Identification of Petri nets from knowledge of their language

Maria Paola Cabasino, Alessandro Giua, Carla Seatzu*

March 13, 2008

Abstract

In this paper we deal with the problem of identifying a Petri net system, given a finite language generated by it. First we consider the problem of identifying a free labeled Petri net system, i.e., all transition labels are distinct. The set of transitions and the number of places is assumed to be known, while the net structure and the initial marking are computed solving an integer programming problem. Then we extend this approach in several ways introducing additional information about the model (structural constraints, conservative components, stationary sequences) or about its initial marking. We also treat the problem of synthesizing a bounded net system starting from an automaton that generates its language. Finally, we show how the approach can also be generalized to the case of labeled Petri nets, where two or more transitions may share the same label. In particular, in this case we impose that the resulting net system is deterministic. In both cases the identification problem can still be solved via an integer programming problem.

Published as:
M.P. Cabasino, A. Giua, C. Seatzu, "Identification of Petri nets from knowledge of their language," *Discrete Event Dynamic Systems*, Vol. 17, No. 4, pp. 447–474, Dec 2007. The original publication is available at www.springerlink.com.

1 Introduction

In this paper we present a linear algebraic approach for the identification of a Petri net system from the knowledge of a finite set of strings that it generates.

Identification is a classical problem in system theory: given a pair of observed input-output signals it consists in determining a system such that the input-output signals approximate the observed ones [22].

*M.P. Cabasino, A. Giua and C. Seatzu are with the Department of Electrical and Electronic Engineering, University of Cagliari, Piazza D'Armi, 09123 Cagliari, Italy. E-mail:{cabasino,giua,seatzu}@diee.unica.it.

In the context of free labeled Petri nets[1], it is common to consider as observed behavior the language of the net, i.e., the set of transition sequences that can be fired starting from the initial marking. Assume that a language $\mathcal{L} \subset T^*$ is given, where T is a given set of n transitions. Let this language be finite, prefix-closed and let k be the length of the longest string it contains. Given a number of places m, the identification problem we consider consists in determining the structure of a net N, i.e., the matrices $Pre, Post \in \mathbb{N}^{m \times n}$, and its initial marking $M_0 \in \mathbb{N}^m$ such that the set of all firable transition sequences of length less than or equal to k is $L_k(N, M_0) = \mathcal{L}$.

Note that the set $\mathcal{L}$ explicitly lists *positive examples*, i.e., strings that are known to belong to the language, but also, implicitly, defines several *counterexamples*, namely all those strings of length less than or equal to k that do not belong to the language.

In a first part of the paper, Section 4, we consider the identification problem applied to free labeled Petri nets. In this case we determine the structure of the net and its initial marking solving an integer programming problem, assuming that the number of places and transitions are known. In our procedure it is also possible to consider an objective function that allows us to find an optimal net according to a given performance index.

We extend this basic approach for the identification of free labeled Petri nets in several ways.

- Assume that some additional information on the structure of the model is given. As an example, one may know that the net admits conservative components or stationary sequences, that it belongs to a particular class (ordinary, marked graph, etc.), that the initial marking belongs to a given set. In Subsection 5.1 we show how to extend the basic procedure to also take into account this additional information.
- Assume that a free labeled Petri net is known to generate a regular language $\mathcal{L}$, and that an automaton generating $\mathcal{L}$ is given. In this case the language may contain words of unbounded length. We show in Subsection 5.2 how the basic procedure can be extended to identify such a net.
- Assume that the number of places m of the net is not specified exactly, but an upper bound $\bar{m}$ on its value is known. In this case, we can solve in one shot a two-criteria optimization problem that first requires identifying a net with the minimal number of places; then, among all those that have a minimal number of places, allows one to optimize for a secondary criterion, such as the number of arcs or of tokens in the initial marking. This extension is presented in Subsection 5.3.

Finally, in Section 6, we generalize this approach to λ-*free labeled* nets, i.e., nets where two or more transitions may share the same label. We assume that the total number of transitions T_e sharing the same label $e \in E$ is known, and show how the constraint set determined for the case of free labeled Petri nets can be modified to account for this more general case. The approach we propose determines a net system that is deterministic, namely at each marking M reachable

[1] A free labeled Petri net is a net where the transitions are not labeled or equivalently the labeling function is an isomorphism.

from the initial one, there cannot exist two or more transitions sharing the same label that are simultaneously enabled at M.

The complexity of the constraint sets we use to characterize the set of admissible solutions is analyzed in the Section 7.

The approach we present is extremely general and, unlike other Petri nets identification approaches, not only can it be applied to λ-free labeled nets, but can be used to determine an optimal net according to a given measure as well. These cases have never been considered in the literature to the best of our knowledge. The main drawback is its computational complexity, in the sense that the number of unknowns grows exponentially with the length k of the longest string.

A preliminary version of this work has been previously presented in [8, 11].

1.1 Related literature

The idea of learning the structure of an automaton from positive examples and from counterexamples has been explored since the early 80's in the formal language domain. As an example, we recall the early work of Gold [12] and Angluin [1].

One of the first original approaches to the identification of safe Petri nets was discussed by Hiraishi [13], who presented an algorithm for the construction of a free labeled Petri net model from the knowledge of a finite set of its firing sequences. In a first phase, a language is identified in the form of a finite state automaton from given firing sequences. In a second phase, the dependency relation is extracted from the language, and the structure of a Petri net is guessed. Provided that the language is generated by a special class of nets, the algorithm uniquely identifies the original net if a sufficiently large set of firing sequences is given.

A different approach is based on the *theory of regions* whose objective is that of deciding whether a given graph is isomorphic to the reachability graph of some free labeled net and then constructing it. An excellent survey of this approach, that also presents some efficient algorithms for net synthesis based on linear algebra, can be found in the paper by Badouel and Darondeau [3]. The type of the net and the language considered in the theory of regions are different from those considered in this paper. For example, in [2] explicit algorithms are given for solving in polynomial time in the size of automata the synthesis problem for pure weighted Petri nets from a restricted class of regular languages or from finite automata; in [9] a method is presented which, given a finite state model, called transition system, synthesizes a safe, place-irredundant Petri net with a reachability graph that is bisimilar to the original transition system. In [4] the authors provide an adaptation of the synthesis algorithm that works in polynomial time with respect to the number of states and to the cardinality of the alphabet for general Petri nets with the sequential firing rule and for Petri nets with step firing rule. The general principle for the synthesis is to inspect regions of the graph representing extensions of places of the candidate nets.

Meda and Mellado [16, 17] have also presented an approach for the identification of free labeled Interpreted Petri nets. Their approach consists in observing the marking of a subset of places and, given some additional information on the dependency between transitions, allows one to reconstruct the part of the net structure related to unobservable places.

Bourdeaud'huy and Yim [6] have presented an approach to reconstruct the incidence matrix and the initial marking of a free labeled net given some structural information on the net, such as the existence of P-invariants or T-invariants. This approach can also deal with positive examples of firing sequences but not with counterexamples. Unlike the approach we present in the following sections, that is based on linear algebraic formalism, the approach of the authors is based on logic constraints.

Dotoli *et al.* in [10] have considered an optimization approach that combines some features of [11, 16, 17]. Their procedure assumes that a production of the net is given, in the sense that it requires not only the knowledge of the sequence of events but also of markings reached during this evolution. Necessary and sufficient conditions for the correct identification of the net are given.

Ru and Hadjicostis [20] have presented an approach for the state estimation of discrete event systems modeled by labeled Petri nets. More specifically, given knowledge of the initial Petri net state, they show that the number of consistent markings in a Petri net with nondeterministic transitions is at most polynomial in the length of the observation sequence, even though the set of possible firing sequences can be exponential in the length of the observation sequence. Li *et al.* in [14] have developed a recursive algorithm for estimating the least-cost transition firing sequence(s) based on the observation of a sequence of labels produced by transition activity in a given labeled Petri net.

A recently published work that is quite similar in spirit and basic methodology to the approach presented in this paper, can be found in [19]. The main difference is that our approach deals with the identification of PN (Petri net) plants, while the approach in [19] deals with the synthesis of PN supervisors.

Finally, in a recent paper Sreenivas [21] dealt with a related topic: the minimization of Petri net models. Given a λ-free labeled Petri net generator and a measure function — that associates to it, say, a non negative integer — the objective is that of finding a Petri net that generates the same language as the original net while minimizing the given measure. In our approach we are able to use as a performance index of the identification procedure some of the measures considered by Sreenivas, thus we can identify a minimal solution among all the possible ones. Note that the undecidability results proved by Sreenivas do not apply to our approach because we only ensure the identity between a given finite language and the set of finite prefixes of the synthesized net language. The example we use in Section 6 is taken from a net discussed in [21].

2 Background on Petri nets

In this section we recall the formalism used in the rest of the paper. For more details on Petri nets we refer to [18].

2.1 Basic definitions

A *Place/Transition net* (P/T net) is a structure $N = (P, T, Pre, Post)$, where P is a set of m places; T is a set of n transitions; $Pre : P \times T \rightarrow \mathbb{N}$ and $Post : P \times T \rightarrow \mathbb{N}$ are the *pre*– and *post*– incidence functions that specify the arcs; $C = Post - Pre$ is the incidence matrix. The *preset* and *postset* of a node $X \in P \cup T$ are denoted ${}^\bullet X$ and $X^\bullet$ while ${}^\bullet X^\bullet = {}^\bullet X \cup X^\bullet$.

A *marking* is a vector $M : P \rightarrow \mathbb{N}$ that assigns to each place of a P/T net a non–negative integer number of tokens, represented by black dots. We denote $M(p)$ the marking of place p. A P/T *system* or *net system* $\langle N, M_0 \rangle$ is a net N with an initial marking M_0.

A transition t is enabled at M iff $M \geq Pre(\cdot\,, t)$ and may fire yielding the marking $M' = M + C(\cdot\,, t)$. We write $M\ [\sigma\rangle$ to denote that the sequence of transitions σ is enabled at M, and we write $M\ [\sigma\rangle\ M'$ to denote that the firing of σ yields M'. Note that in this paper we always assume that two or more transitions cannot simultaneously fire (non-concurrency hypothesis).

A marking M is *reachable* in $\langle N, M_0 \rangle$ iff there exists a firing sequence σ such that $M_0\ [\sigma\rangle\ M$. The set of all markings reachable from M_0 defines the *reachability set* of $\langle N, M_0 \rangle$ and is denoted $R(N, M_0)$.

Given a Petri net system $\langle N, M_0 \rangle$ we define its *free-language* as the set of its firing sequences

$$L(N, M_0) = \{\sigma \in T^* \ \mid \ M_0[\sigma\rangle\}.$$

We also define the set of firing sequences of length less than or equal to $k \in \mathbb{N}$ as:

$$L_k(N, M_0) = \{\sigma \in L(N, M_0) \ \mid \ |\sigma| \leq k\}.$$

2.2 Structural properties

In this section we introduce some structural properties of Petri nets that will be used in Subsection 5.1.

Definition 2.1 *Let us consider a Petri net with m places, n transitions and incidence matrix C. A P-vector $\vec{x} : P \rightarrow \mathbb{N}$, with $\vec{x} \neq \vec{0}$, is called:*

- P-invariant*: if $\vec{x}^T \cdot C = \vec{0}^T$;*

- P-increasing*: if* $\vec{x}^T \cdot C \gneq \vec{0}^T$*;*
- P-decreasing*: if* $\vec{x}^T \cdot C \lneq \vec{0}^T$*.*

It can be shown that if $\vec{x}$ is a P-invariant (resp., P-increasing, P-decreasing) along any evolution the sum of the markings weighted with vector $\vec{x}$ remains constant (resp., does not decrease, does not increase).

A T-vector $\vec{y} : T \to \mathbb{N}$*, with* $\vec{y} \neq \vec{0}$*, is called:*

- T-invariant*: if* $C \cdot \vec{y} = \vec{0}$*;*
- T-increasing*: if* $C \cdot \vec{y} \gneq \vec{0}$*;*
- T-decreasing*: if* $C \cdot \vec{y} \lneq \vec{0}$*.*

It can be shown that if $\vec{y}$ is a T-invariant the firing of a sequence of transitions whose firing vector is $\vec{y}$ does not modify the number of tokens, i.e., it is a stationary sequence. If $\vec{y}$ is a T-increasing the firing of a sequence of transitions whose firing vector is $\vec{y}$ increases the number of tokens, i.e., it is a repetitive non stationary sequence. Finally if $\vec{y}$ is a T-decreasing the firing of a sequence of transitions whose firing vector is $\vec{y}$ decreases the number of tokens.

A Petri net is said *ordinary* if $Pre, Post \in \{0,1\}^{m \times n}$, i.e., if each arc has weight equal to one.

A *marked graph* is an ordinary Petri net such that each place has exactly one input and one output transition.

A *state machine* is an ordinary Petri net where each transition has exactly one input and one output place.

2.3 Labeled Petri nets

When observing the evolution of a net, it is common to assume that each transition t is assigned a label $\varphi(t)$ and the occurrence of t generates an observable output $\varphi(t)$. If $\varphi(t) = \varepsilon$, i.e., if the transition is labeled with the empty string, its firing cannot be observed. This leads to the definition of labeled nets.

Definition 2.2 *Given a Petri net* N *with set of transitions* T*, a* labeling function $\varphi : T \to E \cup \{\varepsilon\}$ *assigns to each transition* $t \in T$ *a symbol, from a given alphabet* E*, or assigns to it the empty string* ε*.*

A labeled *Petri net system is a 3-tuple* $G = \langle N, M_0, \varphi \rangle$ *where* $N = (P, T, Pre, Post)$*,* M_0 *is the initial marking, and* $\varphi : T \to E \cup \{\varepsilon\}$ *is the labeling function.* ■

Four classes of labeling functions may be defined.

Definition 2.3 *The labeling function of a labeled Petri net system* $\langle N, M_0, \varphi \rangle$ *can be classified as follows.*

- Free*: if all transitions are labeled distinctly, namely a different label is associated to each transition, and no transition is labeled with the empty string.*
- Deterministic*: if no transition is labeled with the empty string, and the following condition*[2] *holds: for all* $t, t' \in T$*, with* $t \neq t'$*, and for all* $M \in R(N, M_0)$*:* $M[t\rangle \wedge M[t'\rangle \Rightarrow [\varphi(t) \neq \varphi(t')]$ *i.e., two transitions simultaneously enabled may not share the same label. This ensures that the knowledge of the firing label* $\varphi(t)$ *is sufficient to reconstruct the marking that the firing of* t *yields.*
- λ-free*: if no transition is labeled with the empty string*[3]*.*
- Arbitrary*: if no restriction is posed on the labeling function* φ*.*

■

Each of these types of labeling is a generalization of the previous one. Furthermore all types of labeling only depend on the structure of the net, but for the deterministic labeling that depends both on the structure and on the behavior of the net.

In the particular case in which the labeling function is free, being an isomorphism between the alphabet E and the set of transitions T, it is usual to choose $E = T$, or equivalently to assume that the transitions are not labeled and their firing can be directly observed.

3 Logical constraints transformation

In this section we provide an efficient technique to convert logical *or* constraints into linear algebraic constraints, that is inspired by the work of Bemporad and Morari [5]. In particular, we consider two different cases: *inequality* constraints and *equality* constraints.

[2] A looser condition is sometimes given: for all $t, t' \in T$, with $t \neq t'$, and for all $M \in R(N, M_0)$: $M[t\rangle \wedge M[t'\rangle \Rightarrow [\varphi(t) \neq \varphi(t')] \vee [Post(\cdot, t) - Pre(\cdot, t) = Post(\cdot, t') - Pre(\cdot, t')]$. Thus two transitions with the same label may be simultaneously enabled at a marking M, if the two markings reached from M by firing t and t' are the same.

[3] In the Petri net literature the empty string is denoted λ, while in the formal language literature it is denoted ε. In this paper we denote the empty string ε but, for consistency with the Petri net literature, we still use the term λ*-free* for a non erasing labeling function $\varphi : T \to E$.

3.1 Inequality constraints

Let us consider the following constraint:

$$\bigvee_{i=1}^{r} \vec{a}_i \leq \vec{0}_n \tag{1}$$

where $\vec{a}_i \in \mathbb{R}^n$, $i = 1, \ldots, r$, and $\bigvee$ denotes the logical *or* operator.

Equation (1) can be rewritten in terms of linear algebraic constraints as:

$$\begin{cases} \vec{a}_1 \leq z_1 \cdot \vec{K} \\ \vdots \\ \vec{a}_r \leq z_r \cdot \vec{K} \\ z_1 + \ldots + z_r = r - 1 \\ z_1, \ldots, z_r \in \{0, 1\} \end{cases} \tag{2}$$

where $\vec{K}$ is any constant vector in $\mathbb{R}^n$ that satisfies the following relation

$$K_j > \max_{i \in \{1,\ldots,r\}} a_i(j), \quad j = 1, \ldots, n.$$

In fact, if $z_i = 0$ then the i-th constraint is active, while if $z_i = 1$ it is trivially verified, thus resulting in a redundant constraint. Moreover, the condition $z_1 + \ldots + z_r = r - 1$ implies that one and only one z_i is equal to zero, i.e., only one constraint is active. This means that $\vec{a}_i \leq \vec{0}_n$ for one i, while no condition is imposed for the other i's (in such cases the corresponding constraints may either be violated or satisfied). Obviously, analogous considerations can be repeated if the $\leq$ constraints in (1) are replaced by $\geq$ constraints.

3.2 Equality constraints

Let us now consider the constraint

$$\bigvee_{i=1}^{r} \vec{a}_i = \vec{b}_i \tag{3}$$

where $\vec{a}_i, \vec{b}_i \in \mathbb{R}^n$, $i = 1, \ldots, r$.

Equation (3) can be rewritten in terms of linear algebraic constraints as:

$$\begin{cases} \vec{a}_1 - \vec{b}_1 \leq z_1 \cdot \vec{K} \\ \vec{a}_1 - \vec{b}_1 \geq -z_1 \cdot \vec{K} \\ \vdots \\ \vec{a}_r - \vec{b}_r \leq z_r \cdot \vec{K} \\ \vec{a}_r - \vec{b}_r \geq -z_r \cdot \vec{K} \\ z_1 + \ldots + z_r = r - 1 \\ z_1, \ldots, z_r \in \{0, 1\} \end{cases} \tag{4}$$

where $\vec{K}$ is any constant vector in $\mathbb{R}^n$ such that

$$K_j > \max_{i\in\{1,\dots,r\}} |a_i(j) - b_i(j)|, \quad j = 1,\dots,n.$$

Repeating a similar reasoning as in the previous case, we can immediately observe that, if $z_i = 0$ then

$$\begin{cases} \vec{a}_i - \vec{b}_i \leq \vec{0}_n \\ \vec{a}_i - \vec{b}_i \geq \vec{0}_n \end{cases} \quad \Rightarrow \quad \vec{a}_i = \vec{b}_i.$$

On the contrary, if $z_i = 1$ then

$$\begin{cases} \vec{a}_i - \vec{b}_i \leq \vec{K} \\ \vec{a}_i - \vec{b}_i \geq -\vec{K} \end{cases}$$

that are trivially verified, i.e., they are redundant constraints. Finally, the condition on the sum of z_i's imposes that one constraint is active, i.e., $\vec{a}_i = \vec{b}_i$ for at least one $i \in \{1,\dots,r\}$.

4 Basic identification procedure for free labeled Petri nets

In this section we describe the identification procedure for free labeled Petri nets. As mentioned in Subsection 2.3 for this type of Petri nets we assume $E = T$ without any loss of generality.

The problem we consider in this section can be formally stated as follows.

Problem 4.1 *Let $\mathcal{L} \subset T^*$ be a finite prefix-closed language[4], and*

$$k = \max_{\sigma\in\mathcal{L}} |\sigma|$$

be the length of the longest string in $\mathcal{L}$. Chosen a set of places P of cardinality m we want to identify the structure of a net $N = (P, T, Pre, Post)$ and an initial marking M_0 such that

$$L_k(N, M_0) = \mathcal{L}.$$

We also assume that a nonnegative integer K is given such that the following condition[5] holds:

$$\max_i M_0(p_i) + k \cdot \max_{i,j} Post(i,j) \leq K.$$

The unknowns we want to determine are the elements of the two matrices Pre, $Post \in \mathbb{N}^{m\times n}$ and the elements of the vector $M_0 \in \mathbb{N}^m$. ■

[4] A language $\mathcal{L}$ is said to be *prefix-closed* if for any string $\sigma \in \mathcal{L}$, all prefixes of σ are in $\mathcal{L}$.

[5] This assumption is purely technical, as mentioned in Remark 4.3, and since K can be chosen arbitrarily large does not pose any practical limitation.

A solution to the above identification problem can be computed thanks to the following theorem, that provides a linear algebraic characterization of the place/transition nets with m places and n transitions such that $L_k(N, M_0) = \mathcal{L}$.

Theorem 4.2 *A net system $\langle N, M_0 \rangle$ is a solution of the identification problem (4.1) if and only if it satisfies the following set of linear algebraic constraints*

$$\mathcal{G}_m(\mathcal{E}, \mathcal{D}) \triangleq \begin{cases} M_0 + Post \cdot \vec{\sigma} - Pre \cdot (\vec{\sigma} + \vec{t_j}) \geq \vec{0} & \forall (\sigma, t_j) \in \mathcal{E} \quad (a) \\ -KS_{\sigma,j} + M_0 + Post \cdot \vec{\sigma} - Pre \cdot (\vec{\sigma} + \vec{t_j}) \leq -\vec{1}_m & \forall (\sigma, t_j) \in \mathcal{D} \quad (b) \\ \vec{1}^{\,T} S_{\sigma,j} \leq m - 1 & \forall (\sigma, t_j) \in \mathcal{D} \quad (c) \\ M_0 \in \mathbb{N}^m & (d) \\ Pre, Post \in \mathbb{N}^{m \times n} & (e) \\ S_{\sigma,j} \in \{0,1\}^m & (f) \end{cases} \tag{5}$$

where

$$\mathcal{E} = \{(\sigma, t_j) \mid \sigma \in \mathcal{L}, |\sigma| < k, \sigma t_j \in \mathcal{L}\} \tag{6}$$

and

$$\mathcal{D} = \{(\sigma, t_j) \mid \sigma \in \mathcal{L}, |\sigma| < k, \sigma t_j \notin \mathcal{L}\}. \tag{7}$$

Proof.

- Assume that $\sigma t_j \in \mathcal{L}$, where $\sigma \in T^*$ and $t_j \in T$. Then transition t_j is enabled from the marking $M_\sigma = M_0 + (Post - Pre) \cdot \vec{\sigma}$ and the following relation must hold

  $$M_\sigma \geq Pre(\cdot, t_j).$$

  This relation can be rewritten as

  $$M_0 + Post \cdot \vec{\sigma} - Pre \cdot (\vec{\sigma} + \vec{t_j}) \geq \vec{0}_m. \tag{8}$$

- Assume that $\sigma \in \mathcal{L}$ and $\sigma t_j \notin \mathcal{L}$, where $\sigma \in T^*$ and $t_j \in T$. Then transition t_j is not enabled from the marking

  $$M_\sigma = M_0 + (Post - Pre) \cdot \vec{\sigma},$$

  that is for at least one place p_i it must hold

  $$M_\sigma(p_i) < Pre(p_i, t_j).$$

  We first observe that that each component of M_σ is less than or equal to K, as defined in Problem 4.1. In fact it holds:

  $$\begin{aligned} K &\geq \max_i M_0(p_i) + k \cdot \max_{i,j} Post(i,j) \\ &\geq \max_i M_0(p_i) + |\sigma| \cdot \max_{i,j} Post(i,j) \\ &\geq \max_i M_\sigma(p_i). \end{aligned} \tag{9}$$

We now define a vector

$$S_{\sigma,j} = \begin{bmatrix} s_1 \\ \vdots \\ s_m \end{bmatrix} \in \{0,1\}^m,$$

such that for all $i = 1, \ldots, m$ it holds

$$s_i = 0 \quad \Longrightarrow \quad M_\sigma(p_i) < Pre(p_i, t_j).$$

The i-th component of $S_{\sigma,j}$ (for $i = 1, \ldots, m$) must satisfy the equation

$$-Ks_i + M_\sigma(p_i) - Pre(p_i, t_j) < 0, \tag{10}$$

so that if $s_i = 0$ it must hold $M_\sigma(p_i) - Pre(p_i, t_j) < 0$, while if $s_i = 1$ equation (10) is trivially verified thanks to equation (9). In vector form (and taking into account that all variables are integers) equation (10) rewrites:

$$-KS_{\sigma,j} + M_0 + Post \cdot \vec{\sigma} - Pre \cdot (\vec{\sigma} + \vec{t_j}) \leq -\vec{1}_m. \tag{11}$$

Finally, there exists at least a place that disables t_j if

$$\sum_{i=1}^{m} s_i \leq m - 1, \tag{12}$$

so that at least one s_i is null. In vector form this equation rewrites

$$\vec{1}^{\,T} S_{\sigma,j} \leq m - 1. \tag{13}$$

□

Remark 4.3 *Let us briefly comment about the constant K defined in Problem 4.1.*

In Theorem 4.2, for any pair $(\sigma, t_j) \in \mathcal{D}$ one has to look for at least one place that disables t_j after σ. This or *condition, as discussed in Section 3, may be rewritten in a simpler form if an upper bound on the absolute value of the variables involved in the constraints is given.*

In practice, in Problem 4.1 it is sufficient to pick K very large. We also mention that many software tools allow the definition of an arbitrary large constant.

In general the solution of the set (5) is not unique, thus there exists more than one Petri net system $\langle N, M_0 \rangle$ such that $L_k(N, theM_0) = \mathcal{L}$. To select one among these Petri net systems we choose a given performance index and solving an appropriate IPP (Integer Programming Problem) we determine a Petri net system that minimizes the considered performance index[6]. In particular, if $f(M_0, Pre, Post)$ is the considered performance index, an identification problem can be formally stated as follows.

[6]Clearly, also in this case the solution may be not unique.

Problem 4.4 *Let us consider the identification problem (4.1) and let $f(M_0, Pre, Post)$ be a given performance index. The solution to the identification problem (4.1) that minimizes $f(M_0, Pre, Post)$ can be computed by solving the following IPP*

$$\begin{cases} \min & f(M_0, Pre, Post) \\ s.t. & \mathcal{G}_m(\mathcal{E}, \mathcal{D}). \end{cases} \tag{14}$$

■

Of particular interest are those objective functions that are linear in the unknowns, so that the problem to solve is a linear integer programming problem. As example of a linear objective function, assume we want to determine a Petri net system that minimizes the weighted sum of the tokens in the initial marking and of the arc weights. The general case is:

$$f(M_0, Pre, Post) = \sum_{i=1}^{m} \left(a_i \cdot M_0(p_i) + \left(\sum_{j=1}^{n} b_{i,j} \cdot Pre(p_i, t_j) + c_{i,j} \cdot Post(p_i, t_j) \right) \right), \tag{15}$$

where $a_i, b_{i,j}$ and $c_{i,j}$ are given coefficients.

A typical choice, that we follow in the rest of the paper, is that of choosing all coefficients equal to 1. In this case (15) can be rewritten:

$$f(M_0, Pre, Post) = \vec{1}_m^T \cdot M_0 + \vec{1}_m^T \cdot (Pre + Post) \cdot \vec{1}_n.$$

Example 4.5 Let $\mathcal{L} = \{\varepsilon, t_1, t_1t_1, t_1t_2, t_1t_1t_2, t_1t_2t_1\}$ and $m = 2$, thus $k = 3$. Assume that we want to determine the Petri net system that minimizes the sum of initial tokens and all arcs such that $L_3(N, M_0) = \mathcal{L}$. This requires the solution of an IPP of the form (14) where

$$\mathcal{E} = \{(\varepsilon, t_1), (t_1, t_1), (t_1, t_2), (t_1t_2, t_1), (t_1t_1, t_2)\}$$

and

$$\mathcal{D} = \{(\varepsilon, t_2), (t_1t_2, t_2), (t_1t_1, t_1)\}.$$

The procedure identifies a net system with

$$Pre = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad Post = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad M_0 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

namely the net system in Fig. 1.a. ■

5 Extended identification procedure for free labeled Petri nets

In many cases the available information on the net to identify is not limited to samples of its language. As an example, it may be known that the net has a particular structure, or some partial information on the initial marking (in terms of available resources) may be given.

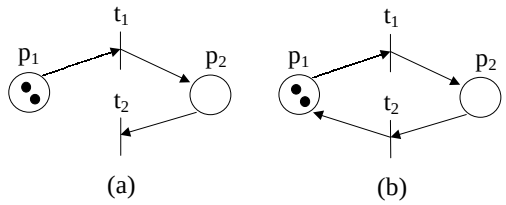

Figure 1: (a) The Petri net system of Example 4.5; (b) the Petri net of the same example when the additional constraint $m_1 + m_2 = \text{const}$ is added.

In this section it is shown how this additional information can easily be incorporated in the identification procedure previously described.

5.1 Structural constraints

P-vectors

Assume that some places of the net are known to belong to a conservative component, i.e., the weighted sum of their tokens in the component remains constant during any evolution. This is equivalent to say that some P-invariants for the net are known (see Definition 2.1).

More generally the knowledge of any P-vector may be taken into account adding to Problem 14 a suitable set of constraints.

- Assume $\vec{x} \in \mathbb{R}^m$ is *P-invariant*. We need to add to Problem 14 the following constraint

$$\vec{x}^T(Post - Pre) = \vec{0}_n^T$$

  that imposes $\vec{x}^T \cdot C = \vec{0}_n^T$.

- Assume $\vec{x} \in \mathbb{R}^m$ is *P-increasing*. We need to add to Problem 14 the following constraints

$$\begin{cases} \vec{x}^T(Post - Pre) \geq \vec{0}_n^T \\ \vec{x}^T(Post - Pre)\vec{1}_n \geq 1 \end{cases}$$

  The first constraint imposes that $\vec{x}^T \cdot C \geq \vec{0}_n^T$ and the second one imposes that $\vec{x}^T \cdot C \neq \vec{0}_n^T$.

- Assume $\vec{x} \in \mathbb{R}^m$ is *P-decreasing*. We need to add to Problem 14 the following constraints

$$\begin{cases} \vec{x}^T(Post - Pre) \leq \vec{0}_n^T \\ \vec{x}^T(Post - Pre)\vec{1}_n \leq -1 \end{cases}$$

  The first constraint imposes that $\vec{x}^T \cdot C \leq \vec{0}_n^T$ and the second one imposes that $\vec{x}^T \cdot C \neq \vec{0}_n^T$.

T-vectors

Assume that a given firing sequence is known to be stationary, i.e., the number of the tokens of the net is not modified by the firing of this sequence. This is equivalent to say that some T-invariants for this net are known (see Definition 2.1).

More generally the knowledge of any T-vector may be taken into account adding to Problem 14 a suitable set of constraints.

- Assume $\vec{y} \in \mathbb{R}^n$ is *T-invariant*. We need to add to Problem 14 the following constraint

$$(Post - Pre)\vec{y} = \vec{0}_m$$

  that imposes $C \cdot \vec{y} = \vec{0}_m$.

- Assume $\vec{y} \in \mathbb{R}^n$ is *T-increasing*. We need to add to Problem 14 the following constraints

$$\begin{cases} (Post - Pre) \cdot \vec{y} \geq \vec{0}_m \\ \vec{1}_m^T (Post - Pre)\vec{y} \geq 1 \end{cases}$$

  The first constraint imposes that $C \cdot \vec{y} \geq \vec{0}_m^T$ and the second one imposes that $C \cdot \vec{y} \neq \vec{0}_m^T$.

- Assume $\vec{x} \in \mathbb{R}^m$ is *T-decreasing*. We need to add to Problem 14 the following constraints

$$\begin{cases} (Post - Pre) \cdot \vec{y} \leq \vec{0}_m \\ \vec{1}_m^T (Post - Pre)\vec{y} \leq -1 \end{cases}$$

  The first constraint imposes that $C \cdot \vec{y} \leq \vec{0}_m^T$ and the second one imposes that $C \cdot \vec{y} \neq \vec{0}_m^T$.

Example 5.1 Let us consider again the case of Example 4.5 but assume the net is known to be conservative. In particular, the sum of the tokens in places p_1 and p_2 remains constant. To this aim we solve an IPP of the form (14) with the addition of a constraint of the form of (5.1), where $\vec{x} = [1 \;\; 1]^T$. We identify a net system with

$$Pre = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad Post = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad M_0 = \begin{bmatrix} 2 \\ 0 \end{bmatrix}$$

namely the net in Fig. 1.b. ■

Net subclasses

In this subsection we consider the constraints that we need to add to Problem 14 to ensure that the identified net belongs to some particular subclasses of nets defined in Subsection 2.2.

- *Ordinary*:
$$Pre, Post \in \{0,1\}^{m \times n}.$$
- *Marked graph*:
$$\begin{cases} Pre \cdot \vec{1}_n = 1 \\ Post \cdot \vec{1}_n = 1. \end{cases}$$
- *State machine*:
$$\begin{cases} \vec{1}_m^T \cdot Pre = 1 \\ \vec{1}_m^T \cdot Post = 1. \end{cases}$$

All these results follow immediately from the definitions in Subsection 2.2.

Constraints on the initial marking

A type of general constraints that can be imposed on the markings of a Petri net is called GMEC (Generalized Mutual Exclusion Constraint) and can be represented by the couple $(\vec{w}, k)$, where $\vec{w} \in \mathbb{Z}^m, k \in \mathbb{Z}$. This constraint defines a set of legal markings:

$$\mathcal{M}(\vec{w}, k) = \{M \in \mathbb{N}^m \mid \vec{w}^T M \leq k\}.$$

If it is known that $M_0 \in \mathcal{M}(\vec{w}, k)$ then the constraint

$$\vec{w}^T M_0 \leq k,$$

should be added to Problem 14.

For example consider a Petri net with an initial marking that can not contain a number of tokens greater than 1 in places p_1 and p_2. In this case we need to impose as additional constraint

$$M(p_1) + M(p_2) \leq 1.$$

Structural decomposition

We can impose a *structural decomposition* of the net in a given number r of subnets. Let

$$P = P_1 \cup P_2 \cup \ldots \cup P_r$$

be a given partition of P. Assume that for all $t \in T$ we are given a set $\Pi(t) \subset \{1, \ldots, r\}$ such that $q \in \Pi(t)$ implies ${}^\bullet t^\bullet \cap P_q = \emptyset$. In plain words, $\Pi(t)$ represents the set of indices of P_q's such that t has no input/output arc from/to a place in P_q.

This can be imposed adding to Problem (14) the following set of linear constraints for all $t \in T$:

$$\sum_{q \in \Pi(t)} \sum_{p \in P_q} (Pre(p,t) + Post(p,t)) = 0.$$

5.2 Synthesis of bounded Petri net systems from regular languages

In this section we assume that the net system we want to synthesize is bounded, and thus its language is regular. The language is given in terms of a finite state automaton $G = (Q, T, \delta, q_0)$ where Q is the set of states, the alphabet T is the set of transitions of the net, $\delta : Q \times T \to Q$ is the transition function, and q_0 is the initial state.

We consider the following problem.

Problem 5.2 *Let $G = (Q, T, \delta, q_0)$ be a given finite state automaton. Chosen a set of places P of cardinality m and a nonnegative integer K, we want to identify the structure of a net $N = (P, T, Pre, Post)$ and an initial marking M_0 such that $L(N, M_0) = \mathcal{L}(G)$, and*

$$\max_i M_0(p_i) + k \cdot \max_{i,j} Post(i, j) \leq K.$$

The unknowns we want to determine are the elements of the two matrices Pre, $Post \in \mathbb{N}^{m \times n}$ and the elements of the vector $M_0 \in \mathbb{N}^m$. ■

The identification procedure previously defined considers sequences of bounded length. An automaton is able to generate sequences of unbounded length every time that there is a cycle. Thus we have to distinguish between sequences that pass through cycles (that can be extended indefinitely) and sequences that do not pass through cycles (whose length is finite).

We say that a firing sequence $\sigma' \prec \sigma$ if σ' is a strict prefix of σ, i.e., if $\sigma = t_{\alpha_1} t_{\alpha_2} \dots t_{\alpha_k}$ and $\sigma' = t_{\alpha_1} t_{\alpha_2} \dots t_{\alpha_r}$ with $r < k$. In following we denote as

$$\Gamma(G) = \{\sigma \in T^* \mid \delta(q, \sigma) = q \ \wedge \ \forall \sigma', \sigma'' \prec \sigma \ \delta(q, \sigma') \neq \delta(q, \sigma'')\} \tag{16}$$

the set of elementary cycles of the automaton. We define the set of the firing vectors associated to the firing sequences in $\Gamma(G)$ as

$$Y(G) = \{\vec{y} \in \mathbb{N}^n \mid \exists \ \sigma \in \Gamma(G) \ : \ \vec{y} = \vec{\sigma}\}. \tag{17}$$

Finally, we define the set of sequences that are generated by the automaton without passing through a cycle as

$$\mathcal{L}_{st}(G) = \{\sigma \in T^* \mid \forall \ u, v \preceq \sigma, \ u \neq v \Rightarrow \ \delta(q_0, u) \neq \delta(q_0, v)\} \subseteq \mathcal{L}(G), \tag{18}$$

where $\mathcal{L}(G)$ denotes the language generated by the automaton, and the subscript *st* denotes the words generated by its spanning tree with root q_0.

Theorem 5.3 *A net system $\langle N, M_0 \rangle$ is a solution of the identification problem (5.2) if and only if it satisfies the following set of linear algebraic constraints*

$$\begin{cases} \mathcal{G}_m(\mathcal{E}, \mathcal{D}) & (a) \\ (Post - Pre) \cdot \vec{y} = \vec{0} \quad \forall \ \vec{y} \in Y(G) & (b) \end{cases} \tag{19}$$

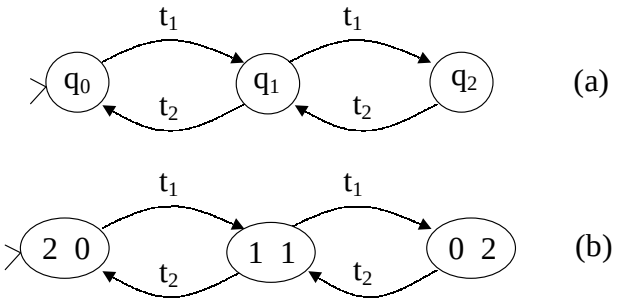

Figure 2: (a) The finite state automaton G of Example 5.4; (b) the reachability graph of the identified net system.

where $\mathcal{E} = \{(\sigma, t) \mid \sigma \in \mathcal{L}_{st}(G), \sigma t \in \mathcal{L}(G)\}$

and $\mathcal{D} = \{(\sigma, t) \mid \sigma \in \mathcal{L}_{st}(G), \sigma t \notin \mathcal{L}\}$.

Proof.

We just give a sketch of the proof. First consider a word $\sigma' = \sigma t$ *where* $\sigma \in \mathcal{L}_{st}(G)$. *Then* $\mathcal{G}_m(\mathcal{E}, \mathcal{D})$ *contains enough constraints to ensure that* σ *and* σ', *or* σ *and not* σ' *are generated by any net solution of (18) according to the case* $\sigma' \in \mathcal{L}(G)$ *or* $\sigma' \notin \mathcal{L}(G)$.

Consider next a word $\sigma' = \sigma t$ *where* $\sigma \in \mathcal{L}(G) \setminus \mathcal{L}_{st}(G)$. *Then* $\sigma = \sigma_1 u \sigma_2$ *where* $u \neq \epsilon$ *and* $\delta(q_0, \sigma_1) = \delta(q_0, \sigma_1 u)$, *hence* $\sigma_1\sigma_2 \in \mathcal{L}(G)$, *and the firing count vector of* u *is a T-invariant of any net solution of (18). In other words, such a net generates* σt *if and only if it generates* $\sigma_1 \sigma_2 t$, *and since* $\sigma_1 \sigma_2$ *is strictly shorter than* σ, *the theorem follows by induction on words.* □

Example 5.4 Let us consider the finite state automaton G in Fig. 2.a. It holds $Y(G) = \{[1\ 1]^T\}$ and $\mathcal{L}_{st}(G) = \{\varepsilon, t_1, t_1t_1\}$ thus $\mathcal{E} = \{(\varepsilon, t_1), (t_1, t_1), (t_1, t_2), (t_1t_1, t_2)\}$ and $\mathcal{D} = \{(\varepsilon, t_2), (t_1t_1, t_1)\}$. Now, assume that we want to determine the Petri net system that minimizes the sum of initial tokens and all arcs.

For $m = 1$ we get no feasible solution, while for $m = 2$ we find the net system in Fig. 1.b whose reachability graph is shown in Fig. 2.b. Note that in this particular case the reachability graph of the net is isomorphic to the given automaton G, but this is not necessarily guaranteed by our procedure. The problem of finding a net whose reachability graph is isomorphic to that of an automaton is addressed in the theory of regions [3]. ■

5.3 Optimizing the number of places

In the previous formulation we assumed that the number m of places is given. In this subsection we remove this assumption and consider the following identification problem.

Problem 5.5 *Let us consider an identification problem in the form (4.1) where* m *is only known*

to be less than or equal to a given value $\bar{m}$, and let $f(m, M_0, Pre, Post)$ be a given performance index. The solution of the identification problem that minimizes $f(m, M_0, Pre, Post)$ with the smallest number of places can be computed solving the following nonlinear IPP

$$\begin{cases} \min\limits_{m \leq \bar{m}} & \min & f(m, M_0, Pre, Post) \\ s.t. & & \mathcal{G}_m(\mathcal{E}, \mathcal{D}). \end{cases} \tag{20}$$

A trivial solution to the identification problem 5.5 consists in solving IPP of the form (14) for increasing values of m, until a feasible solution is obtained.

The following theorem provides an alternative approach to do this, that simply requires the solution of one IPP, while guaranteeing the optimality of the solution both in terms of minimum number of places and in terms of the chosen performance index.

Theorem 5.6 *Solving the identification problem 5.5 is equivalent to solving the following IPP:*

$$\begin{cases} \min & \overline{K} \cdot \vec{1}_{\bar{m}}^T \vec{z} + f(\bar{m}, M_0, Pre, Post) \\ s.t. & \mathcal{G}_{\bar{m}}(\mathcal{E}, \mathcal{D}) \\ & K \cdot \vec{z} - Pre \cdot \vec{1}_n - Post \cdot \vec{1}_n \geq \vec{0}_{\bar{m}} \\ & z_{i+1} \leq z_i, \quad i = 1, \ldots, \bar{m} - 1 \\ & \vec{z} \in \{0,1\}^{\bar{m}} \end{cases} \tag{21}$$

for a sufficiently large constant $\overline{K}$ ($\overline{K}$ must be such that the minimization of the first term of the objective function has priority over the minimization of its second term).

In particular, let us denote as $\vec{z}^*$, $\bar{M}_0^*$, $\overline{Pre}^*$ and $\overline{Post}^*$ the solution of (21), and let m^* be the number of nonzero components of $\vec{z}^*$.

Let M_0^* be the vector obtained from $\bar{M}_0^*$ by only keeping its first m^* components. Analogously, let Pre^* and $Post^*$ be the matrices obtained from $\overline{Pre}^*$ and $\overline{Post}^*$, respectively, by only keeping their first m^* rows.

Then, m^*, M_0^*, Pre^*, $Post^*$ is a solution of the identification problem 5.5.

Proof.

Let us first observe that if $z_i = 1$, then the corresponding constraint

$$K - Pre(p_i, \cdot) \cdot \vec{1}_n - Post(p_i, \cdot) \cdot \vec{1}_n \geq 0$$

is trivially verified being K a very large constant.

On the contrary, if $z_i = 0$, the new constraint becomes

$$-Pre(p_i, \cdot) \cdot \vec{1}_n - Post(p_i, \cdot) \cdot \vec{1}_n \geq 0$$

whose only admissible solution is $Pre(p_i, \cdot) = Post(p_i, \cdot) = \vec{0}_n^T$. Place p_i is in this case redundant and can be removed without affecting the language of the net.

Since our main goal in (21) is that of minimizing $\vec{1}_m^T \vec{z}$, the optimal solution $\vec{z}^*$ will have as many zeros as possible, compatibly with the other constraints. Moreover, being $z_{i+1} \leq z_i, \quad i = 1, \ldots, \bar{m} - 1$, zero is assumed by the last components of $\vec{z}^*$. □

In the previous theorem the chosen performance index allows one to solve in one shot a two-criteria optimization problem using a procedure based on *global priorities* [7]. In this case we have a multi–objective performance in which the goals have different priorities. We first look for all solutions that optimize the first goal, i.e., the number of places, and then among them we look for those that optimize the second goal.

Example 5.7 Let

$$\mathcal{L} = \{\varepsilon, t_1, t_1t_2, t_1t_3, t_1t_2t_1, t_1t_2t_3, t_1t_3t_1, t_1t_3t_2\}$$

thus $k = 3$. Assume that we want to determine the Petri net system that minimizes the sum of initial tokens and all arcs such that $L_3(N, M_0) = \mathcal{L}$. This requires the solution of an IPP of the form (14) where

$$\mathcal{E} = \{(\varepsilon, t_1), (t_1, t_2), (t_1, t_3), (t_1t_2, t_1), (t_1t_2, t_3), (t_1t_3, t_1), (t_1t_3, t_2)\}$$

and

$$\mathcal{D} = \{(\varepsilon, t_2), (\varepsilon, t_3), (t_1, t_1), (t_1t_2, t_2), (t_1t_3, t_3)\}.$$

We assume that the total number of places is bounded by $\bar{m} = 5$ and we choose the constant $K = \bar{m} \cdot n \cdot 10^4 = 15 \cdot 10^4$.

The procedure identifies a net system with $m = 3$ and

$$Pre = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad Post = \begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix},$$

$$M_0 = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix},$$

namely the net system in Figure 3. ■

6 λ-free labeled Petri nets

In this section we show how the above results can be extended to the case of λ-free labeled Petri nets.

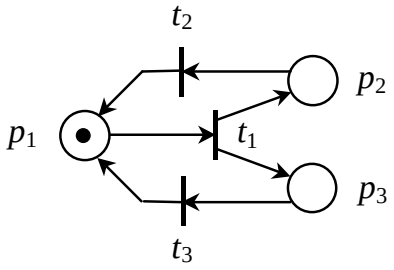

Figure 3: The Petri net system of Example 5.7.

We consider $\varphi : T \to E$ a labeling function over E and we denote the set of transitions that are labeled by symbol e as:

$$T_e = \{t \in T \ \mid \ \varphi(t) = e\} = \{t_1^e, \ldots, t_{n_e}^e\}, \quad e \in E$$

where $n_e = |T_e|$. Obviously it holds

$$T = \bigcup_{e \in E} T_e,$$

i.e., the the labeling equivalence induces a partition of T.

We say that an event e is *ambiguous* if $n_e > 1$, i.e., there exists more than one transition t such that $\varphi(t) = e$, otherwise we say that the event e is *not-ambiguous*. Analogously, we say that a transition t is *ambiguous* if its label is also associated to other transitions, otherwise a transition t is said to be *not-ambiguous*.

We denote $w = \varphi(\sigma)$ the word of events associated to sequence σ.

Given a labeled Petri net system $\langle N, M_0 \rangle$ we define its λ-free labeled language as the set of words in E^* generated from the initial marking M_0, namely,

$$L^E(N, M_0) = \{w \in E^* \ \mid \ \exists \sigma \in T^* : \ M_0[\sigma\rangle, \ \varphi(\sigma) = w\}.$$

We also denote $L_k^E(N, M_0)$ the set of words in $L^E(N, M_0)$ of length less than or equal to $k \in \mathbb{N}$, i.e.,

$$L_k^E(N, M_0) = \{w \in L^E(N, M_0) \ \mid \ |w| \leq k\}.$$

Problem 6.1 *Assume we are given a set of places $P = \{p_1, \ldots, p_m\}$ and a set of transitions $T = \{t_1, \ldots, t_n\}$. Let $\varphi : T \to E$ be a given labeling function over E whose equivalence classes T_e are known. Let $\mathcal{L} \subset E^*$ be a given finite prefix-closed language over E^*, and*

$$k = \max_{w \in \mathcal{L}} |w|$$

be the length of the longest word in $\mathcal{L}$.

We want to identify the structure of a deterministic[7] *net $N = (P, T, Pre, Post)$ labeled by φ and*

[7]Determinism is a desirable property and we assume the net enjoys it. However, it may also be possible to solve this problem without assuming that the net be deterministic.

an initial marking M_0 such that

$$L_k^E(N, M_0) = \mathcal{L}.$$

We also assume that a nonnegative integer K is given such that the following condition holds:

$$\max_i M_0(p_i) + k \cdot \max_{i,j} Post(i,j) \leq K.$$

The unknowns we want to determine are the elements of the two matrices

$$Pre = \{e_{i,j}\} \in \mathbb{N}^{m \times n} \quad and \quad Post = \{o_{i,j}\} \in \mathbb{N}^{m \times n}$$

and the elements of the vector

$$M_0 = \left[\begin{array}{cccc} m_{0,1} & m_{0,2} & \cdots & m_{0,m} \end{array} \right]^T \in \mathbb{N}^m.$$

■

The following theorem provides a linear algebraic characterization of the deterministic labeled Petri net systems with m places, n transitions and labeling function φ such that $L_k^E(N, M_0) = \mathcal{L}$.

Theorem 6.2 *A solution to the identification problem (6.1) satisfies the following set of linear*

algebraic constraints

$\mathcal{G}_m(\mathcal{E},\mathcal{D},\varphi) \triangleq$

$$
\begin{cases}
M_w - Pre(\cdot,t_1^e) \geq -z_1^{e,w}\cdot\vec{K} \\
\quad\vdots \\
M_w - Pre(\cdot,t_{n_e}^e) \geq -z_{n_e}^{e,w}\cdot\vec{K} \\
M_{we} - M_w - Post(\cdot,t_1^e) + Pre(\cdot,t_1^e) \leq z_1^{e,w}\cdot\vec{K} \\
M_{we} - M_w - Post(\cdot,t_1^e) + Pre(\cdot,t_1^e) \geq -z_1^{e,w}\cdot\vec{K} \\
\quad\vdots \\
M_{we} - M_w - Post(\cdot,t_{n_e}^e) + Pre(\cdot,t_{n_e}^e) \leq z_{n_e}^{e,w}\cdot\vec{K} \\
M_{we} - M_w - Post(\cdot,t_{n_e}^e) + Pre(\cdot,t_{n_e}^e) \geq -z_{n_e}^{e,w}\cdot\vec{K} \\
z_1^{e,w} + \ldots + z_{n_e}^{e,w} = n_e - 1 \\
z_1^{e,w},\ldots,z_{n_e}^{e,w} \in \{0,1\} \\
\qquad\qquad \forall (w,e)\in\mathcal{E} & (a) \\
-K\bar{S}(w,t_j^e) + M_w - Pre(\cdot,t_j^e) \leq -\vec{1} \quad \forall (w,e)\in\mathcal{E} \;:\; |T_e|>1,\ \forall t_j^e\in T_e & (b) \\
\vec{1}^T\bar{S}(w,t_j^e) \leq m - z_j^{e,w} \qquad \forall (w,e)\in\mathcal{E} \;:\; |T_e|>1,\ \forall t_j^e\in T_e & (c) \\
-KS(w,t_j^e) + M_w - Pre(\cdot,t_j^e) \leq -\vec{1} \quad \forall (w,e)\in\mathcal{D},\quad \forall t_j^e\in T_e & (d) \\
\vec{1}^{\,T} S(w,t_j^e) \leq m-1 \qquad \forall (w,e)\in\mathcal{D},\quad \forall t_j^e\in T_e & (e) \\
M_w \in \mathbb{N}^m,\quad \forall w\in\mathcal{L} & (f) \\
Pre, Post \in \mathbb{N}^{m\times n} & (g) \\
S(w,t_j^e)\in\{0,1\}^m & (h) \\
\bar{S}(w,t_j^e)\in\{0,1\}^m & (i)
\end{cases}
\tag{22}
$$

where

$$\mathcal{E} = \{(w,e) \mid w\in\mathcal{L}, |w|<k, we\in\mathcal{L}\},$$
$$\mathcal{D} = \{(w,e) \mid w\in\mathcal{L}, |w|<k, we\notin\mathcal{L}\},$$

and $M_\varepsilon = M_0$.

Proof.

- Assume that $we\in\mathcal{L}$, where $w\in E^*$ and $e\in E$. Then *at least* one transition $t_j^e\in T_e$ should be enabled at M_w, or equivalently, for *at least* one $t_j^e\in T_e$ it should hold:

$$M_w \geq Pre(\cdot,t_j^e).$$

Thus, following again the procedure in Section 3 to convert the logical *or* operator in terms

of linear constraints, we can write:

$$\begin{cases} M_w - Pre(\cdot, t_1^e) \geq -z_1^{e,w} \cdot \vec{K} \\ \vdots \\ M_w - Pre(\cdot, t_{n_e}^e) \geq -z_{n_e}^{e,w} \cdot \vec{K} \\ z_1^{e,w} + \ldots + z_{n_e}^{e,w} = n_e - 1 \\ z_1^{e,w}, \ldots, z_{n_e}^{e,w} \in \{0,1\} \end{cases}$$

If $z_j^{e,w} = 0$ it means that $t_j^e \in T_e$ may fire at M_w, and the marking M_{we} reached after its firing is

$$M_{we} = M_w + Post(\cdot, t_j^e) - Pre(\cdot, t_j^e).$$

that satisfies the following set of linear inequalities:

$$\begin{cases} M_{we} - M_w - Post(\cdot, t_1^e) + Pre(\cdot, t_1^e) \leq z_1^{e,w} \cdot \vec{K} \\ M_{we} - M_w - Post(\cdot, t_1^e) + Pre(\cdot, t_1^e) \geq -z_1^{e,w} \cdot \vec{K} \\ \vdots \\ M_{we} - M_w - Post(\cdot, t_{n_e}^e) + Pre(\cdot, t_{n_e}^e) \leq z_{n_e}^{e,w} \cdot \vec{K} \\ M_{we} - M_w - Post(\cdot, t_{n_e}^e) + Pre(\cdot, t_{n_e}^e) \geq -z_{n_e}^{e,w} \cdot \vec{K} \end{cases}$$

Now, if we want the net to be deterministic, we must impose that, whenever $|T_e| > 1$, only one transition $t_j^e \in T_e$ is enabled at M_w.

From the above constraints we know that transition $t_k^e \in T_e$ such that $z_k^{e,w} = 0$ is enabled at M_w. Thus, we need to impose additional constraints in order to be sure that, for all the other transitions t_j^e, $j \neq k$, for which $z_j^{e,w} = 1$, it holds that

$$M_w - Pre(\cdot, t_j^e) \not\geq \vec{0}.$$

In order to do this, for all $t_j^e \in T_e$ we introduce a vector of binary variables $\bar{S}(w, t_j^e)$ that satisfies the following set of linear inequalities:

$$\begin{cases} -K\bar{S}(w, t_j^e) + M_w - Pre(\cdot, t_j^e) \leq -\vec{1} \\ \vec{1}^T \bar{S}(w, t_j^e) \leq m - z_j^{e,w} \end{cases}$$

If $z_j^{e,w} = 0$, then all entries of $\bar{S}(w, t_j^e)$ may be unitary, thus adding no additional constraint (the corresponding inequality is trivially verified). On the contrary, if $z_j^{e,w} = 1$, then at least one entry of $\bar{S}(w, t_j^e)$ is null, thus making t_j^e not enabled at M_w. Being $z_1^{e,w} + \ldots + z_{n_e}^{e,w} = n_e - 1$, we can be sure that only one transition labeled e is enabled at M_w.

- Assume $w \in \mathcal{L}$ and $we \notin \mathcal{L}$. Then for all $t_j^e \in T_e$ the following set of linear constraints should be satisfied:

$$\begin{cases} -K \cdot S(w, t_j^e) + M_w - Pre(\cdot, t_j^e) \leq -\vec{1}_m \\ \vec{1} \cdot S(w, t_j^e) \leq m - 1 \\ S(w, t_j^e) \in \{0,1\}^m. \end{cases}$$

□

As in the free labeled case, it may be possible to associate to our constraints a performance index to solve an integer programming problem and find, if there exists, the optimal solution.

Example 6.3 Let us now consider a numerical example taken from [21] where $m = n = 3$, $L(t_1) = a$, $L(t_2) = L(t_3) = b$ and the net language is $\mathcal{L}' = \{a^r b^q,\ r \geq q \geq 0\}$.

Assume we want to minimize the sum of initial tokens and the sum of all arcs.

Let us first assume that $k = 3$, thus

$$\mathcal{L} = \{\varepsilon, a, aa, ab, aaa, aab\}.$$

This implies that

$$\mathcal{E} = \{(\varepsilon, a), (a, a), (a, b), (aa, a), (aa, b)\}$$

and

$$\mathcal{D} = \{(\varepsilon, b), (ab, a), (ab, b)\}.$$

The resulting net system is such that

$$M_0 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T,$$

$$Pre = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 2 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad Post = \begin{bmatrix} 0 & 0 & 0 \\ 2 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix},$$

namely that represented in Figure 4.a.

Note that another optimal solution is given by the net in figure (b) if we remove the arc from t_2 to p_1 and the arc from p_3 to t_3.

Then, assume $k = 4$, thus

$$\mathcal{L} = \{\varepsilon, a, aa, ab, aaa, aab, aaaa, aaab, aabb\}.$$

This implies that

$$\mathcal{E} = \{(\varepsilon, a), (a, a), (a, b), (aa, a), (aa, b), (aaa, a), (aaa, b), (aab, b)\}$$

and

$$\mathcal{D} = \{(\varepsilon, b), (ab, a), (ab, b), (aab, a)\}.$$

The resulting net system is such that

$$M_0 = \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T,$$

$$Pre = \begin{bmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \\ 0 & 1 & 1 \end{bmatrix}, \quad Post = \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix},$$

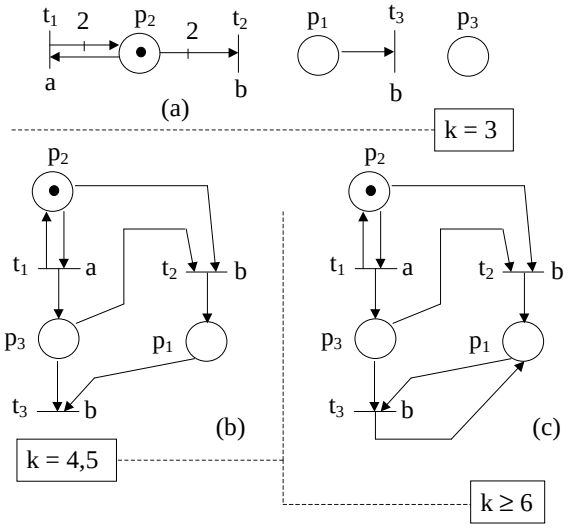

Figure 4: The results of Example 6.3.

namely that represented in Figure 4.b.

The same net system is also obtained if $k = 5$, while the net system in figure (c) is obtained if $k \geq 6$ (that coincides with the net in [21]). ■

Finally, we note that with the technique presented in the previous section we can also lift the requirement that the number of places is known.

It is also possible to deal with the case in which the cardinality of the set T_e for all $e \in E$ is not known a priori but only an upper bound on its value is known. This extension however is not straightforward and it will be left for future research.

7 Complexity of the identification procedure

In this section we discuss the complexity of the IPPs we must solve to identify a net. This complexity is given in terms of number of constraints and number of unknowns. Note however that it is well known that an IPP is an NP-hard problem itself.

7.1 Free labeled nets

Let n be the cardinality of T, k the length of the longest string in $\mathcal{L}$, and ν_r (for $r = 0, \ldots, k$) the number of strings in $\mathcal{L}$ of length r.

Then the constraint set (5) contains $\sum_{r=1}^{k} \nu_r$ constraints of type (a) and $\sum_{r=0}^{k-1}(n\nu_r - \nu_{r+1})$

constraints of type (b) and of type (c). The total number of scalar constraints is thus:

$$c_{\text{free}} = m\left(\sum_{r=1}^{k}\nu_r\right) + (m+1)\left(\sum_{r=0}^{k-1}(n\nu_r - \nu_{r+1})\right).$$

The total number of unknowns is

$$u_{\text{free}} = m + 2(m\times n) + m\left(\sum_{r=0}^{k-1}(n\nu_r - \nu_{r+1})\right).$$

Note that given a value of k and of n, it is possible to find a worst case bound for $\rho = \sum_{r=0}^{k-1}(n\nu_r - \nu_{r+1})$. In fact, it holds:

$$\rho = \textstyle\sum_{r=0}^{k-1}(n\nu_r - \nu_{r+1}) = n\nu_0 + (n-1)\left(\sum_{r=1}^{k-1}\nu_r\right) - \nu_k = n + (n-1)\left(\sum_{r=1}^{k-1}\nu_r\right) - \nu_k.$$

This expression is maximized if we assume $\nu_k = 0$ while all other ν_r must take the largest value, i.e., $\nu_r = n^r$. Hence we have

$$\rho \le n + (n-1)(n + \cdots + n^{k-1}) = n^k,$$

and the total number of unknowns in the worst case is

$$u_{\text{free}} = m + 2(m\times n) + m\ n^k = m(1 + 2n + n^k) = \mathcal{O}(m\ n^k),$$

i.e., it has exponential complexity with respect to k.

7.2 λ-free labeled Petri nets

Let $\tau = \max_{e\in E}|T_e|$, and as we have considered in the previous subsection, k be the length of the longest string in $\mathcal{L}$, and ν_r (for $r = 0,\ldots,k$) be the number of strings in $\mathcal{L}$ of length r.

In the worst case the set (22) has

$$c_{\lambda\text{-free}} = [(4m+1)\tau + 1]\left(\sum_{r=1}^{k}\nu_r\right) + (m+1)\tau\left(\sum_{r=0}^{k-1}(n\nu_r - \nu_{r+1})\right)$$

constraints. Indeed, in such a case, we have $(3m\tau+1)\left(\sum_{r=1}^{k}\nu_r\right)$ constraints of type (a), $(m+1)\tau\left(\sum_{r=1}^{k}\nu_r\right)$ constraints of type (b) plus (c), and $(m+1)\tau\left(\sum_{r=0}^{k-1}(n\nu_r - \nu_{r+1})\right)$ constraints of type (d) and (e).

Moreover, we have that the number of unknowns is

$$u_{\lambda\text{-free}} = m + 2mn + m\left(\sum_{r=1}^{k}\nu_r\right) + \tau\left(\sum_{r=1}^{k}\nu_r\right) + m\tau\left(\sum_{r=1}^{k}\nu_r\right) + m\tau\left(\sum_{r=0}^{k-1}(n\nu_r - \nu_{r+1})\right)$$

where each term corresponds, respectively, to: M_0; Pre and $Post$; M_w; the binary variables $z_j^{e,w}$; the binary vectors $\bar{S}(w, t_j^e)$; the binary vectors $S(w, t_j^e)$.

As shown in the previous subsection we can take:

$$\rho \leq n + (n-1)(n + \cdots + n^{k-1}) = n^k,$$

and then the total number of unknowns in the worst case is

$$u_{\lambda\text{-free}} = \mathcal{O}(m\tau n^k),$$

and keeping in mind that $\tau \leq n$ we can also write

$$u_{\lambda\text{-free}} = \mathcal{O}(mn^{k+1}).$$

Also in this case we have an exponential complexity with respect to k.

8 Conclusions and future work

In this paper we have provided a solution to the problem of identifying a Petri net system that generates a given language, that is based on the solution of IPPs. Both the case of free labeled Petri net systems and the case of λ-free labeled Petri nets are considered. Furthermore we have also considered the problem of synthesizing a net when additional information about the model (structural constraints, conservative components, stationary sequences) or about its initial marking is given. We also treated the problem of synthesizing a bounded net starting from an automaton that generates its language.

Our approach is based on integer programming that is a well accepted methodology for optimization of discrete systems. However we have shown that the computational complexity of the IPPs that describe the problem highly increases with the number of places m, with the number of transitions sharing the same label, and with the length k. This problem may be partially overcome using appropriate heuristics, that compute solutions recursively, with increasing values of k, but that may only provide suboptimal solutions with respect to the chosen performance index. This problem will be the object of our future work in this topic. Another extension of our procedure will be the identification of Petri nets from samples of partially ordered languages [15]. Our main efforts are now devoted to extend the proposed identification procedure to the case of unbounded net systems whose language in completely known.

Acknowledgements

The authors would to thank the anonymous referee that suggested a simple proof for Theorem 5.3.

References

[1] D. Angluin. Inference of reversible languages. *Journal of the ACM*, 29(3):741–765, 1982.

[2] E. Badouel, L. Bernardinello, and P. Darondeau. Polynomial algorithms for the synthesis of bounded nets. *Proceedings of CAAP'95, Lecture Notes in Computer Science*, 915:647–679, 1995.

[3] E. Badouel and P. Darondeau. Theory of regions. *Lecture Notes in Computer Science: Lectures on Petri Nets I: Basic Models*, 1491:529–586, 1998.

[4] Eric Badouel and Philippe Darondeau. On the synthesis of general petri nets. Technical Report 3025, 1996.

[5] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics and constraints. *Automatica*, 35(3):407–429, 1999.

[6] T. Bourdeaud'huy and P. Yim. Synthèse de réseaux de Petri à partir d'exigences. In *Actes de la 5me conf. francophone de Modélisation et Simulation*, pages 413–420, Nantes, France, September 2004.

[7] R.E. Burkard and F. Rendl. Lexicographic bottleneck problems. *Operations Research Letters*, 10:303–308, 1991.

[8] M.P. Cabasino, A. Giua, and C. Seatzu. Identification of deterministic Petri nets. In *Proc. IFAC WODES'06: 8th Work. on Discrete Event Systems*, Ann-Arbor, MI, USA, July 2006.

[9] Jordi Cortadella, Michael Kishinevsky, Luciano Lavagno, and Alexandre Yakovlev. Deriving Petri nets from finite transition systems. *IEEE Transactions on Computers*, 47(8):859–882, 1998.

[10] M. Dotoli, M.P. Fanti, and A.M. Mangini. An optimization approach for identification of Petri nets. In *Proc. IFAC WODES'06: 8th Work. on Discrete Event Systems*, Ann-Arbor, MI, USA, July 2006.

[11] A. Giua and C. Seatzu. Identification of free-labeled Petri nets via integer programming. In *Proc. 44th IEEE Conf. on Decision and Control*, Seville, Spain, December 2005.

[12] E. Mark Gold. Complexity of automaton identification from given data. *Information and Control*, 37(3):302–320, 1978.

[13] K. Hiraishi. Construction of a class of safe Petri nets by presenting firing sequences. In Jensen, K., editor, *Lecture Notes in Computer Science; 13th International Conference on Application and Theory of Petri Nets 1992, Sheffield, UK*, volume 616, pages 244–262. Springer-Verlag, June 1992.

[14] Lingxi Li, Yu Ru, and C. N. Hadjicostis. Least-cost firing sequence estimation in labeled Petri nets. In *Proc. 45th IEEE Conf. on Decision and Control*, San Diego, California USA, December 2006.

[15] R. Lorenz and G. Juhás. Towards synthesis of Petri nets from scenarios. In *Proc. of 27th International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency*, pages 302–321, Turku, Finland, 2006.

[16] M.E. Meda-Campaña and E. López-Mellado. Incremental synthesis of Petri net models for identification of discrete event systems. In *Proc. 41th IEEE Conf. on Decision and Control*, pages 805–810, Las Vegas, Nevada USA, December 2002.

[17] M.E. Meda-Campaña and E. López-Mellado. Required event sequences for identification of discrete event systems. In *Proc. 42th IEEE Conf. on Decision and Control*, pages 3778–3783, Maui, Hawaii, USA, December 2003.

[18] T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.

[19] S. A. Reveliotis and J. Y. Choi. Designing reversibility-enforcing supervisors of polynomial complexity for bounded Petri nets through the theory of regions. In *Proc. of 27th International Conference on Applications and Theory of Petri Nets and Other Models of Concurrency*, pages 322–341, Turku, Finland, 2006.

[20] Y. Ru and C. N. Hadjicostis. State estimation in discrete event systems modeled by labeled Petri nets. In *Proc. 45th IEEE Conf. on Decision and Control*, San Diego, California USA, December 2006.

[21] R.S. Sreenivas. On minimal representations of Petri net languages. In *Proc. WODES'02: 6th Work. on Discrete Event Systems*, pages 237–242, Zaragoza, Spain, October 2002.

[22] J.H. van Schuppen. System theory for system identification. *Journal of Econometrics*, 118(1-2):313–339, January-February 2004.