PN Fluidification revisited: Semantics and Steady State

Laura Recalde [*], Manuel Silva
Departamento de Informática e Ingeniería de Sistemas
Centro Politécnico Superior de Ingenieros, Universidad de Zaragoza
María de Luna 3, E-50015 Zaragoza, Spain
lrecalde@posta.unizar.es, silva@posta.unizar.es

Abstract

Fluidification or continuization of (time-interpreted) Petri nets models may be a very useful relaxation technique when applied to highly populated systems. The idea is to try to overcome, at least partially, the potentially very high computational complexity arising in many practical situations. In this paper we revisit so called first-order fluidification of Petri nets, providing a bridge between classical servers-semantics in *stochastic-discrete* models and *deterministic-continuous* ones. Moreover, connections are set between continuous nets and *linear relaxation* techniques developed for the analysis of discrete event models. Polynomial time algorithms for computing properties related to the steady state, up to the class of bounded equal conflict (EQ) systems (a generalization of free choice and choice free), are presented.

Keywords: Fluidification, discrete and continuous Petri nets, semantics, steady state

1 INTRODUCTION

Fluidification (or continuization) of discrete models is a classical relaxation technique, close to bottleneck analysis, pursuing an approximate analysis of systems at a relatively reduced cost. For example, fluidification has been used when dealing with:

- *population dynamics* systems, for which the Lotka-Volterra classical model is of paradigmatic simplicity.
- *manufacturing* systems [**?**, **?**], particularly when considering long and, eventually, interleaved flow-lines.
- *communication* systems [**?**]
- *traffic flow* systems. It should be pointed out that, in this case, macroscopic continuous models are usually distributed parameter systems, leading to partial (non ordinary) differential equations (see, for example, [**?**]).

In Petri nets fluidification has been introduced independently from three perspectives. *First-order* and *total* (i.e. over all places) fluidification on PN was introduced and developed by R. David and coworkers, starting in 1987 [**?**] (see [**?**] for a recent survey). In this case, fluidification of timed discrete models leads to continuous deterministic models that can be written as a (switching) set of ordinary differential equations. A more recent approach to fluidification is due to K. Trivedi and his group [**?**, **?**]: the fluidification is *partial* (concerns the marking of one or few places), leading to *hybrid* models, and it is of second-order, i.e., models are *stochastic*.

Total fluidification of the marking on the state (or fundamental) equation ($\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \sigma, \mathbf{m} \geq \mathbf{0}, \sigma \geq \mathbf{0}$) of a net system allows the use of *linear* instead of *integer programming* techniques. Thus, the possibility of solving some logical verification or performance evaluation problems in polynomial time appears. The systematic use of linear programming with autonomous net systems was proposed in 1987 (see [**?**] for a revised version of the seminal work, and [**?**] for a recent survey). With respect to timed models, linear programming problems that allow the computation of performance bounds of stochastic PN have also attracted attention from late 80's. In essence, first-order approximations are built using Little-law (see [**?**] for the applications to bounded free-choice nets). Improved bounds have been obtained in several ways, for example in [**?**] by taking into consideration the queueing networks embedded into the PN system, and thus going beyond first-order approximation.

The goal of this paper is twofold:

1. to reencounter two of the most classical deterministic timed interpretations of continuous net models, introduced by R. David and his coworkers (constant and variable speed), starting from discrete Markovian timed net models. The conceptual advantage of this stochastic interpretation is that the discrete marking is a full representation of the (distributed) state, therefore the reachability graph and the Markov chain are isomorphic.
2. to exploit, in an integrated way, qualitative and quantitative analysis techniques. In particular, those developed after the fluidification of the

[*] author to whom correspondence should be addressed
This work was partially supported by the project HF97-169, within the Picasso programme and project CICYT TAP98-0679

marking in the state equation, and relate them to properties of the corresponding continuous net models. We will see that linear programming problems allow to compute the steady state solution for both deterministic timed interpretations.

In this paper we will not consider the general P/T systems, but will restrict to the subclass of equal conflict (EQ) systems [**?**], which is the weighted generalization of free choice.

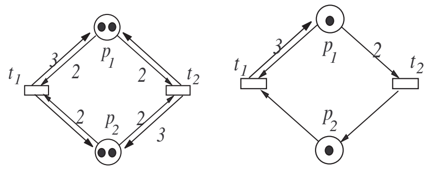
The remainder of the paper is organized as follows: In Section 2 autonomous continuous P/T systems are introduced and some basic results are presented. Section 3 is devoted to discrete markovian EQ systems, recalling the basic concepts of the stochastic interpretation. A continuous first-order deterministic approximation of this interpretation is presented in Section 4, and both single-server and infinite-servers semantics are studied. In Section 5 it is proven that the steady-state throughput for both semantics can be obtained solving a linear programming problem. Moreover, the solution of this LPP is known to be an upper bound for the throughput in the discrete case. Coxian expansion is considered in Section 6.

2 AUTONOMOUS CONTINUOUS P/T SYSTEMS

We assume the reader is familiar with PN (for notations we use the standard one, see for instance [**?**]).

The usual PN system, $\langle \mathcal{N}, \mathbf{m_0} \rangle$, will be said to be *discrete* so as to distinguish it from a *continuous* PN. The only difference between continuous and discrete PN is in the marking, which in a discrete PN is restricted to be integer, while in continuous PN may be any non-negative real number. The firing 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 $\mathbf{m}' = \mathbf{m} + \alpha \cdot \mathbf{C}[P,t]$.

The fluidification of a net system is intended as an *approximation*. A first thing to point out is that not all net systems allow a“reasonable” fluidification. Figure 1 shows two bounded net systems, evidencing that liveness in the original discrete and the continuous relaxed model may be completely unrelated. It can be pointed out that both have “equivalent” token-flow matrices ($\mathbf{C} = \mathbf{Post} - \mathbf{Pre}$), while their behaviour is completely different. The system in 1(a) is non-live as discrete for any initial marking (for instance, firing $\lfloor \mathbf{m_0}[p_1]/2 \rfloor$ times t_2 leads to a deadlock), but never gets completely blocked as continuous unless an infinitely long sequence is considered. On the other hand, the system in 1(b) is live as discrete with this initial marking, but deadlocks as continuous with any initial marking (firing $\mathbf{m_0}[p_1]/2$ times t_2 leads to a deadlock). (See [**?**] for a careful study of autonomous continuous net systems, liveness in particular.)



(a) (b)

Figure 1: Two bounded and strongly connected systems which behave in a very different way if they are considered as discrete or as continuous: (a) is non-live as discrete, but never gets completely blocked as continuous unless an infinitely long sequence is considered; (b) is live as discrete, but non-live as continuous.

Liveness-preservation w.r.t marking-homothecies is a necessary qualitative property in fluidificable net models [**?**]. That is, (discrete) live systems must remain live if the marking of every place is multiplied by the same natural number. This is the case for instance of bounded *equal conflict (EQ) systems*, which are systems in which all conflicts are equal, i.e., ${}^\bullet t \cap {}^\bullet t' \neq \emptyset \Rightarrow \mathbf{Pre}[P,t] = \mathbf{Pre}[P,t']$. In fact, live and bounded EQ systems remain live under any arbitrary increase of the initial marking [**?**]. Therefore, regarding the liveness property, the fluid approximation of those systems cannot present the above mentioned total absence of relationship w.r.t. the behavior of the discrete model. In bounded EQ systems, liveness of the discrete model implies liveness of the continuous one [**?**]. (The reverse may be false, if the initial marking is “too small”.)

The negative effect of fluidification at net level can be easily perceived observing that if the input and output arcs of a transition are multiplied by an arbitrary k, qualitative fireability is preserved. Nevertheless, for the discrete model, multiplying by a large enough k may disable the transition (in fact, if the discrete system is bounded, it may become non live).

Some results that may help to understand the differences between discrete and continuous autonomous net systems are [**?**]:

1. The reachability space for continuous systems is a *convex set*.
2. If all transitions can be fired from the initial marking, a marking can be reached from which all T-semiflows can be fired. Therefore, *structural and behavioural synchronic relations are identical* (e.g. the net is bounded iff structurally bounded, a place is implicit iff structurally implicit and the initial marking is large enough, . . .).
3. A *trap* (i.e. a set of places, P', such that $P'^\bullet \subseteq {}^\bullet P'$) can be, eventually, *emptied at the limit* (i.e. can loss all the fluid or substance contained in it).

Liveness may be extended in a different way assuming that infinitely long sequences can be fired, i.e., asymptotic markings can be reached. Hence, we say that a transition is *non* lim-*live* iff a sequence of successively reachable markings exists which converges to a marking such that none of its successors enables the transition [?]. For bounded discrete EQ systems, liveness is equivalent to deadlock-freeness [?]. This is also true for bounded continuous EQ systems. The following property provides a full characterization of liveness for bounded continuous EQ systems.

Theorem 1. *[?]*
Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a continuous EQ system. Let SEQS *be the quotient set defined by the following equivalence relation: two transitions, t and t', are in* equal conflict *relation when $\mathbf{Pre}[P,t] = \mathbf{Pre}[P,t'] \neq \mathbf{0}$.*
The following conditions are equivalent:

1. *The system is* lim-*live and bounded.*
2. *The system is consistent ($\exists \mathbf{x} > \mathbf{0}$ such that $\mathbf{C} \cdot \mathbf{x} = \mathbf{0}$), conservative, ($\exists \mathbf{y} > \mathbf{0}$ such that $\mathbf{y} \cdot \mathbf{C} = \mathbf{0}$),* $\mathrm{rank}(\mathbf{C}) = |\mathrm{SEQS}| - 1$ *(or, equivalently, it is str. bounded and str. live as discrete [?]) and the support of every P-semiflow is marked, i.e., $\nexists$ $\mathbf{y} \geq \mathbf{0}$ such that $\mathbf{y} \cdot \mathbf{C} = \mathbf{0}$, $\mathbf{y} \cdot \mathbf{m_0} = 0$.*

As a consequence, it can be stated that for structurally bounded EQ nets the existence of a non-null flow vector can be checked in polynomial time. In section 5 it will be shown that the computation of the flow vector can be performed also in polynomial time, at the expense of a small computational effort. In fact, assuming that the net is structurally bounded and live, applying Theorem 1 liveness is characterized by the absence of unmarked P-semiflows. We will see that this last condition appears again in the LPP1 in section 5.1.

3 MARKOVIAN EQ SYSTEMS

If the timing interpretation considered for discrete P/T systems is stochastic and *all* transitions are exponentially distributed, the *memoryless* property of Markovian models makes isomorphic the reachability graph of the autonomous net system and the Markov chain associated to the underlying stochastic model [?]. Moreover, under *infinite-servers semantics* (or *delay node*, using queueing networks terminology) a transition t enabled k times in a marking $\mathbf{m}$ (i.e., $k = \max_{t \in p^\bullet} \{n | \mathbf{m} \geq n\mathbf{Pre}[P,t]\}$) works at k times the speed it would work if it were enabled only once. Of course, an infinite-servers transition can always be constrained to a "k-servers" behaviour by adding one place that is both input and output (self-loop with multiplicity one) for that transition and marking it with k tokens. Other kinds of marking or time dependency of service times are forbidden.

We assume here that *timed* transitions may never be in conflict. Therefore, all timed transitions are *persistent*, i.e., a timed transition will be enabled until it is fired. For the modelling of conflicts we use *immediate* transitions with the addition of (marking and time independent) routing rates $\mathcal{R}$. In other words, for the subset of immediate transitions $\{t_1, \ldots, t_k\} \subset T$ being in conflict at each reachable marking, we suppose that the constants $r_1, \ldots, r_k \in \mathbb{R}^+$ are explicitly defined in the system interpretation in such a way that when $t_1, \ldots, t_k$ are simultaneously enabled, transition t_i ($i = 1, \ldots, k$) fires with probability $r_i / (\sum_{j=1}^{k} r_j)$. Moreover, immediate transitions will only be used for routing, and not to synchronize. In particular, the weight of input and output arcs of immediate transitions is always one. This way, routing is completely decoupled from duration of activities. We will call these systems *markovian P/T systems*, MPT.

As an example, Table 1 shows the size of the reachability set, and the throughput in the steady state, of the system in Figure 2 with mean service time of every transition equal to 1.

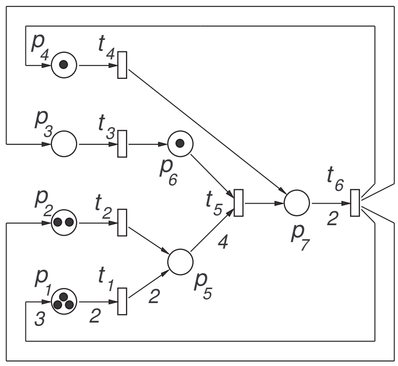


Figure 2: A choice free system modelling a manufacturing application [?]

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. The vector of visit ratios of a live and bounded system must be a T-semiflow (the input weighted flow to each place must be equal to the output weighted flow): $\mathbf{C} \cdot \mathbf{v}^{(i)} = \mathbf{0}, \mathbf{v}^{(i)} \geq \mathbf{0}$. For instance, for the system in Figure 2, $\mathbf{v}^{(6)} = [3/2, 1, 1, 1, 1, 1]$. The visit ratios of transitions in EQ relation, i.e., $\mathbf{Pre}[P,t_j] = \mathbf{Pre}[P,t_k]$, must be proportional to the corresponding routing rates.

In summary, the next result can be stated [?]: The vector of visit ratios with respect to transition t_i of a live and bounded net system must be a solution of:

$$\mathbf{C} \cdot \mathbf{v}^{(i)} = \mathbf{0}; \;\; \mathbf{R} \cdot \mathbf{v}^{(i)} = \mathbf{0}; \;\; \mathbf{v}^{(i)}[t_i] = 1 \qquad (1)$$

where $\mathbf{R}$ is a matrix that relates the visit ratios of transitions in equal conflict relation, according to the corresponding routing rates. Equations in the above statement have been shown to characterize a unique vector for important net subclasses [?, ?].

For stochastic processes, ergodicity is a basic property that guarantees the existence of a steady state. Using the isomorphism between the reachability graph and the Markov chain, it can be proved that live and bounded EQ MPT systems are ergodic.

k	Size of the Reachability Space	Infinite servers $\chi_I[t_6]$	$\chi_I[t_6]/k$	Single-server $\chi_S[t_6]$
1	54	0.2522	0.2522	0.2401
2	1685	0.6493	0.3246	0.4757
3	10354	1.0567	0.3522	0.5805
4	37722	1.4623	0.3656	0.6309
5	103914	1.8671	0.3734	0.6563
...	...	...	...	...
∞	...	∞	LPP1=2/5	LPP2=2/3

Table 1: Throughput of t_6 in Figure 2 for initial marking $k \cdot \mathbf{m_0}$.

Theorem 2. *Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a live and bounded EQ MPT system. Then,*

1. *[**?**] The reachability graph contains home states (states that can be ultimately reached after whichever evolution).*
2. *The underlying Markov chain is ergodic, i.e., a steady state exists.*

4 TWO BASIC SERVERS SEMANTICS FOR CONTINUOUS MODELS

In discrete stochastic net systems, two main semantics are used: *infinite-servers* and *finite-servers*. In this section we will consider deterministic first-order approximations of these semantics for continuous systems.

4.1 General flow constraints

As in discrete systems, the state equation $\mathbf{m} = \mathbf{m_0} + \mathbf{C} \cdot \sigma$ (with $\mathbf{C}$ the token-flow matrix and σ the firing vector) summarizes the way the marking evolves along time. But, in continuous systems, the marking is continuously changing, so we may consider the derivative of $\mathbf{m}$ with respect to time. This way we obtain that $\dot{\mathbf{m}} = \mathbf{C} \cdot \dot{\sigma}$. Let us call $\mathbf{f} = \dot{\sigma}$, since it represents the *flow* through the transitions. In general $\mathbf{f}$ is not constant, but may depend locally on the marking, thus on time. Observe that if a steady state is reached, $\dot{\mathbf{m}} = \mathbf{0}$, and so $\mathbf{C} \cdot \mathbf{f} = \mathbf{0}$. That is, the flow vector in the steady state is a T-semiflow, even if the net is structurally unbounded.

Immediate transitions in discrete systems may be used to determine the relative rate in conflicts resolution. From a continuous point of view conflicts can be seen as "flow splitters", with the relative probability of an outcome giving its relative flow. That is, the marking of the output places of the immediate transitions, $p_1^o, \ldots, p_2^o$ in the system on the left in Figure 3, can be obtained as:

$$\dot{\mathbf{m}}[p_j^o] = \frac{r_j}{r_1 + \cdots + r_k} \cdot \sum_{t \in {}^\bullet p^i} \mathbf{f}[t] \cdot \mathbf{Post}[p^i, t] - \mathbf{f}[t_j^o] \cdot \mathbf{Pre}[p_j^o, t_j^o]$$

Observe that this is equivalent to the system on the right, without immediate transitions. And this system can be transformed into a system with integer arc weights, just multiplying the marking and every arc by $r_1 + \cdots + r_k$.

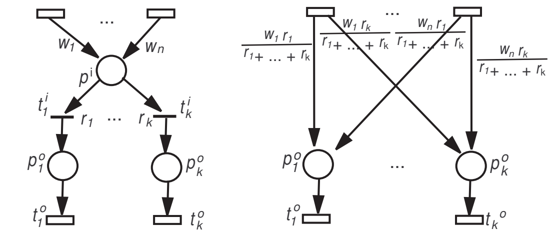


Figure 3: Immediate transitions in an equal conflict.

Lemma 3. *Any EQ continuous MPT system can be transformed into an equivalent CF MPT system without immediate transitions.*

Notice that a timing interpretation may hide qualitative properties of the underlying autonomous system. For instance, the *discrete* —autonomous or markovian— system in Figure 4 is non-live. However, if identical rates are assigned to t_1 and t_2, the continuous deterministic system will never stop.

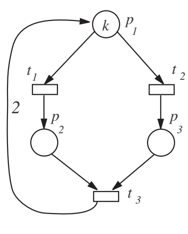


Figure 4: A non-live discrete system that never stops as continuous if the same rates are assigned to t_1 and t_2.

4.2 Infinite-servers semantics

In discrete MPT systems, infinite-servers semantics means that the delay between the enabling and the firing of a transition is the minimum of n ($n = \text{enabling}(t)$) independent exponentially distributed variables, all of them with the same parameter, λ. This is an exponential distribution too, with parameter $n\lambda$ (by *flow additivity* for exponential distribution or, more generally, by *order statistics* [**?**]). If we make a first-order approximation, i.e., we just take into account the mean values, the firing of the transition takes $1/(n\lambda)$ time units. This can be interpreted from a continuous point of view as $\mathbf{f}[t] = \lambda \cdot \text{enabling}(t)$, which corresponds to the *variable speed* of [**?**].

4.3 Finite-servers semantics

In discrete systems, finite-servers semantics can be implemented by means of infinite-servers semantics by adding a place marked with as many tokens as the number of servers. However, this does not represent finite-servers semantics if these tokens are interpreted as fluids. To have "true" finite-servers semantics, the "servers tokens" should be considered different from the tokens in the rest of the marking, because they cannot be split. From a discrete point of view, if the marking is multiplied by a natural number (greater than one), the tokens representing the servers should not be multiplied (i.e., we may have more customers, but keep the number of resources).

Let us consider *single-server* semantics first. In discrete MPT systems, when the transition is enabled, it fires with a rate exponentially distributed with parameter λ. Let x_i be the delay of the i-th firing. Then, average delay after k firings, $\frac{1}{k} \cdot \sum_{i=1}^{k} x_i$, is a k-Erlang distribution with mean $1/\lambda$ (thus variance $1/(\lambda^2 k)$). That is, if we integrate the flow along a large period of time, we obtain approximately the same as if the flow had been constant, because the variance tends to vanish when k is large. So, if the transition is always firing (i.e., it is always enabled), the firing can be approximated by a constant flow with speed λ. If it is not, instead of having idle periods, we may approximate the server by a slower one that is always busy, i.e., one that works at the same speed as the incoming flow of tokens. In any case, if we let the system evolve for a large enough period, the error will be small. This corresponds to the *constant speed* of [**?**], with $V_i = \lambda[t_i]$, that is,

$$\mathbf{f}[t] = \begin{cases} \lambda[t] \quad \text{if } \nexists p \in {}^\bullet t \text{ with } \mathbf{m}[p] = 0 \\ \qquad\qquad (\textit{strong enabling} \text{ in [?]}) \\ \min\limits_{p \in {}^\bullet t} \left\{ \sum\limits_{t' \in {}^\bullet p} \dfrac{\mathbf{f}[t'] \cdot \mathbf{Post}[t',p]}{\mathbf{Pre}[p,t]} \right\} \text{ otherwise} \\ \qquad\qquad (\textit{weak enabling} \text{ in [?]}) \end{cases}$$

Let us consider now the k-servers semantics. In this case, the firing rate of the transition is exponentially distributed with parameter $\lambda \cdot \min\{k, \text{enabling}(t)\}$. If the number of servers, k, is small with respect to the (fluid) marking of the system, it is reasonable to consider that this minimum will (often) be k. Then, the same reasoning as in the previous case can be applied, obtaining a maximal flow of $k\lambda$.

5 Steady state computation for EQ systems

5.1 Throughput bounds for discrete EQ systems

In the case of live and bounded discrete EQ systems (as an instance of subclasses of nets for which the visit ratios can be efficiently obtained) upper (and lower) bounds for the throughput of transition t_i, $\chi[t_i]$, can be computed in terms of the visit ratios [**?**, **?**], independently of the probability density functions (pdfs):

$$\textbf{LPP1: } 1/\chi[t_i] \geq \max\{\mathbf{y} \cdot \mathbf{Pre} \cdot \mathbf{D}^{(i)} \mid \mathbf{y} \cdot \mathbf{C} = \mathbf{0}, \quad \mathbf{y} \cdot \mathbf{m_0} = 1, \quad \mathbf{y} \geq \mathbf{0}\} \tag{2}$$

where $\mathbf{D}^{(i)}$ is the vector of *average service demands*, defined as $\mathbf{D}^{(i)}[t] = \mathbf{v}^{(i)}[t] \cdot \mathbf{s}[t]$, and $\mathbf{s}$ is the vector of mean service times, i.e., $\mathbf{s}[t] = 1/\lambda[t]$.

This upper bound basically consists on finding the slowest isolated subnet among those generated by the elementary P-semiflows ($\mathbf{y}$). In other words, the bound is obtained looking at the bottleneck P-semiflow. Clearly, $\chi[t_i]$ can be computed in polynomial time.

It can be observed that if the initial marking is multiplied by a constant, k, ($\mathbf{m_0}' = k \cdot \mathbf{m_0}$) the solution of the linear programming problem, is also multiplied by k, i.e., $\text{LPP1}(\mathbf{m_0}') = k \cdot \text{LPP1}(\mathbf{m_0})$.

For the system in Figure 2, $\mathbf{D}^{(6)} = [3/2, 1, 1, 1, 1, 1]$, and the set of elementary P-semiflows is $\mathbf{y}_1 = [1, 1, 0, 4, 1, 0, 4]$ and $\mathbf{y}_2 = [0, 0, 1, 1, 0, 1, 1]$. Then, the linear programming problem in (2) reduces to

$$\max\left\{ \frac{\mathbf{y}_1 \cdot \mathbf{Pre} \cdot \mathbf{D}^{(6)}}{\mathbf{y}_1 \cdot \mathbf{m_0}}, \frac{\mathbf{y}_2 \cdot \mathbf{Pre} \cdot \mathbf{D}^{(6)}}{\mathbf{y}_2 \cdot \mathbf{m_0}} \right\} = \\ = \max\{40/18, 5/2\} = 5/2$$

Therefore, $2/5$ is an upper bound of the throughput of t_6. The slowest P-semiflow in this case is $\mathbf{y}_2$, hence the bottleneck conservative component is the P-subnet generated by the support of $\mathbf{y}_2$, $\{p_3, p_4, p_6, p_7\}$, and its input and output transitions. Observe also in Table 1 that $\chi_k[t_6]/k \leq \chi_{k+1}[t_6]/(k+1)$ and $\lim_{k\to\infty} \chi_k/k = 2/3$. We conjecture this property holds in general for EQ systems, and not just for this particular case.

5.2 Continuous EQ models under infinite-servers semantics

For EQ systems, the previous bound for the throughput of a discrete system is equal to the flow of the continuous system in the steady state. The proof is based on the fact that in EQ systems, we may fix one input place for each transition, and the P-subnet these places generate must contain a P-semiflow [**?**]. In particular, the P-subnet defined taking for each synchronization the place that is the "slowest" one in the steady state will contain at least one P-semiflow. Moreover, one of those will be a (the) "bottleneck" P-semiflow, i.e., a P-semiflow associated to the solution of LPP1.

Theorem 4. *Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a live and bounded EQ continuous system with infinite-servers semantics. The flow of the system in the steady state can be computed with the same LPP1 stated in (2) for discrete systems.*

Proof. Applying Lemma 3, it is sufficient to prove the theorem for CF systems.

Let $\mathbf{m}_{ss}$ be the steady state marking of the system. The flow vector associated to this state, $\mathbf{f}_{ss}$, is constant and $\mathbf{C} \cdot \mathbf{f}_{ss} = \mathbf{0}$. Since CF systems have a unique T-semiflow, up to constants, $\mathbf{f}_{ss} = k \cdot \mathbf{v}^{(i)}$, with $\mathbf{v}^{(i)}$ the vector of visit ratios normalized w.r.t. transition t_i. (Observe that $k = \mathbf{f}_{ss}[t_i]$.)

On the other hand, since it is an infinite servers semantics, $\mathbf{f}_{ss}[t_j] = \lambda[t_j] \cdot \min_{p \in {}^\bullet t_j} \{\mathbf{m}_{ss}[p]/\mathbf{Pre}[p,t_j]\}$ and so

$$\mathbf{D}^{(i)}[t_j] = \frac{\mathbf{v}^{(i)}[t_j]}{\lambda[t_j]} = \frac{\mathbf{f}_{ss}[t_j]}{k \cdot \lambda[t_j]} = \frac{1}{k} \cdot \min_{p' \in {}^\bullet t_j} \left\{ \frac{\mathbf{m}_{ss}[p']}{\mathbf{Pre}[p',t_j]} \right\}$$

Let $p \in P$ and $t_j = p^\bullet$. Then

$$\begin{aligned} \mathbf{Pre}[p,T] \cdot \mathbf{D}^{(i)} &= \mathbf{Pre}[p,t_j] \cdot \mathbf{D}^{(i)}[t_j] \\ &= \mathbf{Pre}[p,t] \cdot \frac{1}{k} \cdot \min_{p' \in {}^\bullet t_j} \left\{ \frac{\mathbf{m}_{ss}[p']}{\mathbf{Pre}[p',t_j]} \right\} \\ &\leq \frac{1}{k} \cdot \mathbf{m}_{ss}[p] \end{aligned}$$

For any P-semiflow $\mathbf{y}$ with $\mathbf{y} \cdot \mathbf{m_0} = 1$,

$$\mathbf{y} \cdot \mathbf{Pre} \cdot \mathbf{D}^{(i)} \leq \frac{\mathbf{y} \cdot \mathbf{m}_{ss}}{k} = \frac{\mathbf{y} \cdot \mathbf{m_0}}{k} = \frac{1}{k}$$

Hence the solution of the LPP1 is at most $1/k$. Let us choose for every transition t_j one of its input places, the one for which $\mathbf{m}_{ss}[p]/\mathbf{Pre}[p,t_j]$ is minimum. It can be seen that this set of places contains the support of at least one P-semiflow of the system [**?**], $\mathbf{y}_0$. It is clear that $\mathbf{y}_0 \cdot \mathbf{Pre}[p,T] \cdot \mathbf{D}^{(i)} = 1/k$. Therefore, k is the solution of LPP1.

□

Observe that the throughput in the steady state does not depend on how the initial marking was distributed among the places, but just on the total (weighted) amount of tokens. Nevertheless, $\mathbf{m}_{ss}$ may depend on how is $\mathbf{m_0}$ defined. For instance in the system in Figure 5 the marking in the steady state for $k = u + v > 1$ depends on how were the tokens placed in the initial marking.

Figure 6 shows the evolution of the marking and the throughput of the system in Figure 2 under infinite-servers semantics. Observe that for the chosen $\mathbf{m_0}$ the evolution is non-monotonic towards the steady state.

5.3 Continuous models under finite-servers semantics

With single-server semantics, the flow of a transition t must be always less than or equal to $\lambda[t]$. Moreover, as we have seen before, in the steady state the flow vector

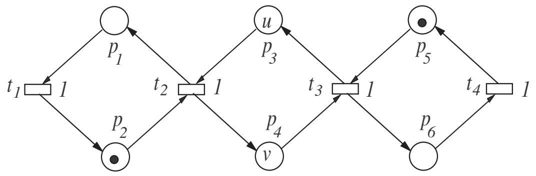


Figure 5: In this live and bounded marked graph, with $\lambda = \mathbf{1}$, the throughput on the steady state depends only on $k = u + v$. However, if $k > 1$ the marking in the steady state is not completely defined by $\lim_{\tau \to \infty} \mathbf{f}(\tau)$.

is a T-semiflow, i.e., proportional to the visit ratio vector $\mathbf{v}^{(i)}$. So, we just need to observe the transitions and find the bottleneck, that is, a certain t_j whose utilization is equal to 1.

Theorem 5. *Let $\langle \mathcal{N}, \mathbf{m_0} \rangle$ be a live and bounded EQ continuous system with finite-servers semantics. The throughput of transition t_i in the steady state is*

$$\textbf{LPP2:}\ \chi[t_i] = \max\{k \mid k \cdot \mathbf{v}^{(i)} \leq \lambda\} \qquad (3)$$

Proof. Applying Lemma 3, it is sufficient to prove the theorem for CF systems.

Let $\mathbf{m}_{ss}$ be the steady state of the system. Then, the flow vector $\mathbf{f}_{ss}$ is constant and $\mathbf{C} \cdot \mathbf{f}_{ss} = \mathbf{0}$. Since CF systems have a unique T-semiflow, up to constants, $\mathbf{f}_{ss} = k\mathbf{v}^{(i)}$, with $\mathbf{v}^{(i)}$ the vector of visit ratios normalized w.r.t. transition t_i, and $k = \mathbf{f}_{ss}[t_i]$.

For every transition t, $\mathbf{f}_{ss}[t] \leq \lambda[t]$, therefore $k \leq \min_{t \in T} \{\lambda[t]/\mathbf{v}^{(i)}[t]\}$.

Assume this minimum is strict. Then for every transition t there exists an empty input place. This set of places contains the support of a P-semiflow [**?**]. So, there exists an empty P-semiflow, contradiction since the system is live. □

Equation (3) can also be deduced from LPP1, if (1) single-server semantics is imposed by adding self-loop places marked with one token to each transition, and (2) the original initial marking is multiplied by a "large enough" constant to distinguish this "divisible" tokens from the "indivisible" servers. Now, we have two kinds of P-semiflows, the "original" ones, and the ones related to self-loops. Since the first kind of P-semiflows have a large amount of tokens, the "bottleneck" P-semiflow will be one associated to a self-loop. In other words, applying LPP1 the slowest transition is detected.

For the system in Figure 2, Figure 7 shows the evolution of the marking and the throughput under single-server semantics. The steady state throughput of t_6 is equal to $2/3$, which is the limit of the corresponding column of Table 1. It can be observed that the steady state throughput under finite- or infinite-servers semantics are different. Even more, from (3) it can be seen that the throughput under finite-servers semantics does *not* depend on the initial marking (as long as the system is live).

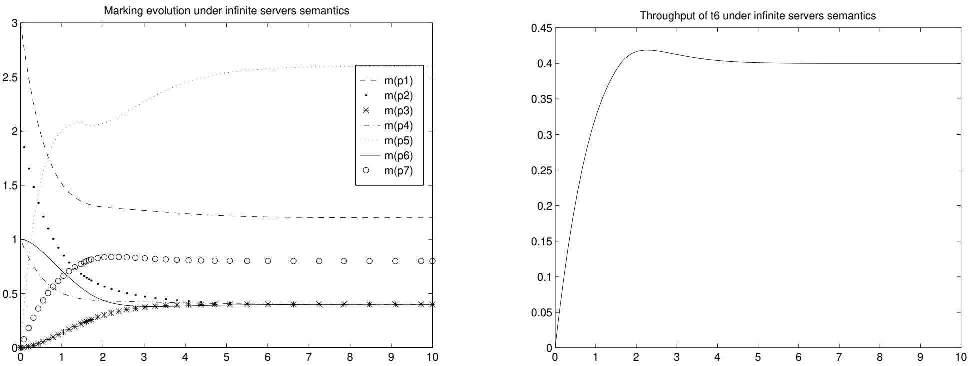


Figure 6: Marking and throughput of the system in Figure 2 under infinite-servers semantics

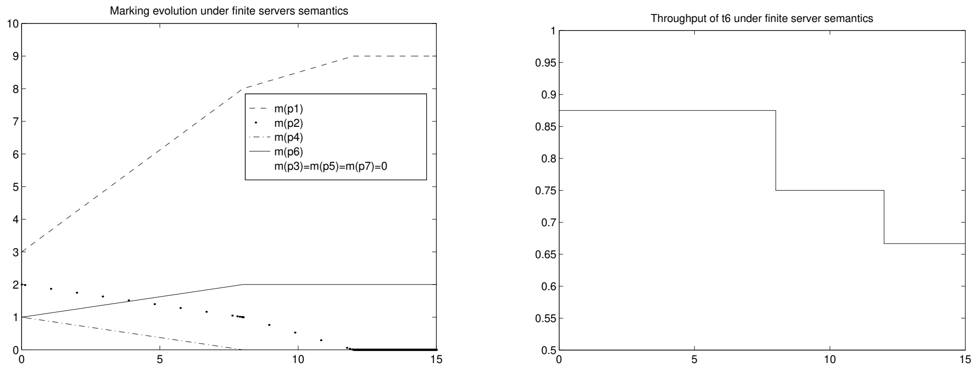


Figure 7: Marking and throughput of the system in Figure 2 under finite-servers semantics

Corollary 6. *The throughput of a live and bounded EQ continuous system under finite-servers semantics does not depend on the initial marking.*

This last property is due to the fact that servers-tokens are indivisible, while the rest of the tokens are fluid (so, a small amount of fluid may represent an extremely large quantity of tokens).

6 COXIAN SERVICE PDFS

It is common in discrete stochastic PN to simulate Coxian distributions by making an expansion into its exponential phases [**?**]. To approximate the discrete PN by a continuous one, we can make a continuous first-order approximation either of the aggregated model, or of the expanded one. Which is the difference (if any) between these two continuous models?

If we consider an infinite-servers semantics, it is immediate to see that the solution of LPP1 will be the same for both systems. That is, the expansion is useless if it's just the steady state what we are interested in. On the other hand, the behaviour in the transient is *not* the same. This can be seen for instance in Figure 8. It shows the marking of the system in Figure 2, with $\mathbf{m_0} = [3,2,1,1,0,0,0]$ and t_3 a 3-Erlang with and without expansion(slower and smoother).

The expansion does not make sense if a finite-servers semantics is considered. For instance, Figure 9 shows the expansion of a 2-Erlang under single servers semantics. But, since the "server token" is continuous, place p is having no effect. So, this expansion will lead to doubling the speed, which certainly is not what is expected.

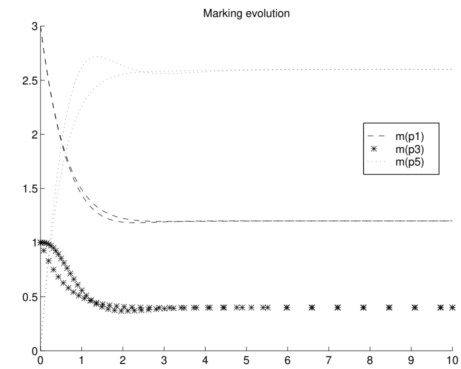


Figure 8: Marking of the system in Figure 2, if t_3 is a 3-Erlang with and without expansion and $\mathbf{m_0} = [3,2,1,1,0,0,0]$.

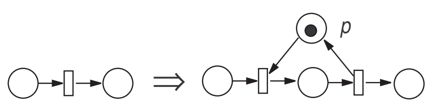


Figure 9: Expansion of a 2-Erlang under single-servers semantics.

7 Concluding remarks

The idea of fluidification, of approaching a discrete model of a system by a relaxed continuous one has been known and used for a long while. Nevertheless, qualitative properties of discrete and continuous models may be non related. For instance, we have shown here that a PN may be live as discrete and non-live as continuous.

In this work we also revisit the evolution semantics for continuous net systems in [?]: their constant-speed corresponds to finite-servers semantics in discrete stochastic PN, while their variable-speed corresponds to infinite-servers semantics. This allows to make a bridge to well-known results for discrete PN, emphasizing the fact that continuous PN belong to the same formal paradigm (leading to conceptual coherence and economy).

The established bridge may also be used to transfer to continuous PN results that have been obtained for discrete PN. For instance, we have shown that a linear programming problem that provides steady state throughput upper bounds for discrete EQ systems, leads to exact values for continuous EQ systems, and therefore to upper bounds for the intermediate hybrid models.

References