcal{D}_{eq}, \mathcal{O}_D \neq \emptyset$	2	0	0	2	4	0	0	4	12

Table 1. Number of solutions depending on different input settings.

To conclude, we notice that providing indecomposable difference vectors has the largest impact on the solution set. However, even if no indecomposable difference vectors can be identified, treating equal difference vectors in the same way and deducing relative reaction rates from the time-scale of the experimental observations leads to a substantial reduction of the solution set, keeping only "biological meaningful" network alternatives.

The further goal is to provide an implementation for the presented reconstruction method, including the option of integrating prior knowledge as additional input during the reconstruction. For that, we will use Answer Set Programming as done for the reconstruction of standard networks in [1]. Finally, we plan to apply the presented reconstruction approach to different biological experimental data. We expect an important impact of Automatic Network Reconstruction

on the integrated experimental and theoretical analysis of biological systems towards their holistic understanding, since computational models derived from experimental data by our exact, exclusively data-driven approach have predictive ability due to completeness of the solution set guaranteed by mathematical proofs.

References

1. M. Durzinsky, W. Marwan, M. Ostrowski, T. Schaub, and A. K. Wagler. Automatic network reconstruction using ASP. *Th. and Pract. of Logic Progr.*, 11:749–66, 2011.
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. *Lecture Notes in Comp. Sc. (special Issue CMSP 2008)*, 5307:328–346, 2008.
5. M. Durzinsky, A. K. Wagler, and R. Weismantel. An algorithmic framework for network reconstruction. *J.Theor. 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 $\mathcal{X}'$ (extended abstract). *CEUR Workshop Proceedings (Special Issue BioPPN 2013)*, 988:45–59, 2013.
7. I. Koch and M. Heiner. Petri nets. In B. H. Junker, F. Schreiber, editor, *Biological Network Analysis*, Wiley Book Series on Bioinformatics, pages 139–179, 2007.
8. T. Lamparter and W. Marwan. Spectroscopic detection of a phytochrome-like photoreceptor in the myxomycete physarum polycephalum and the kinetic mechanism for the photocontrol of sporulation. *Photochem Photobiol*, 73:697–702, 2001.
9. 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.
10. J. W. Pinney, R. D. Westhead, and G. A. McConkey. Petri net representations in systems biology. *Biochem. Soc. Tarns.*, 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.*, 62:930–933, 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, 2010.
14. A. K. Wagler and J.-T. Wegener. On minimality and equivalence of Petri nets (extended abstract). *Fundamenta Informaticae*, 128:209–222, 2013.
15. A. K. Wagler and J.-T. Wegener. Preprocessing for network reconstruction: Feasibility test and handling infeasibilty (extended abstract). *CEUR Workshop Proceedings (Special Issue CS&P 2013)*, (1032):434–47, 2013.