Available online at www.sciencedirect.com
SCIENCE DIRECT®
automatica
Automatica 41 (2005) 605–616
www.elsevier.com/locate/automatica

Steady-state performance evaluation of continuous mono-T-semiflow Petri nets☆

Jorge Júlvez*,[1], Laura Recalde, Manuel Silva

Departamento de Informática e Ingeniería de Sistemas, Centro Politécnico Superior de Ingenieros, Universidad de Zaragoza, María de Luna 3, E-50015 Zaragoza, Spain

Received 9 October 2003; received in revised form 10 September 2004; accepted 4 November 2004

Abstract

The number of states in discrete event systems can increase exponentially with respect to the size of the system. A way to face this *state explosion* problem consists of relaxing the system model, for example by converting it to a continuous one. In the scope of Petri nets, the firing of a transition in a continuous Petri net system is done in a real amount. Hence, the marking (state) of the net system becomes a vector of non-negative real numbers. The main contribution of the paper lies in the computation of throughput bounds for continuous Petri net systems with a single T-semiflow. For that purpose, a branch and bound algorithm is designed. Moreover, it can be relaxed and converted into a linear programming problem. Some conditions, under which the system always reaches the computed bounds, are extracted. The results related to the computation of the bounds can be directly applied to a larger class of nets called *mono T-semiflow reducible*.

Keywords: Performance evaluation; Petri nets; Performance bounds; Piecewise linear systems; Continuous time

1. Introduction

Different approaches have been studied to face the state explosion problem that is inherent to large discrete event systems. A frequent approach is to analyze a relaxed model. A classical relaxation technique consists of *continuizing* the system, i.e. the relaxed model is not discrete any more. The idea of continuizing at net level a discrete Petri net was introduced in the scope of manufacturing systems by David and Alla (1987) (see Alla & David (1998) for a recent survey). Analogously, in parallel, developing structural analysis methods, linear programming techniques were introduced instead of integer programming techniques (Silva & Colom, 1987). However, for quantitative properties, some precision may be lost, and even qualitative properties of the discrete system may not be preserved by the continuous one. For example, in general, the existence of home states or mutual exclusion properties cannot be studied in the continuous relaxation. Liveness (deadlock-freeness also) of the continuous approximation is in general neither necessary nor sufficient for the same property of the original discrete model (Silva & Recalde, 2002). Furthermore, since continuization implies removing a constraint (the integrality of the firing), it could be thought that in timed systems the throughput of the continuous system is an upper bound of the discrete. However, this is not always the case (Silva & Recalde, 2004). Hence, not every Petri net model can be "reasonably continuized", like not every non-linear system can be "reasonably linearized".

This paper focuses on the study of throughput bounds for the subclass of *mono-T-semiflow* (MTS) *Petri nets* (PN)

☆ This paper was not presented at any IFAC meeting. This paper was recommended for publication in revised form by Associate Editor X. Cao under the direction of Editor I. Petersen.

* Corresponding author. Tel.: +34 976762472; fax: +34 976761914.

E-mail addresses: julvez@unizar.es (J. Júlvez), lrecalde@unizar.es (L. Recalde), silva@unizar.es (M. Silva).

[1] Supported by a grant from D.G.A. ref B106/2001. Work partially supported by projects CICYT and FEDER DPI2003-06376 and TIC2001-1819.

0005-1098/$ - see front matter © 2004 Elsevier Ltd. All rights reserved.
doi:10.1016/j.automatica.2004.11.007

(Campos et al., 1991). A Petri net is MTS if it is conservative (i.e., all places are covered by P-semiflows), consistent and has only one T-semiflow (i.e., all transitions are covered by *the* unique minimal T-semiflow). Hence, it can be decided in polynomial time whether a given net, $\mathcal{N}$, is MTS or not. MTS represents an important generalization of *choice-free* nets (Teruel et al., 1997). A subclass of choice-free nets are weighted-T-systems (Teruel et al., 1997), a weighted generalization of the well-known subclass of marked graphs (Commoner et al., 1971).

For the timing interpretation in transitions, a semantics of infinite servers will be used (Silva & Recalde, 2002). Under this firing semantics the continuous Petri net system behaves as a piecewise linear system (Sontag, 1981).

The structure of the paper is the following: In Section 2, timed continuous Petri nets are introduced, some interesting/unexpected behaviors are shown and the class of MTS systems is defined. In Section 3, some techniques to compute throughput bounds are described and applied to a manufacturing example. First, an algorithm based on a branch and bound (b & b) technique is presented to compute upper throughput bounds. A very similar algorithm can be designed to compute lower throughput bounds. Then, it is shown how upper bounds (less tight in general) can be polynomially computed by means of a single linear programming problem. Conditions for reachability of the bounds computed by this last approach are given. Even if MTS systems generalize a certain number of classical net subclasses, including conflicts and synchronizations, Section 4 introduces a larger class of nets, *mono-T-semiflow reducible systems*, to which previous results can be applied. The main feature of a mono-T-semiflow reducible system is that its visit ratio does not depend on the initial marking and can be computed in polynomial time. In Section 5, some conclusions are presented.

2. Timed continuous Petri nets

2.1. Basic definitions

We assume that the reader is familiar with Petri nets (PNs) (see, for example, (Murata, 1989; DiCesare et al., 1993)). The usual PN system, $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ ($\mathcal{N} = \langle P, T, \mathbf{Pre}, \mathbf{Post} \rangle$), will be said to be *discrete* so as to distinguish it from the *continuous* relaxation (Alla & David, 1998; Silva & Recalde, 2002). A first difference between continuous and discrete PNs is in the firing of transitions, which in a discrete PN is restricted to be in the naturals, while in continuous PNs is released into the non-negative real numbers. As a consequence of this, the marking of the net system becomes also continuous. It will be said that a transition t is *enabled* at $\mathbf{m}$ iff for every $p \in {}^{\bullet}t$, $\mathbf{m}[p] > 0$. In other words, for continuous systems a transition is enabled iff every input place is marked. As in discrete systems, the *enabling degree* at $\mathbf{m}$ of a transition measures the maximal amount in which the transition can be fired in one go, i.e., $\mathrm{enab}(t, \mathbf{m}) = \min_{p \in {}^{\bullet}t}\{\mathbf{m}[p]/\mathbf{Pre}[p, t]\}$.

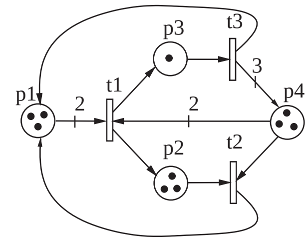

Fig. 1. A continuous Petri net system.

The firing of t in a certain amount $0 < \alpha \leqslant \mathrm{enab}(t, \mathbf{m})$ leads to a new marking $\mathbf{m}' = \mathbf{m} + \alpha \cdot \mathbf{C}[P, t]$, where $\mathbf{C} = \mathbf{Post} - \mathbf{Pre}$ is the token flow matrix. Hence, as in discrete systems, the state (or fundamental) equation ($\mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \boldsymbol{\sigma}$) summarizes the marking evolution.

All the concepts based on the representation of the net as a graph can be directly applied to continuous nets, in particular, the conflict relationships. When $\mathbf{Pre}[P, t] = \mathbf{Pre}[P, t'] \neq \mathbf{0}$, t and t' are in *equal conflict* (*EQ*) relation (Teruel & Silva, 1996). As an immediate generalization, it will be said that t and t' are in *continuous equal conflict* (*CEQ*) relation when there exists $k > 0$ such that $\mathbf{Pre}[P, t] = k \cdot \mathbf{Pre}[P, t'] \neq \mathbf{0}$.

Right and left natural annullers of the token flow matrix $\mathbf{C}$ are called T- and P-semiflows, respectively. We call a semiflow $\mathbf{v}$ *minimal* when its support, $\|\mathbf{v}\|$, i.e., the set of its non-zero components, is not a proper superset of the support of any other semiflow, and the g.c.d. of its elements is one. When $\mathbf{y} \cdot \mathbf{C} = \mathbf{0}$, $\mathbf{y} > \mathbf{0}$ the net is said to be *conservative*, and when $\mathbf{C} \cdot \mathbf{x} = \mathbf{0}$, $\mathbf{x} > \mathbf{0}$ the net is said to be *consistent*.

For the timing interpretation of *continuous* PNs a first-order (or deterministic) approximation of the probability density function will be used, assuming that the delays associated to the firing of transitions can be approximated by their mean values. Here, as in Silva and Recalde (2002), in order to compare behaviors, the discrete models are assumed to be markovian. The state equation of a continuous Petri net has an explicit dependence on time $\mathbf{m}(\tau) = \mathbf{m}_0 + \mathbf{C} \cdot \boldsymbol{\sigma}(\tau)$. Deriving with respect to time, $\dot{\mathbf{m}}(\tau) = \mathbf{C} \cdot \dot{\boldsymbol{\sigma}}(\tau)$ is obtained. Let us denote $\mathbf{f} = \dot{\boldsymbol{\sigma}}$, since it represents the *flow* through the transitions.

Different semantics have been defined for continuous PNs, the most important being *infinite* and *finite* servers (Silva & Recalde, 2002) (*variable* and *constant* speed, respectively (Alla & David, 1998)). Infinite servers semantics will be considered here. Under infinite servers semantics, the flow through a transition t is the product of the *firing speed*, $\boldsymbol{\lambda}[t] > 0$, and the enabling degree of the transition, i.e., $\mathbf{f}[t] = \boldsymbol{\lambda}[t] \cdot \mathrm{enab}(t, \mathbf{m}) = \boldsymbol{\lambda}[t] \cdot \min_{p \in {}^{\bullet}t}\{\mathbf{m}[p]/\mathbf{Pre}[p, t]\}$, leading to piecewise linear systems. A continuous timed system will be represented as $\langle \mathcal{N}, \lambda, \mathbf{m}_0 \rangle$.

As an example, the equations that determine the flow through transitions of the system in Fig. 1 are: $\mathbf{f}[t_1] = \boldsymbol{\lambda}[t_1] \cdot \min\{\mathbf{m}[p_1]/2, \mathbf{m}[p_4]/2\}$, $\mathbf{f}[t_2] = \boldsymbol{\lambda}[t_2] \cdot \min\{\mathbf{m}[p_2], \mathbf{m}[p_4]\}$ and $\mathbf{f}[t_3] = \boldsymbol{\lambda}[t_3] \cdot \mathbf{m}[p_3]$.

Because of the existence of the *min* operator in the equation system, the state equation can be seen as commuting among sets of ordinary linear differential equations (piecewise linearity). If the net is join free (i.e., each transition has at most one input place) a single set of linear differential equations represents the evolution of the marking: $\dot{\mathbf{m}}(\tau)=\mathbf{C}\cdot\mathbf{\Lambda}\cdot\mathbf{m}(\tau)$, $\mathbf{m}(0)=\mathbf{m}_0$ where $\mathbf{\Lambda}[t,p]=\mathbf{\lambda}[t]/\mathbf{Pre}[p,t]$ if $p={}^\bullet t$, and 0 otherwise.

Definition 1. (Campos et al., 1991) A PN is a *mono-T-semiflow* (MTS) net iff it is conservative and has a unique minimal T-semiflow whose support contains all the transitions.

Obviously, for MTS nets $rank(C) = |T| - 1$ (see (Campos et al., 1991)). The results of this paper apply to any MTS system that evolves through a transient and eventually reaches a steady state in which the marking of the places and the flow through transitions remain constant.

A performance measure that is often used in discrete PN systems is the throughput of a transition in the steady state, i.e., the number of firings per time unit. In the continuous approximation, this corresponds to the flow of the transition. Observe that in the steady state $\dot{\mathbf{m}}(\tau) = 0$, and so, from the state equation $\mathbf{C} \cdot \mathbf{f}_{ss} = \mathbf{0}$, where $\mathbf{f}_{ss}$ is the flow vector of the timed system in the steady state, $\mathbf{f}_{ss} = \lim_{\tau\to\infty} \mathbf{f}(\tau)$. Since $\mathbf{f}_{ss} \geqslant \mathbf{0}$, the flow in the steady state is proportional to *the minimal* T-semiflow. The marking at the steady state will be denoted as $\mathbf{m}_{ss}$.

A classical concept in queueing network theory is the "visit ratio". In Petri net terms, the visit ratio of transition t_j with respect to t_i, $\mathbf{v}^{(i)}[t_j]$, is the average number of times t_j is visited (fired), for each visit to (firing of) the reference transition t_i. Observe that $\mathbf{v}^{(i)}$ is a "normalization" ($\mathbf{v}^{(i)}[t_i] = 1$) of the flow vector in the steady state, i.e., $\mathbf{v}^{(i)}[t_j]=\lim_{\tau\to\infty}(\mathbf{f}[t_j](\tau)/\mathbf{f}[t_i](\tau))$. Hence, for any t_i, $\mathbf{f}_{ss}=\chi_i\cdot\mathbf{v}^{(i)}$, with χ_i the throughput of t_i. The vector of visit ratios is a right annuler of the incidence matrix $\mathbf{C}$, and therefore, in MTS systems, proportional to the unique T-semiflow. For MTS systems $\mathbf{f}_{ss} = f_{ss}(\mathcal{N}, \mathbf{\lambda}, \mathbf{m}_0)$, while $\mathbf{v}^{(i)} = v_i(\mathcal{N})$, i.e., the visit ratio does not depend neither on $\mathbf{\lambda}$ nor on $\mathbf{m}_0$.

2.2. Remarkable behaviors of timed continuous systems

In this subsection some *at first glance* unexpected behaviors of continuous MTS systems are briefly shown.

2.2.1. Continuous is not an upper bound of discrete

It could be thought that, since continuization removes some restrictions of the system, the throughput of the continuous system should be at least that of the discrete one. However, the throughput of a continuous PN is *not* in general an upper bound of the throughput of the discrete PN. For instance, in the net system in Fig. 2(a), with $\mathbf{\lambda}=(3, 1, 1, 10)$, the throughput is 0.801 as discrete while it is only 0.535 as continuous. If the continuous marking is seen as a very large

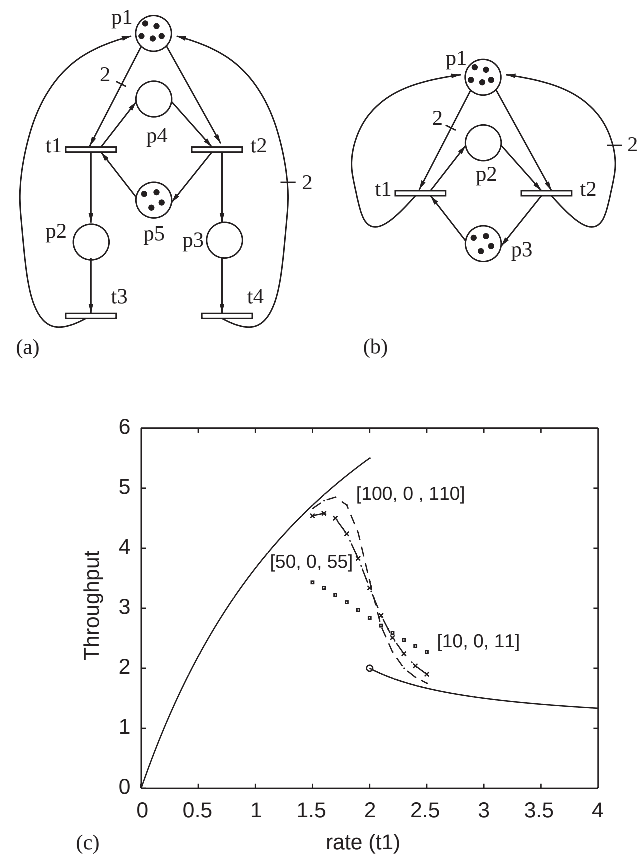

Fig. 2. (a) A net system whose throughput as continuous is not an upper bound for the throughput as discrete, with $\mathbf{\lambda} = (3, 1, 1, 10)$. (b), (c) For this net system, with $\mathbf{\lambda}[t_2] = 1$, increasing the rate of t_1 does not necessarily increase the throughput. Moreover a discontinuity point appears at $\mathbf{\lambda}[t_1] = 2$.

discrete marking, the reason for this "anomaly" lies in the non-monotonicity of the throughput under initial marking scaling (from $\mathbf{m}_0$ to $k \cdot \mathbf{m}_0$, $k > 0$) for discrete systems.

2.2.2. Non-monotonicities

Like in discrete nets, the throughput of a continuous net system does not fulfill in general any monotonicity property, neither w.r.t. the *initial marking*, nor w.r.t. the *structure* of the net, nor w.r.t. the *transitions rates*.

For example, with respect to the initial marking, if in the timed net system of Fig. 2(a) the marking of p_5 is augmented to 5, the systems deadlocks, i.e., the throughput goes down to 0. While if $\mathbf{m}_0[p_5]$ is reduced to 3 the throughput increases from 0.535 to 1.071. Notice that this token (i.e., resource) reduction is equivalent to adding a place "parallel" to p_5 (i.e., with an input arc from t_2 and an output arc to t_1), marked with 3 tokens. Hence, with respect to the net structure, adding constraints may increase the throughput!

Finally, an increase in a transition rate (for example, due to a replacement by a faster machine) may also lead to a decrease in the throughput. Moreover, a very small change may have a large effect. For example, the solid curve in

Fig. 2(c) represents how the throughput of the net system in Fig. 2(b) changes if the rate of t_1 varies from 0 to 4, assuming $\boldsymbol{\lambda}[t_2]=1$. Notice that even a *discontinuity* appears at $\boldsymbol{\lambda}[t_1]=2$. Starting from the discrete underlying net system, this effect can be interpreted as the limit case when the number of discrete tokens goes to infinity (see the dotted lines in Fig. 2(c) for the throughput of the discrete system with initial marking $\mathbf{m}_0=(10\ 0\ 11)$, $\mathbf{m}_0=(50\ 0\ 55)$ and $\mathbf{m}_0=(100\ 0\ 110)$).

3. Performance evaluation bounds

In this section it will be shown how bounds for the throughput of a continuous MTS system can be computed. Section 3.1 presents a non-linear programming problem that can be used to compute the *tight* bound of the system. It will be explained later what *tight* means here. A way to solve that programming problem consists in using a b & b algorithm (Section 3.2). In Section 3.5 the programming problem is *relaxed* leading to a linear programming problem (LPP), although this may lead to a non-tight bound. A sufficient condition for the reachability of the bound computed by the obtained LPP is given in Section 3.5.1.

3.1. Formulating a non-linear programming problem for performance bounds

Let $\mathbf{m}_{ss}$ be the steady-state marking of a continuous net system. For every $\tau>0$, the following equations have to be verified:

$$\dot{\mathbf{m}}(\tau)=\mathbf{C}\cdot\mathbf{f}(\tau), \tag{1}$$

$$\mathbf{f}(\tau)[t]=\boldsymbol{\lambda}[t]\cdot\min_{p\in{}^\bullet t}\left\{\frac{\mathbf{m}(\tau)[p]}{\mathbf{Pre}[p,t]}\right\}\quad\forall t\in T, \tag{2}$$

$$\mathbf{m}(0)=\mathbf{m}_0, \tag{3}$$

$$\mathbf{m}_{ss}=\lim_{\tau\to\infty}\mathbf{m}(\tau). \tag{4}$$

The above equations can be relaxed as follows ($\boldsymbol{\mu}_{ss}$ and $\boldsymbol{\phi}_{ss}$ correspond to $\mathbf{m}_{ss}$ and $\mathbf{f}_{ss}$):

$$\boldsymbol{\mu}_{ss}=\mathbf{m}_0+\mathbf{C}\cdot\boldsymbol{\sigma}, \tag{5}$$

$$\boldsymbol{\phi}_{ss}[t]=\boldsymbol{\lambda}[t]\cdot\min_{p\in{}^\bullet t}\left\{\frac{\boldsymbol{\mu}_{ss}[p]}{\mathbf{Pre}[p,t]}\right\}\quad\forall\ t\in T, \tag{6}$$

$$\mathbf{C}\cdot\boldsymbol{\phi}_{ss}=\mathbf{0}, \tag{7}$$

$$\boldsymbol{\mu}_{ss},\boldsymbol{\sigma}\geqslant\mathbf{0}. \tag{8}$$

Eq. (5) is obtained from (1) and (3), while (6) is a particularization of (2). Eq. (6) may be seen as an application of the underlying idea in the Little's law for queueing systems (let $\theta[t]=1/\boldsymbol{\lambda}[t]$): $\theta[t]\cdot\boldsymbol{\phi}_{ss}[t]=\min_{p\in{}^\bullet t}\{\boldsymbol{\mu}_{ss}[p]/\mathbf{Pre}[p,t]\}$. For JF nets it can be rewritten as: $\boldsymbol{\phi}_{ss}[t]\cdot\mathbf{Pre}[{}^\bullet t,t]\cdot\theta[t]=\boldsymbol{\mu}_{ss}[{}^\bullet t]$, i.e., average flow of tokens by average residence time equals average number of tokens (this idea was first used in the field of Petri nets in Chiola et al., 1993). Since $\mathbf{m}_{ss}$ is a steady state, from Eq. (1), $\mathbf{C}\cdot\mathbf{f}_{ss}=\mathbf{0}$ is deduced, and therefore (7) is immediately obtained.

With this relaxation we have replaced the condition of being a reachable marking with being a solution of (5), the fundamental equation. That is, we lose the information about the feasibility of the transient path. Observe that the system is non-linear (*min* operator) and that it may have several solutions. For example, for the net system in Fig. 1 with $\boldsymbol{\lambda}=(2,1,1)$, any marking $[10-5\cdot\alpha, 4\cdot\alpha-3,\alpha,\alpha]$, with $1\leqslant\alpha\leqslant 5/3$, verifies (5)–(8).

Maximizing the flow of a transition (any of them, since all are related by the T-semiflow), an upper bound of the throughput in the steady state is obtained:

$$\begin{aligned}\max\{&\boldsymbol{\phi}_{ss}[t_1]\mid\boldsymbol{\mu}_{ss}=\mathbf{m}_0+\mathbf{C}\cdot\boldsymbol{\sigma},\\ &\boldsymbol{\phi}_{ss}[t]=\boldsymbol{\lambda}[t]\cdot\min_{p\in{}^\bullet t}\left\{\frac{\boldsymbol{\mu}_{ss}[p]}{\mathbf{Pre}[p,t]}\right\}\quad\forall t\in T,\\ &\mathbf{C}\cdot\boldsymbol{\phi}_{ss}=\mathbf{0},\\ &\boldsymbol{\mu}_{ss},\boldsymbol{\sigma}\geqslant\mathbf{0}\}.\end{aligned} \tag{9}$$

If the flow in (9) is *minimized* (see Section 3.4) instead of maximized a very similar algorithm can be used for the computation of lower bounds.

Let us consider the following proposition that will help to understand better the kind of solutions obtained in (9).

Proposition 2 (Silva et al., 1998). *If $\mathcal{N}$ is consistent and conservative, the following statements are equivalent*:

(1) $\boldsymbol{\mu}_{ss}=\mathbf{m}_0+\mathbf{C}\cdot\boldsymbol{\sigma}$ *and* $\boldsymbol{\sigma}\geqslant\mathbf{0}$,
(2) $\forall\ \mathbf{y}\geqslant\mathbf{0}$ *such that* $\mathbf{y}\cdot\mathbf{C}=\mathbf{0}$, *then* $\mathbf{y}\cdot\boldsymbol{\mu}_{ss}=\mathbf{y}\cdot\mathbf{m}_0$

This means that relaxing the conditions on $\mathbf{m}_{ss}$ to being a solution of the fundamental equation (that is, making the system insensitive to the transient), is equivalent to saying that the solution is insensitive to the initial marking distribution inside the P-semiflows. More precisely, it only depends on the loads of the P-semiflows, $\mathbf{y}\cdot\mathbf{m}=\mathbf{y}\cdot\mathbf{m}_0$.

Notice that the solution of (9) is always "reachable" in the sense that with a suitable initial distribution of the tokens inside the P-semiflows (for instance, with the same steady state distribution), its associated throughput can be obtained. This is why, it will be said that the solution is a *tight* bound.

Nevertheless, the non-linear programming problem in (9) is difficult to solve due to the "min" condition coming from Eq. (6). When a transition t has a single input place, Eq. (6) reduces to Eq. (6′). When t has more than one input place, then Eq. (6) can be relaxed (linearized) as Eq. (6″).

$$\boldsymbol{\phi}_{ss}[t]=\boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}_{ss}[p]}{\mathbf{Pre}[p,t]}\quad\text{if}\ \ p={}^\bullet t, \tag{6′}$$

$$\boldsymbol{\phi}_{ss}[t]\leqslant\boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}_{ss}[p]}{\mathbf{Pre}[p,t]}\quad\forall p\in{}^\bullet t\quad\text{otherwise}. \tag{6″}$$

This way we have a single linear programming problem (LPP) (10) defined by equalities (5), (7) and (6′), and inequalities (8) and (6″), that can be solved in *polynomial time*.

$$\begin{aligned} \max\{&\boldsymbol{\phi}_{ss}[t_1] \mid \boldsymbol{\mu}_{ss} = \mathbf{m}_0 + \mathbf{C}\cdot\boldsymbol{\sigma}, \\ &\boldsymbol{\phi}_{ss}[t] = \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}_{ss}[p]}{\mathbf{Pre}[p,t]} \quad \text{if } p={}^{\bullet}t, \\ &\boldsymbol{\phi}_{ss}[t] \leqslant \boldsymbol{\lambda}[t]\cdot\frac{\boldsymbol{\mu}_{ss}[p]}{\mathbf{Pre}[p,t]} \quad \forall p\in{}^{\bullet}t \text{ otherwise}, \\ &\mathbf{C}\cdot\boldsymbol{\phi}_{ss} = \mathbf{0}, \\ &\boldsymbol{\mu}_{ss}, \boldsymbol{\sigma} \geqslant \mathbf{0}\}. \end{aligned} \tag{10}$$

Unfortunately, this LPP provides in general a non-tight bound, i.e. the solution may be non-reachable for any distribution of the tokens verifying the P-semiflow load conditions, $\mathbf{y}\cdot\mathbf{m}_0$. This occurs when none of the input places of a transition really restricts the flow of that transition. When this happens, the marking does not define the steady state (the flow of that transition would be larger).

For example, for the net system in Fig. 1 with $\boldsymbol{\lambda}=\mathbf{1}$, the optimum of the LPP is $\mathbf{f}_{ss}[t_1]=1.25$. This value is obtained for $\mathbf{m}[p_1]=2.5$, $\mathbf{m}[p_2]=3.25$, $\mathbf{m}[p_3]=1.25$, and $\mathbf{m}[p_4]=2.5$. Under this marking the throughput of t_2 would be 2.5, while for the rest of the transitions, it is 1.25. Since $\mathbf{v}^{(1)}=\mathbf{1}$, this cannot be the steady state. It can be seen that this happens for any maximal solution of this particular LPP. Hence the LPP in this case provides a non-reachable bound of the throughput. In fact, the maximum throughput for this system is 0.75.

3.2. Towards a branch and bound (b & b) algorithm

One way to improve this bound is to *force the equality for at least one place per synchronization*. This corresponds to a correct interpretation of the *min* operator in (9). The problem is that there is no way to know in advance which of the input places should restrict the flow. A b & b algorithm can be used to compute a steady state marking that fulfills what (10) expresses. If the marking solution of (10) does not correspond to a steady state (i.e., there is at least one transition such that all its input places have "more than the necessary" tokens) choose one of the synchronizations and solve the set of LPPs that appear when each one of the input places are assumed to be defining the flow. That is, build a set of LPPs by adding an equation that relates the marking of each input place with the flow of the transition. These subproblems become children of the root search node. The algorithm is applied recursively, generating a tree of subproblems. If an optimal steady state marking is found to a subproblem, it is a feasible steady state marking, but not necessarily globally optimal. Since it is feasible, it can be used to prune the rest of the tree: if the solution of the LPP for a node is smaller than the best known feasible solution, no globally optimal solution can exist in the subspace of the feasible region represented by the node. Therefore, the node can be removed from consideration. The search proceeds until all nodes have been solved or pruned.

The recursive Algorithm 1 sketches how the b & b algorithm works. The inputs of the algorithm are the net, the initial marking and the set of pairs (p,t) such that the marking of p is wanted to define the flow of t in the steady state. This set of pairs is denoted by *eqs* (because it represents the equalities, i.e., the constraints for the firing of transitions) and in the first call to the algorithm will be equal to those (p,t) such that $p={}^{\bullet}t$. Successive calls to the algorithm will increase the set *eqs* in order to force the flow of the rest of transitions to be defined by the marking of an input place. The output of the algorithm is given by the global variable *bound*. The procedure *max_LPP* solves the LPP (10), i.e., Eqs. (5), (7), (6′) applied on the pairs (p,t) in *eqs*, inequality (6″) applied on every input place of the transitions that are not in *eqs*, and inequality (8). We assume that *max_LPP* returns a scalar x corresponding to the solution of the LPP, and a set *nt* (non-satisfied transitions) containing those transitions for which their flow according to x and the vector of visit ratios is less than it should be according to the markings of their input places.

```
Algorithm 1 (b & b, upper bounds)
Global Variable: bound := 0
Branch–Bound(N, m0, eqs)
Begin
(x, nt) := max_LPP(N, m0, eqs)
if x ≤ bound or the LPP was infeasible
    % Do nothing. This node is pruned.
else
    if nt = ∅ % The solution represents a steady state
        bound := max(bound, x)
    else
        take a t ∈ nt do
        for every p ∈ •t do
            eqs := eqs ∪ (p, t)
            Branch–Bound(N, m0, eqs)
        end_for
    end_if
end_if
End
```

The system model in Fig. 3 is a MTS system (but not a marked graph because of the Mi_Idle places). It represents a flexible manufacturing system composed of three machines: M1, M2 and M3. Parts of type A are processed first in machine M1 and then in machine M2, while parts of type B are processed first in M2 and then in M1. The intermediate products are stored in buffer B_1A and B_1B, and the final parts in buffers B_2A and B_2B, respectively. Machine M3 takes a part A and a part B and assembles the final product, that is stored in B_3 until its removal. In B_3 there is space at most for 10 products. There can be at most 10 parts of type A and 10 parts of type B either in B_1A and B_1B, or being processed by M1 and M2. Parts are moved in pallets all along the process, and there are 20 pallets of type

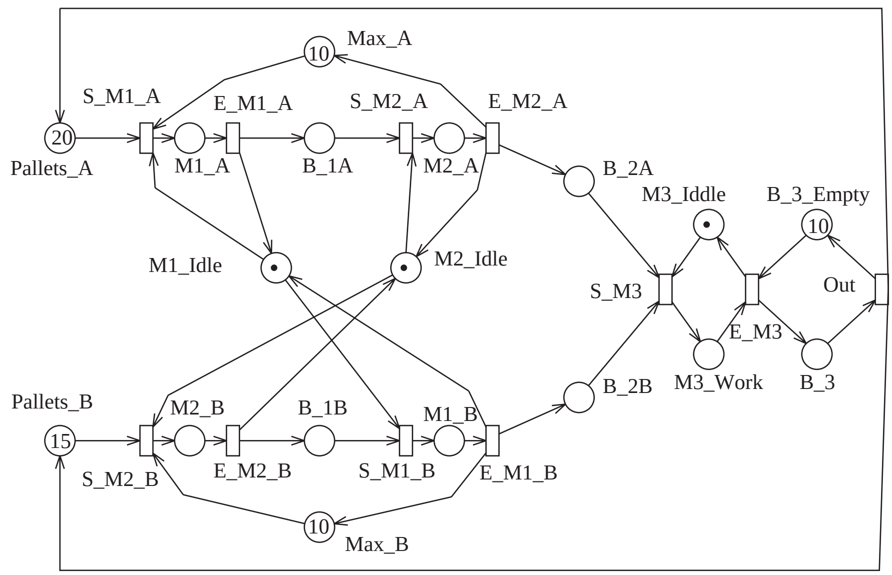

Fig. 3. A PN model of a flexible manufacturing system.

A and 15 pallets of type B. The firing speeds of transitions are: $\boldsymbol{\lambda}$[Out]= $\boldsymbol{\lambda}$[S_M1_A] = $\boldsymbol{\lambda}$[S_M2_A] = $\boldsymbol{\lambda}$[S_M1_B] = $\boldsymbol{\lambda}$[S_M2_B] = $\boldsymbol{\lambda}$[S_M3] = 1, $\boldsymbol{\lambda}$[E_M3] = $\boldsymbol{\lambda}$[E_M2_A] = 1/4, $\boldsymbol{\lambda}$[E_M1_A] = $\boldsymbol{\lambda}$[E_M2_B] = 1/3, $\boldsymbol{\lambda}$[E_M1_B] = 1/5.

The visit ratio of the system is $\mathbf{v}^{(1)} = \mathbf{1}$, that is, in the steady state all the operations have to be executed at the same rate (this is imposed by the assembly of one part A and one part B). A solution of the original LPP (10) is $\mathbf{f}_{ss}$[Out] = 0.111, $\mathbf{m}$[Pallets_A] = 9, $\mathbf{m}$[B_2A] = $\mathbf{m}$[Max_A] = $\mathbf{m}$[B_2B] = $\mathbf{m}$[Max_B] = $\mathbf{m}$[M3_Idle] = $\mathbf{m}$[B_3] = $\mathbf{m}$[M1_Idle] = 0.111, $\mathbf{m}$[M1_A] = 0.333, $\mathbf{m}$[B_1A] = 9.111, $\mathbf{m}$[Pallets_B] = 4, $\mathbf{m}$[M2_A] = 0.444, $\mathbf{m}$[M2_B] = 0.333, $\mathbf{m}$[B_1B] = 9, $\mathbf{m}$[M1_B] = 0.555, $\mathbf{m}$[M2_Idle] = 0.222, $\mathbf{m}$[M3_Work] = 0.888, $\mathbf{m}$[B_3_Empty] = 9.888. This solution corresponds to the root node, see Node 1 in Fig. 4, of the tree that the b & b algorithm computes when applied to the flexible manufacturing system.

According to the value obtained by the LPP, $\mathbf{f}_{ss}$[Out] = 0.111, and the vector of visit ratios, the throughput of all the transitions in the steady state should be 0.111. However, observe that if we consider the markings obtained for the input places of transition S_M2_A and E_M3, the throughput is greater than 0.111 (in the first call to the *branch–bound* algorithm *nt* equals $\{S\_M2\_A, E\_M3\}$). That is, the obtained marking is not a steady-state marking. If we first focus on S_M2_A, we should build two LPPs since it has two input places, adding in each one an equation for S_M2_A. If we force the throughput to be defined by M2_Idle (i.e., $\boldsymbol{\phi}_{ss}[S\_M2\_A] = \boldsymbol{\lambda}[S\_M2\_A] \cdot \boldsymbol{\mu}_{ss}[M2\_Idle]/\mathbf{Pre}[M2\_Idle, S\_M2\_A]$), Node 2, the system is *infeasible*, while if we add a restriction for B_1A, Node 3, the solution is the same but for $\mathbf{m}$[Pallets_A] = 18, $\mathbf{m}$[B_1A] = 0.111, and $\mathbf{m}$[Max_A] = 9.111. Now, the only problem is E_M3. If we add an equation for B_3_Empty, Node 4, the system is *infeasible*. Adding an equation for M3_Work, Node 5, modifies $\mathbf{m}$[Pallets_A] = 18.444, $\mathbf{m}$[Pallets_B] = 4.444, $\mathbf{m}$[M3_Idle] = 0.555, and $\mathbf{m}$[M3_Work] = 0.444. This is a steady-state marking and the throughput associated to it is equal to the one obtained with the original LPP, $\mathbf{f}_{ss}$[Out] = 0.111. No higher throughput may exist and no more branching is needed.

3.3. Pruning nodes in the b & b algorithm

The branching process developed in the algorithm is based on associating transitions with input places, i.e., forcing the throughput of a transition to be defined by one of its input places. The number of nodes of the tree in the worst case is $1 + \sum_{t_i \in T, |^{\bullet}t_i|>1} \prod_{j=1,|^{\bullet}t_j|>1}^{i} |^{\bullet}t_j|$, where the transitions are sorted according to their number of input places ($|^{\bullet}t_1| \geqslant |^{\bullet}t_2| \geqslant |^{\bullet}t_3| \ldots$). Number 1 stands for the root node and each element of the sum stands for one transition (one level of the tree) with several input places. In the worst case the algorithm has to explore the complete tree and solve a linear programming problem, whose complexity is polynomial, per node. However, in most real cases such exhaustive exploration is not required since some branches can be pruned according to the b & b algorithm. Moreover, some considerations will be done in this Subsection that allow us to further reduce the number of nodes to be explored.

Places can be seen as suppliers of fluid, i.e., clients, to their output transitions. Transitions can be seen as stations demanding fluid to their input places. Since we are dealing with MTS systems, in the steady state, the throughput of the transitions has to be proportional to the vector of visit ratios. Clearly, in the steady state, not all the output transitions of a given place, p, are equally *fluid-demanding*, i.e., one output transition t_1 may require a higher marking of p than other output transition t_2, in order to fire according to its visit

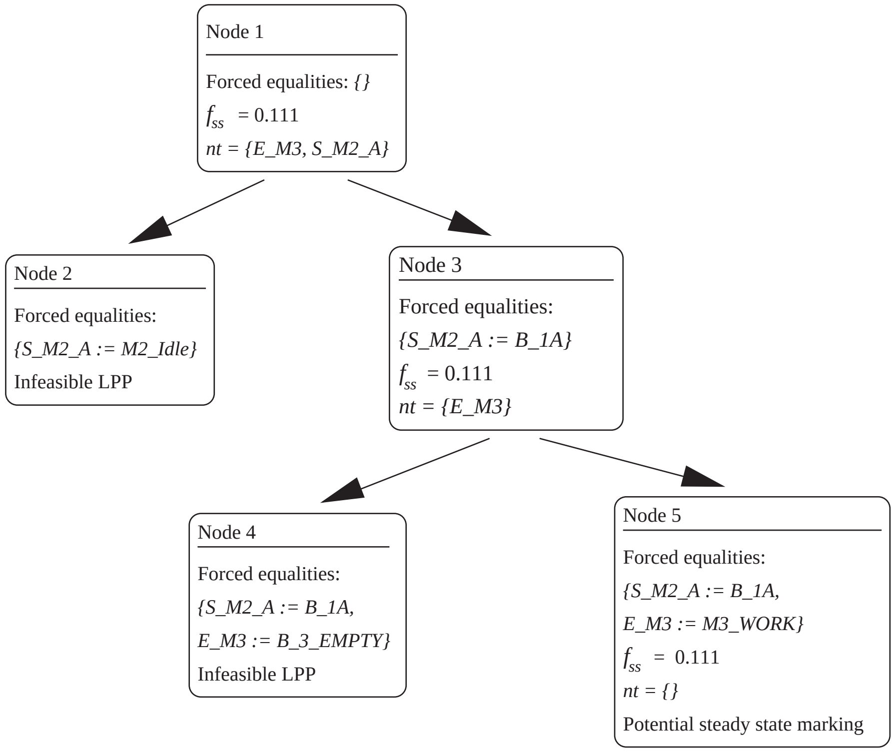

Fig. 4. Tree obtained by the B and B algorithm for the computation of upper bounds applied on the system in Fig. 3.

ratio. In the steady state, a place p can at most define the throughput of that output transition that is demanding the greatest amount of fluid to p. Thus, the b & b algorithm should avoid the exploration of those nodes that associate (the marking of) a place to (the flow of) an output transition that is not its most fluid-demanding transition.

Let us reconsider Eq. (6) in order to compute the most fluid-demanding transition for a given place p in the steady state. A simple relaxation of (6) consists in just looking if each place has enough fluid to fire all its output transitions:

$$\boldsymbol{\mu}_{ss}[p] \geqslant \max_{t \in p^{\bullet}} \left\{ \frac{\mathbf{Pre}[p,t] \cdot \boldsymbol{\phi}_{ss}[t]}{\boldsymbol{\lambda}[t]} \right\}. \tag{11}$$

The right part of Eq. (11) can be seen as the amount of fluid demanded to place p in the steady state by each of its output transitions. The transition giving the maximum is the most fluid-demanding transition, and so, it is the only transition that can be associated to place p in the b & b algorithm.

Let us consider the system in Fig. 1 with $\boldsymbol{\lambda} = \mathbf{1}$ to show how a node can be pruned by using the above reasonings. In principle, in the steady state the throughput of transition t_2 could be defined by any of its input places p_2 or p_4. Notice that p_4 has two output transitions t_1 and t_2, and so in the steady state it has to supply enough fluid for both transitions. The vector of visit ratios of this system is $\mathbf{v}^{(1)} = \mathbf{1}$. Therefore, in the steady state the throughput of every transition is the same. Thus, since $\boldsymbol{\lambda} = \mathbf{1}$ also the enabling degree of every transition should be the same in the steady state. Let us assume that in the steady state the enabling degree of t_2 is given by p_4. This implies that the enabling degree of t_1 is at most half the enabling degree of t_2 since there is an arc with weight 2 going from p_4 to t_2. In other words, t_1 is the most fluid-demanding transition of p_4: t_1 is demanding the double of fluid to p_4 than t_2. Hence, there cannot exist a non-dead steady-state marking at which p_4 is defining the flow of t_2 (and t_1).

Recall that for the system in Fig. 1 the LPP (10) yields $\mathbf{f}_{ss}[t_1] = 1.25$ with $\mathbf{m}[p_1] = 2.5$, $\mathbf{m}[p_2] = 3.25$, $\mathbf{m}[p_3] = 1.25$, and $\mathbf{m}[p_4] = 2.5$. Under this solution the throughput of t_2 would not be 1.25 and therefore it cannot represent a steady-state marking. If we try to force the throughput of t_2 to be defined by p_4, an infeasible LPP will be obtained. Therefore, the b & b algorithm must avoid the computation of the node that associates p_4 to t_2. Forcing the throughput of transition t_2 to be defined by p_2 the solution of the LPP is $\mathbf{f}_{ss}[t_1] = 0.75$ with $\mathbf{m}[p_1] = 5.5$, $\mathbf{m}[p_2] = 0.75$, $\mathbf{m}[p_3] = 0.75$, and $\mathbf{m}[p_4] = 1.5$. That is a steady-state marking, and therefore 0.75 is an upper bound for the throughput of the system.

Taking into account the most fluid-demanding transitions of places the complexity of the b & b algorithm presented in the previous subsection is reduced in many cases. Since a place will be associated only to its most fluid-demanding transition, the higher the number of output transitions of places, the higher the reduction of the complexity of the algorithm. Unfortunately, if all the places have a single output transition the complexity of the algorithm is not reduced.

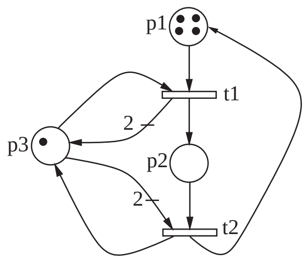

Fig. 5. A PN system with different upper and lower throughput bounds.

3.4. Lower bounds and exact throughput

Lower throughput bounds can be computed in a very similar way to upper bounds by means of a b & b algorithm. In this case, the goal function has to be minimized instead of maximized.

Let us compute the upper and lower bounds for the MTS system in Fig. 5 with initial marking $\mathbf{m}_0 = (4\ 0\ 1)$ and $\boldsymbol{\lambda} = (1\ 2)$. The visit ratio for the underlying net is $\mathbf{v}^{(1)} = (1\ 1)$. For the upper bound, the application of the b & b algorithm discussed in the previous section yields $\mathbf{f}_{ss}[t_1] = 2.5$ with $\mathbf{m}[p_1]=2.5$, $\mathbf{m}[p_2]=1.5$ and $\mathbf{m}[p_3]=2.5$ for the initial LPP (without any branching). That is a steady-state marking, and so, a suitable upper bound has been computed.

Minimizing the throughput of that LPP, we obtain $\mathbf{f}_{ss}[t_1]= 0$ with $\mathbf{m}[p_1] = 1.45$, $\mathbf{m}[p_2] = 2.55$ and $\mathbf{m}[p_3] = 3.55$. At this marking, neither t_1 nor t_2 have a throughput of 0 as established by the solution of the LPP. This is a non-surprising result since every transition has more than one input place and so $\mathbf{0}$ is a trivial solution of the equations in (10). Fig. 6 represents the tree explored by the b & b algorithm to compute the lower throughput bound for the system in Fig. 5.

If we force the throughput of transition t_1 to be defined by the marking of p_3, Node 2, the LPP yields $\mathbf{f}_{ss}[t_1] = 2$ with $\mathbf{m}[p_1]=3$, $\mathbf{m}[p_2]=1$ and $\mathbf{m}[p_3]=2$ which is a steady-state marking. Hence no more branching is required from this node. If transition t_1 is controlled by p_1, Node 3, we obtain $\mathbf{f}_{ss}[t_1]=0$ with $\mathbf{m}[p_1]=0$, $\mathbf{m}[p_2]=4$ and $\mathbf{m}[p_3]=5$. In this case the throughput of t_2 is not defined by any of the markings of its input places. From this node it is possible to associate t_2 to either p_2 or p_3. If t_2 is associated to p_2, Node 5, an infeasible LPP is obtained. If t_2 is associated to p_3, Node 4, the LPP gives $\mathbf{f}_{ss}[t_1] = 2.5$ with $\mathbf{m}[p_1] = 2.5$, $\mathbf{m}[p_2] = 1.5$ and $\mathbf{m}[p_3] = 2.5$, which is a steady-state marking. At this point, the b & b algorithm finishes and it can be concluded that $\mathbf{f}_{ss}[t_1] = 2$ is a suitable lower bound.

Reconsidering the system in Fig. 1, the lower bound obtained after the application of the b & b algorithm is $\mathbf{f}_{ss}[t_1]= 0.75$ with $\mathbf{m}[p_1] = 5.5$, $\mathbf{m}[p_2] = 0.75$, $\mathbf{m}[p_3] = 0.75$, and $\mathbf{m}[p_4]=1.5$. This lower bound is identical to the upper bound obtained for this system in Section 3.3. This means that the exact throughput of the system in the steady state has just been computed. The b & b algorithm focuses on the initial load of the P-semiflows and not on the initial marking of each place, therefore, it holds that for any initial distribution of the given load, the system in Fig. 1 will reach a steady state in which $\mathbf{f}_{ss}[t_1] = 0.75$.

In Section 3.2 the application of the b & b algorithm on the system in Fig. 3 yielded $\mathbf{f}_{ss}[\text{Out}] = 0.111$ as an upper bound for the throughput. If the b & b is applied to compute the lower bound for that system the same value is obtained. That is, $\mathbf{f}_{ss}[\text{Out}]=0.111$ is the exact throughput of the system in the steady state. If the system is considered as a discrete Petri net the number of reachable states is 1357486. The throughput of the discrete system computed by solving the associated Markov chain is $\mathbf{f}_{ss}[\text{Out}] = 0.1090$.

3.5. Branching elimination for the computation of upper bounds

Let us consider again the problem defined by the LPP in (10). As it has been done in Section 3.3, Eq. (6) can be relaxed to (11). The following single LPP can be obtained to compute an upper throughput bound:

$$\begin{aligned}\max\{&\boldsymbol{\phi}_{ss}[t_1] \mid \boldsymbol{\mu}_{ss} = \mathbf{m}_0 + \mathbf{C}\cdot\boldsymbol{\sigma},\\ &\boldsymbol{\mu}_{ss}[p] \geqslant \max_{t\in p^\bullet}\left\{\frac{\mathbf{Pre}[p,t]\cdot\boldsymbol{\phi}_{ss}[t]}{\boldsymbol{\lambda}[t]}\right\} \quad \forall p\in P,\\ &\mathbf{C}\cdot\boldsymbol{\phi}_{ss} = \mathbf{0},\\ &\boldsymbol{\sigma},\boldsymbol{\mu}_{ss} \geqslant \mathbf{0}\}.\end{aligned} \tag{12}$$

Since in MTS systems $\mathbf{v}^{(1)}$ is completely defined, if $\boldsymbol{\phi}_{ss} = \chi\cdot\mathbf{v}^{(1)}$, LPP (12) can be written as

$$\begin{aligned}\max\{&\chi \mid \boldsymbol{\mu}_{ss} = \mathbf{m}_0 + \mathbf{C}\cdot\boldsymbol{\sigma}\\ &\boldsymbol{\mu}_{ss} \geqslant \chi\cdot\mathbf{PD}, \quad \boldsymbol{\sigma},\boldsymbol{\mu}_{ss}\geqslant\mathbf{0}\},\end{aligned} \tag{13}$$

where $\mathbf{PD}[p] = \max_{t\in p^\bullet}\{\mathbf{Pre}[p,t]\cdot\mathbf{v}^{(1)}[t]/\boldsymbol{\lambda}[t]\}$.

Defining $\alpha = 1/\chi$ and $\boldsymbol{\sigma}' = 1/\chi\cdot\boldsymbol{\sigma}$, (13) reduces to

$$\min\{\alpha \mid \alpha\cdot\mathbf{m}_0 + \mathbf{C}\cdot\boldsymbol{\sigma}' \geqslant \mathbf{PD}, \quad \boldsymbol{\sigma}'\geqslant\mathbf{0}\} \tag{14}$$

The dual of this LPP is

$$\max\{\mathbf{y}\cdot\mathbf{PD} \mid \mathbf{y}\cdot\mathbf{C}\leqslant\mathbf{0}, \quad \mathbf{y}\cdot\mathbf{m}_0\leqslant 1, \quad \mathbf{y}\geqslant\mathbf{0}\}. \tag{15}$$

One of the formulations of the *alternatives theorem* (Murty, 1983) states that the following two statements are equivalent:

$$\begin{aligned}&(1)\quad \exists\ \mathbf{x}>\mathbf{0} \text{ such that } \mathbf{C}\cdot\mathbf{x}\geqslant\mathbf{0} \text{ and}\\ &(2)\quad \forall\ \mathbf{y}\geqslant\mathbf{0} \text{ such that } \mathbf{y}\cdot\mathbf{C}\leqslant\mathbf{0} \text{ then } \mathbf{y}\cdot\mathbf{C}=\mathbf{0}.\end{aligned} \tag{16}$$

Since MTS nets are consistent, (16) is true and $\mathbf{y}\cdot\mathbf{C}\leqslant\mathbf{0}$, $\mathbf{y}\geqslant\mathbf{0}$ can be replaced by $\mathbf{y}\cdot\mathbf{C}=\mathbf{0}$, $\mathbf{y}\geqslant\mathbf{0}$. Moreover, since we are maximising $\mathbf{y}\cdot\mathbf{PD}$, the solution must verify $\mathbf{y}\cdot\mathbf{m}_0 = 1$ (otherwise a better result can be obtained with $\beta\cdot\mathbf{y}$, $\beta = 1/(\mathbf{y}\cdot\mathbf{m}_0)$).

Proposition 3. *Let γ be the solution of*

$$\begin{aligned}\gamma = \max\{&\mathbf{y}\cdot\mathbf{PD} \mid \mathbf{y}\cdot\mathbf{C}=\mathbf{0}\\ &\mathbf{y}\cdot\mathbf{m}_0 = 1, \quad \mathbf{y}\geqslant\mathbf{0}\}.\end{aligned} \tag{17}$$

The throughput in the steady state verifies $\mathbf{f}_{ss}\leqslant(1/\gamma)\mathbf{v}^{(1)}$.

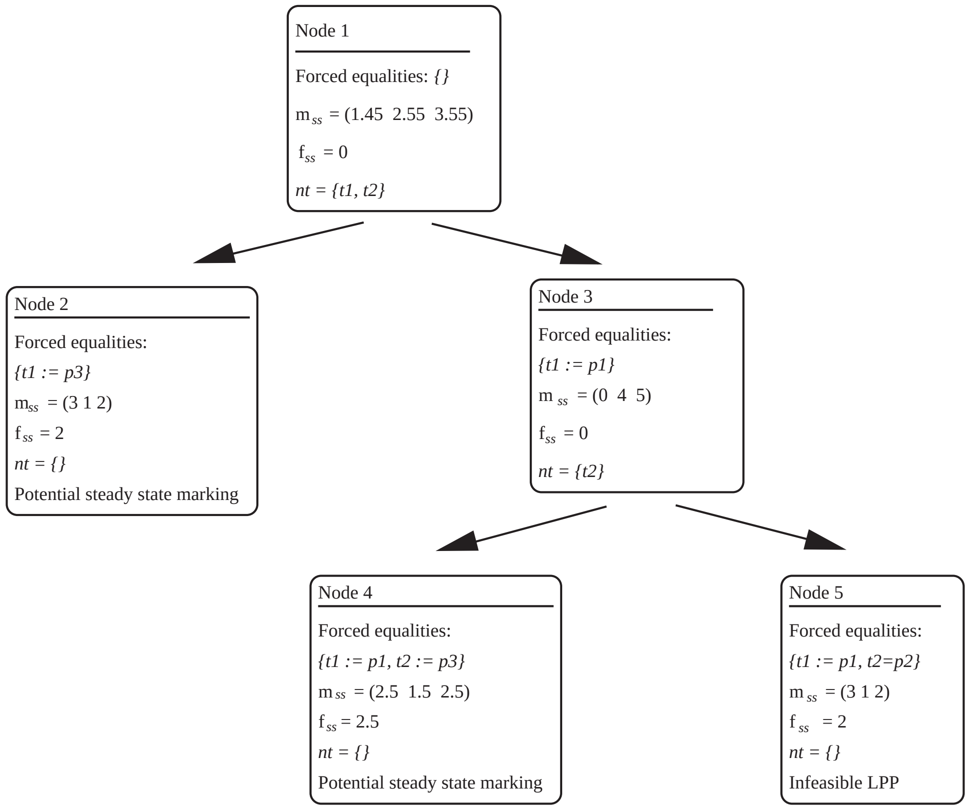

Fig. 6. Tree obtained by the B and B algorithm for the computation of lower bounds applied on the system in Fig. 5.

Intuitively, the idea of (17) is to find the slowest isolated subnet among those generated by the elementary P-semiflows. In other words, the bound is obtained by looking at the bottleneck P-semiflow. The complexity of finding the bottleneck P-semiflow is polynomial since it is obtained by solving a LPP. In (Campos & Silva, 1992) a similar result was obtained for discrete systems in which conflicts were forbidden except among immediate transitions.

3.5.1. Reachability of the bound

For each marking $\mathbf{m}$, its T-coverture (T-cov($\mathbf{m}$)) is defined as the set of places that restrict the flow of the transitions.

Definition 4. Given a net system, the *T-coverture* at a marking $\mathbf{m}$, is

$$\text{T-cov}(\mathbf{m}) = \{p \mid \exists t \in p^\bullet \text{ such that } \mathbf{f}[t] = \boldsymbol{\lambda}[t] \cdot \mathbf{m}[p]/\mathbf{Pre}[p,t]\}.$$

A characterization can be obtained for the solution of (17) being the exact value. Given a vector $\mathbf{v}$, let us denote as $\|\mathbf{v}\|$ its support, i.e., the set of its non-zero components.

Proposition 5. *Let* $\langle \mathcal{N}, \mathbf{m}_0 \rangle$ *be a MTS continuous system.*

The flow computed with (17) (*or* (13)) *is the flow in the steady state iff the T-coverture at the steady state,* T-cov($\mathbf{m}_{\text{ss}}$), *contains the support of a P-semiflow.*

Moreover, the maximum of (17) *is reached for the P-semiflow contained in the T-coverture.*

Proof. Let $\mathbf{m}_{\text{ss}}$ be the steady-state marking, $\mathbf{f}_{\text{ss}} = \chi_1 \cdot \mathbf{v}^{(1)}$ the flow vector associated to this state, and γ the solution of (17). Applying (17), $\chi_1 \leqslant 1/\gamma$.

For "$\Rightarrow$", assume that $\mathbf{y}_0$ is a P-semiflow such that the maximum of the LPP is reached. If its support is not contained in T-cov($\mathbf{m}_{\text{ss}}$), a place $p \in \|\mathbf{y}_0\|$ exists such that $\mathbf{m}_{\text{ss}}[p] > \max_{t\in p^\bullet}\{\mathbf{Pre}[p,t]\cdot\chi_1\cdot\mathbf{v}^{(1)}[t]/\boldsymbol{\lambda}[t]\} = \chi_1 \cdot \mathbf{PD}[p]$. Hence, $\mathbf{y}_0 \cdot \mathbf{m}_{\text{ss}} > \chi_1 \cdot \mathbf{y}_0 \cdot \mathbf{PD}$, and $1/\chi_1 > \mathbf{y}_0 \cdot \mathbf{PD} = \gamma$, contradiction.

For "$\Leftarrow$", let $\mathbf{y}_0$ be a P-semiflow such that $\|\mathbf{y}_0\| \subseteq$ T-cov($\mathbf{m}_{\text{ss}}$) and $\mathbf{y}_0 \cdot \mathbf{m}_0 = 1$. Then, for every $p \in \|\mathbf{y}_0\|$, a transition $t \in p^\bullet$ exists such that $\mathbf{m}_{\text{ss}}[p] = \mathbf{Pre}[p,t] \cdot \chi_1 \cdot \mathbf{v}^{(1)}[t]/\boldsymbol{\lambda}[t]$. Hence, $\mathbf{m}_{\text{ss}}[p] = \chi_1 \cdot \max_{t\in p^\bullet}\{\mathbf{Pre}[p,t]\cdot \mathbf{v}^{(1)}[t]/\boldsymbol{\lambda}[t]\} = \chi_1 \cdot \mathbf{PD}[p]$.

Therefore $\gamma \geqslant \mathbf{y}_0 \cdot \mathbf{PD} = \mathbf{y}_0 \cdot \mathbf{m}_{\text{ss}}/\chi_1 = 1/\chi_1$. Then $1/\chi_1 = \gamma$. $\square$

From Proposition 5 the following corollary is obtained:

Corollary 6. *Let* $\mathcal{N}$ *be a MTS continuous net. If the P-subnet defined by any T-coverture contains a P-semiflow, then the flow at the steady state can be computed in polynomial time with the LPP* (17).

4. Extending the subclass of nets: MTS reducible nets

One interesting property of MTS nets is that the vector of visit ratios only depends on the net structure, i.e., $\mathbf{v}^{(1)} = v^{(1)}(\mathcal{N})$. Therefore, as it has been seen, knowing the flow in the steady state of one transition, the flow of the rest of transitions is trivially computed. However, the subclass of nets for which the vector of visit ratios does not depend on the initial marking is larger than the class of MTS nets. If we consider the net in Fig. 7(a) with $\boldsymbol{\lambda} = (1\ 1\ 1\ 1)$, we will realize that the flow through transitions in the steady state is always proportional to the vector (2 1 2 1), that is the flow through transitions t_2 and t_4 is double than the flow through transitions t_1 and t_3, independently of the initial marking. The reason for this fact is that given a continuous net (not necessarily MTS), the following is verified:

$$\frac{\mathbf{f}[t_i]}{\mathbf{Pre}[P, t_i] \cdot \boldsymbol{\lambda}[t_i]} = \frac{\mathbf{f}[t_j]}{\mathbf{Pre}[P, t_j] \cdot \boldsymbol{\lambda}[t_j]} \quad \forall t_i, t_j \text{ in CEQ relation.} \tag{18}$$

In this section, the results obtained in Section 3 will be extended to a larger class of nets, the class of *mono-T-semiflow reducible* nets (MTSR), for which $\mathbf{v}^{(1)} = v^{(1)}(\mathcal{N}, \boldsymbol{\lambda})$, i.e., $\mathbf{v}^{(1)}$ does not depend on the initial marking.

Definition 7. Let $\mathcal{N}$ be a consistent and conservative PN and $\boldsymbol{\lambda}$ a speeds vector. We will say that $\langle \mathcal{N}, \lambda \rangle$ is *mono T-semiflow reducible* (MTSR) if the following system has a unique solution:

$$\mathbf{C} \cdot \mathbf{v}^{(1)} = \mathbf{0},$$
$$\frac{\mathbf{v}^{(1)}[t_i]}{\mathbf{Pre}[P, t_i] \cdot \boldsymbol{\lambda}[t_i]} = \frac{\mathbf{v}^{(1)}[t_j]}{\mathbf{Pre}[P, t_j] \cdot \boldsymbol{\lambda}[t_j]} \quad \forall\, t_i, t_j \text{ in CEQ relation}, \ \mathbf{v}^{(1)}[t_1] = 1. \tag{19}$$

Every continuous MTSR net can be *reduced* to an "equivalent" MTS net with identical behavior. The *reduction rule* consists in merging those transitions in CEQ relation into only one flow-equivalent transition. The arcs and the firing speed, $\boldsymbol{\lambda}$, of the equivalent transition have to be such that they preserve the evolution of their input and output places. This can be achieved with simple arithmetic operations on the weights of the input/output arcs and the firing speeds of the original transitions. Fig. 8 sketches how two transitions in CEQ relation can be merged to a single one. It can be checked that the evolution of the input places is preserved and so is the flow associated to the output arcs. An iterative merger on every couple of transitions in CEQ relation leads to a net without CEQs and to a MTS if the original net was MTSR. Notice that the arc weights of the resulting net may not be natural numbers. Nevertheless, this is not a problem for any of the properties being considered.

Observe that if the input and output arc weights of a transition are multiplied by a constant, the evolution of the input and output places is the same. However, the flow through the transition varies in an inverse proportion to the constant. Therefore, by varying that constant, it is possible to reduce a MTSR net to an infinite number of equivalent MTS nets.

The net in Fig. 7(a) is MTSR (for $\boldsymbol{\lambda} = \mathbf{1}$ its equivalent MTS net is depicted in Fig. 7(b)) but not MTS since it has two T-semiflows. The net in Fig. 7(c) with $\boldsymbol{\lambda} = (1\ 1\ 1)$

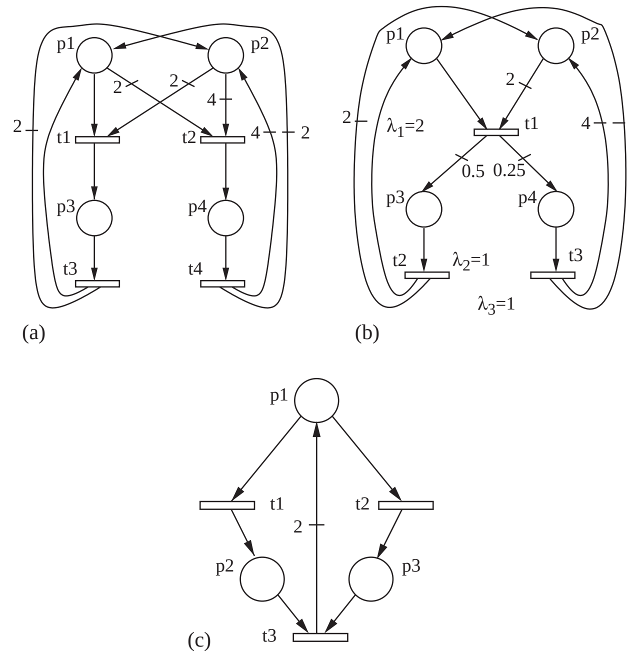

Fig. 7. (a) A MTSR (but not MTS net) for every $\boldsymbol{\lambda} > \mathbf{0}$. (b) Equivalent MTS net given $\boldsymbol{\lambda} = \mathbf{1}$. (c) A MTS net that belongs to MTSR iff $\boldsymbol{\lambda}_1 = \boldsymbol{\lambda}_2$.

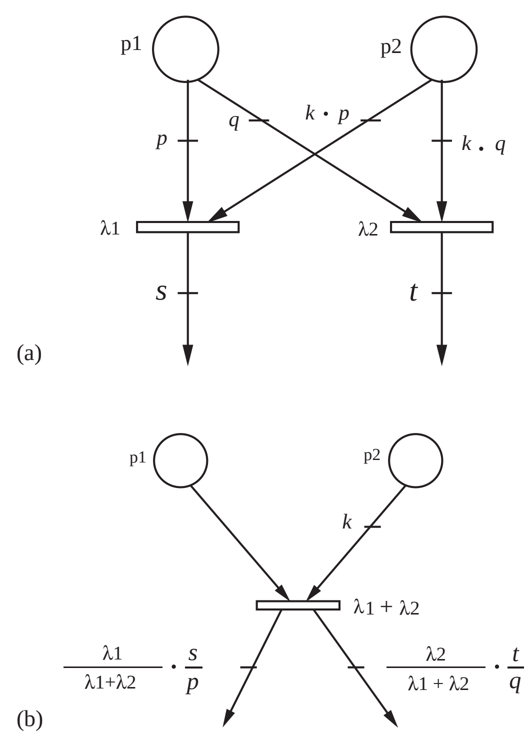

Fig. 8. Two transitions in CEQ relation and an equivalent one.

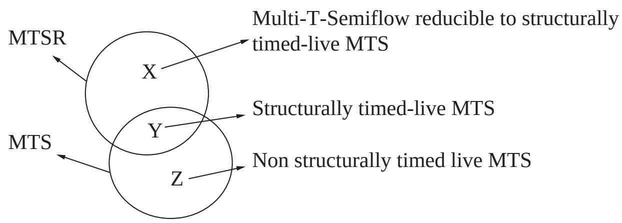

Fig. 9. Relationship between MTS and MTSR nets. MTSR only does not include the non str. timed-live MTS nets.

belongs to both MTS and MTSR. Nevertheless, it should be noticed that the class MTSR does not include the class MTS: The net in Fig. 7(c) with $\boldsymbol{\lambda} = (1\ 2\ 1)$ belongs to MTS but not to MTSR. However, disregarding those nets that belong only to MTS does not imply a loss of generality since their steady state throughput is zero, that is, they are not *structurally timed-live*. Therefore, it has no sense computing throughput bounds for systems like the one in Fig. 7(c) with $\boldsymbol{\lambda} = (1\ 2\ 1)$, because it is null. The diagram in Fig. 9 shows the relationship between the classes MTS and MTSR.

Extending the results of Section 3 to MTSR is almost immediate. For MTSR, the relaxation of the "min" condition in the non-linear programming problem (9) yields (6′), (6″) and (18). This last equation is necessary to fulfill the flow proportions between transitions in CEQ. Once Eq. (18) is added, the b & b algorithm of Section 3.2 can be applied directly to MTSR systems.

In the same way, Eq. (18) must also be added to the programming problem (12) for MTSR systems. And with identical reasoning of Section 3.5, same Eqs. (13)–(15) and (17) are obtained. Every reasoning and result of Section 3.5.1 is also directly applicable on MTSR systems.

It is interesting to remark that the class of MTSR nets offer a significant modeling power from a practical point of view. Focusing on live and bounded systems, the class of MTSR nets includes the class of equal conflict (EQ) nets (Teruel & Silva, 1996), which is a superset of the classes of free-choice (FC), choice-free (CF) (Teruel et al., 1997), weighted T-systems (WTS) and marked graphs (MG) nets (Commoner et al., 1971) (being MG a generalization of PERT charts). Fig. 10 shows the inclusion relationships among the mentioned classes.

With respect to the reachability of the upper bound computed by the LPP in (17) for MTSR systems, Corollary 6, it has to be noticed that it suffices to prove that the minimal T-covertures contain the support of a P-semiflow. This condition is in general difficult to solve since the number of minimal T-covertures may be very large. Nevertheless, Corollary 6 holds for instance for *str.* lim-*live and str. bounded EQ nets* (or equivalently (Silva et al., 1998) EQ nets that are consistent, conservative and the rank of the token flow matrix is upper bounded by the number of conflicts). More general classes exist for which this result holds too. For instance, it holds for the net system in Fig. 3.

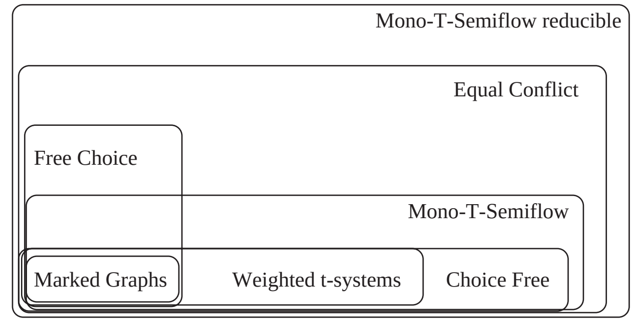

Fig. 10. Live and bounded net classes included in the class of MTSR nets.

5. Conclusions

Continuous Petri nets were introduced in order to overcome the state explosion problem of high traffic or highly populated discrete systems. Here, the attention is focused first on the subclass of MTS nets. This subclass is polynomially characterizable and offers a reasonable modeling power. Removing the constraint on conservativeness is not technically difficult and can be done by following the concepts presented in Campos et al., 1992.

The continuized model does not always faithfully represent the original discrete model, and even for the MTS subclass of net models some unexpected results may happen. This work presents a study of the throughput bounds (upper, lower, reachability) in the steady state.

In MTS systems the vector of visit ratios does only depend on the structure of the net. Therefore, once the steady state flow of one transition is known, it is immediate to compute the flow for the rest of transitions. Upper and lower bounds for the throughput of the system in the steady state can be computed by b & b algorithms. Relaxing some conditions an upper bound can be computed by a single LPP (17). This LPP is based on a search for the *slowest* P-semiflow of the system and it is the continuous *version* of the one in Campos & Silva, 1992. It has been shown that the bound computed by the LPP will be reached iff the set of places that are determining the flow of the system in the steady state (T-coverture) contains a P-semiflow.

The class of *mono T-semiflow reducible* (MTSR) nets considers those continuous nets whose visit ratio does only depend on the structure and the speeds of transitions (not on the initial marking), i.e., $\mathbf{v}^{(1)} = v^{(1)}(\mathcal{N}, \boldsymbol{\lambda})$. In this case the obtained results concerning the computation and reachability of the bounds for timed MTS systems are directly applicable.

References

Alla, H., & David, R. (1998). Continuous and hybrid Petri nets. *Journal of Circuits, Systems and Computers*, *8*(1), 159–188.

Campos, J., Chiola, G., Colom, J. M., & Silva, M. (1992). Properties and performance bounds for timed marked graphs. *IEEE Transactions on*

Circuits and Systems-I: Fundamental Theory and Applications, *39*(5), 386–401.

Campos, J., Chiola, G., & Silva, M. (1991). Ergodicity and throughput bounds of Petri net with unique consistent firing count vector. *IEEE Transactions on Software Engineering*, *17*(2), 117–125.

Campos, J., Silva, M. (1992). Structural techniques and performance bounds of stochastic Petri net models. In G. Rozenberg (Ed.), *Advances in Petri nets 1992*, Lecture Notes in Computer Science, Vol. 609 (pp. 352–391). Berlin: Springer.

Chiola, G., Anglano, C., Campos, J., Colom, J. M., & Silva, M (October 1993). Operational analysis of timed Petri nets and application to the computation of performance bounds. In *Proceedings of the fifth international workshop on Petri nets and performance models*, Toulouse, France (pp. 128–137). Silver Spring: IEEE Computer Society Press.

Commoner, F., Holt, A. W., Even, S., & Pnueli, A. (1971). Marked directed graphs. *Journal on Computer Systems Science*, *5*, 72–79.

David, R., & Alla, H. (1987). Continuous Petri nets. In *Proceedings of the eighth European workshop on application and theory of Petri nets*, Zaragoza, Spain (pp. 275–294).

DiCesare, F., Harhalakis, G., Proth, J. M., Silva, M., & Vernadat, F. B. (1993). *Practice of Petri nets in manufacturing*. London: Chapman & Hall.

Murata, T. (1989). Petri nets: properties, analysis and applications. *Proceedings of the IEEE*, *77*(4), 541–580.

Murty, K. G. (1983). *Linear programming*. New York: Wiley.

Silva, M., & Colom, J. M. (1987). On the structural computation of synchronic invariants in P/T nets. In *Proceedings of the eighth European workshop on application and theory of Petri nets*, Zaragoza, Spain (pp. 237–258).

Silva, M., & Recalde, L. (2002). Petri nets and integrality relaxations: a view of continuous Petri nets. *IEEE Transactions on Systems Man and Cybernetics*, *32*(4), 314–327.

Silva, M., & Recalde, L. (2004). On fluidification of Petri net models: from discrete to hybrid and continuous models. *Annual Reviews in Control*, 2004, 28, 253–266.

Silva, M., Teruel, E., & Colom, J. M. (1998). Linear algebraic and linear programming techniques for the analysis of net systems. In G. Rozenberg, & W. Reisig (Eds.), *Lectures in Petri nets*. *I*: *Basic models*, Lecture Notes in Computer Science, Vol. 1491, Springer (pp. 309–373).

Sontag, E. D. (1981). Nonlinear regulation: The piecewise linear approach. *IEEE Transactions on Automatic Control*, *26*(2), 346–358.

Teruel, E., Colom, J. M., & Silva, M. (1997). Choice-free Petri nets: a model for deterministic concurrent systems with bulk services and arrivals. *IEEE Transactions on Systems Man and Cybernetics*, *27*(1), 73–83.

Teruel, E., & Silva, M. (1996). Structure theory of equal conflict systems. *Theoretical Computer Science*, *153*(1–2), 271–300.

Jorge Júlvez received his Computer Engineering degree from the University of Zaragoza in 1998. In 2001 he joined, as a Ph.D. student, the Group of Discrete Event Systems Engineering in the Departamento de Informática e Ingeniería de Sistemas, University of Zaragoza. His research is related to the study of qualitative and quantitative properties of continuous Petri nets.

Laura Recalde received the M.S. degree in Mathematics and the Ph.D. degree in Computer Engineering from the University of Zaragoza in 1995 and 1998, respectively. She is currently Assistant Professor in the Departamento de Informática e Ingeniería de Sistemas, University of Zaragoza, where she is in charge of courses in Systems Theory and Discrete Event Systems. Her main research interest is the modeling and analysis of distributed concurrent systems using Petri nets.

Manuel Silva received the Industrial-Chemical Engineering degree from the University of Sevilla in 1974 and the Post-graduate and Ph.D. degrees in Control Engineering from the Institut National Polytechnique de Grenoble in 1975 and 1978, respectively. From 1975 to 1978, he worked for the Centre National de la Recherche Scientifique at the Laboratoire d'Automatique de Grenoble. In 1978 he started the group of Systems Engineering and Computer Science at the University of Zaragoza, where he was named Professor in Systems Engineering and Automatic Control in 1982. His research interests include modeling, validation, performance evaluation, and implementation of distributed concurrent systems using Petri nets, binary decision graphs, and robots programming and control. He is the author of the book *Las Redes de Petri en la Automática y la Informática* (AC, 1985; reprinted: Thomson, 2003), coauthor of the book *Practice of Petri Nets in Manufacturing* (Chapman & Hall, 1993), and co-editor of one special issue on CIM of the IEEE Transactions on Robotics and Automation. Prof. Silva was dean of the Centro Politécnico Superior, University of Zaragoza, from 1986 to 1992 and president of the Aragonese Research Council (CONAI) and of the Research and Innovation Committee of the French–Spanish Comisión de Trabajo de los Pirineos (CTP) from 1993 to 1995. He has been associate editor of the IEEE Transactions on Robotics and Automation and associate editor of the European Journal of Control. He is advisory member of the IEICE Transactions on Fundamentals on Electronics, Communications and Computer Sciences, member of the Steering Committees of the International Conferences on Application and Theory of Petri Nets, WODES and QEST, and founder of member of the Asociación Española de Robótica. Interested in the History of Technology and Engineering. He has been distinguished with a medal from Lille and by the Association of Telecommunication Engineers of Aragón. He is a member of the Royal Academy of Engineering of Spain.