# Basic Server Semantics and Performance Monotonicity of Continuous Petri Nets

Cristian Mahulea, Laura Recalde, and Manuel Silva

**Abstract**

Continuous Petri nets were introduced as an approximation to deal with the state explosion problem which can appear in discrete event models. When time is introduced, the flow through a fluidified transition can be defined in many ways. The most used in literature are constant and variable speed [8], which can be seen as some kind of finite and infinite server interpretations of the transitions behavior, and derived from stochastic (discrete) Petri nets [18]. In this paper we will compare the results obtained with both relaxations for the broad class of mono-T-semiflow reducible nets, and prove that, under some frequently true conditions, infinite server semantics offers a throughput which is closer to the throughput of the discrete system in steady state. In the second part, it will be proved that the throughput of mono-T-semiflow reducible net systems is monotone with respect to the speed of the transitions and the initial marking of the net.

## I. Introduction

Petri nets (PNs) are a well-known formalism to deal with discrete event systems (DES) [9]. The state explosion problem appears frequently in such systems, making the enumerative analysis methods inapplicable in many practical cases. Fluidification is a classical approximation technique that relaxes the description of the system by removing the integrality constraints. Applying this idea to discrete PNs, the firing of transitions is not limited to natural numbers but to positive real numbers leading to continuous Petri nets (contPNs) [8], [17].

In the last years, contPNs have been used in many applications in which the number of clients is relatively big. The main application has been in manufacturing systems [14], [4], [3] but have been used

To appear as: C. Mahulea, L. Recalde, M. Silva, "Basic Server Semantics and Performance Monotonicity of Continuous Petri Nets," *Discrete Event Dynamic Systems*, 2008.

C. Mahulea, L. Recalde and M. Silva are with Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza, María de Luna 1, E-50018 Zaragoza, Spain, {cmahulea,lrecalde,silva}@unizar.es.

also in traffic systems [11] or supply chains [10], for example. Fluid models may be studied by means of net based structural analysis techniques [12], [5].

As in the discrete case, contPNs can be autonomous (untimed) or can have time associated with the transitions or places, namely timed contPNs. In the literature, in the continuous case, timing is mainly associated to transitions and two basically different ways of defining their firing are often used: *finite server semantics* (or *constant speed*) and *infinite server semantics* (or *variable speed*) [8], [18], [17].

For discrete PNs infinite server semantics is *more general*, since it can simulate finite server semantics [1]. However, the continuous approximation of this model under infinite server semantics is not equivalent to using finite server semantics in the original fluidified model. In the continuous case the two semantics are in fact related to conceptually and technically different relaxations of the model. Since these two semantics are different, the immediate question is, given a net system, which one is better? Up to now, the first reference to this problem we were able to find in the literature is a remark in [2], where the authors mention having observed that in several cases *infinite server semantics* provides a very accurate approximation of discrete PNs.

One of the goals of this paper is to provide some results about the use of these semantics. In general, is shown that it is not possible to say that one of them is always better than the other. Hence, we will concentrate on a subclass of nets, hopefully with a very significant modeling power from a practical point of view. The subclass is called mono-T-semiflow reducible [12], [13]. Focusing on live and bounded systems, this class includes mono-T-semiflow nets [12] and equal conflict nets [20] (thus free-choice and choice-free [19]). We will prove that for mono-T-semiflow reducible nets infinite server semantics provides a throughput which is closer to the throughput of the discrete system in steady state, under quite conditions in practice.

Another good indicator of the practical interest of a class of net models is the kind of properties it verifies. In the last part of the paper *monotonicity* of the throughput w.r.t. the firing speed of transitions and the initial marking is studied. It is proved that under a quite general condition the class of mono-T-semiflow reducible nets enjoys these monotonicity properties. This means, for example, that if a machine is replaced by a faster one or more resources are available, the production rate will never decrease.

A purely structural property that will be necessary to check is whether the places that restrict the firing of transitions in the steady state contain the support of a P-semiflow. Two algorithms are presented for this.

The paper is organized as follows: in Section II basic definitions of timed contPN are given. In particular, the procedures to compute the evolution of continuous net systems under finite and infinite

server semantics are recalled. Section III is devoted to the comparison and a more in depth study of the two semantics, concentrating on the class of mono-T-semiflow reducible nets. Sufficient conditions that guarantee steady state performance monotonicity w.r.t. the speed of transitions and w.r.t. the initial marking are proved in Section IV. Two algorithms to check structural properties sufficient for monotonicity are shown in Section V. A manufacturing system is taken as case study in Section VI and it is shown how the previous results can be used for its analysis. Some conclusions are given in Section VII.

## II. Preliminaries

### A. Untimed continuous Petri nets

***Definition** 2.1:* A *contPN system* is a pair $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$, where: $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle$ is a *net structure* (with set of places $P$, set of transitions $T$, the pre and post incidence matrices $\boldsymbol{Pre}, \boldsymbol{Post} : P \times T \to \mathbb{N}$) and $\boldsymbol{m_0}$ is the *initial marking*.

The token load of the place $p_i$ at marking $\boldsymbol{m}$ is denoted by $m_i$ and *preset* and *postset* of a node $X \in P \cup T$ are denoted by ${}^\bullet X$ and $X^\bullet$, respectively.

A transition $t_j \in T$ is *enabled* at $\boldsymbol{m}$ iff $\forall p_i \in {}^\bullet t_j$, $m_i > 0$ and its enabling degree is $enab(t_j, \boldsymbol{m}) = \min\limits_{p_i \in {}^\bullet t_j} \left\{ \frac{m_i}{\boldsymbol{Pre}(p_i, t_j)} \right\}$. An enabled transition $t$ can fire in any real amount $0 < \alpha < enab(t, \boldsymbol{m})$ leading to a new marking $\boldsymbol{m'} = \boldsymbol{m} + \alpha \boldsymbol{C}(\cdot, t)$, where $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is the token-flow matrix. If $\boldsymbol{m}$ is reachable from $\boldsymbol{m_0}$ through a finite sequence $\sigma$, a *state (or fundamental) equation* can be written: $\boldsymbol{m} = \boldsymbol{m_0} + \boldsymbol{C} \cdot \boldsymbol{\sigma}$, where $\boldsymbol{\sigma} \in \mathbb{N}^{|T|}$ is the firing count vector.

A contPN is *bounded* when every place is bounded ($\forall p \in P, \exists b_p \in \mathbb{R}_{\geq 0}$ with $m(p) \leq b_p$ at every reachable marking $\boldsymbol{m}$). It is *live* when every transition is *live* (it can ultimately occur from every reachable marking).

Right and left non negative annullers of the token flow matrix $\boldsymbol{C}$ are called T- and P-semiflows, respectively. If non negativity is not required, the annullers are called T- and P-flows. When $\boldsymbol{y} \cdot \boldsymbol{C} = 0$, $\boldsymbol{y} > 0$ the net is said to be *conservative*, and when $\boldsymbol{C} \cdot \boldsymbol{x} = 0$, $\boldsymbol{x} > 0$ the net is said to be *consistent*.

Two transitions $t_1$ and $t_2$ are said to be in *conflict relation* if ${}^\bullet t_1 \cap {}^\bullet t_2 \neq \emptyset$. Two transitions $t_1$ and $t_2$ are in *continuous equal conflict* (CEQ) relation when there exists $k > 0$ such that $\boldsymbol{Pre}[P, t_1] = k \cdot \boldsymbol{Pre}[P, t_2] \neq 0$.

### B. Timed continuous Petri nets

Under any timed interpretation of the net model, the fundamental equation depends on time: $\boldsymbol{m}(\tau) = \boldsymbol{m_0} + \boldsymbol{C} \cdot \boldsymbol{\sigma}(\tau)$. Differentiating with respect to it, the following equation is obtained: $\dot{\boldsymbol{m}}(\tau) = \boldsymbol{C} \cdot \dot{\boldsymbol{\sigma}}(\tau)$.

| Clients | Servers | Semantics of the transition |
|---|---|---|
| Many(C) | Many(C) | infinite server semantics |
| Many(C) | Few(D) | finite server semantics |
| Few(D) | Few(D) | discrete transitions |
| Few(D) | Many(C) | discrete transitions |

TABLE I

ON THE FLUIDIFICATION OF A TRANSITION [18]

The derivative of the firing sequence will be called the *(firing) flow* of the timed model: $\boldsymbol{f}(\tau) = \dot{\boldsymbol{\sigma}}(\tau)$.

Different definitions of the flow of continuous timed transitions have been given, the two most important being *finite server* (or *constant speed*) and *infinite server* (or *variable speed*) [2], [17].

*1) Preliminaries for the server semantics:* In general, transitions are interpreted as *stations* (in QN terminology), where *servers* and *clients* meet. Thus, the more appropriate firing relaxation depends on the relative number of servers and clients in the discrete model we want to approximate. Assuming, qualitatively speaking, that there may be "many" or "few" of each one of them, fluidification can be considered for clients, for servers or for both. Table I represents the four theoretically possible cases. If the number of clients is "small" (Few-Few and Few-Many in Table I), the system is not really loaded, the transitions "should" remain discrete and the fluidification may be unsuitable. If there are many clients and a few servers (Many-Few) the relaxation is only at the level of clients, and finite server semantics can provide a good approximation. On the other hand, in the case of many clients and many servers (Many-Many), a continuous model with infinite server semantics seems reasonable, since there are so many servers that there is no need to make them explicit.

It is important to notice that finite server semantics corresponds at conceptual level to a *hybrid* behavior: fluidification is applied only to clients, while servers are kept as discrete, counted as a finite number (used to define the bound of firing speed of the transition). On the other hand, infinite server semantics really relaxes clients and servers, being the firing speed driven by the enabling degree of the transition, like for stochastic-markovian (discrete) PNs.

*2) Finite Server Semantics:* Under *finite server* (constant speed), each transition $t_j$ has associated a real positive number, $\lambda_j$, called *maximal firing speed*. If the markings of the input places of the transition are strictly greater than zero (*strongly enabled*), its flow will be constant, equal with this value (all servers working at full speed). Otherwise (*weakly enabled*), the flow will be the minimum between its

*maximal firing speed* and the total input flow to the places with zero marking. With this definition, $\lambda_j$ represents the product of the number of servers in the transition and their speed. The instantaneous flow of a transition $t_j$ is given by:

$$f_j = \begin{cases} \lambda_j, \text{ if } \nexists p_i \in {}^\bullet t \text{ with } m_i = 0 \\ \min\left\{ \min\limits_{p_i \in {}^\bullet t | m_i = 0} \left\{ \sum\limits_{t' \in {}^\bullet p_i} \frac{f[t'] \cdot Post[t', p_i]}{Pre[p_i, t_j]} \right\}, \lambda_j \right\} \text{ otherwise} \end{cases} \tag{1}$$

Observe that (1) is not defining completely the flow of a contPN under finite server semantics. In the case of conflict, a resolution policy should be specified, otherwise many solutions of the flows are possible [5]. Therefore, this semantics is non-deterministic as defined in (1).

*3) Infinite Server Semantics:* Under *infinite server* (variable speed) the flow of a transition $t_j$ is:

$$f_j = \lambda_j \cdot enab(t_j, \boldsymbol{m}) = \lambda_j \cdot \min_{p_i \in {}^\bullet t_j} \left\{ \frac{m_i}{\boldsymbol{Pre}[p_i, t_j]} \right\} \tag{2}$$

The *enabling degree* of the transition $t_j$ represents the number of active servers for that transition at $\boldsymbol{m}$. The flow will be the number of active servers times the work each one does per time unit ($\lambda_j$). Notice that the number of active servers in a transition (station) depends only on the marking of its input places.

***Definition** 2.2:* [15] A *configuration* $\mathcal{C}_i$ of $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ at $\boldsymbol{m}$ is a set of $(p, t)$ arcs, one per transition, such that $p \in {}^\bullet t$ and

$$\boldsymbol{f}[t] = \boldsymbol{\lambda}[t] \cdot \frac{\boldsymbol{m}[p]}{\boldsymbol{Pre}[p, t]}.$$

The number of configurations is upper bounded by: $\prod\limits_{t \in T} |{}^\bullet t|$. In general, we are interested in the set of places that limit the enabling degree. So, associated to a configuration $\mathcal{C}_i$, we define the corresponding *P-configuration*, the set of places from which there exist at least one output arc belonging to $\mathcal{C}_i$. Formally, the P-configuration of $\mathcal{C}_i$, is:

$$P - \mathcal{C}_i = \{p_j | \exists (p_j, t_k) \in \mathcal{C}_i\}.$$

A contPN system evolves and may reach a steady state (i.e. a marking such that $\dot{\boldsymbol{m}}(\tau) = 0$). The (P-)configuration of the steady state marking will be called the *steady state (P-)configuration*. This steady state marking, if it exists, has to fulfill certain conditions: the flow it defines has to be a T-semiflow (because $\dot{\boldsymbol{m}} = \boldsymbol{C} \cdot \boldsymbol{f} = \boldsymbol{0}$), and has to verify the equations defined by the P-flows (see [15] for more details). The solutions of these equations represent all the possible ways of distributing the tokens of the P-flows so that the system remains on that marking. However, it may happen that several markings fulfill these conditions. We will refer to all these possible steady states markings as *possible equilibrium markings*. All their (P-)configurations will be called *possible equilibrium (P-)configurations*.

Through this paper we will assume that the net system will reach a steady state after some time, and try to derive some results about this steady state. However, it may happen that the system does not approach to steady state, even being a mono-T-semiflow net system.

We want to stress that under both semantics, the timed system is defined by the untimed net system and a positive vector, $\boldsymbol{\lambda}$ (i.e. here it is assumed that there are no immediate transitions), although $\boldsymbol{\lambda}$ has a different meaning under each semantics: it is the firing rate of a transition in the case of infinite server semantics, and it is a maximal firing speed in the case of finite server semantics (the product of the number of servers and the firing rate of one server).

### C. Mono-T-semiflow reducible nets

As in queueing network theory, the *visit ratio* of transition $t_j$ with respect to $t_i$, $\boldsymbol{v}^{(i)}[t_j]$, is the average number of times $t_j$ is visited (fired) for each visit to (firing of) the reference transition $t_i$ in steady-state. *Mono-T-semiflow reducible* nets is a class for which the visit ratio vector depends only on the net and the firing rate vector, but not on $\boldsymbol{m_0}$, i.e., $\boldsymbol{v}^{(i)} = \boldsymbol{v}^{(i)}(\mathcal{N}, \boldsymbol{\lambda})$ [12].

***Definition** 2.3:* [12] $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ is a *mono-T-semiflow reducible* net if it is consistent, conservative and the following system has a unique solution (its visit ratio):

$$\begin{cases} \boldsymbol{C} \cdot \boldsymbol{v}^{(1)} = \boldsymbol{0} \\ \frac{\boldsymbol{v}^{(1)}[t_i]}{\boldsymbol{Pre}[p,t_i] \cdot \lambda[t_i]} = \frac{\boldsymbol{v}^{(1)}[t_j]}{\boldsymbol{Pre}[p,t_j] \cdot \lambda[t_j]} \quad \forall t_i, t_j \text{ in CEQ relation}, \forall p \in {}^{\bullet}t_i \\ \boldsymbol{v}^{(1)}[t_1] = 1 \end{cases} \tag{3}$$

Based on the above definition, an immediate result is:

***Proposition** 2.4:* Let $\langle \mathcal{N}, \boldsymbol{\lambda}_1, \boldsymbol{m}_0 \rangle$ be a mono-T-semiflow reducible net system, and $\boldsymbol{\lambda}_2$ a rates vector such that $\boldsymbol{\lambda}_1$ and $\boldsymbol{\lambda}_2$ keep the same proportion in continuous equal conflicts, i.e., for every pair $t_i, t_j$ in CEQ relation $\boldsymbol{\lambda}_1[t_i]/\boldsymbol{\lambda}_1[t_j] = \boldsymbol{\lambda}_2[t_i]/\boldsymbol{\lambda}_2[t_j]$. The timed contPN systems $\langle \mathcal{N}, \boldsymbol{\lambda}_1, \boldsymbol{m_0} \rangle$ and $\langle \mathcal{N}, \boldsymbol{\lambda}_2, \boldsymbol{m_0} \rangle$ have the same visit ratio vector: $\boldsymbol{v}^{(i)}(\mathcal{N}, \boldsymbol{\lambda}_1) = \boldsymbol{v}^{(i)}(\mathcal{N}, \boldsymbol{\lambda}_2)$.

Through this paper, if $\boldsymbol{\lambda}_1$, $\boldsymbol{\lambda}_2$ satisfy the above condition, we will say that they are *visit ratio preserving*. This can be extended to sets of rates.

***Example** 2.5:* The mono-T-semiflow reducible net in Fig. 1 represents a queuing network, adapted from [7]. It has four minimal T-semiflows: $\boldsymbol{x_1} = t_1 + t_2 + t_3$, $\boldsymbol{x_2} = t_1 + t_2 + t_4 + t_6 + t_8 + t_{11}$, $\boldsymbol{x_3} = t_6 + t_8 + t_{10}$ and $\boldsymbol{x_4} = t_1 + t_2 + t_5 + t_7 + t_9 + t_{12}$. The values of $\lambda_3$, $\lambda_4$ and $\lambda_5$ will determine the splitting of the flow entering in $p_3$ (because $t_3$, $t_4$ and $t_5$ are in free-choice) while $\lambda_{10}$ and $\lambda_{11}$ will define the splitting of the flow entering in $p_{11}$. For example, for the particular value of $\boldsymbol{\lambda} = \boldsymbol{1}$, the

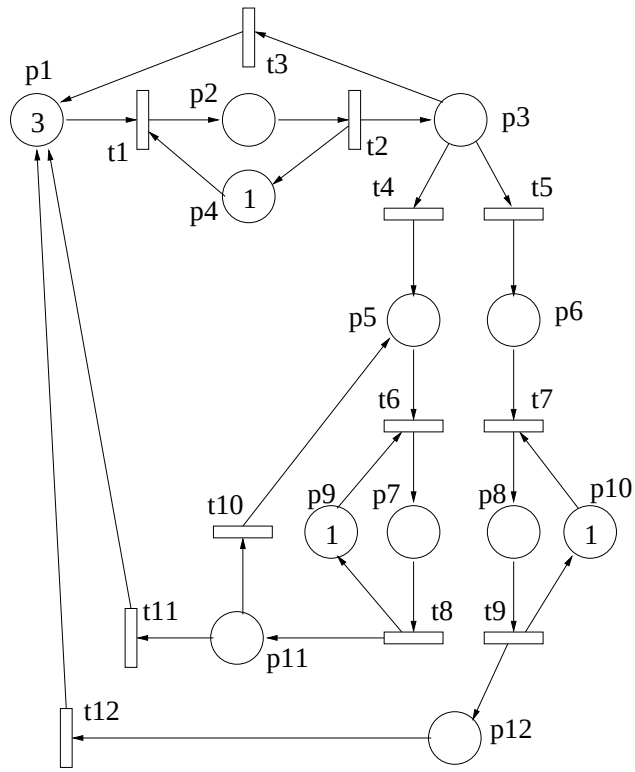


Fig. 1. Continuous mono-T-semiflow reducible net system, adapted from a queuing network in [7]. In fact, it is an EQ net; $\{t_3, t_4, t_5\}$ and $\{t_{10}, t_{11}\}$ are equal conflicts.

visit ratio vector normalized for $t_3$ is $\boldsymbol{v}^{(3)} = [3, 3, 1, 1, 1, 2, 1, 2, 1, 1, 1, 1]^T$ (the addition of the minimal T-semiflows). ■

## III. Finite server semantics vs. infinite server semantics

### A. Motivation Example

Let us consider the contPN system in Fig. 2, modeling a shared resource (place $p_6$) among two processes and let us see its evolution with both continuous approximations: finite and infinite server semantics. Let $\boldsymbol{\lambda} = [1, 2, 1, 1, 0.5]$. Observe that in this case the behavior of the discrete PN is the same for finite and infinite server semantics because the (single) servers are implicit in the model. On the contrary, continuous finite and infinite server semantics do not lead to the same values.

*Infinite server semantics.* First, observe that $p_2$ is implicit (i.e. it is never the only place to constrain the firing of $t_4$ [9]) and its marking verifies: $m_2(\tau) = m_4(\tau) + m_6(\tau)$. Hence, $f_4 = \min\{m_2, m_6\} = m_6$, and so only two configurations can govern the system evolution. At $\tau = 0$, $m_3 < m_6$, therefore the

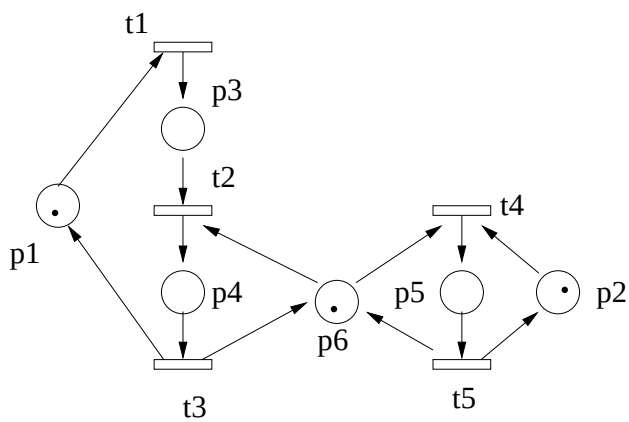


Fig. 2. ContPN system used in Section III-A.

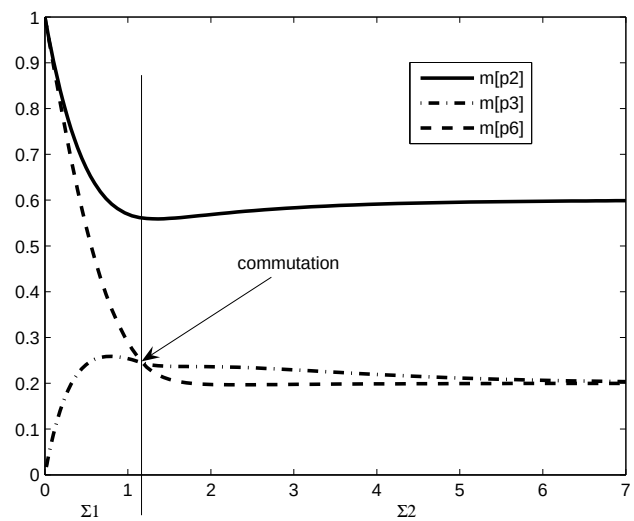


Fig. 3. Evolution of contPN in Fig. 2 with $\boldsymbol{\lambda} = [1, 2, 1, 1, 0.5]^T$ under infinite server semantics.

evolution is governed by the configuration $\mathcal{C}_1 = \{(p_1, t_1), (p_3, t_2), (p_4, t_3),\ (p_6, t_4), (p_5, t_5)\}$. The linear system in $\mathcal{C}_i$ will be called $\Sigma_i$.

The evolution of the contPN system is sketched in Fig. 3. It evolves according to $\Sigma_1$ until $\tau \simeq 1.14$ t.u. when $m_3(\tau) = m_6(\tau)$. At that point, a switch occurs and the new governing configuration is: $\mathcal{C}_2 = \{(p_1, t_1),\ (p_6, t_2),\ (p_4, t_3),\ (p_6, t_4),\ (p_5, t_5)\}$. The system evolves according to $\Sigma_2$ and reaches the steady state marking $[0.4, 0.6, 0.2, 0.4, 0.4, 0.2]$ with the corresponding flow: $[0.4, 0.4, 0.4, 0.2, 0.2]$.

*Finite server semantics.* The evolution of the system under finite server semantics is presented in Fig. 4.

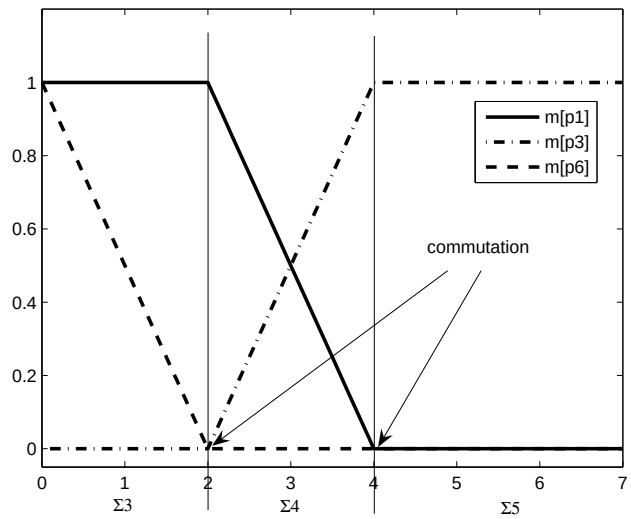


Fig. 4. Evolution of contPN in Fig. 2 with $\boldsymbol{\lambda} = [1, 2, 1, 1, 0.5]^T$ under finite server semantics.

At $\boldsymbol{m_0}$, the input places of $t_1$ and $t_4$ are marked, therefore $t_1$ and $t_4$ are strongly enabled and $f_1 = f_4 = 1$. The other transitions are weakly enabled and their flow depends on the input flows to the empty input places. For $t_2$, the input flow to $p_3$ (the only empty input place) is 1, hence $f_2 = \min\{\lambda_2, 1\} = 1$. Transition $t_3$ will work at the maximum speed because the input flow in $p_4$ is $f_2 = 1$, equal to $\lambda_1$, its maximal firing speed. For $t_5$, the input flow to $p_5$ is 1, then its flow is limited by its maximal firing speed that is $0.5$. According with these flow equations, the evolution of the system will be governed by a linear system, $\Sigma_3$, until $\tau = 2$, when $m_6$ and $m_2$ become empty. At this time, the marking is $[1, 0, 0, 0, 1, 0]$. Now, $t_1$ and $t_5$ are strongly enabled, therefore $f_1 = 1$ and $f_5 = 0.5$. The weakly enabled transitions $t_2$ and $t_4$ are in conflict and a resolution policy should be specified. Assume, for example, that the flow of $t_2$ is equal to the flow of $t_4$. Moreover, the output flows of all the empty places are upper bounded by the input flows. The solution is $f_2 = f_3 = f_4 = 0.5$. So, the system of equations that defines the evolution after $\tau = 2$ is a new linear system $\Sigma_4$.

At $\tau = 4$, $p_4$ is emptied and a new flow computation has to be done. The current marking is $[0, 0, 1, 0, 1, 0]$. The only strongly enabled transition is $t_5$, hence $f_5 = 1$. Solving the flows, $f_1 = f_2 = f_3 = f_4 = 0.5$ is obtained. These values correspond to a steady state marking ($\dot{\boldsymbol{m}}(\tau) = 0$).

Clearly, the evolution of a contPN system is quite different under both semantics: different transitory regimes and steady-state markings are obtained.

### B. Comparison of server semantics for mono-T-semiflow reducible nets

In this subsection we study both continuous semantics to see which one approximates better the steady state throughput of the discrete net. In [12] a branch and bound algorithm is used to compute upper bounds of the steady state throughput for continuous systems under infinite server semantics, each node corresponding to a LPP. In the case of mono-T-semiflow reducible nets, the bounds can be computed solving only one simple LPP. The linear programming problem is the "continuous version" of the bounds obtained in [6] for discrete nets[1].

Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m}_0 \rangle$ be a mono-T-semiflow reducible net and $\gamma_i$ the solution of the linear programming problem:

$$\gamma_i = \max \{ \boldsymbol{y} \cdot \boldsymbol{PD}_i | \boldsymbol{y} \cdot \boldsymbol{C} = 0, \boldsymbol{y} \cdot \boldsymbol{m_0} = 1, \boldsymbol{y} \geq 0 \} \tag{4}$$

where $\boldsymbol{PD}(p) = \max_{t \in p^\bullet} \frac{\boldsymbol{Pre}(p,t) \cdot \boldsymbol{v}^{(i)}(t)}{\boldsymbol{\lambda}(t)}$ and $\boldsymbol{v}^{(i)}$ is the visit ratio vector normalized for transition $t_i$.

According to [12] the flow of the continuous system under infinite server semantics, $\boldsymbol{f}$, verifies $\boldsymbol{f} \leq \frac{1}{\gamma_i} \boldsymbol{v}^{(i)}$. Moreover, this bound is reached (i.e. $\boldsymbol{f} = \frac{1}{\gamma_i} \boldsymbol{v}^{(i)}$) iff the steady state P-configuration contains the support of a P-semiflow [12].

On the other hand, in [6] it is given an upper bound of the throughput in steady state of the stochastic (discrete) Petri net: $\chi \leq \frac{1}{\gamma_i} \cdot \boldsymbol{v}^{(i)}$, where $\gamma_i$ and $\boldsymbol{v}^{(i)}$ are the same as in continuous case. This bound is valid for arbitrary probability distribution functions of service (including deterministic). Hence, if the steady state P-configuration contains the support of a P-semiflow, the throughput of the continuous net under infinite server semantics is an upper bound of the throughput of the discrete net system.

We will now prove that the throughput of continuous net systems under finite server semantics is greater than or equal to the throughput under infinite server semantics when the servers are made explicit, by adding the corresponding place self-loops (otherwise the comparison is inappropriate). Also, we assume that the conflict resolution in the case of finite server semantics is done according to the same visit ratio vector, as in discrete and infinite server semantics.

The comparison is done under the liveness hypothesis of the untimed contPN system. Liveness is used to ensure that at any time instant the continuous system under finite server semantics has at least one strongly-enabled transition. Otherwise, according to the flow definition (1) every transition has one empty input place and the net is not live as untimed. Liveness analysis of autonomous and timed mono-T-semiflow reducible nets is studied in [13]. It is proved that deadlock freeness is equivalent to liveness

[1] In [6] the mono-T-semiflow reducible nets are called Free Related T-semiflows nets (FRT)

for this subclass of models. Also a necessary condition for the existence of a marking that makes the system lim-live is given, which can be checked in polynomial time: every transition has at least one place that is not input of any other transition.

***Proposition** 3.1:* Let $\langle \mathcal{N}, \boldsymbol{m_0} \rangle$ be a live mono-T-semiflow reducible contPN system. For any $\boldsymbol{\lambda}$, the flow in steady state under finite server semantics is greater than or equal to the flow under infinite server semantics.

*Proof:* The net is mono-T-semiflow reducible so the throughput in steady state will be proportional to the visit ratio vector for both finite and infinite server semantics ($\boldsymbol{f_F} = \alpha_F \cdot \boldsymbol{v}^{(i)}$, $\boldsymbol{f_\infty} = \alpha_\infty \cdot \boldsymbol{v}^{(i)}$). Under finite server semantics at least one transition will be strongly enabled in steady state (otherwise the net is not live). Let $t_i$ be one of those transitions, and assume we normalize the visit ratio with respect to it. Then $\alpha_F = n_i \cdot \boldsymbol{\lambda}[t_i]$ where $n_i$ is the number of servers for transition $t_i$. Under infinite servers, if $\boldsymbol{m}$ is the steady state marking, it verifies that $\alpha_\infty = \min\limits_{p \in {}^\bullet t_i} \frac{\boldsymbol{m}[p]}{\boldsymbol{Pre}[p,t_i]} \cdot \boldsymbol{\lambda}[t_i] \leq \min\limits_{p \in {}^\bullet t_i} \boldsymbol{m}[p] \cdot \boldsymbol{\lambda}[t_i]$ ($\boldsymbol{Pre}[p,t_i] \in \mathbb{N}_{>0}$). Since the place that models the servers of $t_i$ is input of the transition, $\min\limits_{p \in {}^\bullet t_i} \boldsymbol{m}[p] \leq n_i$ and so $\alpha_\infty \cdot \boldsymbol{v}^{(i)} = \boldsymbol{f_\infty}[t_i] \leq \boldsymbol{f_F}[t_i] = \alpha_F \cdot \boldsymbol{v}^{(i)}$. □

Therefore:

***Theorem** 3.2:* Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a live mono-T-semiflow reducible Petri net system with every *possible equilibrium P-configuration* containing the support of a P-semiflow. For any probability distribution function for the firing of the transitions, the steady state throughput of the discrete model is better approximated by the continuous relaxation with infinite server semantics than with finite server semantics.

*Proof:* According to Proposition 5 in [12], the steady state throughput of the discrete net system is upper bounded by the steady-state throughput of the continuous one under infinite server semantics. Using Prop. 3.1, the steady state throughput of the continuous net system with finite server semantics is greater or equal than the one with infinite server semantics. □

***Example** 3.3:* Let us consider again the queuing network presented in Fig. 1. We have computed simulations for the discrete stochastic model (exponential distribution for servers) and the continuous model under infinite and finite (single-server) server semantics, using $\boldsymbol{\lambda} = \mathbf{1}$ (the servers are not represented in the figure to simplify it). In this net, every P-configuration contains the support of a P-semiflow (this will be proved in Example 4.12), so infinite server semantics will fit better. Simulating the model and measuring the flow of transition $t_1$, the following results are obtained: the throughput is $0.1337$ for the discrete model, $0.1667$ for the continuous model under infinite server semantics, and $0.3333$ for the continuous model under finite server semantics (i.e. two times bigger). ■

In general, proving that every P-configuration contains the support of a P-semiflow may be computa-

tionally expensive since the number of P-configurations may be large. However, there are net subclasses for which it is immediate. Take for example EQ nets (if ${}^\bullet t_i \cap {}^\bullet t_j \neq \emptyset$, $\boldsymbol{Pre}[\cdot, t_i] = \boldsymbol{Pre}[\cdot, t_j]$). For this class, every P-configuration contains a P-semiflow iff it is conservative, consistent and the rank of the token flow matrix equals to the number of equal conflicts ($|SEQS|$) minus one [20] ($|SEQS|$ is defined as the number of equivalence classes defined by the continuous equal conflict relation with $k = 1$). Hence it can be verified in polynomial time. Moreover, if the initial marking is such that every P-semiflow is marked, the net is live as continuous [16].

***Corollary** 3.4:* Let $\mathcal{N}$ be an EQ net system that is consistent, conservative, $rank(\boldsymbol{C}) = |SEQS| - 1$ and $\boldsymbol{m_0}$ such that every P-semiflow is marked. In steady state, the continuous system $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ with infinite server semantics provides a throughput closer to the throughput of the discrete system than finite server semantics.

### *C. Discussion*

Resources (and in particular servers) are usually shared among different operations. This leads to intuitively assert that finite server semantics approximation may lead to quite optimistic results. However, it may happen that it provides a throughput closer to the throughput of the discrete system.

***Example** 3.5:* Let us consider the net system in Fig. 5 with $\boldsymbol{m_0} = [15, 1, 1, 0]$. Assume that each transition has two dedicated servers and the speed of servers are $2, 1, 1$. In steady state the throughput of all the transitions will be the same (the incidence matrix has only one minimal right annuller, $\boldsymbol{x} = [1, 1, 1]^T$, and the flow in steady state must be a T-semiflow). As discrete, the steady state throughput is $1.55$ under exponential assumption for all transitions. If this model is seen as continuous with finite server semantics the maximum firing speed $\boldsymbol{\lambda} = 2 \cdot [2, 1, 1] = [4, 2, 2]$ (because each transition has 2 servers), and the steady state flow is equal to 2. Under infinite server semantics $\boldsymbol{\lambda} = [2, 1, 1]$ and the steady state flow is 0.75. Clearly, finite server semantics provides a throughput which is closer to the throughput of the discrete system in steady state. However, we will see later that $P - \mathcal{C}_3 = \{p_2, p_3, p_4\}$ is an equilibrium P-configuration that does not contain any P-semiflow, and Th. 3.2 does not apply.

This example also shows that the flow of the continuous model under infinite server semantics is not necessarily an upper bound of the throughput as discrete, what may came as a surprise: fluidification does not improve performance. ■

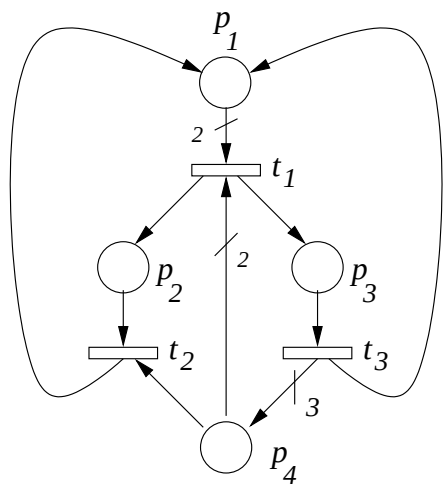


Fig. 5. Mono-T-semiflow net ($\boldsymbol{x} = [1, 1, 1]^T$) used in Example 3.5.

## IV. Monotonicity under infinite server semantics

### A. Monotonicity results

In this section we will focus our attention on the timed contPN system under infinite server semantics and study some properties related to monotonicity. From the flow definition in (2) it is easy to observe that if the vector $\boldsymbol{\lambda}$ is multiplied by a constant $k > 0$ then at any reachable marking the flow will be also multiplied by $k$. Analogously, when the initial marking is multiplied by $k$, the system will be $k$ times faster. But, what happens if only some components of $\boldsymbol{\lambda}$ or only some components of $\boldsymbol{m}_0$ are increased? In general, as happens for discrete nets, increasing the rate of a transition or the initial marking of a place may lead to a slower system (see examples in Subsection 2.2 in [12]). This unexpected behavior is usually not desirable. For example, replacing a machine by a faster one or adding new machines in a production system should not decrease the throughput. In this section we will see that mono-T-semiflow reducible nets, under quite general conditions, have the a priori expected monotonicity property.

***Theorem** 4.1:* Let $\langle \mathcal{N}, \boldsymbol{\lambda_1}, \boldsymbol{m_0} \rangle$ and $\langle \mathcal{N}, \boldsymbol{\lambda_2}, \boldsymbol{m_0} \rangle$ be mono-T-semiflow reducible contPN systems with $\boldsymbol{\lambda_1} \leq \boldsymbol{\lambda_2}$, but visit ratio preserving. If the systems reach a steady state and both steady state P-configurations contain the support of a P-semiflow then the steady state flows verify $\boldsymbol{f_1} \leq \boldsymbol{f_2}$.

*Proof:* For $i = 1, 2$, let $\boldsymbol{m_i}$ be the steady state marking of $\langle \mathcal{N}, \boldsymbol{\lambda_i}, \boldsymbol{m_0} \rangle$, and $\boldsymbol{y_i}$ a P-semiflow whose support is contained in one P-configuration defined by $\boldsymbol{m_i}$. Obviously, both $\boldsymbol{m_1}$ and $\boldsymbol{m_2}$ can be reached in the autonomous net system, thus $\boldsymbol{y_i} \cdot \boldsymbol{m_1} = \boldsymbol{y_i} \cdot \boldsymbol{m_2}$.

Let us focus on $\boldsymbol{f_2}$ and $\boldsymbol{y_2}$. Every place $p_{j2} \in ||\boldsymbol{y_2}||$ restricts the flow of at least one of its one output transitions, denoted by $t_{j2}$, i.e.:

$$\boldsymbol{f_2}[t_{j2}] = \boldsymbol{\lambda_2}[t_{j2}] \cdot \frac{\boldsymbol{m_2}[p_{j2}]}{\boldsymbol{Pre}[p_{j2}, t_{j2}]} \tag{5}$$

Using the P-semiflow $\boldsymbol{y_2}$, we can write the token conservation law for $\boldsymbol{m_1}$ and $\boldsymbol{m_2}$, and taking $\boldsymbol{m_2}$ from the previous equation:

$$\sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \frac{\boldsymbol{Pre}[p_{j2}, t_{j_2}] \cdot \boldsymbol{f_2}[t_{j2}]}{\boldsymbol{\lambda_2}[t_{j2}]} = \sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \boldsymbol{m_1}[p_{j2}] \tag{6}$$

Now, $\boldsymbol{m_1}[p_{j2}]$ can be replaced using $\boldsymbol{f_1}$ because

$$\boldsymbol{f_1}[t_{j2}] = \boldsymbol{\lambda_1}[t_{j2}] \cdot enab(t_{j2}, \boldsymbol{m_1}) \leq \boldsymbol{\lambda_1}[t_{j2}] \cdot \frac{\boldsymbol{m_1}[p_{j2}]}{\boldsymbol{Pre}[p_{j2}, t_{j2}]}. \tag{7}$$

Moreover, $\boldsymbol{\lambda_1} \leq \boldsymbol{\lambda_2}$, so:

$$\begin{aligned} &\sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \boldsymbol{m_1}[p_{j2}] \geq \\ \geq &\sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \frac{\boldsymbol{Pre}[p_{j2}, t_{j_2}] \cdot \boldsymbol{f_1}[t_{j2}]}{\boldsymbol{\lambda_1}[t_{j2}]} \geq \\ \geq &\sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \frac{\boldsymbol{Pre}[p_{j2}, t_{j_2}] \cdot \boldsymbol{f_1}[t_{j2}]}{\boldsymbol{\lambda_2}[t_{j2}]} \end{aligned} \tag{8}$$

The net is mono-T-semiflow reducible, and $\boldsymbol{\lambda_1}$ and $\boldsymbol{\lambda_2}$ keep the proportion of flows in continuous equal conflicts. Hence, both visit ratios will be the same, $\boldsymbol{v}^{(1)} > 0$. Let $\boldsymbol{f_1} = k_1 \cdot \boldsymbol{v}^{(1)}$ and $\boldsymbol{f_2} = k_2 \cdot \boldsymbol{v}^{(1)}$. Therefore, merging (7) and (8):

$$\sum_{p_{j2} \in ||\boldsymbol{y_2}||} \boldsymbol{y_2}[p_{j2}] \cdot \frac{\boldsymbol{Pre}[p_{j2}, t_{j_2}] \cdot \boldsymbol{v}^{(1)}[t_{j2}]}{\boldsymbol{\lambda_2}[t_{j2}]} \cdot (k_2 - k_1) \geq 0 \tag{9}$$

And so $k_2 \geq k_1$. □

This result an be extended to sets of rates.

***Definition** 4.2:* Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a timed contPN system and let $\mathcal{L} \subseteq \mathbb{R}_{>0}^{|T|}$. The system $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ is said to be *monotone for $t_j$ in steady state w.r.t. $\boldsymbol{\lambda} \in \mathcal{L}$* if $\forall \boldsymbol{\lambda_1}, \boldsymbol{\lambda_2} \in \mathcal{L}$ with $\boldsymbol{\lambda_1} \leq \boldsymbol{\lambda_2}$, if these systems have steady states, the associated steady state flows verify $\boldsymbol{f_1}[t_j] \leq \boldsymbol{f_2}[t_j]$.

In the case of mono-T-semiflow reducible net systems, since the steady state flow is proportional to the visit ratio vector defined by the net structure and routing (speeds at CEQs), if a system is monotone for a given $t_j$, it is monotone for $\forall t \in T$. Therefore, it can be said that the net system is monotone (i.e. it is monotone for all transitions).

***Theorem** 4.3:* Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a timed mono-T-semiflow reducible contPN system under infinite server semantics and $\mathcal{L} \subseteq \mathbb{R}_{>0}^{|T|}$ visit ratio preserving. If for all $\boldsymbol{\lambda} \in \mathcal{L}$ the *possible equilibrium P-configurations* of $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ contain the support of a P-semiflow then the timed system is monotone in steady state with respect to $\boldsymbol{\lambda} \in \mathcal{L}$.

*Proof:* Applying to all $\boldsymbol{\lambda_1}, \boldsymbol{\lambda_2} \in \mathcal{L}$ Th. 4.1, the conclusion is verified. □

***Example** 4.4:* Let us consider the contPN in Fig. 5. This net is mono-T-semiflow with $\boldsymbol{x} = [1, 1, 1]$ and has 4 P-configurations: $P - \mathcal{C}_1 = \{p_1, p_2, p_3\}$, $P - \mathcal{C}_2 = \{p_1, p_4, p_3\}$, $P - \mathcal{C}_3 = \{p_4, p_2, p_3\}$ and $P - \mathcal{C}_4 = \{p_4, p_3\}$, and two minimal P-semiflows: $\boldsymbol{y_1} = p_1 + p_2 + p_3$ and $\boldsymbol{y_2} = p_1 + 4 \cdot p_3 + p_4$. Therefore there are two P-configurations that contain a P-semiflow, $P - \mathcal{C}_1$ and $P - \mathcal{C}_2$, one P-configuration that contains the support of a P-flow, $P - \mathcal{C}_3$ ($\boldsymbol{y_3} = \boldsymbol{y_1} - \boldsymbol{y_2}$) but no P-semiflow, and one P-configuration that does not contain the support of any P-flow, $P - \mathcal{C}_4$.

*1)* Let $\boldsymbol{m_0} = [1, 1, 0, 15]$. We will see that for all $\boldsymbol{\lambda} \in \mathbb{R}_{>0}^{|T|}$ the only possible equilibrium P-configurations are $P - \mathcal{C}_1$ and $P - \mathcal{C}_2$. Both contain P-semiflows, so using Theorem 4.3 the timed system is monotone with respect to $\boldsymbol{\lambda} \in \mathbb{R}_{>0}^{|T|}$ (for this initial marking).

First, let us observe that $P - \mathcal{C}_3$ is not a possible equilibrium P-configuration. To be an equilibrium P-configuration, the following system of equations should be satisfied:

$$\begin{cases} m_1 + m_2 + m_3 = 2 & \text{(a) first P-semiflow} \\ m_1 + 4 \cdot m_3 + m_4 = 16 & \text{(b) second P-semiflow} \\ m_1 \geq m_4 & \text{(c) } f_1 \text{ restricted by } p_4 \\ m_4 \geq m_2 & \text{(d) } f_2 \text{ restricted by } p_2 \end{cases} \tag{10}$$

where $m_1, m_2, m_3, m_4$ and $f_1, f_2, f_3$ is a possible steady state in $P - \mathcal{C}_3$. From (10.a) $\Longrightarrow m_1 \leq 2$ and $m_3 \leq 2$. Moreover, (10.c) implies $m_4 \leq m_1 \leq 2$. Therefore, $m_1 + 4 \cdot m_3 + m_4 \leq 2 + 8 + 2 = 12$ and so (10.b) cannot be satisfied.

Assume now that $P - \mathcal{C}_4$ is the steady state P-configuration, therefore the following system should be satisfied:

$$\begin{cases} m_1 + m_2 + m_3 = 2 & \text{(a) first P-semiflow} \\ m_1 + 4 \cdot m_3 + m_4 = 16 & \text{(b) second P-semiflow} \\ m_1 \geq m_4 & \text{(c) } f_1 \text{ restricted by } p_4 \\ m_2 \geq m_4 & \text{(d) } f_2 \text{ restricted by } p_4 \end{cases} \tag{11}$$

According to (11.a) and (11.b), $m_4 \geq 8$ and $m_2 \leq 2$ (because of (11.a)); thus (11.d) is not true. Therefore (11) has no solution and $P - \mathcal{C}_4$ cannot be an equilibrium P-configuration.

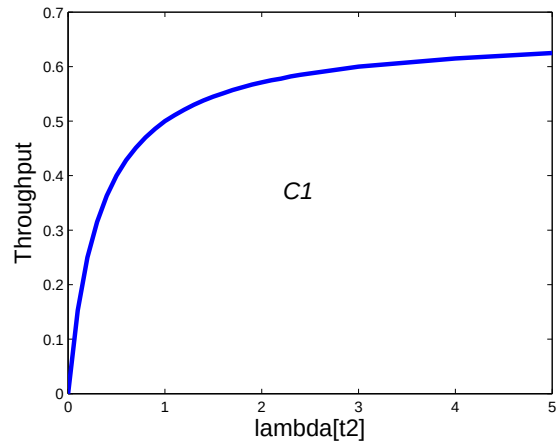


Fig. 6. Throughput of the contPN system in fig. 5 for $\boldsymbol{m_0} = [1, 1, 0, 15]^T$ and different values of $\lambda_2$ ($\lambda_1 = \lambda_3 = 1$).

Thus for $\boldsymbol{m_0} = [1, 1, 0, 15]^T$ the only possible equilibrium P-configurations, for any $\boldsymbol{\lambda} \in \mathbb{R}^{|T|}_{>0}$, are $P - \mathcal{C}_1$ and $P - \mathcal{C}_2$. Both contain the support of a P-semiflow, thus the timed system is monotone w.r.t. $\boldsymbol{\lambda} \in \mathbb{R}^{|T|}_{>0}$ (Theorem 4.3). In Figure 6 it is shown the non decreasing monotonicity of the throughput w.r.t. $\lambda_2$.

*2)* Let $\boldsymbol{m_0} = [15, 1, 1, 0]$. For this marking, $P - \mathcal{C}_2$, $P - \mathcal{C}_3$, $P - \mathcal{C}_4$ can be equilibrium configurations if appropriate values for $\boldsymbol{\lambda}$ are chosen. For example: $\boldsymbol{m_2} = [1, 13, 3, 6]^T$ corresponding to $P - \mathcal{C}_2$ is an equilibrium marking for $\boldsymbol{\lambda_2} = [6, 1, 2]^T$ with $\boldsymbol{f_2} = \mathbf{6}$; $\boldsymbol{m_3} = [13.67, 3, 0.33, 4]^T$ corresponding to $P - \mathcal{C}_3$ is an equilibrium marking for $\boldsymbol{\lambda_3} = [3, 2, 18]^T$ with $\boldsymbol{f_3} = \mathbf{1}$; $\boldsymbol{m_4} = [10.5, 4.5, 2, 0.5]^T$ corresponding to $P - \mathcal{C}_4$ is an equilibrium marking for $\boldsymbol{\lambda_4} = [8, 4, 1]^T$ with $\boldsymbol{f_4} = \mathbf{2}$. Since some possible equilibrium P-configurations do not contain the support of a P-semiflow, monotony cannot be guaranteed.

In Figure 7 it is sketched the evolution of the throughput of the net system for $\boldsymbol{m_0} = [15, 1, 1, 0]^T$ and $\boldsymbol{\lambda} = [1, \lambda_2, 1]^T$ with $0 < \lambda_2 \leq 5$. It can be seen that it is not monotonic. Even a discontinuity exists at $\lambda_2 = 0.5$. When $0 < \lambda_2 < 0.5$, the equilibrium P-configuration is $P - \mathcal{C}_2$ and the throughput is increasing with $\lambda_2$. For $\lambda_2 \geq 0.5$ the steady state P-configuration is $P - \mathcal{C}_3$ that contains the support of a P-flow (but not of a P-semiflow) and the steady state throughput is decreasing (the conditions of Theorem 4.3 are not satisfied). Therefore, for $\boldsymbol{m_0} = [15, 1, 1, 0]^T$ the system is not monotone in steady state w.r.t. $\boldsymbol{\lambda} \in \mathbb{R}^{|T|}_{>0}$.

3) Let us now study the monotonicity of the same contPN with $\boldsymbol{m_0} = [15, 1, 1, 0]^T$ w.r.t $\boldsymbol{\lambda} \in \mathcal{L}_1$ where $\mathcal{L}_1 = \{[\lambda_1, \lambda_2, \lambda_3]^T | \lambda_1 = \lambda_3 = 1, 0 < \lambda_2 < 0.5\}$.

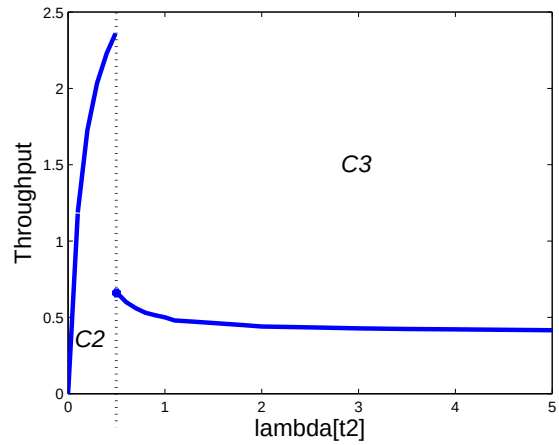


Fig. 7. Throughput of the contPN system in fig. 5 for $\boldsymbol{m_0} = [15, 1, 1, 0]^T$ and different values of $\lambda_2$.

First, $P - \mathcal{C}_4$ cannot be an equilibrium P-configuration. If it were, $p_4$ restricts the flows of $t_1$ and $t_2$ in steady state, and so since the steady state flow verifies $f_1 = f_2 = f_3$, $\lambda_2 \cdot m_4 = \lambda_1 \cdot \frac{m_4}{2} = \frac{m_4}{2}$. Because we are assuming $\lambda_2 < 0.5$, $m_4 = 0$ and $f_1 = f_2 = f_3 = 0$, i.e., the system is in a deadlock. Notice that $f_3$ is defined by $m_3$, so $m_3 = 0$. Using the P-semiflows: $m_1 + m_2 = 17$ and $m_1 = 19$ which cannot be satisfied.

If $P - \mathcal{C}_3$ were the equilibrium P-configuration, since the steady state flow verifies $f_1 = f_2 = f_3$, then $\lambda_1 \cdot \frac{m_4}{2} = \lambda_2 \cdot m_2 = \lambda_3 \cdot m_3$. But $\lambda_1 = \lambda_3 = 1$, and so $\frac{m_4}{2} = \lambda_2 \cdot m_2 = m_3$. Hence, the following system of equations should have a solution:

$$\begin{cases} m_1 + m_2 + m_3 = 17 & \text{(a) first P-semiflow} \\ m_1 + 4 \cdot m_3 + m_4 = 19 & \text{(b) second P-semiflow} \\ m_1 \geq m_4 & \text{(c) } f_1 \text{ restricted by } p_4 \\ m_4 \geq m_2 & \text{(d) } f_2 \text{ restricted by } p_2 \\ \frac{m_4}{2} = \lambda_2 \cdot m_2 = m_3 & \text{(e) steady state flows} \end{cases} \tag{12}$$

Using (12.e), $m_4 = 2\lambda_2 \cdot m_2$ and considering (12.d): $2\lambda_2 \cdot m_2 \geq m_2$. But $m_2 > 0$ because the system cannot deadlock (see the reasoning for $\mathcal{C}_4$) therefore $2 \cdot \lambda_2 \geq 1$ that cannot be satisfied since $0 < \lambda_2 < 0.5$. Hence, the only possible equilibrium P-configurations are $P - \mathcal{C}_1$ and $P - \mathcal{C}_2$ that contain the support of a P-semiflow. Therefore the time contPN system is monotone w.r.t. $\boldsymbol{\lambda} \in \mathcal{L}_1$. ■

Let us now consider monotonicity w.r.t. the initial marking.

***Theorem** 4.5:* Let $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ be a mono-T-semiflow reducible contPN under infinite server semantics, and let $\boldsymbol{m_1} \leq \boldsymbol{m_2}$. If for $i = 1, 2$ $\langle \mathcal{N}, \lambda, \boldsymbol{m_i} \rangle$ reaches a steady state and the steady state P-configuration

contains the support of a P-semiflow, then the steady state flows verify $\boldsymbol{f_1} \leq \boldsymbol{f_2}$.

*Proof:* According to Prop. 5 of [12], $\boldsymbol{f_i} = \frac{1}{\gamma_i} \cdot \boldsymbol{v}^{(1)}$ with $\gamma_i = \max\{\boldsymbol{y} \cdot \boldsymbol{PD} | \boldsymbol{y} \cdot \boldsymbol{C} = \boldsymbol{0}, \boldsymbol{y} \cdot \boldsymbol{m_i} = \boldsymbol{1}, \boldsymbol{y} \geq \boldsymbol{0}\} = \max\left\{\frac{1}{\boldsymbol{y} \cdot \boldsymbol{m_i}} \cdot \boldsymbol{y} \cdot \boldsymbol{PD} | \boldsymbol{y} \cdot \boldsymbol{C} = \boldsymbol{0}, \boldsymbol{y} \geq \boldsymbol{0}\right\}$. Let us assume $\boldsymbol{m_{10}} \leq \boldsymbol{m_{20}}$. Then, for every P-semiflow $\boldsymbol{y}$, $\boldsymbol{y} \cdot \boldsymbol{m_1} \leq \boldsymbol{y} \cdot \boldsymbol{m_2}$ and so, $\gamma_1 \geq \gamma_2$. Therefore, $\boldsymbol{f_1} \leq \boldsymbol{f_2}$. □

Monotonicity w.r.t. the initial marking can be studied also on a region as done in the case of the monotonicity w.r.t. $\boldsymbol{\lambda}$.

***Definition** 4.6:* Let $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$ be a timed contPN and let $\mathcal{M} \subseteq \mathbb{R}_{>0}^{|P|}$. The system $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ is said to be *monotone for $t_j$ in steady state w.r.t.* $\boldsymbol{m_0} \in \mathcal{M}$ if for all $\boldsymbol{m_1}, \boldsymbol{m_2} \in \mathcal{M}$ with $\boldsymbol{m_1} \leq \boldsymbol{m_2}$, these systems have steady states and the associated steady state flows verify $\boldsymbol{f_1}[t_j] \leq \boldsymbol{f_2}[t_j]$.

As in the case of the monotony w.r.t. $\boldsymbol{\lambda}$, Theorem 4.7 can be used to derive the following sufficient condition for monotonicity of a mono-T-semiflow net system w.r.t. the initial marking.

***Theorem** 4.7:* Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a timed mono-T-semiflow reducible contPN system under infinite server semantics. If for all $\boldsymbol{m_0} \in \mathcal{M}$ the *possible equilibrium P-configurations* contain the support of a P-semiflow, then $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ is monotone in steady state w.r.t. $\boldsymbol{m_0} \in \mathcal{M}$.

*Proof:* Applying to all $\boldsymbol{m_1}, \boldsymbol{m_2} \in \mathcal{M}$ Th. 4.5, the conclusion is verified. □

***Example** 4.8:* Let us consider the contPN in figure 5. Assume $\boldsymbol{\lambda} = [1, 1, 1]^T$.

1) Let us study monotonicity w.r.t. $\boldsymbol{m_0} \in \mathcal{M}_1 = \{\boldsymbol{m} | m_1 = 0, m_2 = m_3 = 1, m_4 > 0\}$.

$P - \mathcal{C}_4$ cannot be an equilibrium P-configuration. If it were, in steady state, $p_4$ limits the flow of $t_1$ and $t_2$, but taking into account the T-semiflow: $f_1 = f_2$, thus $\frac{m_4}{2} = m_4$ (because $\lambda_1 = \lambda_2 = 1$), and the only solution is $m_4 = 0$. Therefore, from the T-semiflow, $m_3 = 0$. Writing the P-semiflows we have: $m_1 + m_2 = 2$ and $m_1 = 4 + z$, with $z$ the initial marking of $p_4$. Since $z \geq 0$ these equations cannot be satisfied. Therefore $P - \mathcal{C}_4$ cannot be an equilibrium P-configuration.

For $\mathcal{C}_3$, the following system of equations should have a solution:

$$\begin{cases} m_1 + m_2 + m_3 = 2 & \text{(a) first P-semiflow} \\ m_1 + 4m_3 + m_4 = 4 + z & \text{(b) second P-semiflow} \\ \frac{m_4}{2} = m_2 = m_3 & \text{(c) steady state flow} \\ m_1 \geq m_4 & \text{(d) } f_1 \text{ restricted by } p_4 \\ m_4 \geq m_2 & \text{(e) } f_2 \text{ restricted by } p_2 \end{cases} \tag{13}$$

where $z$ is the initial marking of $p_4$. From (13.b) - (13.a): $3 \cdot m_3 - m_2 + m_4 = z + 2$. Replacing $m_3$ and $m_4$ using (13.c), $3 \cdot m_2 - m_2 + 2 \cdot m_2 = z + 2$, thus $m_2 = \frac{z+2}{4}$. Hence, any solution should be of the form: $[m_1, \frac{z+2}{4}, \frac{z+2}{4}, \frac{z+2}{2}]^T$ with $m_1 \geq \frac{z+2}{2}$ (13.d). But (13.a) implies: $m_1 = 2 - \frac{z+2}{4} - \frac{z+2}{4} = \frac{-z+2}{2}$

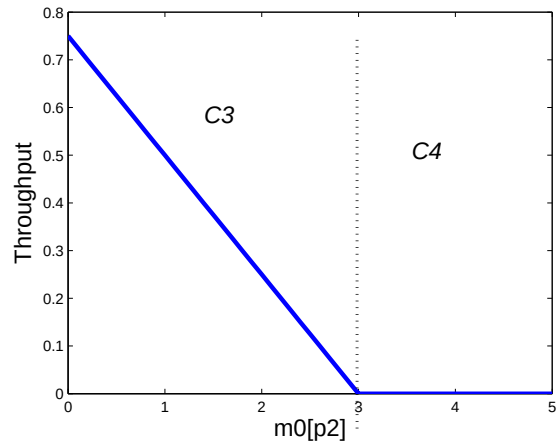


Fig. 8. Throughput of the contPN system in fig. 5 for $\boldsymbol{m_0} = [15, z, 1, 0]^T$ and $\boldsymbol{\lambda} = [1, 1, 1]^T$.

and combining with (13.d) results $\frac{-z+2}{2} \geq \frac{z+2}{2} \rightarrow 2 \cdot z \leq 0 \rightarrow z \leq 0$ which is impossible because $z$ is the initial marking of $p_4$.

Thus neither $P - \mathcal{C}_4$ nor $P - \mathcal{C}_3$ can be equilibrium P-configurations. Using Theorem 4.7 the timed net system is monotone w.r.t. the initial marking $\boldsymbol{m_0}$ in $\mathcal{M}_1$.

2) Let us consider $\mathcal{M}_2 = \{[m_1, m_2, m_3, m_4]^T | m_1 = 15, m_3 = 1, m_4 = 0, m_2 > 0\}$. $P - \mathcal{C}_3$ and $P - \mathcal{C}_4$ can be equilibrium P-configurations and monotonicity can be lost. Indeed, simulating for $\boldsymbol{m_0}[p_2] \in [0, 5]$ (Figure 8), the throughput decreases, even deadlock is reached for $\boldsymbol{m_0}[p_2] \geq 3$. Hence the timed net system with $\boldsymbol{m_0} \in \mathcal{M}_2$ is not monotonic in steady state. ■

### *B. Some properties of non monotonicity*

As an immediate consequence of Theorems 4.3 and 4.7, if all P-configurations defined by $\mathcal{N}$ (i.e., independently of $\boldsymbol{m_0}$), contain a P-semiflow, then the underlying net system is monotone in steady state w.r.t. the set of $\boldsymbol{\lambda} \in \mathbb{R}_{>0}^{|T|}$ that impose the same routing (visiting ratio), and w.r.t. $\boldsymbol{m_0}$ in $\mathbb{R}_{>0}^{|P|}$. Moreover, we will prove that this P-semiflow condition when asked to all the P-configurations is in fact equivalent to an analogous P-flows condition (Corollary 4.11). Let us first consider the following lemma.

***Lemma** 4.9:* Let $\mathcal{N}$ be a consistent join free net (i.e., $\forall t \in T$, $|{}^\bullet t| \leq 1$). For every P-flow $\boldsymbol{y}$ there exist a P-semiflow $\boldsymbol{y'}$ such that $||\boldsymbol{y'}|| \subseteq ||\boldsymbol{y}||$.

*Proof:* Dual of Theorem 9 of [19] (T-flows of CF nets). □

***Theorem** 4.10:* Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a mono-T-semiflow reducible contPN system under infinite server semantics. If there exists a set $\mathcal{L}$ of rates, $\mathcal{L}$ visit ratio preserving, or a set of initial markings $\mathcal{M}$, such

that $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ is not monotone in steady state w.r.t. $\boldsymbol{\lambda} \in \mathcal{L}$ or w.r.t. $\boldsymbol{m_0} \in \mathcal{M}$, then there exists a P-configuration that does not contain any P-flow.

*Proof:* If $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ is not monotone, applying Theorem 4.3 or Theorem 4.7, an equilibrium P-configuration exists that does not contain a P-semiflow. If this equilibrium P-configuration does not contain the support of any P-flow this is the requested P-configuration. Otherwise, assume that this equilibrium P-configuration contains one P-flow (or more). Let us consider the subnet $\mathcal{N}'$ defined by the set of all transitions together with the places that limit their flow in steady state, and let us call $\boldsymbol{C'}$ the token flow matrix of this subnet. Since the original net is mono-T-semiflow, it is consistent. The same T-semiflow of $\mathcal{N}$ is also a right annuller of $\boldsymbol{C'}$, hence $\mathcal{N}'$ is consistent.

If $\mathcal{N}'$ were a JF net, using Lemma 4.9 the support of a P-semiflow should be included in $\mathcal{N}'$, but that is impossible by assumption. Hence $\mathcal{N}'$ must have at least one join/synchronization. Let us call $t_k$ this transition. Let $p_i$ and $p_j$ be the input places of $t_k$ that belong to the P-configuration. Obviously, only one place restricts the flow of $t_k$. Let us assume $p_i$ to be this one.

If we consider now that $t_k$ is restricted by $p_j$ and the other transitions are restricted by the same places as before, we obtain a new P-configuration (possibly a non-equilibrium one) in which place $p_i$ has been removed. If this P-configuration contains a P-flow, repeat the reasoning. In the end this procedure will define a P-configuration that does not contain any P-semiflow (from hypothesis) or P-flow. □

***Corollary** 4.11:* Let $\langle \mathcal{N}, \boldsymbol{\lambda}, \boldsymbol{m_0} \rangle$ be a continuous mono-T-semiflow reducible net under infinite server semantics. If all the P-configurations contain the support of a P-flow, then the underlying net system is monotone in steady state w.r.t. $\boldsymbol{m_0} \in \mathcal{M}$, $\forall \mathcal{M} \subseteq \mathbb{R}_{>0}^{|P|}$, and w.r.t. $\boldsymbol{\lambda} \in \mathcal{L}$, $\forall \mathcal{L} \subseteq \mathbb{R}_{>0}^{|T|}$ visit ratio preserving.

***Example** 4.12:* Let us consider again the queuing network presented in Fig. 1. It has the following P-semiflows: $\boldsymbol{y_1} = p_2 + p_4$, $\boldsymbol{y_2} = p_7 + p_9$, $\boldsymbol{y_3} = p_8 + p_{10}$ and $\boldsymbol{y_4} = p_1 + p_2 + p_3 + p_5 + p_6 + p_7 + p_8 + p_{11} + p_{12}$. Since this net is EQ, all P-configurations contain the support of a P-semiflow [20]. Hence, both kinds of monotonicity properties hold. ■

## V. Checking structural monotonicity

This section deals with monotonicity results valid for family of $\boldsymbol{m}_0$, thus partially independent on the initial marking, based on the structural timed net $\langle \mathcal{N}, \boldsymbol{\lambda} \rangle$.

In Section IV, monotonicity of the throughput w.r.t. $\boldsymbol{\lambda}$ or $\boldsymbol{m}_0$ is proved for mono-T-semiflow reducible nets whose possible equilibrium P-configurations contain the support of a P-semiflow. Under the same hypothesis, plus liveness, in Section III-B it is proved that infinite server semantics provides a throughput closer to that of the discrete system in steady state than finite server semantics. Therefore, it is interesting

to know the P-configurations that do not contain the support of a P-semiflow, because if they are possible equilibrium P-configurations the previous two properties may not hold.

The idea is to use boolean equations to represent the conditions the places in a P-configuration have to fulfill. Let $\gamma_i$ be a boolean variable such that $\gamma_i = 1$ iff $p_i$ belongs to one P-configuration. Any P-semiflow will provide a boolean equation: the product of the boolean variables associated to the support of the P-semiflow should be 0 if the P-semiflow does not belong to the P-configuration. Moreover, if a transition is not a join ($|{}^\bullet t_i| = 1$) its input place must belong to all P-configurations (it is an essential cover), therefore the boolean variables associated to those places are 1. For every synchronization, we should have another boolean equation ensuring that at least one input place is taken so that the solution is (or contains) a P-configuration. The solutions of this system of boolean equations provide P-configurations that do not contain the support of a P-semiflow.

***Example** 5.1:* Let us consider the net in Fig. 5. This net has been used in Examples 4.4 and 4.8 where the P-configurations that do not contain the support of a P-semiflow were supposed to be known. These P-configurations can be computed, for example, using the above algorithm. This net has two P-semiflows: $\boldsymbol{y_1} = p_1 + p_2 + p_3$ and $\boldsymbol{y_2} = p_1 + 4 \cdot p_3 + p_4$, so the corresponding boolean equations are: $\gamma_1 \cdot \gamma_2 \cdot \gamma_3 = 0$ and $\gamma_1 \cdot \gamma_3 \cdot \gamma_4 = 0$. In order to cover transitions we need additional equations. For $t_1$: $\gamma_1 + \gamma_4 = 1$ ($m_1$ or $m_4$ limit the flow of $t_1$) and the same for $t_2$: $\gamma_2 + \gamma_4 = 1$ ($m_2$ or $m_4$ limit the flow of $t_2$). Clearly, $p_3$ being the only input place in $t_3$ is an essential cover, thus $\gamma_3 = 1$. The following system of boolean equations is obtained:

$$\begin{cases} \gamma_1 \cdot \gamma_2 \cdot \gamma_3 = 0 & \text{(a) First P-semiflow is not contained} \\ \gamma_1 \cdot \gamma_3 \cdot \gamma_4 = 0 & \text{(b) Second P-semiflow is not contained} \\ \gamma_1 + \gamma_4 = 1 & \text{(c) } t_1 \text{ is covered} \\ \gamma_2 + \gamma_4 = 1 & \text{(d) } t_2 \text{ is covered} \\ \gamma_3 = 1 & \text{(e) } t_3 \text{ is covered} \end{cases} \tag{14}$$

From (14.e), taking into account (14.a) and (14.b), $\gamma_1 \cdot (\gamma_2 + \gamma_4) = 0$. Because of (14.d): $\gamma_1 = 0$; thus $\gamma_4 = 1$ and $\gamma_2 = \oslash$. Summarizing, $\gamma_1 = 0$, $\gamma_2 = \oslash$, $\gamma_3 = 1$ and $\gamma_4 = 1$. Thus the net has two P-configurations ($\mathcal{C}_1 = \{p_3, p_4\}$, $\mathcal{C}_2 = \{p_2, p_3, p_4\}$) that do not contain the support of a P-semiflow. Depending on the initial marking and the firing rates these P-configurations can or cannot be equilibrium P-configurations and the monotonicity may be lost (see Example 4.4). ■

***Remark** 5.2:* It can be seen in (14) that for every transition with only one input place (there $|{}^\bullet t_3| = |\{p_3\}| = 1$) a boolean equation $\gamma_3 = 1$ is introduced. Before solving the equations, these variables can be removed reducing the number of variables. The interpretation in the PN is that $t_3$ and $p_3$ can be removed

in the model because $p_3$ belongs to all P-configurations. If all transitions with only one input place are removed, the net will contain only join transitions and the order of the system may be much lower.

According to Corollary 4.11, if all P-configurations contain the support of a P-flow, the timed net system is monotonic in steady state w.r.t. $\boldsymbol{m_0} \in \mathbb{R}_{>0}^{|P|}$ and w.r.t. $\boldsymbol{\lambda} \in \mathcal{L}$, $\forall \mathcal{L} \subseteq \mathbb{R}_{>0}^{|T|}$ visit ratio preserving.

A second algorithm can be used to check if all P-configurations contain the support of a P-flow. This second algorithm may fail, but this does not imply that the net system is not monotonic, since the net may include P-configurations not containing the support of a P-semiflow but which cannot be equilibrium P-configurations for a specific initial marking and transition rates (see Example 4.4). In fact, with this algorithm it is monotonicity for any subset of $\boldsymbol{\lambda}$ and $\boldsymbol{m_0}$ that is obtained.

The algorithm consists in: first, all P-configurations are computed (which is exponential) and then, each P-configuration is checked to see whether it contains the support of a P-flow. For the second step, let the token flow matrix of a P-configuration $P - \mathcal{C}_j$ be denoted by $\boldsymbol{C_j}$ ($\boldsymbol{C_j}$ is obtained from the token flow matrix of the original net by removing the rows not corresponding to $P - \mathcal{C}_j$). To check if $P - \mathcal{C}_j$ contains the support of a P-flow, is equivalent to check if $\exists \boldsymbol{y} \neq \boldsymbol{0}$ such that $\boldsymbol{y} \cdot \boldsymbol{C_j} = 0$ (polynomial time complexity).

Using this algorithm, we can check if all P-configurations contain the support of a P-flow. If the answer is negative, we can compute P-configurations not containing the support of a P-semiflow checking for each P-configuration $P - \mathcal{C}_j$ with the associated token flow matrix $\boldsymbol{C_j}$ if the following system has a solution: $\exists \boldsymbol{y} \geq 0$, $\boldsymbol{y} \neq 0$ such that $\boldsymbol{y} \cdot \boldsymbol{C_j} = 0$. If we prove that they cannot be equilibrium P-configurations, the system is monotone.

## VI. Case Study

The Petri net system in Fig. 9 represents an assembly line with kanban strategy (see [21]). The system has two stages connected by transition $t_{14}$. The first one is composed by three lines starting with $p_2$, $p_3$ and $p_4$, three machines $p_{23}$, $p_{24}$ and $p_{25}$ and three buffers $p_{26}$, $p_{27}$ and $p_{28}$. The second stage has two lines that share the resource $p_{18}$. The number of kanban cards are given by the markings of $p_2$, $p_3$, $p_4$, $p_{32}$.

The number of reachable markings for the initial marking in the figure is 209704. Having so many states, we want to approximate the model with its continuous approximation. Considering that the firing of every transition takes 1 t.u., which firing semantics will approximates better the throughput in steady state?

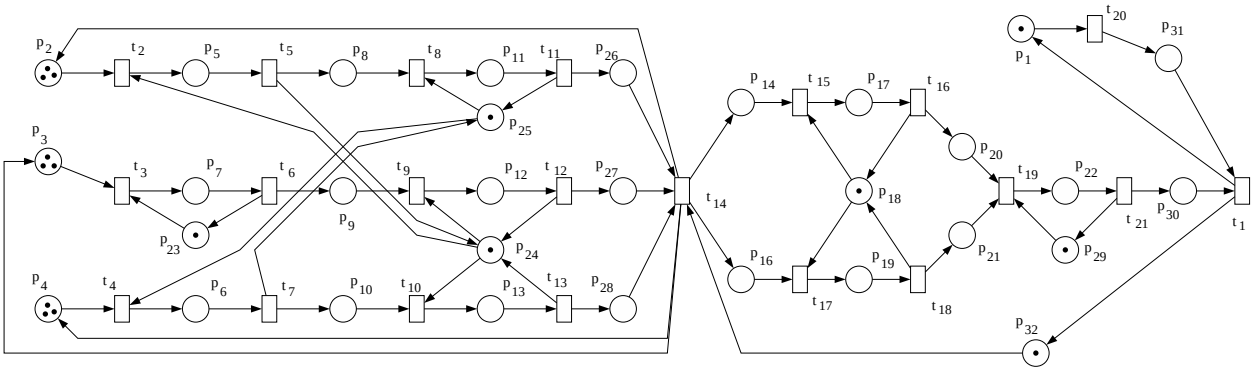


Fig. 9. An assembly line with kanban strategy. The net is mono-T-semiflow (There exists no equal conflict to be reduced).

First, it is easy to observe that every transition has at least one place with its marking not greater than one at every time moment, hence the discrete net has the same behavior under finite and infinite server semantics. Thus it is not necessary to make explicit the servers to compare the continuous approximations. This PN has 12 minimal P-semiflows covering all places so it is conservative and has one minimal T-semiflow covering all transitions so it is consistent, and mono-T-semiflow. Applying any algorithm from Section V we obtain that every P-configuration contains the support of a P-semiflow. Applying Theorem 3.2, the throughput in steady state of continuous model with infinite server semantics will be closer than finite server semantics to the throughput of the deterministic discrete net. This is obtained also by simulations. The throughput of the (discrete) timed PN is $0.125$, of the continuous net with infinite server semantics is $0.167$ while with finite server semantics is $1$.

Observe that the conditions of Theorem 4.3 are also satisfied. Therefore, increasing the speed of some transitions or the marking of some places, the throughput of the contPN with infinite server semantics will be greater or equal.

## VII. Conclusions

Mono-T-semiflow reducible contPN have been discussed in this paper. First, a comparison between finite and infinite firing semantics, the two most used for timed contPN systems is provided. A “good” firing semantics for timed continuous Petri nets should provide a time evolution “similar” to the discrete model. Being a relaxation, an identical result is practically impossible to obtain. In the actual level of knowledge, in general it is difficult to answer to this question, although, in practice, infinite server semantics is usually better than finite server semantics. In this paper we have proved that for mono-T-

semiflow reducible contPN systems with all the equilibrium P-configurations containing the support of a P-semiflow, the throughput in steady state with infinite server semantics is always closer to the throughput of the discrete net. Moreover, under the same conditions, monotonicity with respect to the firing rate $\boldsymbol{\lambda}$ and with respect to the initial marking $\boldsymbol{m_0}$ is obtained. Finally, two methods to check the conditions are presented.

## REFERENCES


[1] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. Wiley, 1995.

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

[3] M. Allam and H. Alla. Modeling and simulation of an electronic component manufacturing systems using hybrid Petri nets. *IEEE Trans. on Semiconductor Manufacturing*, 11(3):374–383, 1998.

[4] A. Amrah, N. Zerhouni, and A. El Moudni. On the control of manufacturing lines modelled by controlled continuous Petri nets. *International Journal of Systems Science*, 29(2):127–137, 1998.

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

[6] J. Campos and M. Silva. Structural techniques and performance bounds of stochastic Petri net models. In G. Rozenberg, editor, *Advances in Petri Nets 1992*, volume 609 of *Lecture Notes in Computer Science*, pages 352–391. Springer, 1992.

[7] C. G. Cassandras. *Discrete Event Systems. Modeling and Performace Analysis*. Asken Associates, 1993.

[8] R. David and H. Alla. *Discrete, Continuous and Hybrid Petri Nets*. Springer-Verlag, 2005.

[9] F. DiCesare, G. Harhalakis, J. M. Proth, M. Silva, and F. B. Vernadat. *Practice of Petri Nets in Manufacturing*. Chapman & Hall, 1993.

[10] M. Dotoli, M.P. Fanti, A. Giua, and C. Seatzu. Modelling systems by hybrid Petri nets. An application to supply chains. In *Petri Net, Theory and Application*. ARS Publishing. to appear.

[11] J. Julvez and R. Boel. Modelling and controlling traffic behaviour with continuous Petri nets. In $16^{th}$ *IFAC World Congress*, Prague, Czech Republic, July 2005.

[12] J. Júlvez, L. Recalde, and M. Silva. Steady-state performance evaluation of continuous mono-T-semiflow Petri nets. *Automatica*, 41(4):605–616, 2005.

[13] J. Júlvez, L. Recalde, and M. Silva. Deadlock-freeness analysis of continuous mono-T-semiflow Petri nets. *IEEE Trans. on Automatic Control*, 51(9):1472–1481, 2006.

[14] D. Lefebvre. Feedback control design for manufacturing systems modeled by continuous Petri nets. *International Journal of Systems Science*, 30(6):591–600, 1999.

[15] C. Mahulea, A. Ramírez, L. Recalde, and M. Silva. Steady state control reference and token conservation laws in continuous Petri net systems. *IEEE Transactions on Automation Science and Engineering*, 5(2): 307–320, 2008.

[16] L. Recalde, E. Teruel, and M. Silva. Autonomous continuous P/T systems. In J. Kleijn S. Donatelli, editor, *Application and Theory of Petri Nets 1999*, volume 1639 of *Lecture Notes in Computer Science*, pages 107–126. Springer, 1999.

[17] M. Silva and L. Recalde. Petri nets and integrality relaxations: A view of continuous Petri nets. *IEEE Trans. on Systems, Man, and Cybernetics*, 32(4):314–327, 2002.

[18] M. Silva and L. Recalde. On fluidification of Petri net models: from discrete to hybrid and continuous models. *Annual Reviews in Control*, 28(2):253–266, 2004.

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

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

[21] A. Zimmermann, D. Rodríguez, and M. Silva. A Two Phase Optimisation Method for Petri Net Models of Manufacturing Systems. *Journal of Intelligent Manufacturing*, 12(5):421–432, October 2001.