Identification of deterministic Petri nets

Maria Paola Cabasino, Alessandro Giua, Carla Seatzu

***Abstract*— In a previous paper we presented an approach to identify a Petri net system, given a finite language that it generates. 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.**

In this paper we extend this approach in two ways. Firstly, we consider the case in which the number of places of the net is not given but only an upper bound on its value is known. Secondly, we show how the approach can be extended to the case of deterministic 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.

I. Introduction

In a previous paper [7] we presented a linear algebraic approach for the identification of a Petri net 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 [15].

In the context of Petri nets, the observed behavior is 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 fixed number of places m, the identification problem we considered in [7] consisted 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 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 or equal to k that do not belong to the language.

In this paper we extended this approach in two ways.

Firstly we show that the number of places m needs not be specified exactly, but it is only sufficient to know an upper bound $\bar{m}$ on its value. In this case, we can also 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.

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 {cabasino,giua,seatzu}@diee.unica.it

Secondly, let us observe that the approach presented in [7] applies to *free labeled* nets, i.e., nets where each transition is assigned a unique label: in this case, the set of transitions T coincides with the sets of event labels E. In this paper we extend 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 previously determined 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 from the initial one, there cannot exist transitions with the same label that are simultaneously enabled at M.

The approach we present is extremely general and, unlike other PN identification approaches, can also be applied to λ-free labeled nets: this case has never been considered in the literature to the best of our knowledge. Another original feature is the fact that, by choosing a suitable objective function, it can also be used to determine a minimal net according to a given measure. The main drawback is its computational complexity, in the sense that the number of unknowns grows with the number of counterexamples and (in the case of λ-free labeled nets) with the occurrence of events labeling more that one transition.

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

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 (e.g., we recall the early work of Gold [8] and Angluin [1]).

Some original approaches to the identification of Petri nets have been reviewed in [7]. Among them we recall the work of Hiraishi [9] on safe Petri nets and by Meda and López [10], [11] on free labeled Interpreted Petri nets. Bourdeaud'huy and Yim [4] have presented an approach based on logic constraints that can deal with positive examples of firing sequences but not with counterexamples.

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 [2].

Finally, in a recent paper Sreenivas [14] 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 of the original net while minimizing the given measure. The example we use in Section V is taken from a net discussed in [14].

II. Background on Petri nets

In this section we recall the formalism used in the paper. For more details on Petri nets we address to [12].

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; $Pre : P \times T \to \mathbb{N}$ and $Post : P \times T \to \mathbb{N}$ are the pre– and $post$– incidence functions that specify the arcs; $C = Post - Pre$ is the incidence matrix.

A *marking* is a vector $M : P \to \mathbb{N}$ 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*[1] 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:

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

A. λ-free labeled Petri nets

A *labeling function* $\varphi : T \to E$ assigns to each transition $t \in T$ a symbol from a given alphabet E. Note that the same label $e \in E$ may be associated to more than one transition while no transition may be labeled with the empty string ε[2]. Using the notation of [6] and [13] we say that this labeling function is λ*-free*.

In this paper we use the following notation:

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

where $n_e = |T_e|$. We say that a transition t is *nondeterministic* if its label is also associated to other transitions, otherwise a transition t is said to be *deterministic*. Analogously, we say that an event e is *deterministic* if there exists only one transition t such that $\varphi(t) = e$, otherwise we say that the event e is *nondeterministic*.

Definition 2.1: A Petri net system $\langle N, M_0 \rangle$ with λ-free labeling function $\varphi : T \to E$ is *deterministic* if $\forall\ M \in R(N, M_0)$ and for any two transitions $t, t' \in T$:

$$t \neq t', \ \varphi(t) = \varphi(t'), \ M[t\rangle \implies \neg M[t'\rangle,$$

i.e., if two transitions are labeled with the same symbol they cannot simultaneously be enabled at M. ■

We denote as w the word of events associated to the sequence σ, i.e., $w = \varphi(\sigma)$. Moreover, we denote as ε the empty word associated to the word of null length.

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

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

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

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

III. 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 [3].

A. 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

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

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.

[1] As it will appear in the next subsection, *free* specifies that no labeling function is assigned to the considered the Petri net system.

[2] 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 the labeling function.

B. Equality constraints

Let us now consider the constraint

$$\bigvee_{i=1}^{r} \vec{a}_i = \vec{b}_i \qquad (3)$$

where $\vec{a}_i, \vec{b}_i \in \mathbb{R}^n$, $i = 1, \dots, r$. Equation (3) can be rewritten in terms of linear algebraic constraints as:

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

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

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

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

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

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 $i \in \{1, \dots, r\}$.

IV. Free-labeled Petri nets

A. Previous approach

In [7] we considered the following problem.

***Problem** 4.1:* Assume we are given a set of places $P = \{p_1, \dots, p_m\}$ and a set of transitions $T = \{t_1, \dots, t_n\}$. Let $\mathcal{L} \subset T^*$ be a finite prefix-closed language[3] over T, and

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

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

We want to identify the structure of a net $N = (P, T, Pre, Post)$ and an initial marking M_0 such that

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

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

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

and the elements of the vector

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

■

In [7] we proved that 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:* [7] A solution to the identification problem (4.1) satisfies the following set of linear algebraic constraints

$$\mathcal{G}(m, T, \mathcal{L}) \triangleq$$

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

where

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

and

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

$\vec{\varepsilon}_j$ is the j-th canonical basis vector, and K is a very large constant.

Constraints (a) are the *enabling constraints*, i.e., a transition t_j is enabled at $M_0 + (Post - Pre) \cdot \vec{\sigma}$ if and only if $M_0 + (Post - Pre) \cdot \vec{\sigma} \geq Pre \cdot \vec{\varepsilon}_j$.

Constraints (b) and (c) are the *disabling constraints*: if a transition t_j is disabled at $M_0 + (Post - Pre) \cdot \vec{\sigma}$ then there exists at least one place $p \in P$ such that

$$M_0(p) + (Post(p, \cdot) - Pre(p, \cdot)) \cdot \vec{\sigma} \leq Pre(p, \cdot) \cdot \vec{\varepsilon}_j - 1. \quad (6)$$

Indeed, by constraint (c) at least one entry of $S(\sigma, t_j)$ is null, thus eq. (6) holds for at least one $p \in P$. On the contrary, no constraint is given for the other places to which it correspond a non null entry of $S(\sigma, t_j)$ because in this case constraint (b) is redundant.

In general the set (5) is not a singleton, thus there exists more than one Petri net system $\langle N, M_0 \rangle$ such that $L_k(N, M_0) = \mathcal{L}$. To select one among these Petri net systems we choose a given performance index and solving an appropriate IPP we determine a Petri net system that minimizes the considered performance index[4]. In particular, if $f(m, M_0, Pre, Post)$ is the considered performance

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

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

index, an identification problem can be formally stated as follows.

***Problem** 4.3:* Let us consider the identification problem (4.1) and let $f(m, M_0, Pre, Post)$ be a given performance index. The solution to the identification problem (4.1) that minimizes $f(m, M_0, Pre, Post)$ can be computed by solving the following IPP

$$\left\{\begin{array}{ll} \min & f(m, M_0, Pre, Post) \\ \text{s.t.} & \mathcal{G}(m, T, \mathcal{L}). \end{array}\right. \tag{7}$$

■

As an example, assume we want to determine a Petri net system that minimizes the sum of the tokens in the initial marking and of the arc weights. In such a case we choose

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

B. Optimizing the number of places

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

***Problem** 4.4:* Let us consider an identification problem in the form (4.1) where m is only known to be less or equal to a given value $\bar{m}$, and let $f(m, M_0, Pre, Post)$ be a given performance index. The solution to 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

$$\left\{\begin{array}{lll} \min\limits_{m \le \bar{m}} & \min & f(m, M_0, Pre, Post) \\ \text{s.t.} & & \mathcal{G}(m, T, \mathcal{L}). \end{array}\right. \tag{8}$$

A trivial solution to the above identification problem 4.4 consists in solving IPP of the form (7) 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** 4.5:* Solving the identification problem 4.4 is equivalent to solving the following IPP:

$$\left\{\begin{array}{ll} \min & \bar{K} \cdot \vec{1}_m^T \cdot \vec{z} + f(\bar{m}, M_0, Pre, Post) \\ \text{s.t.} & \mathcal{G}(\bar{m}, T, \mathcal{L}) \\ & K \cdot \vec{z} - Pre \cdot \vec{1}_n - Post \cdot \vec{1}_n \ge \vec{0}_m \\ & z_{i+1} \le z_i, \quad i = 1, \ldots, \bar{m} - 1 \\ & \vec{z} \in \{0,1\}^{\bar{m}} \end{array}\right. \tag{9}$$

for a sufficiently large constant $\bar{K}$.

In particular, let us denote as $\vec{z}^*$, $\bar{M}_0^*$, $\overline{Pre}^*$ and $\overline{Post}^*$ the solution of (9), 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.

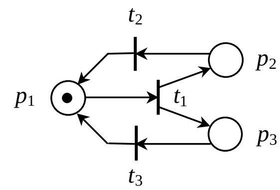

Fig. 1. The Petri net system of Example 4.6.

Then, m^*, M_0^*, Pre^*, $Post^*$ is a solution of the identification problem 4.4.

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

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

is trivially verified being K a very large constant.

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

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

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 (9) is that of minimizing $\vec{1}_m^T \cdot \vec{z}$, the optimal solution $\vec{z}^*$ will have as many zeros as possible, compatibly with the other constraints. Moreover, being $z_{i+1} \le z_i, \quad i = 1, \ldots, \bar{m} - 1$, 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* [5]. 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** 4.6:* 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 (7) where

$$\begin{array}{rl} \mathcal{E} = & \{(\varepsilon, t_1), (t_1, t_2), (t_1, t_3), (t_1t_2, t_1), (t_1t_2, t_3), \\ & (t_1t_3, t_1), (t_1t_3, t_2)\}, \end{array}$$

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 $\bar{m} = 5$.

The procedure identifies the net system in Figure 1. ■

V. 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.

***Problem** 5.1:* 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

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

and $\varphi : T \to E$ be a labeling function over E. 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*[5] net $N = (P, T, Pre, Post)$ labeled by φ and an initial marking M_0 such that

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

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

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

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** 5.2:* A solution to the identification problem 5.1 satisfies the following set of linear algebraic constraints

$$\mathcal{G}(m, T, \mathcal{L}, \varphi) \triangleq$$

$$\left\{\begin{array}{ll}
M_w - Pre(\cdot, t_1^e) \geq -z_1^{e,w} \cdot \vec{K} & \\
\quad\vdots & \\
M_w - Pre(\cdot, t_{n_e}^e) \geq -z_{n_e}^{e,w} \cdot \vec{K} & \\
M_{we} - M_w - Post(\cdot, t_1^e) + Pre(\cdot, t_1^e) \leq z_1^{e,w} \cdot \vec{K} & \\
M_{we} - M_w - Post(\cdot, t_1^e) + Pre(\cdot, t_1^e) \geq -z_1^{e,w} \cdot \vec{K} & \\
\quad\vdots & \\
M_{we} - M_w - Post(\cdot, t_{n_e}^e) + Pre(\cdot, t_{n_e}^e) \leq z_{n_e}^{e,w} \cdot \vec{K} & \\
M_{we} - M_w - Post(\cdot, t_{n_e}^e) + Pre(\cdot, t_{n_e}^e) \geq -z_{n_e}^{e,w} \cdot \vec{K} & \\
z_1^{e,w} + \ldots + z_{n_e}^{e,w} = n_e - 1 & \\
z_1^{e,w}, \ldots, z_{n_e}^{e,w} \in \{0,1\} & \\
\qquad \forall (w,e) \in \mathcal{E} & (a) \\
-K\bar{S}(w, t_j^e) + M_w - Pre(\cdot, t_j^e) \leq -\vec{1} & \\
\qquad \forall (w,e) \in \mathcal{E} \ : \ |T_e| > 1, \ \forall t_j^e \in T_e & (b) \\
\vec{1}^T \bar{S}(w, t_j^e) \leq m - z_j^{e,w} & \\
\qquad \forall (w,e) \in \mathcal{E} \ : \ |T_e| > 1, \ \forall t_j^e \in T_e & (c) \\
-KS(w, t_j^e) + M_w - Pre(\cdot, t_j^e) \leq -\vec{1} & \\
\qquad \forall (w,e) \in \mathcal{D}, \quad \forall t_j^e \in T_e & (d) \\
\vec{1}^{\ T} S(w, t_j^e) \leq m - 1 & \\
\qquad \forall (w,e) \in \mathcal{D}, \quad \forall t_j^e \in T_e & (e) \\
M_w \in \mathbb{N}^m, \quad \forall w \in \mathcal{L} & (f) \\
Pre, Post \in \mathbb{N}^{m \times n} & (g) \\
S(w, t_j^e) \in \{0,1\}^m & (h) \\
\bar{S}(w, t_j^e) \in \{0,1\}^m & (i)
\end{array}\right. \qquad (10)$$

where

$$\begin{array}{l}
\mathcal{E} = \{(w,e) \mid w \in \mathcal{L}, |w| < k, we \in \mathcal{L}\}, \\
\mathcal{D} = \{(w,e) \mid w \in \mathcal{L}, |w| < k, we \notin \mathcal{L}\}, \\
M_\varepsilon = M_0
\end{array}$$

.

Proof: — Assume that $we \in \mathcal{L}$. 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 III to convert the logical *or* operator in terms of linear constraints, we can write:

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

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

If $z^e_{j,w} = 0$ it means that $t^e_j \in T_e$ may fire at M_w, and the marking M_{we} reached after its firing is

$$M_{we} = M_w + Post(\cdot, t^e_j) - Pre(\cdot, t^e_j)$$

that satisfies the following set of linear inequalities:

$$\begin{cases} M_{we} - M_w \\ \qquad -Post(\cdot, t^e_1) + Pre(\cdot, t^e_1) \leq z^e_{1,w} \cdot \vec{K} \\ M_{we} - M_w \\ \qquad -Post(\cdot, t^e_1) + Pre(\cdot, t^e_1) \geq -z^e_{1,w} \cdot \vec{K} \\ \vdots \\ M_{we} - M_w \\ \qquad -Post(\cdot, t^e_{n_e}) + Pre(\cdot, t^e_{n_e}) \leq z^e_{n_e,w} \cdot \vec{K} \\ M_{we} - M_w \\ \qquad -Post(\cdot, t^e_{n_e}) + Pre(\cdot, t^e_{n_e}) \geq -z^e_{n_e,w} \cdot \vec{K} \end{cases}$$

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

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

$$M_w - Pre(\cdot, t^e_j) \ngeq \vec{0}.$$

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

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

If $z^{e,w}_j = 0$, then all entries of $\bar{S}(w, t^e_j)$ may be unitary, thus adding no additional constraint (the corresponding inequality is trivially verified). On the contrary, if $z^{e,w}_j = 1$, then at least one entry of $\bar{S}(w, t^e_j)$ is null, thus making t^e_j not enabled at M_w. Being $z^e_{1,w} + \ldots + z^e_{n_e,w} = 1$, 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 $t^e_j \in T_e$ the following set of linear constraints should be satisfied:

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

Note that, as in the previous case, for determining the value of K it not necessary that the net be K-bounded. It is sufficient to take a value

$$\begin{aligned} K \quad & \geq \max_i M_0(p_i) + k \cdot \max_{i,j} Post(i, j) \\ & \geq \max_i M(p_i) + |w| \cdot \max_{i,j} Post(i, j) \\ & \geq \max_i M_w(p_i). \end{aligned}$$

□

As in the free labeled case, the above linear algebraic characterization enables us to solve identification problems via IPP.

***Problem** 5.3:* Let us consider the identification problem 5.1 and let $f(m, M_0, Pre, Post)$ be a given performance index. The solution to the identification problem 5.1 that minimizes $f(m, M_0, Pre, Post)$ can be computed by solving the following IPP

$$\begin{cases} \min & f(m, M_0, Pre, Post) \\ \text{s.t.} & \mathcal{G}(m, T, \mathcal{L}, \varphi). \end{cases} \quad (11)$$

■

***Example** 5.4:* Let us now consider a numerical example taken from [14] 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

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

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

The resulting net system is that represented in Figure 2.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

$$\begin{aligned} \mathcal{E} = \quad & \{(\varepsilon, a), (a, a), (a, b), (aa, a), (aa, b), \\ & (aaa, a), (aaa, b), (aab, b)\}, \end{aligned}$$

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

The resulting net system is that represented in Figure 2.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 [14]). ■

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. We will left this extension for future research.

Complexity of (11)

Let $\tau = \max_{e \in E} |T_e|$, 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.

In the worst case the set (10) has

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

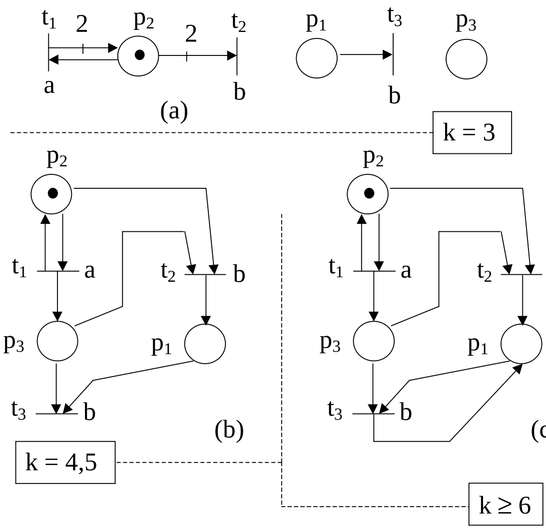

Fig. 2. The results of Example 5.4.

constraints. In fact, 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

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

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 $S(w, t_j^e)$.

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:

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

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

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

and the total number of unknowns in the worst case is

$$u = \mathcal{O}(m\tau n^k),$$

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

$$u = \mathcal{O}(mn^{k+1}).$$

This has exponential complexity with respect to k.

VI. Conclusions and Future Work

In this paper we provided a solution to the problem of identifying a Petri net system that generates a given language, that is based on the solution of appropriate IPP. Both the case of free labeled Petri net systems and the case of λ-free labeled Petri nets are considered.

Our future work in this topic will be twofold.

We plan to derive appropriate heuristics in order to overcome problems related to the computational complexity. Secondly, we plan to characterize some cases in which the knowledge of a finite prefix $L_k(N, M_0)$ — plus eventually some structural properties such as P or T invariants — is guaranteed to univocally identify the net $\langle N, M_0\rangle$ if k is sufficiently large.

References

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

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

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

[4] 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.

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

[6] S. Gaubert and A. Giua. Petri net languages and infinite subsets of $\mathbb{N}^m$. *Journal of Computer and System Sciences*, 59(3):373–391, December 1999.

[7] 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.

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

[9] 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.

[10] 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.

[11] 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.

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

[13] J. L. Peterson. *Petri Net Theory and the Modeling of Systems*. Englewood Cliffs, New Jersey: Prentice Hall, Inc., 1981.

[14] 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.

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