STEADY STATE PERFORMANCE EVALUATION FOR SOME CONTINUOUS PETRI NETS

Laura Recalde *,1 **Jorge Júlvez** *,2 **Manuel Silva** *,1

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

Abstract: This paper addresses the computation of the steady state behaviour of a subclass of continuous Petri nets under infinite servers semantics. For this class, the visit ratio is fixed by the net structure and the rates associated to the transitions. The class generalises well known net subclasses, as marked graphs or equal conflict net systems.
A programming problem allows to obtain a tight bound of the throughput in the steady state. Its relaxation leads to a linear programming problem (LPP) that, for particular net subclasses, provides the exact value. Some of the "paradoxes" (essentially non-monotonicities) that the behaviour of those models may exhibit are pointed out.

Keywords: Petri nets, continuous systems, steady state throughput, linear programming

1. INTRODUCTION

An important drawback in the analysis of discrete event systems is the complexity associated to the exponential growth in the number of states. A way to try to overcome this difficulty is to *continuise* the system (total or partially) and apply different analysis techniques that may provide an approximation to the original behaviour. In the manufacturing systems domain, and using Petri net models, this idea has been applied, for instance, in (Alla and David, 1998; Balduzzi *et al.*, 2000).

This paper will mainly concentrate in the analysis of the steady state behaviour of continuous net systems, in particular the computation of the throughput. In previous work (Recalde and Silva, 2001), all the conflicts in the net were forced to be free (i.e., locally solved). This means that, from a performance point of view, the conflicts were not relevant once the rates had been set, and the system could be represented in fact as a net without conflicts. The present paper extends the approach by allowing conflicts, as long as the relationship among the throughput of the different transitions in the steady state is fixed by means of the structure and the firing rates. That is, nets for which a unique repetitive behaviour exists. For this class it is "only" a multiplying constant that has to be computed to completely define the timed behaviour.

The formalism and the basic concepts will be introduced in Section 2. In Section 3 the class of nets we will restrict to, and some concepts related to the steady state behaviour are defined. Some examples in Section 4 show that the throughput of continuous systems is not in general an upper bound of the throughput of the discrete system, and that all the non monotonicities that exist in discrete systems may appear also in their continuous relaxations. In Section 5 a programming problem is deduced that computes a tight upper bound of the steady state throughput for the subclass of continuous systems under study. Unfortunately, to solve this problem a tree structure of linear programming problems (LPP) is generated. This programming problem can be relaxed into a single LPP providing an, in general, less tight upper bound. Section 6 relates this LPP with another one that is classically used for the analysis of discrete net systems. A sufficient condition for checking the reachability of the bound is also obtained.

[1] This work has been partially supported by Project CICYT TAP98-0679.
[2] Supported by a grant from D.G.A.

2. TIMED CONTINUOUS PETRI NETS: FIRING SEMANTICS

The reader is assumed to be familiar with Petri nets (PNs) (see (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 a *continuous* PN. The main difference between continuous and discrete PNs is in the marking, which in a discrete PN is restricted to be integer, while in continuous PNs is released into the non-negative real numbers. This is a consequence of the firing, which is modified in the same way. Hence, a transition t is *enabled* at $\mathbf{m}$ iff for every $p \in {}^\bullet t$, $\mathbf{m}[p] > 0$, and its *enabling degree* is $\text{enab}(t, \mathbf{m}) = \min_{p \in {}^\bullet t}\{\mathbf{m}[p]/\mathbf{Pre}[p,t]\}$. The firing of t in a certain amount $\alpha \leq \text{enab}(t, \mathbf{m})$ 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 ($\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}$) summarises the way the marking evolves along time.

All the concepts based on the representation of the P/T net as a graph can be directly applied to continuous P/T nets, in particular, the definitions based on the annullers of the token-flow matrix. Right and left natural annullers are called T- and P-semiflows, respectively. 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 we will use a first order (or deterministic) approximation of the discrete case (Recalde and Silva, 2001), assuming that the delays associated to the firing of transitions can be approximated by their mean values. Then, the state equation 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 servers* (or *variable speed*) and *finite servers* (or *constant speed*) (Alla and David, 1998; Recalde and Silva, 2001). Infinite servers semantics will be considered here. Under infinite servers semantics, the flow through a transition t is the product of the speed, $\boldsymbol{\lambda}[t]$, and the enabling of the transition, i.e., $\mathbf{f}[t] = \boldsymbol{\lambda}[t] \cdot \text{enab}(t, \mathbf{m}) = \boldsymbol{\lambda}[t] \cdot \min_{p \in {}^\bullet t}\{\mathbf{m}[p]/\mathbf{Pre}[p,t]\}$, leading to non-linear systems.

For example, for the net system in Fig. 1,

$$\begin{aligned} \mathbf{f}[t_1] &= \boldsymbol{\lambda}[t_1] \cdot \min\{\mathbf{m}[p_1]/2, \mathbf{m}[p_4]/2\} \\ \mathbf{f}[t_2] &= \boldsymbol{\lambda}[t_2] \cdot \min\{\mathbf{m}[p_2], \mathbf{m}[p_4]\} \\ \mathbf{f}[t_3] &= \boldsymbol{\lambda}[t_3] \cdot \mathbf{m}[p_3] \end{aligned} \qquad (1)$$

Observe that in this case the state equation can be seen as switching among sets of ordinary linear differential equations. 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:

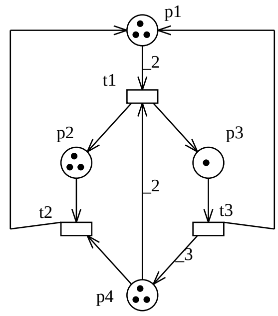


Fig. 1. A net system for which the solution of (18) with $\boldsymbol{\lambda} = \mathbf{1}$ is not reachable.

$$\begin{aligned} \dot{\mathbf{m}}(\tau) &= \mathbf{C} \cdot \boldsymbol{\Lambda} \cdot \mathbf{m}(\tau) \\ \mathbf{m}(0) &= \mathbf{m_0} \end{aligned}$$

where $\boldsymbol{\Lambda}[t,p] = \boldsymbol{\lambda}[t]/\mathbf{Pre}[p,t]$ if $p = {}^\bullet t$, and 0 otherwise. If the net system is conservative, $\mathbf{C}$ has redundant rows. Hence, some variables can be expressed in terms of the rest of the variables and the initial marking values.

If more complex semantics are used (for example, markings products) highly non linear systems of differential equations may appear. In that case it may happen that the system does not converge to a steady state, but that it approaches to a limit cycle, or even has a chaotic behaviour (Silva and Recalde, 2000).

3. STEADY STATE FLOW

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, $\mathbf{C} \cdot \mathbf{f} = \mathbf{0}$. Since $\mathbf{f} \geq \mathbf{0}$, the flow in the steady state is a T-semiflow. We will denote as $\mathbf{f}_{ss}$ the flow vector in the steady state.

A classical concept in queueing network theory is the *visit ratio*. 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 “normalisation” 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.

Two transitions, t and t', are said to be in *conflict relation* if ${}^\bullet t \cap {}^\bullet t' \neq \emptyset$. They are said to be in *equal conflict* (EQ) relation when $\mathbf{Pre}[P,t] = \mathbf{Pre}[P,t'] \neq \mathbf{0}$. A net is said to be an *Equal Conflict* (EQ) net iff for all $t, t' \in T$ such that ${}^\bullet t \cap {}^\bullet t' \neq \emptyset$, $\mathbf{Pre}[P,t] = \mathbf{Pre}[P,t']$. EQ nets are the weighted generalisation of free choice nets (Teruel and Silva, 1996).

If t_i, t_j are in EQ relation, $\dfrac{\mathbf{f}[t_i]}{\boldsymbol{\lambda}[t_i]} = \dfrac{\mathbf{f}[t_j]}{\boldsymbol{\lambda}[t_j]}$. So, in particular, $\boldsymbol{\lambda}[t_j] \cdot \mathbf{f}_{ss}[t_i] = \boldsymbol{\lambda}[t_i] \cdot \mathbf{f}_{ss}[t_j]$.

We will define the class of nets for which the visit ratio does not depend on the initial marking, but is defined by the net structure and the rates of the transitions.

Definition 1. Let $\mathcal{N}$ be a PN and $\boldsymbol{\lambda}$ a rates vector. We will say that this system is *mono T-semiflow reducible* if the following system has a unique solution:

$$\begin{aligned} &\mathbf{C} \cdot \mathbf{v}^{(1)} = \mathbf{0} \\ &\frac{\mathbf{v}^{(1)}[t_i]}{\boldsymbol{\lambda}[t_i]} = \frac{\mathbf{v}^{(1)}[t_j]}{\boldsymbol{\lambda}[t_j]} \ \ \forall\ t_i, t_j \text{ in EQ relation} \end{aligned} \tag{2}$$

For this class, only the constant χ_1 has to be obtained to know the flow vector of the system in the steady state. In this paper we will concentrate on the study of mono T-semiflow reducible systems. A particular case of mono T-semiflow reducible nets are EQ nets that allow a live and bounded marking, i.e., structurally live and bounded EQ nets (Teruel and Silva, 1996).

Let us consider an example of a flexible manufacturing system (Fig. 2), composed of three machines. 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 A and 15 pallets of type B.

The net in Fig. 2 is mono T-semiflow reducible (more precisely, it is a marked graph with two shared resources, M1 and M2). The visit ratio of this PN 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). If this system is seen as discrete, the reachability space is very large, and the exact computation of the throughput becomes unfeasible. However, just because the number of tokens is quite big, studying the system as a continuous PN may be a good approximation. In the following sections we will analyse this system in more detail. First, several kinds of behavioural "anomalies" will be pointed out in Section 4.

4. "UNEXPECTED" BEHAVIOURS

4.1 Continuous is not an upper bound of discrete

It could be thought that, since continuisation 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. 3(a), with $\boldsymbol{\lambda} = [3, 1, 1, 10]$, the throughput is 0.801 as discrete while it is only 0.535 as continuous.

4.2 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. 3(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, Fig. 3(c) shows how the throughput of the net system in Fig. 3(b) changes if the rate of t_1 varies from 0 to 5, assuming $\boldsymbol{\lambda}[t_2] = 1$. Notice that even a *discontinuity* (!) appears at $\boldsymbol{\lambda}[t_1] = 2$.

5. COMPUTATION OF STEADY STATE FLOW BOUNDS

Let $\mathbf{m}_{ss}$ be the steady state marking of a continuous net system. Then, for every $\tau > 0$ it must verify:

$$\dot{\mathbf{m}}(\tau) = \mathbf{C} \cdot \mathbf{f}(\tau) \tag{3}$$

$$\mathbf{f}(\tau)[t] = \boldsymbol{\lambda}[t] \cdot \min_{p \in {}^\bullet t} \left\{ \frac{\mathbf{m}(\tau)[p]}{\mathbf{Pre}[p,t]} \right\} \ \ \forall t \in T \tag{4}$$

$$\mathbf{m}(0) = \mathbf{m_0} \tag{5}$$

$$\mathbf{m}_{ss} = \lim_{\tau \to \infty} \mathbf{m}(\tau) \tag{6}$$

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

$$\boldsymbol{\mu}_{ss} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}, \tag{7}$$

$$\boldsymbol{\phi}_{ss}[t] = \boldsymbol{\lambda}[t] \cdot \min_{p \in {}^\bullet t} \left\{ \frac{\boldsymbol{\mu}_{ss}[p]}{\mathbf{Pre}[p,t]} \right\} \ \ \forall t \in T \tag{8}$$

$$\mathbf{C} \cdot \boldsymbol{\phi}_{ss} = \mathbf{0} \tag{9}$$

$$\boldsymbol{\mu}_{ss}, \boldsymbol{\sigma} \geq \mathbf{0} \tag{10}$$

Equation (7) is obtained from (3) and (5), while (8) is a particularisation of (4). Since $\mathbf{m}_{ss}$ is a steady state, from (3), (9) is deduced. With this relaxation we have replaced the condition of being a reachable marking with being a solution of (7), the fundamental equation. That is, we are loosing 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 $1 \leq \alpha \leq 5/3$, verifies (7-10).

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

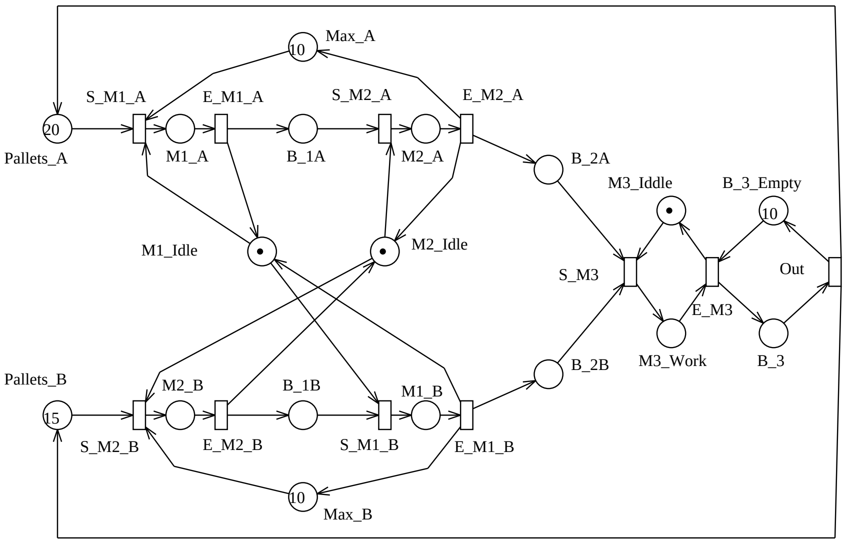
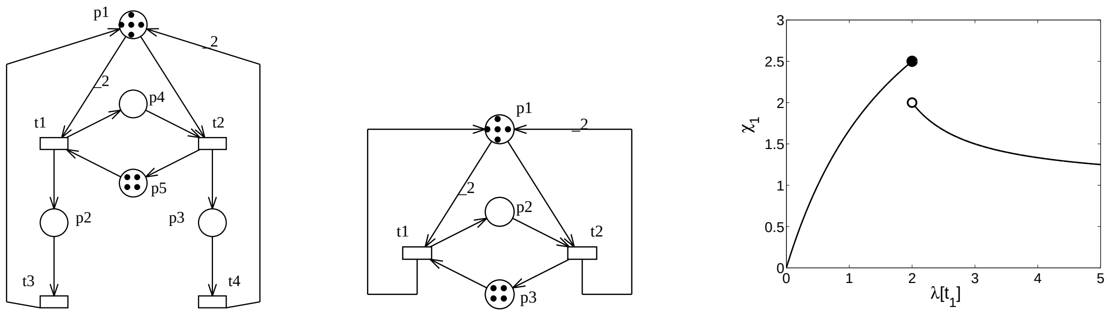


Fig. 2. A PN model of the manufacturing system described in Section 3



(a) (b) (c)

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

$$\max\{\boldsymbol{\phi}_{ss}[t_1] \mid \boldsymbol{\mu}_{ss} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}$$
$$\boldsymbol{\phi}_{ss}[t] = \boldsymbol{\lambda}[t] \cdot \min_{p \in {}^\bullet t} \left\{ \frac{\boldsymbol{\mu}_{ss}[p]}{\mathbf{Pre}[p,t]} \right\} \; \forall t \in T, \quad (11)$$
$$\mathbf{C} \cdot \boldsymbol{\phi}_{ss} = \mathbf{0}, \;\; \boldsymbol{\mu}_{ss}, \boldsymbol{\sigma} \geq \mathbf{0}\}$$

Proposition 2. Let $\mathcal{N}$ be consistent and conservative. Conditions (7) and $\boldsymbol{\sigma} \geq \mathbf{0}$ are equivalent to

$$\forall \mathbf{y} \geq \mathbf{0} \text{ such that } \mathbf{y} \cdot \mathbf{C} = \mathbf{0}, \text{ then } \mathbf{y} \cdot \boldsymbol{\mu}_{ss} = \mathbf{y} \cdot \mathbf{m_0}$$

PROOF. "⇒", is immediate. For "⇐", since the net is conservative, a base of non-negative left annullers of $\mathbf{C}$ exists. For every basis vector $\mathbf{y}$, $\mathbf{y} \cdot (\boldsymbol{\mu}_{ss} - \mathbf{m_0}) = \mathbf{0}$. Therefore, $\boldsymbol{\sigma}$ exists such that $\boldsymbol{\mu}_{ss} - \mathbf{m_0} = \mathbf{C} \cdot \boldsymbol{\sigma}$ and, since it is consistent, we can assume that $\boldsymbol{\sigma} \geq \mathbf{0}$. ☐

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. Otherwise stated, 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 (11) is always "reachable" in the sense that with a suitable initial distribution of the tokens inside the P-semiflows, this throughput can be obtained (for instance, with the same steady state distribution). In this sense, we will say that the solution is a tight bound.

Nevertheless, the programming problem in (11) is difficult to solve due to the "min" condition, that makes it non linear. This condition can be relaxed (linearised) as follows:

$$\boldsymbol{\phi}_{ss}[t] = \boldsymbol{\lambda}[t] \cdot \frac{\boldsymbol{\mu}_{ss}[p]}{\mathbf{Pre}[p,t]} \quad \text{if} \quad p = {}^\bullet t \quad (12)$$

$$\boldsymbol{\phi}_{ss}[t] \leq \boldsymbol{\lambda}[t] \cdot \frac{\boldsymbol{\mu}_{ss}[p]}{\mathbf{Pre}[p,t]} \quad \forall p \in {}^\bullet t \quad \text{otherwise} \quad (13)$$

$$\frac{\boldsymbol{\phi}_{ss}[t_i]}{\boldsymbol{\lambda}[t_i]} = \frac{\boldsymbol{\phi}_{ss}[t_j]}{\boldsymbol{\lambda}[t_j]} \quad \forall \; t_i, t_j \text{ in EQ relation} \quad (14)$$

This way we have a linear programming problem (LPP) defined by equations (7), (9), (12), and (14), and inequalities (10) and (13), that can be solved in polynomial time.

For the system in Fig. 2, let the delays associated to the operations be: $\boldsymbol{\lambda}$[S_M1_A] $=$ $\boldsymbol{\lambda}$[S_M2_A] $= 1$ t.u., $\boldsymbol{\lambda}$[S_M1_B] $=$ $\boldsymbol{\lambda}$[S_M2_B] $= 1$ t.u., $\boldsymbol{\lambda}$[S_M3] $=$ $\boldsymbol{\lambda}$[Out] $= 1$ t.u., $\boldsymbol{\lambda}$[E_M1_A] $=$ $\boldsymbol{\lambda}$[E_M2_B] $= 3$ t.u.,

$\boldsymbol{\lambda}$[E_M1_B] $= 5$ t.u., and $\boldsymbol{\lambda}$[E_M2_A] $= \boldsymbol{\lambda}$[E_M3] $=$ 4 t.u. The solution of the LPP is 0.1111.

Unfortunately, the 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 may occur because it may be the case that for that solution none of the input places of a synchronisation really restricts the flow of that transition. When this happens, the marking cannot 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, the throughput would be 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.

The way to improve this bound is to force the equality for at least one place per synchronisation. This corresponds to a correct interpretation of the *min* operator in (11). The problem is that there is no way to know in advance which of the input places should restrict the flow, and in general the number of linear problems to be solved is $\Pi_{t \in T} |^\bullet t|$. A branch and bound algorithm can be used to improve the bound, computing exactly what (11) expresses. The idea is to solve the LPP defined by the system of (in)-equalities (7), (9), (10), (12), (13) and (14). If the marking does not correspond to a steady state (i.e., there is at least one transition such that all its input places have "too many" 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 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 possible 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.

For example, the visit ratio of the system in Fig. 2 is equal to $\mathbf{1}$. A solution of the original LPP 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] $= 0.222$, $\mathbf{m}$[B_3_Empty] $= 9.888$, $\mathbf{m}$[M3_Work] $= 0.888$. Observe that the throughput of transitions S_M2_A and E_M3 does not fit the marking of their input places. So, we should build two LPPs, adding in each one an equation for S_M2_A. If we force the throughput to be defined by M2_Idle, the linear system is *unfeasible*, and if we add a restriction for B_1A 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 the system is *unfeasible*. Adding an equation for M3_Work 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 can be a steady state marking, and no larger throughput may exist (observe that in this case we have obtained the same throughput as in the original LPP). Hence, in this case it is not necessary to go on verifying the other synchronisations.

6. ON DUALITY AND BRANCHING ELIMINATION

Let us consider again the problem defined by (in)-equalities (7), (8), (9) and (10). A simple relaxation of (8) consists in just looking if each place has enough tokens to fire all its output transitions according to the flow vector. We add the equation that relates the throughput of transitions in EQ relation, which can also be deduced from (8). This can be written as a single LPP:

$$\begin{aligned}\max\{\boldsymbol{\phi}_{ss}[t_1] \mid\ & \boldsymbol{\mu}_{ss} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \\ & \boldsymbol{\mu}_{ss} \geq \max_{t \in p^\bullet} \left\{ \frac{\mathbf{Pre}[p,t] \cdot \boldsymbol{\phi}_{ss}[t]}{\boldsymbol{\lambda}[t]} \right\} \ \forall p \in P \\ & \mathbf{C} \cdot \boldsymbol{\phi}_{ss} = \mathbf{0} \\ & \frac{\boldsymbol{\phi}_{ss}[t_i]}{\boldsymbol{\lambda}[t_i]} = \frac{\boldsymbol{\phi}_{ss}[t_j]}{\boldsymbol{\lambda}[t_j]} \quad \forall\ t_i, t_j \text{ in EQ relation} \\ & \boldsymbol{\sigma}, \boldsymbol{\mu}_{ss} \geq \mathbf{0}\}\end{aligned}$$

Since in mono T-semiflow reducible systems $\mathbf{v}^{(1)}$ is completely defined, if $\boldsymbol{\phi}_{ss} = \chi \cdot \mathbf{v}^{(1)}$, this LPP can be written as:

$$\begin{aligned}\max\{\chi \mid\ & \boldsymbol{\mu}_{ss} = \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma} \\ & \boldsymbol{\mu}_{ss} \geq \chi \cdot \mathbf{PD}, \quad \boldsymbol{\sigma}, \boldsymbol{\mu}_{ss} \geq \mathbf{0}\}\end{aligned} \tag{15}$$

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

Defining $\alpha = 1/\chi$ and $\boldsymbol{\sigma}' = 1/\chi \cdot \boldsymbol{\sigma}$, (15) reduces to:

$$\min\{\alpha \mid \alpha \cdot \mathbf{m_0} + \mathbf{C} \cdot \boldsymbol{\sigma}' \geq \mathbf{PD}, \quad \boldsymbol{\sigma}' \geq \mathbf{0}\} \tag{16}$$

The dual of this LPP is:

$$\max \{\mathbf{y} \cdot \mathbf{PD} \mid \mathbf{y} \cdot \mathbf{C} \leq \mathbf{0},\ \mathbf{y} \cdot \mathbf{m_0} \leq 1,\ \mathbf{y} \geq \mathbf{0}\} \tag{17}$$

One of the formulations of the alternatives theorem states that it is equivalent $\exists \mathbf{x} > \mathbf{0}$ such that $\mathbf{C} \cdot \mathbf{x} \geq \mathbf{0}$, or $\forall \mathbf{y} \geq \mathbf{0}$ such that $\mathbf{y} \cdot \mathbf{C} \leq \mathbf{0}$ then $\mathbf{y} \cdot \mathbf{C} = \mathbf{0}$ (see for example (Murty, 1983)). Since the net is consistent, $\mathbf{y} \cdot \mathbf{C} \leq \mathbf{0}, \mathbf{y} \geq \mathbf{0}$ can be replaced with $\mathbf{y} \cdot \mathbf{C} = \mathbf{0}, \mathbf{y} \geq \mathbf{0}$. 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 this LPP:

$$\gamma = \max \{\mathbf{y} \cdot \mathbf{PD} \mid \mathbf{y} \cdot \mathbf{C} = \mathbf{0} \quad \mathbf{y} \cdot \mathbf{m_0} = 1, \;\; \mathbf{y} \geq \mathbf{0}\} \tag{18}$$

Then, the throughput in the steady state verifies

$$\mathbf{f}_{ss} \leq \frac{1}{\gamma} \mathbf{v}^{(1)}$$

Intuitively, the idea of (18) is to find the slowest isolated subnet among those generated by the elementary P-semiflows. In other words, the bound is obtained looking at the bottleneck P-semiflow. It generalises the kind of result in (Campos and Silva, 1992) for discrete systems, where conflicts were forbidden except among immediate transitions.

For each marking $\mathbf{m}$, we will define its T-coverture (T-cov$(\mathbf{m})$) 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

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

A characterisation can be obtained for the solution of (18) being the exact value. Given a vector $\mathbf{v}$, let us denote as $\|\mathbf{v}\|$ the set of its non-zero components.

Proposition 5. Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a continuous system and $\boldsymbol{\lambda}$ a set of rates for which the system is mono T-semiflow reducible.

The flow computed with (18) (or (15)) is the flow in the steady state iff the T-coverture at the steady state, T-cov$(\mathbf{m}_{ss})$, contains the support of a P-semiflow.

Moreover, the maximum of (18) is reached for the P-semiflow contained in the T-coverture.

PROOF. Let $\mathbf{m}_{ss}$ be the steady state marking of the system, $\mathbf{f}_{ss} = \chi_1 \cdot \mathbf{v}^{(1)}$ the flow vector associated to this state, and γ the solution of (18). Applying (18), $\chi_1 \leq 1/\gamma$.

For "⇒", 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$(\mathbf{m}_{ss})$, a place $p \in \|\mathbf{y}_0\|$ exists such that $\mathbf{m}_{ss}[p] > \max_{t \in p^\bullet} \{\mathbf{Pre}[p,t] \cdot \chi_1 \cdot \mathbf{v}^{(1)}[t]/\boldsymbol{\lambda}[t]\} = \chi_1 \cdot \mathbf{PD}[p]$. Hence, $\mathbf{y}_0 \cdot \mathbf{m}_{ss} > \chi_1 \cdot \mathbf{y}_0 \cdot \mathbf{PD}$, and $1/\chi_1 > \mathbf{y}_0 \cdot \mathbf{PD} = \gamma$, contradiction.

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

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

Clearly, if the structure of the net is such that every T-coverture contains a P-semiflow, Prop. 5 holds.

Corollary 6. Let $\mathcal{N}$ be a continuous net, and $\boldsymbol{\lambda}$ a set of rates for which it is mono T-semiflow reducible. 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 of (18).

Observe that the checking of some of the T-covertures can be spared. It is sufficient to prove that the minimal T-covertures contain the support of a P-semiflow. However, this condition is in general difficult to solve since the number of covertures may be very large. Nevertheless, corollary 6 holds for instance for str. live and str. bounded EQ nets, or equivalently, EQ nets that are consistent, conservative and the rank of the token flow matrix is upper bounded by the number of conflicts (Teruel and Silva, 1996). More general classes exist for which this result holds too. For instance, it holds for the net system in Fig. 2.

7. CONCLUDING REMARKS

The main contribution of this paper is a new approach for the computation of the steady state flow of mono T-semiflow reducible nets. A tight bound is obtained solving a non-linear programming problem. A relaxation leads to a single LPP, and a sufficient condition for the solution being a bound of the flow in the steady state is presented. As a manufacturing application, the steady state flow of a system, modelled with a marked graph with two interleaved resources, is computed in polynomial time.

REFERENCES

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

Balduzzi, F., G. Menga and A. Giua (2000). First-order hybrid Petri nets: a model for optimization and control. *IEEE Trans. on Robotics and Automation* **16**(4), 382–399.

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

Dicesare, F., G. Harhalakis, J. M. Proth, M. Silva and F. B. Vernadat (1993). *Practice of Petri Nets in Manufacturing*. 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*. Wiley and Sons.

Recalde, L. and M. Silva (2001). Petri Nets fluidification revisited: Semantics and steady state. *APII-JESA* **35**(4), 435–449.

Silva, M. and L. Recalde (2000). Réseaux de Petri et relaxations de l'integralité: Une vision des réseaux continus. In: *Conférence Internationale Francophone d'Automatique (CIFA 2000)*. pp. 37–48.

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