Identification of unbounded Petri nets from their coverability graph*

Maria Paola Cabasino, Alessandro Giua, Carla Seatzu

Dip. Ingegneria Elettrica ed Elettronica, Università di Cagliari,
Piazza d'Armi, 09123 Cagliari, Italy.
Email: {cabasino,giua,seatzu}@diee.unica.it

Abstract

We solve the following problem: given an automaton that represents the coverability graph of a net, determine a net system whose coverability graph is isomorph to the automaton. Our approach requires solving an integer programming problem whose set of unknowns contains the elements of the pre and post incidence matrices and the initial marking of the net.

1 Introduction

In this paper we consider the problem of *identifying* a Petri net system $\langle N, M_0 \rangle$, where N is the net structure and M_0 the initial marking, from the knowledge of its *coverability graph* (CG). More precisely, we present a procedure that given a CG $\mathcal{G}$ determines a net system whose CG is isomorphic to $\mathcal{G}$.

Some original approaches to the identification of a Petri net have been reviewed in [4]. Among them we recall: [5] on safe Petri nets; [8] on Interpreted Petri nets; [2] dealing with logical constraints; and [1] on the theory of regions.

In our previous work on this topic [4], we presented an approach based on *integer programming* to identify a net.

Here we turn our attention to identify unbounded Petri nets. In this case, the language of the net may not be regular and thus it may not be representable using a finite automaton.

A standard technique for the analysis of unbounded nets consists in the construction of a CG [7]. Consider the simple motivating example in Figure 1.(a) that represents an unbounded queue: t_1 represents the arrival of the customers, t_2 the departure of the customers after service. In (b) we have represented the (infinite) reachability graph of the net that describes all possible firing sequences. In (c) we have represented the (finite) CG of the net, where a component labeled ω denotes a place whose token content can grow unbounded.

It is well know that the CG is an automaton that generates a language that is a superset of the net language. As an example, the CG in the figure generates the firing sequences t_1 and $t_1 t_2$ but also strings of the form $t_1 t_2^k$ with $k \geq 2$ that are not firing sequences of the net.

Secondly, we observe that the CG contains structural information on the net that goes beyond the language. The first type of structural information concerns the existence of ω-*increasing* productions: observing the graph one can see that production $\pi : q_0 t_1 q_1$ yields a marking with

*Published as: M.P. Cabasino, A. Giua, C. Seatzu, "Identification of unbounded Petri nets from their coverability graph," CDC06: 45th IEEE Conf. on Decision and Control (San Diego, CA, USA), Dec 2006.

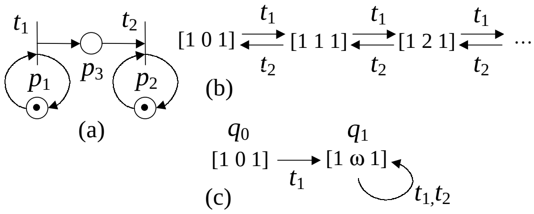

Figure 1: A motivating example.

a number of ω components greater than that of its predecessor. The second type of structural information concerns the existence of *ω-stationary* productions: one can see that production $\pi' : q_1 t_2 q_1$ does not increase the number of ω components and keeps the token content of all other places constant. It is important to stress that this information can be extracted from the unlabeled graph, i.e., from a graph obtained from the one in Figure 1.(c) removing the labels [1 0 1] and [1 ω 1].

Since on one hand the language of a CG does not exactly characterize the net language, but on the other hand the graph contains structural information on the net, we believe it is meaningful in the case of unbounded nets to solve the following problem.

- *Problem 3*: given an automaton $\mathcal{G}$ that represents the CG of a net, determine a net system $\langle N, M_0 \rangle$ whose CG is isomorph to $\mathcal{G}$.

This problem is solved in this paper using an approach based on integer programming. It is well known that integer programming has exponential complexity and it gets harder as the number of integer variables increases. A discussion of the complexity of our approach in terms of number of binary variables and constraints can be found in [3, 4]: since most of the constraints we use in these papers are similar to those used in [3, 4], for sake of brevity we omit this discussion.

Methods based on integer programming have often been frowned upon within the control community. We agree that these methods are unpractical for on-line control computation. However, when used off-line they represent a precious tool and as such they are routinely used to solve large scale problems in all application areas of operations research [10] and several commercial software packages are readily available.

2 Background

In this section we recall the formalism used in the paper. For more details on Petri nets we address to [9].

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'.

A marking M is *reachable* in $\langle N, M_0 \rangle$ iff there exists a firing sequence σ such that $M_0 [\sigma\rangle M$. In such a case it holds: $M = M_0 + C \cdot \vec{\sigma}$, where $\vec{\sigma} \in \mathbb{N}^n$ denotes *firing vector* (i.e., the Parikh vector) of sequence σ. 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 *language* as the set of its firing sequences $L(N, M_0) = \{\sigma \in T^* \mid M_0[\sigma\rangle\}$.

Definition 2.1 A sequence $\sigma \in T^*$ is called *repetitive* if there exists a marking $M \in R(N, M_0)$ such that

$$M_1[\sigma\rangle M_2[\sigma\rangle M_3[\sigma\rangle \cdots \tag{1}$$

i.e., if it can fire infinitely often starting from M_1. It is possible to distinguish two different types of repetitive sequences:

- *stationary* sequence: if in (1) it holds $M_i = M_{i+1}$ for all $i = 1, 2, \ldots$.
- *increasing* sequence: if in (1) it holds $M_i \lneq M_{i+1}$ for all $i = 1, 2, \ldots$.

■

There exists a simple structural condition to characterize repetitive sequences.

Fact 2.2 If sequence σ is enabled at M_1, a necessary and sufficient condition for being repetitive is that in (1) it holds $M_i \leq M_{i+1}$ for all $i = 1, 2, \ldots$, or equivalently $C \cdot \vec{\sigma} \geq \vec{0}$.

Furthermore if $C \cdot \vec{\sigma} = \vec{0}$ the sequence is stationary, else if $C\vec{\sigma} \gneq \vec{0}$ it is increasing. ■

3 Coverability graph and properties

One technique used for the analysis of unbounded Petri nets is based on the construction of the coverability tree/graph (see also [9]).

Algorithm 3.1 ***Construction of the coverability tree for $\langle N, M_0 \rangle$.***

1. *Label the root node q_0 with the initial marking M_0 and tag it "new".*
2. ***While** a node tagged "new" exists **do***
   (a) *Select a node q tagged "new" and let M be its label.*
   (b) ***For** all t enabled at M:*
      i. *Let $M' = M + C(\cdot, t)$ be the marking reached from M firing t.*
      ii. *Let $\bar{q}$ be the first node met on the backward path from q to q_0 whose label is $\bar{M} \lneq M'$. **If** such a node exists **then** for all $p \in P$ such that $M'(p) > \bar{M}(p)$ let $M'(p) = \omega$.*
      iii. *Add a new node q' and label it M'.*
      iv. *Add an arc labeled t from q to q'.*
      v. ***If** there exists already in the tree a node with label M', **then** tag node q' "duplicate", **else** tag it "new".*
   (c) *Untag node q.* ■

From the coverability tree (CT) one can obtain the CG by fusing duplicate nodes with the untagged node with the same label: one can always convert a CT in a graph and viz.

In the construction of the CT the existence of a sequence σ that leads from a marking $\bar{M}$ to a greater marking M' is identified at step 2.(b).ii. The components that by the repeated firing of such a sequence σ grow unbounded are denoted with a special symbol ω that represents infinity[1].

Note that if $\bar{M}$ contains no ω components then σ is an increasing sequence. However, if $\bar{M}$ contains ω components we can only say that σ is increasing for all places p such that $\bar{M}(p) < \omega$: nothing can be said for the remaining places.

Definition 3.2 Let us now consider a node q' labeled with a marking M' that has one or more components changed to ω at step 2.(b).ii of Algorithm 3.1. With the notation used in step 2.(b).ii, we also denote $\bar{q}$ the node covered by q' and q the node father of q'.

- Node q' is called an *ω-increasing node* and the corresponding marking is called an *ω-increasing marking.*
- The production associated to the path on the graph

$$\pi : \ \bar{q} \xrightarrow{\sigma'} q \xrightarrow{t} q' \tag{2}$$

  is called an *ω-increasing production.*
- We also define for any production of the form in eq. (2): $\nu(\pi)$ *the set of all nodes* of the production, $s(\pi) = \bar{q}$ the *start node* of the production, $l(\pi) = q$ the *last-but-one node*, $e(\pi) = q'$ the *end node*, and $\ell(\pi) = \sigma' t$ the corresponding sequence. ■

Example 3.3 *Let us consider the net in Figure 2.(a), whose CT is given in Figure 2.(b). The CG is shown in Figure 2.(c). This net and the successive ones do not have a particular physical meaning; we only use them to demonstrate properties of interest.*

Sequence t_1t_2 is a repetitive sequence that from $M_0 = [1\ 0\ 0]^T$ yields $[1\ 0\ 1]^T$ increasing the marking of place p_3: hence in the CG we have an ω-increasing production $\pi : q_0t_1q_1t_2q_2$ that from $s(\pi) = q_0$ leads to node $e(\pi) = q_2$ with label $M_2 = [1\ 0\ \omega]^T$.

Sequence t_3 from $[1\ 0\ \omega]^T$ yields $[1\ \omega\ \omega]^T$ increasing the marking of place p_2: hence in the CG we have an ω-increasing production $\pi : q_2t_3q_4$ that from $s(\pi) = q_2$ leads to node $e(\pi) = q_4$ with label $M_4 = [1\ \omega\ \omega]^T$ (in this case $l(\pi) = s(\pi)$). Note that although t_3 is an ω-increasing sequence, it is not a repetitive sequence because it decreases the marking of place p_3. ■

Now, let us show how it is possible to recognize those markings that are ω-increasing and the corresponding ω-increasing productions.

Proposition 3.4 In a CG $\mathcal{G}$ of a Petri net a node q' is ω-increasing and π is the corresponding ω-increasing production if and only if there exists in the graph a node $\bar{q} \neq q'$ such that:

$$s(\pi) = \bar{q}, \quad e(\pi) = q'$$

and there exists another production π' such that:

$$s(\pi') = e(\pi') = q', \ \ell(\pi) = \ell(\pi') = \sigma, \ \nu(\pi) \cap \nu(\pi') = q'.$$

Proof. As explained in the construction of the CT, a node q' is labeled by an ω-increasing marking M' if and only if there exists a node $\bar{q}$ labeled by a smaller marking $\bar{M}$ and a sequence σ

[1] For all $n \in \mathbb{N}$ it holds $\omega > n$ and $\omega \pm n = \omega$.

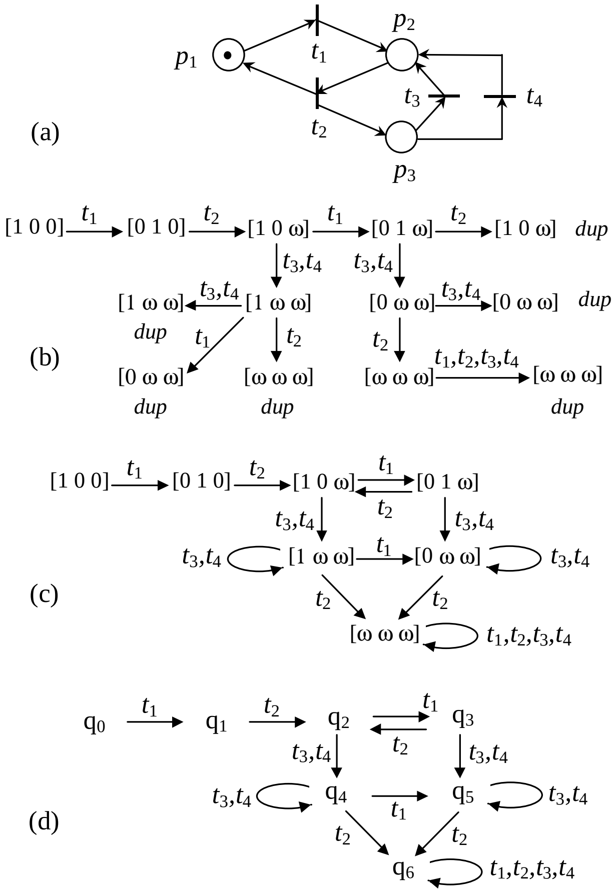

Figure 2: Net in Example 3.3.

such that $\bar{M}[\sigma\rangle\bar{M}'$ where $\bar{M} \lneq \bar{M}'$ and $M'(p) = \omega$ for all those places p such that $\bar{M}'(p) > \bar{M}(p)$, and $M'(p) = \bar{M}'(p)$ for all those places such that $\bar{M}'(p) \neq \omega$. Now since σ is enabled at $\bar{M}$ it is also enabled at the greater marking M'; however since its firing does not change the marking of a place p such that $M'(p) < \omega$, its firing from M' leads back to M'. Finally, we observe that π is $\omega-$increasing iff $e(\pi)$ (hence[2] all nodes in $\nu(\pi')$) contains at least one more $\omega-$component with respect to all nodes in $\nu(\pi) \setminus e(\pi)$; this implies that $\nu(\pi) \cap \nu(\pi') = e(\pi)$. □

This result enables us to introduce the following notation.

- $\mathcal{O}$ is the set of all ω-increasing markings,
- Π is the set of ω-increasing productions,
- Π_k is the set of ω-increasing productions that end in q_k.

Example 3.5 *From the graph in Figure 2.(d) we recognize the following nodes as associated to ω-increasing markings:*

- q_2. *With the notation used in the proof of the previous proposition, $\bar{q} = q_0$ and the corresponding ω-increasing production is $\pi = q_0t_1q_1t_2q_2$.*
- q_4 *(resp., q_5). Here $\bar{q} = q_2$ (resp., $\bar{q} = q_3$) and the two corresponding ω-increasing productions are $\pi = q_2t_3q_4$ or $\pi' = q_2t_4q_4$ (resp., $\pi = q_3t_3q_5$ or $\pi' = q_3t_4q_5$).*
- q_6. *We have two choices for $\bar{q}$: $\bar{q} = q_4$ and $\bar{q} = q_5$. The corresponding ω-productions are $\pi = q_4t_2q_6$ and $\pi' = q_5t_2q_6$.* ■

4 Equivalence classes

In this section we introduce some equivalence classes that will be useful in the identification procedure.

Definition 4.1 (Set of ω components) Given a node q of a CG we define

$$\Omega(q) = \{p \in P \mid q \text{ is labeled with } M, M(p) = \omega\}$$

the set of places associated to ω components in node q. ■

The set of ω components induces two relations on the nodes of a CG.

Definition 4.2 Given two nodes q and q' we say that:

- $q \equiv q'$ if $\Omega(q) = \Omega(q')$.
- $q \preceq q'$ (resp., $q \prec q'$) if $\Omega(q) \subseteq \Omega(q')$ (resp., $\Omega(q) \subsetneq \Omega(q')$). ■

One can immediately verify that the first one is an equivalence relation while the second one is a partial order relation. It is thus possible to partition the set of nodes according to the equivalence classes of $\equiv$ and to order them according to $\prec$.

Our identification procedure requires at first to partition the nodes of the graph into equivalence classes for the $\equiv$ relation (we call this partition $\mathcal{Q}$) and to order them. We first observe that although the unlabeled graph does not contain enough information to exactly reconstruct such a partition, it allows one to determine a partition $\hat{\mathcal{Q}}$ that refines[3] $\mathcal{Q}$.

[2] This result is formally proved by Proposition 4.4.

[3] Partition $\hat{\mathcal{Q}}$ refines partition $\mathcal{Q}$ iff for all q it holds $\hat{\mathcal{Q}}[q] \subseteq \mathcal{Q}[q]$, where $\mathcal{Q}[q]$ denotes the class that contains q.

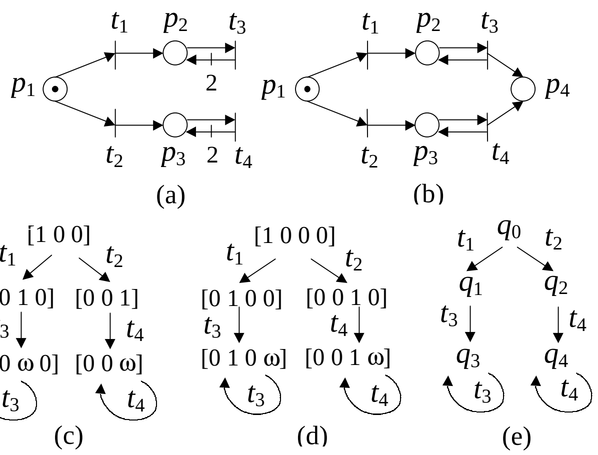

Figure 3: Nets in Example 4.3.

Example 4.3 *Let us consider the nets in Figures 3.(a) and (b). The CG of these nets are given in (c) and (d), respectively, while the unlabled graph is reported in (e) and is the same in the two cases. In the case of the net in Figure 3.(a) the equivalence classes are:* $\mathcal{Q}_0 = \{q_0, q_1, q_2\}$, $\mathcal{Q}_1 = \{q_3\}$, $\mathcal{Q}_2 = \{q_4\}$, *while in the case of the net in Figure 3.(b) the equivalence classes are:* $\mathcal{Q}_0 = \{q_0, q_1, q_2\}$, $\mathcal{Q}_1 = \{q_3, q_4\}$.

Thus in this case we cannot exactly reconstruct the equivalence classes by simply looking at the unlabeled graph. In particular, our algorithm always finds the final partition of the CG of the net in Figure 3.(a), that is a refinement of the CG of the net in Figure 3.(b). ■

The following elementary observation also holds.

Proposition 4.4 Let us consider two nodes q_i and q_j such that

$$q_i \xrightarrow{t} q_j$$

then the following results hold:

(i) $q_i \preceq q_j$,

(ii) $q_i \prec q_j$ iff $\exists\ \pi \in \Pi_j$ and $l(\pi) = q_i$.

Proof. (i) Assume nodes q_i and q_j are labeled, respectively, M and M'. In the construction of the CT whenever a marking M is such that $M(p) = \omega$, then the firing of an enabled transition t from M leads to $M' = M + C(\cdot, t)$, hence $M'(p) = M(p) + C(p, t) = \omega + C(p, t) = \omega$.

(ii) Follows from the fact that an ω is introduced in the graph only by an ω-increasing production. □

This means that along any path of a CG the nodes that one encounters are ordered with respect to (wrt) $\preceq$. Moreover, the number of ω-components only increases when reaching an ω-increasing marking from an ω-increasing sequence.

We can finally state the procedure to partition an unlabeled CG in equivalence classes. In the following algorithm we will say that given two subsets of nodes $\hat{\mathcal{Q}}_i$ and $\hat{\mathcal{Q}}_j$ (with $i \neq j$) the following predicate holds:

- $c(i,j)$: **if** there exist two nodes $q' \in \hat{\mathcal{Q}}_i$ and $q'' \in \hat{\mathcal{Q}}_j$ such that $\delta(q',t) = q''$ for some transition $t \in T$ **and** it does not exist an ω-increasing production $\pi \in \Pi$ such that $l(\pi) = q'$ and $e(\pi) = q''$.

Algorithm 4.5 ***Partition of an unlabeled coverability tree***

1. *Consider an initial partition of the graph in strongly connected components,*

$$\hat{\mathcal{Q}}_0 \cup \hat{\mathcal{Q}}_1 \cup \cdots \cup \hat{\mathcal{Q}}_k$$

where $\hat{\mathcal{Q}}_0$ is the component containing the initial node q_0 and $k+1$ is the number of such components.

2. ***While** there exist $\hat{\mathcal{Q}}_i$ and $\hat{\mathcal{Q}}_j$ such that $c(i,j)$ **do** merge $\hat{\mathcal{Q}}_i$ and $\hat{\mathcal{Q}}_j$.*

3. *The final partition is*

$$\hat{\mathcal{Q}}_0 \cup \hat{\mathcal{Q}}_1 \cup \cdots \cup \hat{\mathcal{Q}}_r, \qquad r \leq k.$$

■

Example 4.6 Consider the graph in Figure 2.(d). The initial partition is $\hat{\mathcal{Q}}_0 = \{q_0\}$, $\hat{\mathcal{Q}}_1 = \{q_1\}$, $\hat{\mathcal{Q}}_2 = \{q_2, q_3\}$, $\hat{\mathcal{Q}}_3 = \{q_4\}$, $\hat{\mathcal{Q}}_4 = \{q_5\}$ and $\hat{\mathcal{Q}}_5 = \{q_6\}$.

At step 2 of the algorithm we merge $\hat{\mathcal{Q}}_0$ with $\hat{\mathcal{Q}}_1$ and $\hat{\mathcal{Q}}_3$ with $\hat{\mathcal{Q}}_4$. The resulting final partition is $\hat{\mathcal{Q}}_0 = \{q_0, q_1\}$, $\hat{\mathcal{Q}}_1 = \{q_2, q_3\}$, $\hat{\mathcal{Q}}_2 = \{q_4, q_5\}$, $\hat{\mathcal{Q}}_3 = \{q_6\}$. ■

In the previous example the algorithm determines the exact partition $\mathcal{Q}$ in equivalence classes. In general the following result holds.

Proposition 4.7 *The partition $\hat{\mathcal{Q}}$ determined by Algorithm 4.5 is a refinement of the partition $\mathcal{Q}$ in equivalence classes for the $\equiv$ relation.*

Proof. We first note that the initial partition refines $\mathcal{Q}$. In fact, according to Proposition 4.4.i if two nodes q_k and q_j belong to the same strongly connected component, then $q_k \preceq q_j$ and $q_j \preceq q_k$, hence $q_k \equiv q_j$.

Secondly, we observe that the classes that are merged at step 2 of the algorithm belong to the same equivalence class being joined by a transition that is not the terminal path of an ω-increasing production, hence no component is changed to ω according to Proposition 4.4.ii. □

We finally introduce the notion of ω-stationary sequence.

Definition 4.8 A sequence σ is *ω-stationary* wrt $P' \subseteq P$ if $C \cdot \vec{\sigma} \uparrow_{P'} = \vec{0}$. ■

Proposition 4.9 In a CG $\mathcal{G}$ of a Petri net, a production π with $s(\pi) = q$ corresponds to a sequence ω-stationary wrt $\Omega(q)$ iff $e(\pi) = q$.

Proof. Follows from the previous definition, because a sequence σ is ω-stationary wrt $\Omega(q)$ if and only if it does not modify the token content of places that are not associated to ω components, i.e., if and only if $e(\pi) = q$. □

A production π such that $\sigma = \ell(\pi)$ is a stationary sequence, is an ω-stationary production. However, there may also exist ω-stationary productions that do not correspond to stationary sequences.

Example 4.10 Let us consider the net in Figure 2. Here $\sigma = t_3$ is an ω-stationary sequence wrt $\Omega(q_4) = \Omega(q_5) = \{p_2, p_3\}$ and wrt $\Omega(q_6) = \{p_1, p_2, p_3\}$. The firing of t_3 from any of these nodes corresponds to a cycle.

■

5 Synthesis of a PN system from its unlabeled coverability graph

Problem 5.1 *Let $\mathcal{G} = (Q, T, \delta, q_0)$ be a given finite state automaton. Chosen a set of places P of cardinality m, we want to identify the structure of a free-labeled Petri net $N = (P, T, Pre, Post)$ and an initial marking M_0 such that the CG of $\langle N, M_0 \rangle$ is isomorphic to $\mathcal{G}$.*

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 $M_0 \in \mathbb{N}^m$. ■

In this section we provide a set of linear algebraic constraints and we prove that a net system $\langle N, M_0 \rangle$ is a solution of Problem 5.1 iff it satisfies the given set of constraints. The proof will be mostly discursive, in the sense that we explain the meaning of each type of constraints.

In the previous section we have characterized the information on the net that can be extracted from the CG in terms of its language and of ω-increasing and ω-stationary sequences. However, to ensure that the synthesized net has a CG isomorphic to the given one, it is also necessary to impose two additional types of constraints.

The first type of constraints requires that if in the graph two sequences σ_k and σ'_k lead to the same node q_k, then for all places $p \notin \Omega(q_k)$ it should hold

$$M_0(p) + C(p, \cdot) \cdot \vec{\sigma}_k = M_0(p) + C(p, \cdot) \cdot \vec{\sigma}'_k. \tag{3}$$

To do this we introduce the following definition.

Definition 5.2 Given a node $q_k \in Q$ we denote π_k a minimal[4] production starting from q_0 and ending in q_k. We also denote $\sigma_k = \ell(\pi_k)$ the associated sequence and $\vec{\sigma}_k$ the corresponding firing vector, that will be used to represent the marking $M_k = M_0 + C \cdot \vec{\sigma}_k$ associated to node q_k.

Sequence σ_k will be used to identify node q_k while other sequences σ'_k yielding the same marking will have to satisfy (3).

The second condition is the dual of the previous one. Assume that the graph contains two nodes q_k and q_j that are equivalent in the sense that all productions that start from them cannot be distinguished neither in terms of language nor in terms of ω-increasing nor of ω-stationary sequences. To make sure that in the graph of the synthesized net these two nodes are not collapsed into a single one, we need to specify that either the two nodes belong to two different classes, i.e., $\Omega(q_k) \neq \Omega(q_j)$, or they differ for at least a component different from ω, i.e., there exists $p \notin \Omega(q_k) \cup \Omega(q_j)$ such that

$$M_0(p) + C(p, \cdot) \cdot \vec{\sigma}_k \neq M_0(p) + C(p, \cdot) \cdot \vec{\sigma}_j. \tag{4}$$

The nodes that must be distinguished are the nodes that satisfy the following notion of bisimilarity.

Definition 5.3 Given a finite state automaton $\mathcal{G} = (Q, T, \delta, q_0)$. Let $q,\ q' \in Q$. We say that q is simulated by q' if the following conditions hold.

- *(Language equivalence.)* $\delta(q, \sigma)$ is defined $\Rightarrow$ $\delta(q', \sigma)$ is defined, for any $\sigma \in T^*$.
- *(ω-increasing equivalence.)* If π_1 is a production and π_2 is an ω-increasing production such that $s(\pi_1) = q$ and $e(\pi_1) = s(\pi_2)$ $\Rightarrow$ there exists a production π'_1 and an ω-increasing production π'_2 such that $s(\pi'_1) = q'$, $e(\pi'_1) = s(\pi'_2)$, $\ell(\pi_1) = \ell(\pi'_1)$ and $\ell(\pi_2) = \ell(\pi'_2)$.

[4] By minimal we mean that the production does not contain twice the same node. More than one such production may exist: we arbitrarily chose one.

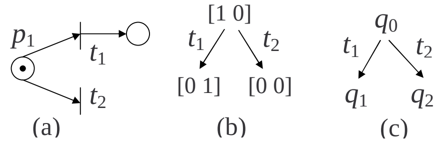

Figure 4: The resulting net in Example 5.4.

- *(ω-stationary equivalence.)* If π_1 is a production and π_2 is an ω-stationary production such that $s(\pi_1) = q$ and $e(\pi_1) = s(\pi_2) \Rightarrow$ there exists a production π_1' and an ω-stationary production π_2' such that $s(\pi_1') = q'$, $e(\pi_1') = s(\pi_2')$, $\ell(\pi_1) = \ell(\pi_1')$ and $\ell(\pi_2) = \ell(\pi_2')$.

We say that $q,\ q' \in Q$ are bisimilar if q is simulated by q' and q' is simulated by q. ■

Example 5.4 *Let us consider the net system in Figure 4.(a) whose CG is shown in (b), while the unlabeled graph is reported in (c). It is immediate to verify that nodes q_1 and q_2 are bisimilar.* ■

We finally introduce the notation to describe enabling and disabling of transitions following [4].

- $\mathcal{E} = \{(q,t) \in Q \times T \mid \delta(q,t)$ is defined$\}$ is the set of couples (state q – transition t) such that t is enabled at the state q of $\mathcal{G}$.
- $\mathcal{D} = \{(q,t) \in Q \times T \mid \delta(q,t)$ is not defined$\}$ is the set of couples (state q – transition t) such that t is *not* enabled at the state q of $\mathcal{G}$.

The following theorem characterizes the set of solutions to Problem 5.1.

Theorem 5.5 *A net system $\langle N, M_0\rangle$ is a solution of the identification problem 5.1 if and only if it satisfies the following linear algebraic constraints.*

(a) Enabling constraints: $\forall\ (q_k, t_j) \in \mathcal{E}$ *let*

$$M_0 + C \cdot \vec{\sigma}_k + K \cdot \vec{s}_{c(k)} \geq Pre \cdot \vec{\varepsilon}_j$$

where $\vec{\varepsilon}_j$ is the j-th canonical basis vector and $\vec{\sigma}_k$ is chosen as in Definition 5.2.

(b) Constraints related to ω-increasing sequences: $\forall\ q_k \in \mathcal{O}$ *and* $\forall\ \pi \in \Pi_k$ *let*

$$\begin{cases} C \cdot \vec{\sigma} - \vec{s}_{c(k)} + K \cdot \vec{s}_{c(i)} \geq \vec{0} & (b1) \\ -C \cdot \vec{\sigma} + K \cdot \vec{s}_{c(k)} \geq \vec{0} & (b2) \\ \vec{s}_{c(k)} \geq \vec{s}_{c(i)} & (b3) \\ \vec{1}^T \cdot \vec{s}_{c(k)} > \vec{1}^T \cdot \vec{s}_{c(i)} & (b4) \end{cases}$$

where $\vec{\sigma}$ is the firing vector associated to the generic production π and $q_i = l(\pi)$.

(c) Constraints related to ω-stationary sequences: $\forall i = 0, 1, \ldots, r$ *and* $\forall \vec{\sigma} \in \mathcal{S}_i = \{\vec{\sigma} \in \mathbb{N}^n \mid \exists \pi : \nu(\pi) \subseteq \hat{\mathcal{Q}}_i, s(\pi) = e(\pi), \ell(\pi) = \sigma\}$

$$\begin{cases} C \cdot \vec{\sigma} + K \cdot \vec{s}_i \geq \vec{0} & (c1) \\ -C \cdot \vec{\sigma} + K \cdot \vec{s}_i \geq \vec{0} & (c2) \end{cases}$$

where $\mathcal{S}_i$ is the set of firing vectors corresponding to cycles in component $\hat{\mathcal{Q}}_i$.

(d) Blocking constraints: $\forall\ (q_k, t_j) \in \mathcal{D}$ *let*

$$\begin{cases} M_0 + C \cdot \vec{\sigma}_k - K \cdot \vec{s}_{k,j} < Pre \cdot \vec{\varepsilon}_j & (d1) \\ \vec{1}^T \cdot \vec{s}_{k,j} \leq m - 1 & (d2) \\ \vec{s}_{k,j} \geq \vec{s}_{c(k)} & (d3) \end{cases}$$

(e) Equivalence constraints: *Assume that the minimal production reaching q_k, as in Definition 5.2, is $\pi_k = q_0 \longrightarrow \cdots \longrightarrow q_{r(k)} \stackrel{t_{r(k)}}{\longrightarrow} q_k$. Then we define $\mathcal{I}(q_k) = \{(q,t) \mid q \stackrel{t}{\longrightarrow} q_k \wedge (q,t) \neq (q_{r(k)}, t_{r(k)})\}$. In other words $\mathcal{I}(q_k)$ is the set of couples (state q - transition t) that lead to q_k except the couple $(q_{r(k)}, t_{r(k)})$ that has already been considered in constraints (a).*

$\forall (q_i, t) \in \mathcal{I}(q_k)$ *let*

$$\begin{cases} C \cdot \vec{\sigma}_k - C \cdot \vec{\sigma} + K \cdot \vec{s}_{c(k)} \geq \vec{0} & (e1) \\ -C \cdot \vec{\sigma}_k + C \cdot \vec{\sigma} + K \cdot \vec{s}_{c(k)} \geq \vec{0} & (e2) \end{cases}$$

where $\sigma = \sigma_i t$, and σ_i is the sequence associated to node q_i, as in Definition 5.2.

(f) Discriminating constraints: *for all bisimilar nodes q_k, q_j that belong to the same equivalence class $\hat{\mathcal{Q}}_i$, let*

$$\begin{cases} C \cdot \vec{\sigma}_k - C \cdot \vec{\sigma}_j + K \cdot \vec{l}_{j,k} \geq \vec{1} & (f1) \\ C \cdot \vec{\sigma}_k - C \cdot \vec{\sigma}_j - K \cdot \vec{l}_{k,j} \leq -\vec{1} & (f2) \\ \vec{1}^T \cdot \left(\vec{l}_{j,k} + \vec{l}_{k,j}\right) \leq 2m - 1 & (f3) \\ \vec{l}_{j,k},\ \vec{l}_{k,j} \geq \vec{s}_i & (f4) \\ \vec{l}_{k,j},\ \vec{l}_{j,k} \in \{0,1\}^m & (f5) \end{cases}$$

(g) Discriminating constraints: *for all bisimilar nodes q_k, q_j that belong to equivalence classes among which an ordering does not exist, let*

$$\begin{cases} \vec{s}_{c(k)} - \vec{s}_{c(j)} + K \cdot \vec{v}_{j,k} + K \cdot z_1 \cdot \vec{1} \geq \vec{1} & (g1) \\ \vec{s}_{c(k)} - \vec{s}_{c(j)} - K \cdot \vec{v}_{k,j} + K \cdot z_1 \cdot \vec{1} \leq -\vec{1} & (g2) \\ C \cdot \vec{\sigma}_k - C \cdot \vec{\sigma}_j + K \cdot \vec{l}_{j,k} + K \cdot z_2 \cdot \vec{1} \geq \vec{1} & (g3) \\ C \cdot \vec{\sigma}_k - C \cdot \vec{\sigma}_j - K \cdot \vec{l}_{k,j} - K \cdot z_2 \cdot \vec{1} \leq -\vec{1} & (g4) \\ \vec{1}^T \cdot (\vec{v}_{j,k} + \vec{v}_{k,j}) \leq 2m - 1 & (g5) \\ \vec{1}^T \cdot \left(\vec{l}_{j,k} + \vec{l}_{k,j}\right) \leq 2m - 1 & (g6) \\ \vec{l}_{j,k},\ \vec{l}_{k,j} \geq \vec{s}_{c(k)} & (g7) \\ \vec{l}_{j,k},\ \vec{l}_{k,j} \geq \vec{s}_{c(j)} & (g8) \\ \vec{l}_{k,j},\ \vec{l}_{j,k}, \vec{v}_{k,j},\ \vec{v}_{j,k} \in \{0,1\}^m & (g9) \\ z_1 + z_2 \leq 1 & (g10) \\ z_1,\ z_2 \in \{0,1\} & (g11) \end{cases}$$

(h) Integrity constraints.

$$\begin{cases} M_0 \in \mathbb{N}^m & (h1) \\ C = Post - Pre & (h2) \\ Pre, Post \in \mathbb{N}^{m \times n} & (h3) \\ \vec{s}_i \in \{0,1\}^m,\ \forall\ \hat{\mathcal{Q}}_i & (h4) \\ \vec{s}_{k,j} \in \{0,1\}^m,\ \forall\ (q_k, t_j) \in \mathcal{D} & (h5) \end{cases}$$

In the following we denote as $\mathcal{C}(\mathcal{G}, P)$ the set of constraints (a) to (h) associated to the unlabeled graph $\mathcal{G}$ and to the set of places P.

Proof. We just present a sketch of the proof.

• *Constraints (a).* To each equivalence class $\hat{\mathcal{Q}}_i$ we associate a vector $\vec{s}_i \in \{0,1\}^m$ such that $s_i(p) = 1 \quad \Leftrightarrow \quad \forall\, q \in \hat{\mathcal{Q}}_i,\ p \in \Omega(q)$.

If $(q_k, t_j) \in \mathcal{E}$, then the marking $M_k = M_0 + C \cdot \vec{\sigma}_k$ is such that $M_k(p) \geq Pre(p,t)$ for all p such that $s_{c(k)}(p) = 0$. On the contrary, if $s_{c(k)}(p) = 1$ regardless of the value of $M_k(p)$, t_j is enabled.

• *Constraints (b).* Let q_k be any node in $\mathcal{O}$, π a production in Π_k, and $q_i = l(\pi)$.

We first observe that if there exists a place $p \in \Omega(q_i)$, then by Proposition 3.4 $p \in \Omega(q)$ for any $q \in \hat{\mathcal{Q}}_{c(k)}$. Thus, $\vec{s}_{c(i)} \geq \vec{s}_{c(k)}$.

Therefore, if $s_{c(i)}(p) = 1$ then $s_{c(k)}(p) = 1$, and constraints (b1) and (b2) are trivially verified.

If $s_{c(i)}(p) = 0$ it may either be $s_{c(k)}(p) = 0$ or $s_{c(k)}(p) = 1$. In the first case the firing of the sequence associated to π does not increase the token contents of p as imposed by constraints (b1) and (b2). In the second case, it must hold $C(p,\cdot) \cdot \vec{\sigma} > 0$ that is equivalent to impose constraint (b1), while (b2) is trivially verified.

• *Constraints (c).* Let π be a production whose characteristic vector is in $\mathcal{S}_i$, namely a production relative to an ω-stationary sequence for $\hat{\mathcal{Q}}_i$.

By definition π should not change the content of all places $p \in \Omega(q)$ for any $q \in \hat{\mathcal{Q}}_i$, while no constraint should be imposed on the other places. This is actually the meaning of constraints (c). In fact, if $s_i(p) = 0$, then $C(p,\cdot) \cdot \vec{\sigma} = 0$; if $s_i(p) = 1$, then constraints (c) are trivially verified.

• *Constraints (d).* If transition t_j is not enabled at $M_k = M_0 + C \cdot \vec{\sigma}_k$, then for at least one place p it must hold $M_k(p) < Pre \cdot \vec{\varepsilon}_j$.

We now define a vector $\vec{s}_{k,j} \in \{0,1\}^m$ such that $s_{k,j}(p) = 0 \quad \Leftrightarrow \quad M_k(p) < Pre \cdot \vec{\varepsilon}_j$.

Assume that each component of M_k is less or equal to K. Then the component of $\vec{s}_{k,j}$ relative to the generic place p must satisfy the equation

$$M_k(p) - K \cdot s_{k,j}(p) < Pre \cdot \vec{\varepsilon}_j, \tag{5}$$

so that if $s_{k,j}(p) = 0$ it must hold $M_k(p) < Pre \cdot \vec{\varepsilon}_j$, while if $s_{k,j}(p) = 1$, equation (5) is trivially verified. In vector form (and taking into account that all variables are integers) this equation rewrites as (d1).

Note that there exists at least one place that disables t_j if $\vec{1}^T \cdot \vec{s}_{k,j} \leq m - 1$ so that at least one $s_{k,j}(p)$ is null.

Finally, the constraint $\vec{s}_{k,j} \geq \vec{s}_{c(k)}$ imposes that, if $s_{c(k)}(p) = 1$ then $s_{k,j}(p) = 1$. That is to say , t_j cannot be disabled by a place $p \in \Omega(q_k)$.

• *Constraints (e).* Assume that there exists a production π that, as π_k, reaches node q_k from q_0. Assume also that $\pi = q_0 \longrightarrow \ldots \longrightarrow q_i \stackrel{t}{\longrightarrow} q_k$ with $(q_i, t) \in \mathcal{I}(q_k)$. Then for all places $p \notin \Omega(q_k)$ it should be $C(p,\cdot) \cdot \vec{\sigma}_k = C(p,\cdot) \cdot \vec{\sigma}$, while for the other places no relationship can be deduced from the CG. This is exactly the meaning of constraints (e1) and (e2). In fact, if $s_{c(k)}(p) = 0$, then $C(p,\cdot) \cdot \vec{\sigma}_k = C(p,\cdot) \cdot \vec{\sigma}$; otherwise we get two constraints that are trivially verified.

• *Constraints (f).* Let q_k and q_j be two nodes that are bisimilar and that belong to the same equivalence class $\hat{\mathcal{Q}}_i$. To distinguish between these nodes we impose that there exists at least one place p, to which it does not correspond ω in $\hat{\mathcal{Q}}_i$, such that $C(p,\cdot) \cdot \vec{\sigma}_k \neq C(p,\cdot) \cdot \vec{\sigma}_j$.

We define two vectors $\vec{l}_{k,j},\ \vec{l}_{j,k} \in \{0,1\}^m$ such that $\vec{l}_{k,j},\ \vec{l}_{j,k} \geq \vec{s}_i$.

If $l_{j,k}(p) = l_{k,j}(p) = 1$ constraints (f1) and (f2) are trivially verified, and this occurs for all places $p \in \Omega(q_k) = \Omega(q_j)$ being $\vec{l}_{j,k},\ \vec{l}_{k,j} \geq \vec{s}_i$. If $l_{j,k}(p) = 1$ and $l_{k,j}(p) = 0$ then $C(p,\cdot) \cdot \vec{\sigma}_k \leq C(p,\cdot) \cdot \vec{\sigma}_j - 1$. If $l_{k,j}(p) = 1$ and $l_{j,k}(p) = 0$ then $C(p,\cdot) \cdot \vec{\sigma}_k \geq C(p,\cdot) \cdot \vec{\sigma}_j + 1$.

Note that one of the above two cases always occur being by (f3), $\vec{1}^T \cdot \left(\vec{l}_{j,k} + \vec{l}_{k,j}\right) \leq 2m - 1$.

• *Constraints (g).* Let q_k and q_j be two nodes that are bisimilar and that belong to equivalence classes among which an ordering does not exist. To distinguish between these nodes we impose that at least one of the following conditions hold: (I) q_k and q_j differ in a place not containing ω, (II) $\Omega(q_k) \neq \Omega(q_j)$, i.e., $\vec{s}_{c(k)} \neq \vec{s}_{c(j)}$.

Now, being $z_1 + z_2 \leq 1$, with $z_1,\ z_2 \in \{0, 1\}$, three different cases may occur: $z_1 = 1$ and $z_2 = 0$, $z_1 = 0$ and $z_2 = 1$, $z_1 = z_2 = 0$.

— Assume that $z_1 = 1$ and $z_2 = 0$. In such a case constraints (g1) and (g2) are trivially verified, and the only significant constraints are (g3), (g4), (g6) to (g8) that are analogous to constraints (f): they impose that q_k and q_j differ in a place not containing ω (case (I) above).

— Assume that $z_1 = 0$ and $z_2 = 1$. In such a case constraints (g3) and (g4) are trivially verified, and the only significant constraints are (g1), (g2) and (g5). Using the same reasoning as above, it is easy to verify that these constraints impose that $s_{c(k)}(p) \neq s_{c(j)}(p)$ for at least one place $p \in P$ (case (II) above).

— Assume that $z_1 = z_2 = 0$. In such a case no constraint in (g) is trivial, thus q_k and q_j have ω in different places, and they also differ in some place not containing ω. □

Example 5.6 *Let us consider the unlabeled graph $\mathcal{G}$ in Figure 2.(d). We want to determine a net system $\langle N, M_0 \rangle$ with $N = (P, T, Pre, Post)$ and $m = 3$, whose reachability graph is isomorphic to $\mathcal{G}$. In particular, we want to minimize the tokens in the initial marking and the arc weights.*

The set of ω-increasing nodes is $\mathcal{O} = \{q_2, q_3, q_4, q_5, q_6\}$. The set of ω-increasing productions that ends in q_2 is $\Pi_2 = \{\pi_2\}$, with $\ell(\pi_2) = t_1 t_2$. Then $\Pi_4 = \{\pi_4', \pi_4''\}$, with $\ell(\pi_4') = t_3$ and $\ell(\pi_4'') = t_4$; $\Pi_5 = \{\pi_5', \pi_5''\}$, with $\ell(\pi_5') = t_3$ and $\ell(\pi_5'') = t_4$; $\Pi_6 = \{\pi_6\}$ with $\ell(\pi_6) = t_2$.

*We only have ω-stationary productions associated to the equivalence class $\mathcal{Q}_3 = \{q_6\}$ and the set of firing vectors is $\mathcal{S}_6 = \{\vec{\sigma}_6', \vec{\sigma}_6'', \vec{\sigma}_6'''\}$ where $\sigma_6' = t_1, \sigma_6'' = t_3, \sigma_6''' = t_4$. Then, $\mathcal{E} = \{(q_0, t_1), (q_1, t_2), (q_2, t_1), (q_2, t_3), (q_2, t_4), (q_3, t_2), (q_3, t_3), (q_3, t_4), (q_4, t_1), (q_4, t_2), (q_4, t_3), (q_4, t_4), (q_5, t_2), (q_5, t_3), (q_5, t_4),$
$(q_6, t_1), (q_6, t_2), (q_6, t_3), (q_6, t_4)\}$, and $\mathcal{D} = \{(q_0, t_2), (q_0, t_3), (q_0, t_4), (q_1, t_1), (q_1, t_3), (q_1, t_4), (q_2, t_2), (q_3, t_1), (q_5, t_1)\}$.*

Moreover, the set of firing vectors associated to minimal sequences that enable to reach the different nodes of the graph are: $\Sigma_1 = \{\vec{\varepsilon}_1\}$, $\Sigma_2 = \{\vec{\varepsilon}_1 + \vec{\varepsilon}_2\}$, $\Sigma_3 = \{2\vec{\varepsilon}_1 + \vec{\varepsilon}_2\}$, $\Sigma_4 = \{\vec{\varepsilon}_1 + \vec{\varepsilon}_2 + \vec{\varepsilon}_3,\ \vec{\varepsilon}_1 + \vec{\varepsilon}_2 + \vec{\varepsilon}_4\}$, $\Sigma_5 = \{2\vec{\varepsilon}_1 + \vec{\varepsilon}_2 + \vec{\varepsilon}_3,\ 2\vec{\varepsilon}_1 + \vec{\varepsilon}_2 + \vec{\varepsilon}_4\}$, $\Sigma_6 = \{\vec{\varepsilon}_1 + 2\vec{\varepsilon}_2 + \vec{\varepsilon}_3,\ \vec{\varepsilon}_1 + 2\vec{\varepsilon}_2 + \vec{\varepsilon}_4\}$.

Finally we observe that there are no bisimilar nodes, thus we have no constraints of the form (f) and (g).

The set of constraints $\mathcal{C}(\mathcal{G}, P)$ is not reported here for sake of brevity but can be found at [6], together with the file LINDO we used to solve the resulting IPP.

We identify the net system in Figure 2.(a). ■

6 Conclusions

In this paper we considered the problem of identifying an unbounded Petri net system given its unlabeled coverability graph, namely an automaton $\mathcal{G}$ whose arcs are labeled with the transitions of the net. The solution we propose ensures that the coverability graph of the resulting net system is isomorphic to $\mathcal{G}$, and is based on a linear algebraic characterization of the net systems whose coverability graph is isomorphic to $\mathcal{G}$. Therefore, the identification problem is written in terms of an IPP.

References

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

[2] T. Bourdeaud'huy and P. Yim. Synthse de rseaux de Petri partir d'exigences. In *Actes de la 5me conf. francophone de Modlisation et Simulation*, pages 413–420, Nantes, France, September 2004.

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

[4] 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.

[5] 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.

[6] http:\\bode.diee.unica.it\dehs\CDC06.

[7] R.M. Karp and R.E. Miller. Parallel program schemata. *Journ. Computer and System Sciences 3*, (2):147–195, May 1969.

[8] M.E. Meda-Campaa and E. Lpez-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.

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

[10] L.A. Wolsey. *Integer Programming.* John Wiley & Sons, 1998.