Noname manuscript No.
(will be inserted by the editor)

Fluid approximation of Petri net models with relatively small populations

Estíbaliz Fraca · Jorge Júlvez · Manuel Silva

the date of receipt and acceptance should be inserted later

Abstract Fluidization is an appealing relaxation technique based on the removal of integrality constraints in order to ease the analysis of discrete Petri nets. The result of fluidifying discrete Petri nets are the so called *Fluid* or *Continuous* Petri nets. As with any relaxation technique, discrepancies among the behaviours of the discrete and the relaxed model may appear. Moreover, such discrepancies may have a comparatively bigger effect when the population of the system, the marking in Petri net terms, is "relatively" small. This paper proposes two complementary approaches to obtain a better fluid approximation of discrete Petri nets. The first one focuses on untimed systems and is based on the addition of places that are *implicit* in the untimed discrete system but not in the continuous. The idea is to *cut* undesired *spurious* solutions whose existence worsens the fluidization. The second one focuses on a particular situation that can severely affect the quality of fluidization in timed systems. Namely, such a situation arises when the enabling degree of a transition is equal to 1. This last approach aims to alleviate such a state of affairs, which is termed the *bound reaching problem*, on systems under *infinite servers* semantics.

1 Introduction

Petri Nets (PN) is a well known family of formalisms for the modelling of Discrete Event Systems (DES). As any other formalism for DES, they suffer from the well known *state explosion problem*. Such a problem appears both during the analysis (e.g., to decide if the system is bounded or not) and the synthesis (e.g., designing a controller) of the system, and it affects both the *untimed* and the *timed* model.

Instituto de Investigación en Ingeniería de Aragón (I3A), Universidad de Zaragoza,
María de Luna 3, E-50018 Zaragoza, Spain (e-mail: {efraca, julvez, silva}@unizar.es).

Many different time interpretations can be adopted for the timing of PN. Nevertheless, without any doubt, one of the "most basic and classical" interpretation for performance evaluation and control consists of: (1) associating exponential probability distributions to the delay of the atomic firing of transitions ; and (2) solving conflicts by a race policy (see, for example, [18, 1, 4]). In the sequel, we will assume this time interpretation for discrete models. The resulting stochastic nets will be referred as *Markovian Petri Nets* (MPN).

An interesting technique to overcome the above mentioned state explosion problem is known as *fluidization*. The fluidization of a transition consists of relaxing its firing amount (and thus the marking of its neighbour places) to the non-negative real quantities. If all transitions are fluidized, the result is a *fluid* or *continuous* PN (CPN) [10, 21, 20]. By fluidization, more efficient analysis techniques can be developed at the price of losing some fidelity. In particular, the CPN may not preserve some qualitative or quantitative properties of the original discrete one [22, 21]. In other words, this issue is an instance of the classical trade-off between "accuracy" and "computational complexity".

Similarly to the *linearization* of any continuous nonlinear time-driven dynamical system, the *fluidization* of DES (untimed or timed) requires some conditions to be of reasonable quality; for example, to satisfy the *marking homothetic monotonicity* property [14]. If such property holds and the marking is large, then the results obtained with fluidization are frequently very good. With respect to timed net models, some functional extensions of the *law of large numbers* lead to the legitimization of the deterministic continuous PN approximation (see Subsection 2.3). This last relaxed model can be expressed as a set of *Ordinary Differential Equations* (ODEs). If the marking is not large, then some functional extensions of the *Central Limit Theorem* can be helpful, leading to *Stochastically Differential Equations* (SDEs) [25, 5]. In the sequel we limit to *deterministic* relaxations.

Synchronizations in PN can be expressed with two complementary constructions: (1) *rendez-vous* (or joins); and (2) weights in arcs going from places to transitions. At this point it should be pointed out that if the marking is "very large", the effect of those weights on arcs is not "seen" [20] (intuitively speaking, if the marking of the place at the origin of the k-weighted arc is $1000k$ –i.e, relatively very big– the enabling is 1000, so the continuous approximation is valid). However, if the marking is not "very large", the relative errors may be higher (intuitively speaking, rounding the number 1.5 to 1 lead to a relative error three orders of magnitude bigger than rounding 1000.5 to 1000). Thus, appropriate fluidization techniques are required for systems whose marking is "relatively" small (and hence cannot be fluidified properly), yet large enough to make its study a discrete system computationally prohibitive.

This paper deals with techniques to improve the fluidization process, what is specially interesting when the population is "relatively" small, at least in some parts of the system. We do not consider neither very small populations (in which fluidization is *frequently* not needed) nor very large ones (in which fluidization usually provides a good approximation of the original PN).

Some of the differences between discrete and continuous systems appear because solutions of the fundamental equation which are *spurious* in the untimed discrete system (i.e., analytical solutions of the *fundamental* or *state-transition* equation that are not reachable on the net model) may become reachable by the continuous relaxation. Therefore, some transformations on the discrete PN system are firstly proposed. They improve the fluidization process for untimed PN, thus potentially *for any timed interpretation*. In particular, the steady state throughput of the MPN will be better approximated by the continuous approximation, i.e., by the timed CPN (TCPN).

These transformations are based on the addition of some places which are *implicit* in the discrete system [23] but they constraint the behaviour of the continuous one. In particular, such *cutting* implicit places [23] remove some spurious solutions. The key issue here is that any continuous (possibly integer) spurious deadlock can be removed (the main differences between the discrete model and the continuous approximation is caused by spurious deadlocks). The elimination of a given spurious deadlock is a computational problem of *polynomial time* complexity. Unfortunately, the number of spurious deadlocks may be theoretically exponential. Nevertheless, this is not a frequent case in practice. Let us remark that improvements in both timed and untimed models are obtained by these techniques.

With respect to timed models, we focus on a particular situation in which the fluidized system does not approximate certain quantitative properties of the original one. It is the case of PN systems in which the enabling bound of a transition is equal to 1, and hence the probability of firing that transition may be very low in the discrete case but not so "difficult" on the continuous approximation. This problem is denoted as the *Bound Reaching Problem* (BRP) [13]. The BRP is a challenging problem that may appear in many practical cases. It can arise in systems in which relatively small and large populations are combined in a given model, and also when *inhibitor arcs* of a bounded system are removed and simulated with regular arcs and places.

Among the different concerns related to the BRP, in general terms, the approximation of the throughput of a discrete Markovian PN (MPN) by a Timed Continuous PN (TCPN) under *infinite server* semantics (ISS) was considered in [21]. Here we extend such an approximation to *join* or *rendez-vous* transitions by means of some *representative* places which implement the concept of *linear enabling functions* [24, 7].

The rest of the paper is organized as follows. Section 2 recalls basic definitions. In Section 3, we concentrate on the addition of *cutting* implicit places with the goal of improving the untimed continuous approximation. The *bound reaching problem* is introduced in Section 4. In Section 5, a method derived from ISS, here denoted as *ρ-semantics*, is proposed for the firing of transitions involved in the BRP. Section 6 deals with the extension of previous results to the most frequent class of synchronizations: *rendez-vous*. A case study is discussed in Section 7, and Section 8 concludes the paper.

2 Definitions and previous concepts

The main concepts related to discrete and continuous PN are recalled here, both as untimed and timed formalisms. The relationship between the timed interpretations when the system population tends to infinity is also established. In the following, it is assumed that the reader is familiar with Petri nets (see [11, 10] for a gentle introduction).

2.1 Discrete Petri nets

A Petri net is a tuple $\mathcal{N} = \langle P, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle$ where $P = \{p_1, p_2, ..., p_n\}$ and $T = \{t_1, t_2, ..., t_m\}$ are disjoint and finite sets of places and transitions, and $\boldsymbol{Pre}$, $\boldsymbol{Post}$ are $|P| \times |T|$ sized, natural valued, incidence matrices. The *preset* and *postset* of a node $u \in P \cup T$ are denoted by $^\bullet u$ and $u^\bullet$, respectively. A discrete PN system is a tuple $\langle \mathcal{N}, \boldsymbol{M}_0 \rangle$ where $\mathcal{N}$ is the net structure and $\boldsymbol{M}_0 \in \mathbb{N}^{|P|}_{\geq 0}$ is the initial marking (denoted in upper case $\boldsymbol{M}$ for the discrete system).

The enabling degree of transition t_i at marking $\boldsymbol{M}$ is defined as:

$$Enab(t_i, \boldsymbol{M}) = \min_{p_j \in {}^\bullet t_i} \lfloor \frac{\boldsymbol{M}[p_j]}{\boldsymbol{Pre}[p_j, t_i]} \rfloor \tag{1}$$

The firing of t_i in a certain *natural* amount $\alpha \leq Enab(t_i, \boldsymbol{M})$ leads to a new marking $\boldsymbol{M}'$, which is denoted as $\boldsymbol{M} \xrightarrow{\alpha t_i} \boldsymbol{M}'$, and $\boldsymbol{M}' = \boldsymbol{M} + \alpha \cdot \boldsymbol{C}[P, t_i]$, where $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$ is the *token flow* matrix (*incidence* matrix if $\mathcal{N}$ is self-loop free) and $\boldsymbol{C}[P, t_i]$ denotes the i^{th} column in $\boldsymbol{C}$. Hence, $\boldsymbol{M} = \boldsymbol{M}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$, the *state-transition* (or *fundamental*) equation summarizes the marking evolution; where $\boldsymbol{\sigma}$ is the firing count vector associated to the fired sequence.

Right and left natural annullers of the token flow matrix are called T- and P-semiflows, respectively. When $\exists \mathbf{y} > \mathbf{0}$, $\mathbf{y} \cdot \boldsymbol{C} = \mathbf{0}$, the net is said to be *conservative*, and when $\exists \mathbf{x} > \mathbf{0}$, $\boldsymbol{C} \cdot \mathbf{x} = \mathbf{0}$, the net is said to be *consistent*. A nonempty set of places Θ is a *trap* if $\Theta^\bullet \subseteq {}^\bullet\Theta$, while a nonempty set of places Σ is a *siphon* if ${}^\bullet\Sigma \subseteq \Sigma^\bullet$.

The set of all the reachable markings of $\langle \mathcal{N}, \boldsymbol{M}_0 \rangle$ is denoted as $RS_D(\mathcal{N}, \boldsymbol{M}_0)$. Its linearised reachability set (LRS) contains the markings which fulfill the fundamental equation (even if they are not reachable) [23]. In this work, the LRS is defined on the real numbers ($\boldsymbol{m} \in \mathbb{R}^{|P|}_{\geq 0}$):

$$LRS(\mathcal{N}, \boldsymbol{M}_0) = \{\boldsymbol{m} | \boldsymbol{m} = \boldsymbol{M}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}, \boldsymbol{m} \in \mathbb{R}^{|P|}_{\geq 0}, \boldsymbol{\sigma} \in \mathbb{R}^{|T|}_{\geq 0}\} \tag{2}$$

A marking $\boldsymbol{M}$ is *spurious* if it is a non reachable solution of the state-transition equation, i.e., $\boldsymbol{M} \in LRS(\mathcal{N}, \boldsymbol{M}_0)$ but $\boldsymbol{M} \notin RS_D(\mathcal{N}, \boldsymbol{M}_0)$. The *structural bound* of a place p_j, and the *structural enabling bound* of a transition t_i are integer values defined as:

$$SB(p_j) = \lfloor max\{\boldsymbol{M}[p_j] \mid \boldsymbol{M} = \boldsymbol{M}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}, \boldsymbol{M}, \boldsymbol{\sigma} \geq \mathbf{0}\} \rfloor \tag{3}$$

$$SEB(t_i) = \lfloor max\{e \mid \forall p \in {}^\bullet t_i, e \leq \frac{\boldsymbol{M}[p_j]}{\boldsymbol{Pre}[p_j, t_i]}, \\ \boldsymbol{M} = \boldsymbol{M}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}, \boldsymbol{M}, \boldsymbol{\sigma} \geq \boldsymbol{0}\} \rfloor \tag{4}$$

A Markovian Petri net system (MPN) is a particular time stochastic interpretation [18, 1, 4], in which the time to fire a transition t_i follows an exponentially distributed function with parameter $\lambda_i \cdot Enab(t_i, \boldsymbol{M})$, where λ_i is the firing rate associated to t_i. More formally, a MPN is a tuple $\langle \mathcal{N}, \boldsymbol{M}_0, \boldsymbol{\lambda} \rangle$, where $\boldsymbol{\lambda} \in \mathbb{R}_{>0}^{|T|}$ is the vector of rates associated to the transitions.

Given a bounded and ergodic MPN system, the steady state throughput of a transition t_i, denoted as $\chi_{MPN}(t_i)$, provides a meaningful measure for its long-term performance. It is defined as the limit of the average number of times t_i fires per time unit when time tends to infinity [8]:

$$\chi_{MPN}(t_i) = \lim_{\tau \to \infty} \frac{\sigma_i(\tau)}{\tau} \tag{5}$$

where τ is the time variable and $\sigma_i(\tau)$ is the firing count of transition t_i at time instant τ.

2.2 Continuous Petri nets

The main difference between continuous and discrete PN is in the firing amounts and consequently in the marking, which in discrete PN are restricted to be in the naturals, while in continuous PN are relaxed into the non-negative real numbers. Thus, a *continuous* PN system (CPN) is understood as a relaxation of a *discrete* one.

A continuous PN system is a tuple $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ where $\mathcal{N}$ is the net structure (as defined for discrete PN) and $\boldsymbol{m}_0 \in \mathbb{R}_{\geq 0}^{|P|}$ is the initial marking. The enabling degree of a continuous transition t_i at marking $\boldsymbol{m}$ is defined as:

$$enab(t_i, \boldsymbol{m}) = \min_{p_j \in {}^\bullet t_i} \left\{ \frac{\boldsymbol{m}[p_j]}{\boldsymbol{Pre}[p_j, t_i]} \right\} \geq Enab(t_i, \boldsymbol{m}) \tag{6}$$

The firing of t_i in a certain *real* amount $\alpha \leq enab(t_i, \boldsymbol{m})$ leads to a new marking $\boldsymbol{m}'$ that satisfies $\boldsymbol{m}' = \boldsymbol{m} + \alpha \cdot \boldsymbol{C}[P, t_i]$. Notice that in contrast to discrete PN, a continuous transition can fire if all its input places are positively marked, i.e., $enab(t_i, \boldsymbol{m}) > 0$, regardless of the input arc weights. Its set of reachable markings is denoted as $RS_C(\mathcal{N}, \boldsymbol{m}_0)$ [21]. And its LRS coincides with the LRS of the discrete system. It holds that:

$$LRS(\mathcal{N}, \boldsymbol{M}_0) \supseteq RS_C(\mathcal{N}, \boldsymbol{M}_0) \supseteq RS_D(\mathcal{N}, \boldsymbol{M}_0) \tag{7}$$

As in discrete PN, the equation $\boldsymbol{m} = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}$ summarizes the system evolution. The derivative of this equation with respect to time is $\dot{\boldsymbol{m}} = \boldsymbol{C} \cdot \boldsymbol{f}$ where $\boldsymbol{f} = \dot{\boldsymbol{\sigma}}$ is the vector of instantaneous flows of transitions.

A Timed Continuous Petri Net (TCPN) is a continuous PN together with a vector $\boldsymbol{\lambda} \in \mathbb{R}_{>0}^{|T|}$ defining the speed associated to transitions, denoted as $\langle \mathcal{N}, \boldsymbol{m}_0, \boldsymbol{\lambda} \rangle$. One of the most used semantics is *infinite server* semantics (ISS) (proved to be specially interesting for engineering applications [17]). Moreover, *product semantics* may be also considered [22] for population dynamics or (bio)chemistry, and *finite server semantics* has been also considered in some works [10]. Alternatively, stochastic time interpretations are proposed in the literature such as Stochastic Continuous PN [25].

In the sequel, ISS is considered. According to ISS, the flow through a continuous timed transition t_i is defined as follows:

$$f_i = \lambda_i \cdot enab(t_i, \boldsymbol{m}) = \lambda_i \cdot \min_{p_j \in {}^{\bullet}t_i} \left\{ \frac{\boldsymbol{m}[p_j]}{\boldsymbol{Pre}[p_j, t_i]} \right\} \tag{8}$$

If there exists a steady state in a TCPN system, the throughput of t_i, denoted as $\chi_{TCPN}(t_i)$, is equal to its steady state flow f_i [21]:

$$\chi_{TCPN}(t_i) = \lim_{\tau \to \infty} f_i(\tau) \tag{9}$$

where $f_i(\tau)$ is the flow of transition t_i at time instant τ.

2.3 Deterministic limit of MPN

The *deterministic limit* of a system [16] describes the trajectory towards which the population densities of a discrete Markovian system converge as its size tends to infinity. Let us consider a MPN with initial marking $\boldsymbol{M}_0 = k \cdot \boldsymbol{\mu}_0 \in \mathbb{N}_{\geq 0}^{|P|}$ where $\boldsymbol{\mu}_0 \in \mathbb{R}_{\geq 0}^{|P|}$ represents the initial marking density of the system, and $k \in \mathbb{R}_{>0}$ represents the relative system size (or volume).

The vector field for place p_j is defined as $F_j(\boldsymbol{\mu}) = \sum_{t_i \in ({}^{\bullet}p_j \cup p_j{}^{\bullet})} \boldsymbol{C}[p_j, t_i] \cdot f_i$, where $f_i = \lambda_i \cdot enab(t_i, \boldsymbol{\mu})$ (notice that F_j is a nonnegative function of real arguments on the system densities). Let $F(\boldsymbol{\mu})$ be a vector composed of the vector field functions $F_j(\boldsymbol{\mu})$ of every place p_j. The two following conditions can be easily checked:

a) $F(\boldsymbol{\mu})$ is Lipschitz continuous, i.e., $\exists H \geq 0$ s.t. $|F(\boldsymbol{\mu}) - F(\boldsymbol{\nu})| \leq H \cdot |\boldsymbol{\mu} - \boldsymbol{\nu}|$;
b) $\sum_{t_i \in ({}^{\bullet}p_j \cup p_j{}^{\bullet})} |\boldsymbol{C}[p_j, t_i]| \cdot f_i(\boldsymbol{\mu}) < \infty$.

Then, the deterministic limit behaviour of the marking densities $\boldsymbol{\mu}$ of the MPN when k tends to infinity is given by the following set of differential equations [12, 16]: $\dot{\boldsymbol{\mu}} = F(\boldsymbol{\mu}) = \boldsymbol{C} \cdot \boldsymbol{f}$.

Thus, the deterministic limit of a MPN matches with the time evolution defined for TCPN. Therefore, a TCPN faithfully captures the behaviour of a MPN with "infinitely" large markings.

3 Transformations on the untimed discrete PN: addition of cutting implicit places

By fluidization, *spurious* markings of a discrete PN system may become reachable in the continuous one [22]. Some transformations of the net system are proposed here to avoid those markings, thus obtaining a more faithful approximation.

The *spurious* markings can be either *integer* or *not*, and it is specially interesting to avoid them when they are deadlocks. Two techniques have been proposed in the literature to avoid *integer* spurious markings. The first one, considered in [23, 21], avoids markings in which a trap is emptied by adding a polynomially calculable implicit place. Because a trap cannot be emptied in a discrete PN system, the avoided marking is *spurious* in the discrete system. The second technique [14] proposes to avoid a marking that empties a *siphon* by adding a place. However, it cannot assure that the *avoided marking* was spurious (otherwise stated, the added place is implicit). Thus, this technique can only be applied if it is previously known to be a *spurious* solution.

Among *non-integer* spurious markings which can be removed, those which are vertices of $LRS(\mathcal{N}, \boldsymbol{m}_0)$ are particularly interesting. Some classical works aim to remove the *non-integer* vertices of a polytope, such as the Gomory-Chvàtal cuts. Given a polytope on the reals, they *cut* the markings outside the *integer hull* of the polytope [3, 9]. This method could be used to remove undesired non integer spurious markings. Although Gomory cuts are tractable for a given set of equations, finding a good family of cuts in the general case requires further investigation [9].

We propose to implement some polynomial time *cuts* on the polytope defined by $LRS(\mathcal{N}, \boldsymbol{M}_0)$, considering the PN structure. Those *cuts* aim to avoid spurious markings, and they are obtained by means of *implicit* places which force some marking relations. We propose three different kinds of *implicit* places to avoid such non-integer vertices of the polytope: *vertex cutting* places avoid those vertices which are non-integer; *marking truncation* places are a particular case but more efficient to compute; *enabling truncation* places do not modify the set of reachable markings, but they can modify the firing amounts of the transitions.

A place p is said to be an *implicit place* if it does not constrain the behaviour of the discrete system.

Definition 1 Given a PN system $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$:

- A place p is *implicit* in $\langle \mathcal{N}, m_0 \rangle$ if it is never the unique place that prevents the firing of a transition.
- A place p is *structural implicit* in $\mathcal{N}$ if there exists $\boldsymbol{m}_0$ for which p is implicit.

A characterization of the structural implicit places is given in [23]:

Proposition 1 *Let* $\mathcal{N} = \langle P \cup \{p\}, T, \boldsymbol{Pre}, \boldsymbol{Post} \rangle$. *Place* p *is structurally implicit iff (equivalently):*

1. *A* $\boldsymbol{y} \geq \mathbf{0}$ *exists such that* $\boldsymbol{C}[p,T] \geq \boldsymbol{y}^T \cdot \boldsymbol{C}[P,T]$
2. *No* $\boldsymbol{x} \geq \mathbf{0}$ *exists such that* $\boldsymbol{C}[P,T] \cdot \boldsymbol{x} \geq \mathbf{0}$ *and* $\boldsymbol{C}[p,T] \cdot \boldsymbol{x} < \mathbf{0}$

Here, we refer to *concurrent implicit* places, which preserve not only the firing sequences, but also the steps [23, 15]. The removal of a *concurrent implicit* place allows the timed performance measures to be preserved.

Proposition 2 *Given a net system* $\langle \mathcal{N}, \boldsymbol{M}_0 \rangle$*, and* $\langle \mathcal{N}', \boldsymbol{M}'_0 \rangle$ *the same net system without place* p*, then* p *is a* concurrent implicit *place [15] if* $\boldsymbol{M}_0[p] > \gamma - 1$*, where* γ *can be computed as:*

$$\begin{aligned}\gamma = min\{\mathbf{y}^T \cdot \boldsymbol{m}'_0 + \mu |\ & \mathbf{y}^T \cdot \boldsymbol{C}' \leq \boldsymbol{C}[p,T] \\ & \mathbf{z}^T \cdot \boldsymbol{Pre}'[P', p^\bullet] + \mu \cdot \mathbf{1}^T \geq \boldsymbol{Pre}[p, p^\bullet] \\ & \mathbf{y} \geq \mathbf{z} \geq \mathbf{0}, \mu \leq 0\}\end{aligned} \tag{10}$$

If p is implicit in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ as a continuous system, then p is also implicit in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ as a discrete system. Assume p_k is not implicit in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$ considered as a discrete system. Then, there exists a marking $\boldsymbol{m}$ reachable with discrete firings at which p_k constraints the enabling of an output transition t. Since $\boldsymbol{m}$ is also reachable in the system as continuous, p_k is not implicit in the continuous system either.

3.1 Vertex cutting place

The aim of this technique is to *cut* non-integer vertices of $LRS(\mathcal{N}, \boldsymbol{M}_0)$, that are not reachable on the discrete model. Let us explain the technique through an example, before introducing the method in a formal way.

Consider the example in Fig. 1(a) without the grey place, v. As discrete, it is deadlock-free and it has four reachable markings: $\boldsymbol{M}_0 = (1,0,1,0,0)$, $\boldsymbol{M}_1 = (0,1,0,3,0)$, $\boldsymbol{M}_2 = (0,0,1,0,3)$, and $\boldsymbol{M}_3 = (0,0,1,1,1)$. The polytope defined by $LRS(\mathcal{N}, \boldsymbol{M}_0)$ is the convex set defined by those vertices and $(0,0,1,1.5,0)$. As continuous, the deadlock $\boldsymbol{m}_d = (0,0,1,1.5,0) \in LRS(\mathcal{N}, \boldsymbol{M}_0)$ is reachable (by firing t_2 followed by $1.5t_3$ from $\boldsymbol{m}_0$). Thus, the deadlock-freeness property is lost. Marking $\boldsymbol{m}_d$ is outside its "integer hull", so it is not reachable in the discrete PN (in particular, $\boldsymbol{m}_d \notin \mathbb{N}^{|P|}$).

Consider the marking $\boldsymbol{m}_d$ which is a vertex and it is not reachable. Then, there exists at least two other markings which are vertices and in which at least one of the places which are empty at $\boldsymbol{m}_d$ (i.e., p_1, p_2 and p_5) are marked (with a marking equal or greater than 1, in the discrete model), otherwise it would not be a vertex. Hence, we can assure that the following inequality holds for every discrete reachable marking: $\boldsymbol{m}[p_1] + \boldsymbol{m}[p_2] + \boldsymbol{m}[p_5] \geq 1$. This inequality can be forced by the addition of a place v which is *implicit* in the discrete (but not in the continuous) PN: $\boldsymbol{m}[p_1] + \boldsymbol{m}[p_2] + \boldsymbol{m}[p_5] - \boldsymbol{m}[v] = 1$. From this equation, place v is defined as $\boldsymbol{C}[v,T] = \boldsymbol{C}[p_1,T] + \boldsymbol{C}[p_2,T] + \boldsymbol{C}[p_5,T]$, and $\boldsymbol{m}_0[v] =$

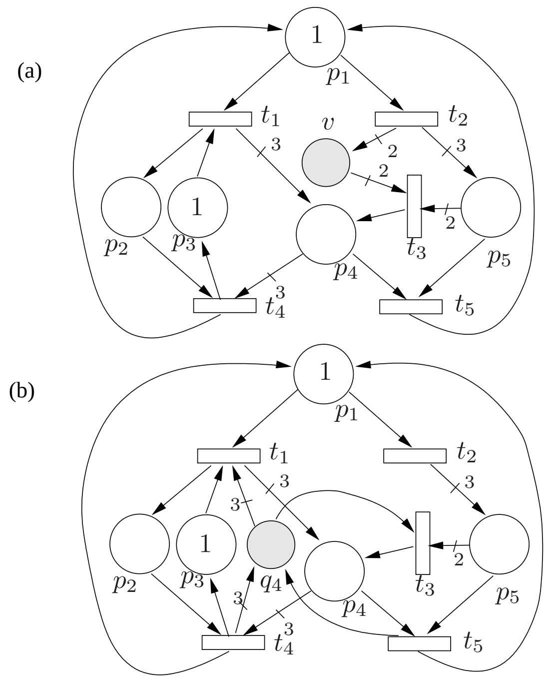


Fig. 1 (a) The *vertex cutting* place v *cuts* the *spurious* deadlock $\boldsymbol{m}_d = (0, 0, 1, 1.5, 0)$. (b) The *marking truncation* place q_4 does not *cut* $\boldsymbol{m}_d$.

$\boldsymbol{m}_0[p_1] + \boldsymbol{m}_0[p_2] + \boldsymbol{m}_0[p_5] - 1$, as depicted in Fig. 1(a). Place v, here denoted as *vertex cutting* place, adds the invariant $\boldsymbol{m}[p_1]+2\cdot\boldsymbol{m}[p_3]+\boldsymbol{m}[p_4]+\boldsymbol{m}[v] = 3$ to the net, and $\boldsymbol{m}_d$ becomes not reachable in the CPN. In this example, the added place v leads to a continuous system which preserves the deadlock-freeness property of the original discrete one.

Definition 2 Vertex cutting place. Given a non-integer vertex $\boldsymbol{m}_v$, a vertex cutting place v is the place which forces the following relation:

$$\sum_{i|\boldsymbol{m}_v[p_i]=0} \boldsymbol{m}[p_i] \geq 1.$$

The obtained *vertex cutting* place v is implicit in the discrete PN and it cuts the marking $\boldsymbol{m}_v$ in the continuous system.

More formally, given a PN system $\langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}; \boldsymbol{m}_0\rangle$ and a marking $\boldsymbol{m}_v$, the system $\langle P', T', \boldsymbol{Pre}', \boldsymbol{Post}'; \boldsymbol{m}_0'\rangle$ resulting of adding vertex v can be obtained by Algorithm 1.

Input: PN system $\langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}; \boldsymbol{m}_0\rangle$, marking $\boldsymbol{m}_v$
Output: PN system $\langle P', T', \boldsymbol{Pre}', \boldsymbol{Post}'; \boldsymbol{m}_0'\rangle$
1) $P' = P \cup \{v\}$;
2) $T' = T$;
3) $\boldsymbol{Pre}'[P, T] = \boldsymbol{Pre}[P, T]$;
4) $\boldsymbol{Pre}'[v, T] = \sum_{i|\boldsymbol{m}_v[p_i]=0} \boldsymbol{Pre}[p_i, T]$;
5) $\boldsymbol{Post}'[P, T] = \boldsymbol{Post}[P, T]$;
6) $\boldsymbol{Post}'[v, T] = \sum_{i|\boldsymbol{m}_v[p_i]=0} \boldsymbol{Post}[p_i, T]$;
7) $\boldsymbol{m}_0'[P] = \boldsymbol{m}_0[P]$;
8) $\boldsymbol{m}_0'[v] = \sum_{i|\boldsymbol{m}_v[p_i]=0} \boldsymbol{m}_0[v] - 1$;

Algorithm 1: Addition of vertex cutting place v.

Checking if a given solution $\boldsymbol{m}_v$ is a vertex of a polytope can be done in polynomial time (as explained in Appendix A). However, enumerating all the vertices of the polytope is computationally costly [2].

Let us consider a spurious deadlock $\boldsymbol{m}_d$ (the continuous enabling degree of all transitions at $\boldsymbol{m}_d$ is 0; i.e., $\boldsymbol{m}_d$ is a deadlock in the continuous PN). If $\boldsymbol{m}_d$ is a vertex of $LRS(\mathcal{N}, \boldsymbol{M}_0)$, it can be removed with the technique presented in this section. If $\boldsymbol{m}_d$ is not a vertex of $LRS(\mathcal{N}, \boldsymbol{M}_0)$, it must be a convex combination of two or more vertices of the $LRS(\mathcal{N}, \boldsymbol{M}_0)$. Notice that the null components of $\boldsymbol{m}_d$ are also null in such vertices, and that at least one of such vertices is not reachable (if every vertex is reachable, then every linear combination, and in particular $\boldsymbol{m}_d$, is also reachable [22]). Therefore, at least one of such vertices is a spurious deadlock that can be removed with the presented technique.

The repeated execution of this procedure, i.e., the removal of spurious deadlock vertices, reduces the size of $LRS(\mathcal{N}, \boldsymbol{M}_0)$ and, therefore, produces a better approximation of the discrete PN.

3.2 Marking truncation places

This subsection introduces a particular class of *vertex cutting* places, which is called *marking truncation* places, that can be computed and added very efficiently. Because a marking of a discrete PN system belongs to $\mathbb{N}^{|P|}$, a place p_j will never have more tokens than its *structural enabling bound* $SB(p_j)$. However, this bound can be overlooked by the continuous system.

The addition of an implicit place q_j is proposed, which is a complementary place of p_j which truncates the marking of p_j to the highest possible integer, and consequently, it limits the firing of the transitions in ${}^\bullet p_j$ and $p_j{}^\bullet$ (in the continuous system) and avoids undesired markings.

Definition 3 Given a place p_j, its marking truncation place q_j is obtained as: $\boldsymbol{Pre}[q_j, T] = \boldsymbol{Post}[p_j, T]$ and $\boldsymbol{Post}[q_j, T] = \boldsymbol{Pre}[p_j, T]$. Its initial marking is obtained as: $\boldsymbol{m}_0[q_j] = SB(p_j) - \boldsymbol{m}_0[p_j]$.

Given a PN system $\langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}; \boldsymbol{m}_0\rangle$, their marking truncation places can be added by Algorithm 2.

Input: PN system $\langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}; \boldsymbol{m}_0\rangle$.
Output: PN system $\langle P', T', \boldsymbol{Pre}', \boldsymbol{Post}'; \boldsymbol{m}'_0\rangle$

1) $P' = P$; $T' = T$;
2) $\boldsymbol{Pre}'[P, T] = \boldsymbol{Pre}[P, T]$; $\boldsymbol{Post}'[P, T] = \boldsymbol{Post}[P, T]$;
3) $\boldsymbol{m}'_0[P] = \boldsymbol{m}_0[P]$;
4) **for every** $p_j \in P$;
5) $\quad P' = P' \cup \{q_j\}$;
6) $\quad \boldsymbol{Pre}'[q_j, T] = \boldsymbol{Post}[p_j, T]$;
7) $\quad \boldsymbol{Post}'[q_j, T] = \boldsymbol{Pre}[p_j, T]$;
8) $\quad \boldsymbol{m}'_0[q_j] = SB(p_j) - \boldsymbol{m}_0[p_j]$ where $SB(p_j)$ is obtained from (3)

Algorithm 2: Addition of marking truncation places.

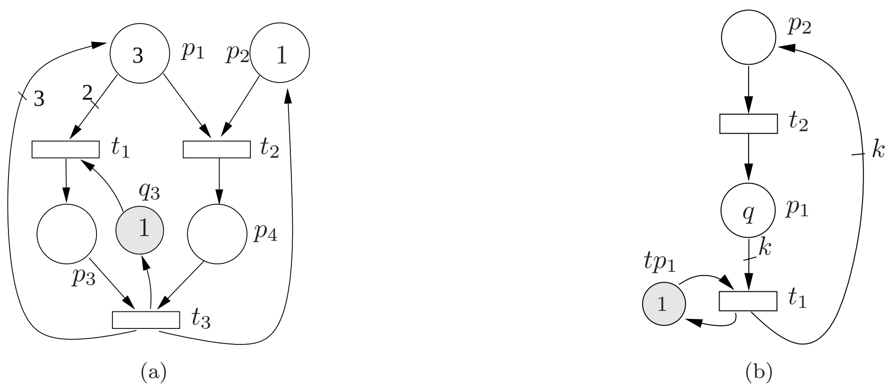


Fig. 2 (a) The *marking truncation* place q_3 removes the undesired *spurious* marking $\boldsymbol{m}_d = (0, 1, 1.5, 0)$. (b) With $k < q < 2 \cdot k$, the *enabling truncation* place tp_1 improves the throughput approximation, but does not change $RS_C(\mathcal{N}, \boldsymbol{m}_0)$. If $q = k$, then t_1 suffers the BRP (see Section 4).

For example, consider the PN in Fig. 2(a), which is also deadlock-free as discrete but not as continuous. The *marking truncation* place q_3 is added, which is *complementary* to p_3, whose initial marking is $\boldsymbol{M}_0[q_3] = \lfloor 1.5 \rfloor - 0 = 1$. It is implicit in the discrete system, but it modifies the $LRS(\mathcal{N}, \boldsymbol{M}_0)$. It avoids having more than 1 token in p_3, hence it avoids the spurious deadlock $\boldsymbol{m}_d = (0, 1, 1.5, 0)$, in which $\boldsymbol{m}_d[p_3] = 1.5$ is higher than $SB(p_3) = 1$. The resulting system is deadlock-free as continuous. Moreover, the addition of q_3, with $\boldsymbol{m}_0[q_3] = 1$, makes the timed approximation of the original system as MPN more accurate. The improvement is not significant when $\lambda_1 = \lambda_2$ (see first column of Table 1), but it is specially relevant when $\lambda_1 > \lambda_2$ (see the second column of the table), because the steady state of the TCPN is near to $\boldsymbol{m}_d$.

Table 1 $\chi(t_1)$ of the PN in Fig. 2(a), with different $\boldsymbol{\lambda}$.

Method	$\chi(t_1)$ $\boldsymbol{\lambda} = (5, 5, 1)$	$\chi(t_1)$ $\boldsymbol{\lambda'} = (5, 1, 1)$
MPN	0.769	0.492
TCPN	0.833	0.025
TCPN+impl	0.833	0.499

The addition of every *marking* truncation place to the system can be done in polynomial time (because computing $SB(p)$ is polynomial). However, this technique does not always obtain the "integer hull" of the polytope. For example, consider again the PN system in Fig. 1(b). The *marking truncation* place q_4 is added to limit the structural bound of p_4 $SB(p_4)$. However, $\boldsymbol{m}_d$ is still reachable by the continuous PN by firing $t_2, 1.5t_3$ from $\boldsymbol{m}_0$ (the rest of places q_j would not avoid it either), and the addition of the *vertex cutting* place is needed.

3.3 Enabling truncation places

Finally, the *enabling truncation* places do not modify the $LRS(\mathcal{N}, \boldsymbol{M}_0)$, but they limit the flow at a given marking if time is considered. They can have an effect on the throughput, even if all the $SB(p)$ are integer (the marking truncation place would have no effect).

Analogously to the marking of a place, a transition t_i will never be fired in an amount higher than its *structural enabling bound* $SEB(t_i)$. The addition of an implicit place tp_i is proposed. It would truncate the maximal possible firing of the transition to the highest possible integer.

Definition 4 Enabling truncation place. Given a transition t_i, its enabling truncation place tp_i is a self-loop place of transition t_i whose initial marking is $\boldsymbol{m}_0[tp_i] = SEB(t_i)$.

Given a PN system $\langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}; \boldsymbol{m}_0 \rangle$, their enabling truncation places can be added by Algorithm 3.

Input: PN system $\langle P, T, \boldsymbol{Pre}, \boldsymbol{Post}; \boldsymbol{m}_0 \rangle$.
Output: PN system $\langle P', T', \boldsymbol{Pre}', \boldsymbol{Post}'; \boldsymbol{m}'_0 \rangle$
1) $P' = P$; $T' = T$;
2) $\boldsymbol{Pre}'[P,T] = \boldsymbol{Pre}[P,T]$; $\boldsymbol{Post}'[P,T] = \boldsymbol{Post}[P,T]$;
3) $\boldsymbol{m}'_0[P] = \boldsymbol{m}_0[P]$;
4) **for every** $t_i \in T$;
5) $P' = P' \cup \{tp_i\}$;
6) $\boldsymbol{Pre}'[tp_i, T \setminus t_i] = 0$;
7) $\boldsymbol{Pre}'[tp_i, t_i] = 1$;
8) $\boldsymbol{Post}'[tp_i, T \setminus t_i] = 0$;
9) $\boldsymbol{Post}'[tp_i, t_i] = 1$;
10) $\boldsymbol{m}'_0[tp_i] = SEB(t_i)$ where $SEB(t_i)$ is obtained from (4)

Algorithm 3: Addition of enabling truncation places.

Consider the PN in Fig. 2(b) with $k = 5$, $q = 8$ and $\boldsymbol{\lambda} = (1, 5)$. The structural enabling bound of t_1 is $SEB(t_1) = \lfloor 1.6 \rfloor = 1$. Hence, an *enabling truncation* place tp_1 with initial marking equal to 1 is added (see place tp_1, drawn in grey colour). The throughput of t_1 in the discrete PN is $\chi_{MPN}(t_1) = 0.90$. The *concurrent implicit* place tp_1 *forces* the continuous system to be more faithful to the original discrete system and produces the throughput $\chi_{TCPN+impl}(t_1) = 1.00$, which is a better approximation than the throughput of the original continuous net system $\chi_{TCPN}(t_1) = 1.33$. Only tp_1 has been added because place tp_2 would not affect to the behaviour of the continuous PN.

4 The bound reaching problem (BRP)

TCPN under ISS approximate reasonably well the behaviour of MPN when the population is relatively large, as pointed out in Section 2.3. However, the *Bound Reaching Problem* (BRP) identifies a particular but important situation in which the quality of the approximation may be significantly worse [13].

Definition 5 A transition t_i in $\langle \mathcal{N}, \boldsymbol{M}_0 \rangle$ is said to suffer from the BRP if $SEB(t_i) = 1$. The set of transitions suffering from the BRP in a PN system $\langle \mathcal{N}, \boldsymbol{M}_0 \rangle$ is denoted as *Bound Reaching Transition Set*: $BRTS = \{t_i \mid SEB(t_i) = 1\}$.

The differences between discrete and continuous behaviour are due to the fact that *synchronizations* (arc *weights* and therefore *joins*) are strongly relaxed when the net system is fluidified. Consider transition t_1 in Fig. 2(b) with $q = k$ (and without tp_1). Thus, $SEB(t_1) = 1$. Seen as discrete, t_1 is only enabled when $\boldsymbol{M}[p_1] = k$. However, as continuous, t_1 is enabled for any positive amount of tokens $0 < \boldsymbol{m}[p_1] \leq k$, regardless of the arc weight k.

Considering the net system as a MPN, the firing time distribution of t_1 from $\boldsymbol{m}$ follows an exponential probability function with parameter $\lambda_1 \cdot \lfloor \frac{\boldsymbol{m}[p_1]}{k} \rfloor$. After the first firing of t_1, only t_2 can be fired, and it keeps on firing until the

Table 2 Throughput of t_1 in Fig. 2(b), with $k = q$, and $\boldsymbol{\lambda} = (10, 1)$.

Method	$k = 1$	$k = 2$	$k = 4$	$k = 8$	k=16
$\chi_{MPN}(t_1)$	0.909	0.517	0.420	0.355	0.287

k tokens are again at p_1. Then, t_1 becomes enabled again. If k increases, the probability of having k tokens in p_1 decreases, and also its steady state throughput. Its average *cycletime* is obtained from order statistics as $\Theta = \frac{1}{\lambda_1} + \frac{1}{\lambda_2} \cdot \sum_{i=1}^{k} \frac{1}{i}$, and its mean throughput is $\frac{1}{\Theta}$, i.e., $\chi_{MPN}(t_1) = \lambda_2 / (\sum_{i=1}^{k} \frac{1}{i} + \frac{\lambda_2}{\lambda_1})$.

Notice that $\chi_{MPN}(t_1)$ is the product of a dimensionless coefficient depending on k and $\frac{\lambda_2}{\lambda_1}$, multiplied by λ_2 that defines a *time scale* (time homothecy [22]). Thus, we can normalize $\lambda_2 = 1$, and the normalized $\chi_{MPN}(t_1)$ depends on k and λ_1. The steady state throughput of t_1 for different values of k is shown in Table 2 for $\boldsymbol{\lambda} = (10, 1)$.

Considered as a TCPN, t_1 is enabled for any marking $\boldsymbol{m}[p_1] > 0$. Moreover, the firing of t_1, and hence the behaviour of the system, is not modified by k. The throughput of t_1 as a TCPN is *time homothetic* (i.e., its steady state flow is proportional to $\boldsymbol{\lambda}$), and it is equal to: $\chi_{TCPN}(t_1) = \lambda_2 / (1 + \frac{\lambda_2}{\lambda_1})$.

Considering $\boldsymbol{\lambda} = (10, 1)$, $\chi_{TCPN}(t_1) = 0.909$ for any value of k. The continuous throughput coincides with the discrete one for $k = 1$, but it provides a bad approximation for $k > 1$ (see Table 2), which gets worse when k grows.

In order to overcome this lack of accuracy due to the BRP, different techniques can be investigated. They can range from fully continuous to hybrid. A first possibility is to modify the flow of the continuous transition. For example, an "ad hoc" continuous flow estimation is explained in [13] as an alternative to ISS. However, it has some disadvantages such as no *time homothecy*.

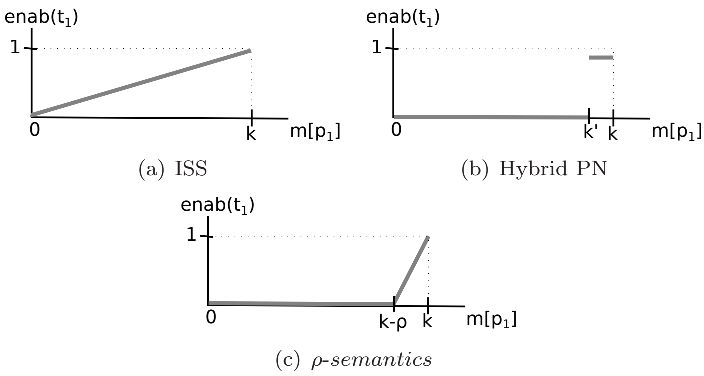


Fig. 3 Enabling degree of t_1 in Fig. 2(b) with $q = k$. (a) TCPN under ISS (continuous and differentiable flow); (b) Hybrid PN (see Fig.4(a)), t_1 discrete and approximately scaled arcs; (c) ρ-*semantics* (piecewise function, based on ISS).

5 A new semantics to approach the bound reaching problem

The aim of this section is to propose an alternative fluidization technique to tackle the BRP. The resulting method will be a deterministic approach based on ISS.

Consider again the PN in Fig. 2(b), with $q = k$. A key difference between the behaviour of the MPN and the TCPN under ISS is that in the MPN, t_1 can fire only when the k tokens are in p_1; while in the continuous case, it is not needed to "wait until the k tokens" are in p_1 to fire t_1.

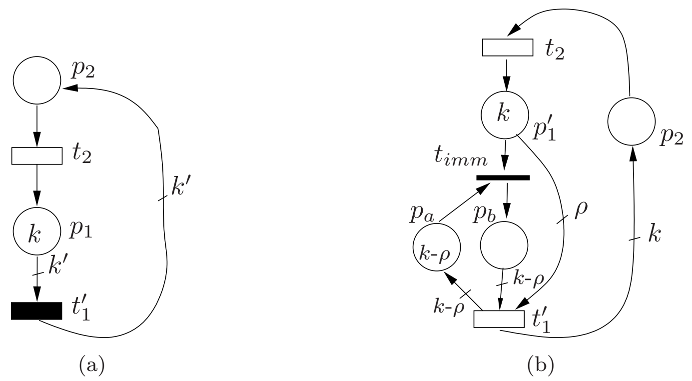


Fig. 4 (a) Hybrid PN system in which t'_1 is discrete (black transition) and t_2 is continuous, arc weights are modified to k', (b)Transformation of the PN in Fig. 2(b), with $q = k$. Transitions t_1 and t_2 are continuous under ISS, and t_{imm} is *immediate* (thin black transition).

In a TCPN, waiting until p_1 has k tokens would take infinite time. Hence, it makes sense to wait until some other smaller value, such as $k - \rho$ (where ρ comes from "the rest"). This behaviour can be obtained by transforming p_1-t_1 (see Fig. 2(b)) to a subnet composed of p'_1, t'_1, p_a, p_b, t_{imm} (see Fig. 4(b)), such that t'_1 is not enabled for "the first" $k - \rho$ tokens, and it is enabled for higher amounts.

Immediate transitions are difficult to handle in TCPN [19]. A first approximation can be to consider immediate transitions as timed ones which are several orders of magnitude faster than the other transitions (for example, $\lambda_{imm} = 10000$ if $\boldsymbol{\lambda} = (10, 1)$). However, this has some disadvantages: If λ_{imm} is relatively not very high, then the steady state might not be the desired one; while for very high relative values of λ_{imm}, *stiffness* problems would appear.

In this particular construction, we can abstract the structure given by p'_1, t'_1, p_a, p_b, t_{imm} by a unique transition which compacts the desired behaviour, resulting in the denoted *ρ-semantics*, based on ISS (see [13] and Fig. 3(c) for more details). Its flow is given by:

$$f_1{=}\lambda_1 \cdot \begin{cases} 0 & \text{if } \boldsymbol{m}[p_1] \leq \boldsymbol{Pre}[p_1,t_1]\text{-}\rho \\ \frac{\boldsymbol{m}[p_1]\text{-}(\boldsymbol{Pre}[p_1,t_1]\text{-}\rho)}{\rho} & \text{otherwise} \end{cases} \tag{11}$$

The transient flow of t_1 is still a continuous function but piecewise defined, which introduces certain "hybridization" in the behaviour of the transition (see Fig. 3(c)). If ρ tends to 0, the flow tends to a step function of values from 0 to λ_1, while if ρ tends to $\boldsymbol{Pre}[p_1, t_1]$, the flow tends to ISS. The computation done by this approach is local to transition t_1 (the transition in $BRTS$), and it is simple and fast to calculate. It inherits some basic properties of ISS, such as homothetic monotonicity w.r.t. the firing rates.

With the proposed ρ-*semantics*, the throughput of the system can be "tuned" from 0 (when $\rho \sim 0$) to the throughput of the TCPN (when ρ is "equal" to $\boldsymbol{Pre}[p_1, t_1]$). The challenge is how to select ρ to approximate the steady state throughput of the MPN.

Let us first compute ρ for the PN in Fig. 2(b) with $q = k$, and then apply that *heuristics* on any PN system with a similar structure. The throughput of t_1 at the steady stated can be symbolically computed, considering ρ-*semantics* for the flow of t_1 (i.e., equation (11), with ρ as a parameter), and ISS for t_2 (i.e., equation (8)). Because of its p-semiflow, it holds $\boldsymbol{m}[p_1] + \boldsymbol{m}[p_2] = k$. At the steady state, the equality $\boldsymbol{C} \cdot \boldsymbol{f}_{ss} = \boldsymbol{0}$ must be satisfied, and hence $\chi_\rho(t_2) = k \cdot \chi_\rho(t_1)$. From these equalities, the value of $\chi_\rho(t_1)$ for the PN in Fig.2(b) with $q = k$ is:

$$\chi_\rho(t_1) = \lambda_2 \cdot \frac{\rho}{k + \rho \cdot \frac{\lambda_2}{\lambda_1}} \tag{12}$$

Forcing $\chi_{MPN}(t_1) = \chi_\rho(t_1)$, an analytical formula for the value of ρ is obtained, that fortunately depends only on k (not on $\boldsymbol{\lambda}$):

$$\rho = \frac{k}{\sum_{i=1}^{k} \frac{1}{i}} \tag{13}$$

The PN model in Fig. 2(b) can be seen as a simplification of any net system with analogous structure. Hence, it will be possible to use the heuristic formula (13) in other transitions suffering from the BRP such that $|{}^\bullet t| = 1$.

6 Generalization of the ρ-*semantics* to join transitions

As introduced in Section 5, the ρ-*semantics* has been designed for transitions with a unique input arc. The aim of this section is to generalize it to transitions with more than one input place (*rendez-vous* or *join* transitions).

First, linear enabling functions for *join* transitions are presented. Then, the addition of the *representative* places is introduced. Finally, the application of the ρ-*semantics* to *join* transitions is proposed.

6.1 Linear enabling functions

Linear Enabling Functions (LEF) were introduced for discrete PN in [24, 7] to characterize the enabling of a transition by a single linear expression.

The enabling of a transition t_i can be represented with a single LEF if every $p \in {}^\bullet t_i$ except at most one (denoted as π), satisfies the equality $SB(p) = \boldsymbol{Pre}[p, t_i]$, i.e,

$$\exists \pi \in {}^\bullet t_i \text{ s.t. } \forall p \in \{{}^\bullet t_i \setminus \pi\}, SB(p) = \boldsymbol{Pre}[p, t_i] \tag{14}$$

A particular case appears if every place $p \in {}^\bullet t_i$ satisfies $SB(p) = \boldsymbol{Pre}[p, t_i]$. If $SB(p) > \boldsymbol{Pre}[p, t_i]$ holds for more than one of its input places p, its enabling cannot be directly represented by a single LEF, and some previous transformations of the PN should be done (see [24]).

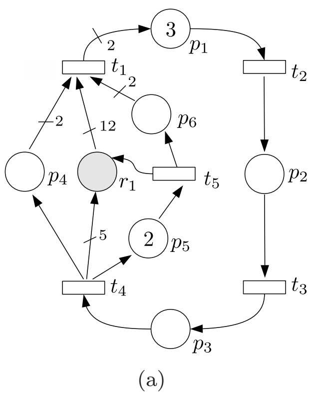
Given a transition t_i which holds equation (14) it will be enabled when:

$$\boldsymbol{M}[\pi] + SB(\pi) \cdot \sum_{p \in \{{}^\bullet t_i \setminus \pi\}} \boldsymbol{M}[p] \geq \boldsymbol{Pre}[\pi, t_i] + SB(\pi) \cdot \sum_{p \in \{{}^\bullet t_i \setminus \pi\}} \boldsymbol{Pre}[p, t_i] \tag{15}$$

Consider the PN in Fig. 5(a) (without the grey place, r_1). Transition t_1 holds equation (14) with $\pi = p_6$ (because $SB(p_6) = 5 > \boldsymbol{Pre}[p_6, t_1] = 2$, $\{{}^\bullet t_1 \setminus \pi\} = \{p_4\}$, and $SB(p_4) = \boldsymbol{Pre}[p_4, t_1] = 2$). The enabling of t_1 (in the discrete system) is given by: $\boldsymbol{M}[p_6] + SB(p_6) \cdot \boldsymbol{M}[p_4] \geq \boldsymbol{Pre}[p_6, t_1] + SB(p_6) \cdot \boldsymbol{Pre}[p_4, t_1]$.

6.2 Representative places for *join* transitions

The LEF of a discrete transition can be represented in the PN system with an implicit place which is *representative* of its enabling [7].



(a)

Method	$\chi(t_1)$
MPN	0.396
TCPN	0.484
ρ-semantics	0.410

(b)

Fig. 5 (a)The grey place r_1 is the *representative place* of transition t_1. The ρ-*semantics* is applied to transition t_1. (b) Steady state throughput of t_1, with $\boldsymbol{\lambda} = (10, 1, 1, 1, 1)$.

Given a PN system $\langle \mathcal{N}, \boldsymbol{M}_0 \rangle$ which satisfies (14), a representative place r_i can be added to the system, which is computed as a linear non-negative combination of the places in ${}^\bullet t_i$. Place r_i is built as:

$$\boldsymbol{C}[r_i, T] = \boldsymbol{C}[\pi, T] + SB(\pi) \cdot \sum_{p \in \{{}^\bullet t_i \setminus \pi\}} \boldsymbol{C}[p, T] \tag{16}$$

And its initial marking is computed as:

$$\boldsymbol{M}_0[r_i] = \boldsymbol{M}_0[\pi] + SB(\pi) \cdot \sum_{p \in \{{}^\bullet t_i \setminus \pi\}} \boldsymbol{M}_0[p] \tag{17}$$

Consider again t_1 in Fig. 5(a). The *representative* place r_1 is added, computed as $\boldsymbol{C}[r_1, T] = \boldsymbol{C}[p_6, T] + SB(p_6) \cdot \boldsymbol{C}[p_4, T]$, where $SB(p_6) = 5$ (see the grey place r_1 in Fig.5(a)). Its initial marking is computed as $\boldsymbol{M}_0[r_1] = \boldsymbol{M}_0[p_6] + SB(p_6) \cdot \boldsymbol{M}_0[p_4] = 0$. In general, the added place r_i (the marking of r_i) is a non-negative linear combination of the places in ${}^\bullet t_i$ (their markings). Hence, r_i will never be the unique place constraining the enabling degree of t_i, and it is implicit in the discrete and the continuous systems.

The original places in ${}^\bullet t_i$ become implicit in the discrete PN system when r_i is added. However, they do not become implicit in the continuous one. For example, once the implicit place r_1 has been added in Fig. 5(a), p_4 and p_6 become implicit in the discrete net system. However, $\boldsymbol{m}'_1 = (3, 0, 0, 0, 1, 1, 0)$ is reachable in $\langle \mathcal{N}, \boldsymbol{m}_0 \rangle$, and the enabling degree of t_1 is restricted by p_4 at $\boldsymbol{m}'_1$ (and not by r_1). Analogously, place p_6 does not constrain the enabling degree of t_1 in the reachable marking $\boldsymbol{m}'_2 = (2, 0, 0, 1, 3, 0, 5)$. Hence, p_4 is not implicit in the continuous system and p_4, p_6 cannot be removed.

6.3 Generalization of the ρ-*semantics*

The flow of the *join* transition t_i is defined by using the ρ-*semantics* for place r_i, and ISS for the rest of the places in ${}^\bullet t_i$.

The value of ρ_i is computed from ρ as defined in equation (13). If place r_i was obtained as a linear combination of those in ${}^\bullet t_i$, then ρ_i is obtained with the same linear combination of $\rho(\boldsymbol{Pre}[p_i, t_j])$ for the places $p_j \in {}^\bullet t_i$, where $\rho(\boldsymbol{Pre}[p_i, t_j])$ is computed from equation (13), with $k = \boldsymbol{Pre}[p_i, t_j]$.

Given a representative place r_i which has been obtained as $\boldsymbol{C}[r_i, T] = \boldsymbol{C}[\pi, T] + SB(\pi) \cdot \sum_{p \in \{{}^\bullet t_i \setminus \pi\}} \boldsymbol{C}[p, T]$, then ρ_i is obtained as follows:

$$\begin{aligned} \rho_i = \rho(\boldsymbol{Pre}[\pi, t_i]) + SB(\pi) \cdot \sum_{p \in \{{}^\bullet t_i \setminus \pi\}} \rho(\boldsymbol{Pre}[p, t_i]) = \\ \frac{\boldsymbol{Pre}[\pi, t_i]}{\sum_{n=1}^{\boldsymbol{Pre}[\pi, t_i]} \frac{1}{n}} + SB(\pi) \cdot \sum_{p \in \{{}^\bullet t_i \setminus \pi\}} \frac{\boldsymbol{Pre}[p, t_i]}{\sum_{n=1}^{\boldsymbol{Pre}[p, t_i]} \frac{1}{n}} \end{aligned} \tag{18}$$

The flow of the transition (defined in equation (11)) is generalized below to *join* transitions, in which *an analogous term* is used for the *representative place*, and ISS is used for the rest of the places in ${}^{\bullet}t_i$:

$$f_i{=}\lambda_i \cdot \begin{cases} 0 & \text{if } \boldsymbol{m}[r_i] \leq \boldsymbol{Pre}[r_i,t_i]\text{-}\rho_i \\ min\{\frac{\boldsymbol{m}[r_i]\text{-}(\boldsymbol{Pre}[r_i,t_i]\text{-}\rho_i)}{\rho_i}, \underset{p\in{}^{\bullet}t_i}{min}\{\frac{\boldsymbol{m}[p]}{\boldsymbol{Pre}[p,t_i]}\}\} & \text{otherwise} \end{cases} \tag{19}$$

For example, consider transition t_1 in the PN in Fig.5(a). In this case, ${}^{\bullet}t_1 = \{p_1, p_2\}$, so $|{}^{\bullet}t_1| > 1$ and the ρ-*semantics* cannot be directly applied. The first step is to add the *representative place* r_1 (see Section 6.2). Then, the value of ρ_1 can be obtained from equation (18). Finally, the ρ-*semantics* proposed in equation (19) can be applied to t_1. Considering $\boldsymbol{\lambda} = (10, 1, 1, 1, 1)$, the ρ-*semantics* ($\chi_\rho(t_1) = 0.41$, see the Table in Fig. 5(b)) gives a better approximation than the TCPN ($\chi_{TCPN}(t_1) = 0.484$) to the original MPN ($\chi_{MPN}(t_1) = 0.396$).

The ρ-*semantics* has been presented here for *join* transitions, eventually with weights in the arcs. However, the behaviour of those transitions in which *joins* and *choices* are combined requires further investigation.

7 Case study

The aim of this section is to apply the techniques presented in this paper to an example from the literature. The Petri net example shown in Fig. 6 is obtained from [6]. It represents a supervisory control system for a distributed manufacturing process.

In [6], the system is modelled with UML, and then a discrete PN system is derived. The system represents a production line process, in which two components interact. The left part of the Petri net (with places and transitions labelled by B) represents a *belt*, while the right part represents some *film*. Both parts are mutually synchronized.

In this work, we modify the production line, allowing the system to produce k jobs at the same time. In order to obtain that, we allow the belt to hold k jobs, so we modify the initial marking of B_{out} to k. These jobs will be synchronized in transition B_{new}, so also the weight of the arcs around this transition are modified to k. Moreover, k jobs will be wrapped by using the film (right part of the figure) at the same time. The k tokens in $Fout$ are synchronized in transition F_{new}.

Let us consider the computation of the steady state throughput of transition B_{new}. It can be efficiently done for low values of k for the discrete PN, however, the state explosion problem raises for bigger values of k. For example, the Markov chain obtained for $k = 1$ has 16 states, for $k = 3$ it has 673 states, and for $k = 5$ it grows to 7322 states. It grows exponentially for bigger values of k. Hence, the fluidization of the system is interesting in this case.

Consider a small value of k such as $k = 3$, and $\boldsymbol{\lambda} = (1, 1, 1, 1, 10, 1, 1, 1, 1, 10)$. In this case, $\chi_{SPN}(B_{new}) = 0.356$ for the MPN. This value is not well ap-

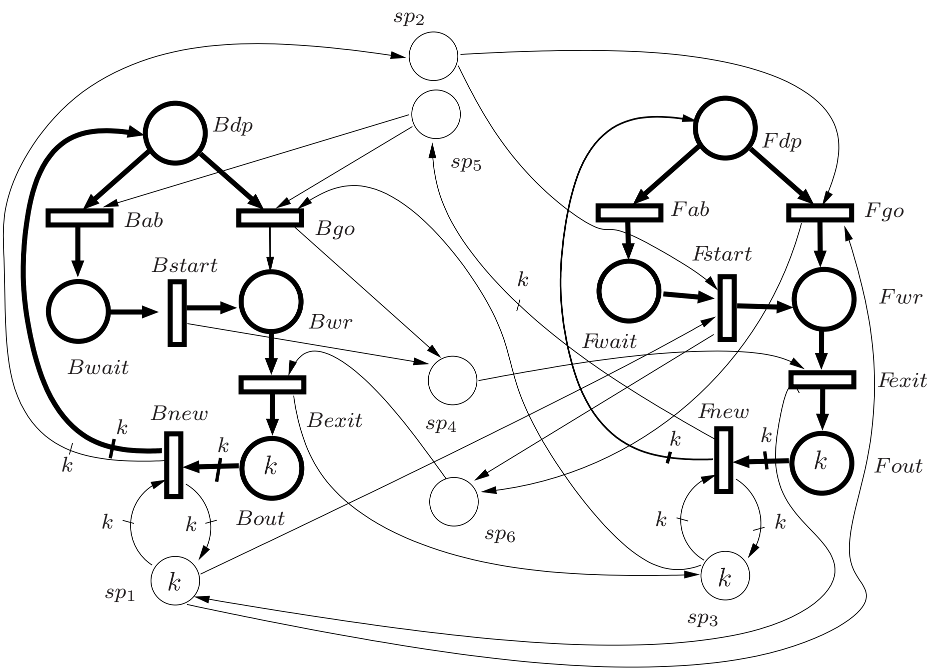


Fig. 6 Case study. Petri net which models a supervisory control system (obtained from [6]).

proximated by the continuous PN, $\chi_{ISS}(B_{new}) = 0.714$, but it is well approximated by applying the ρ-*semantics* to transitions B_{new}, in which the obtained throughput is $\chi_\rho(B_{new}) = 0.398$. A relative error (computed as $|\chi_{SPN} - \chi_\rho|/\chi_{SPN}$) of 11.8% is obtained. This result was obtained in 3.32 seconds.

Table 3 Steady state throughput of transition F_{new} in the PN in Fig. 6.

Method	χ_{SPN}	χ_{ISS}	χ_ρ	error
$k = 3$	0.356	0.714	0.398	11.8%
$k = 10$	0.860	2.381	0.839	2.4%
$k = 100$	6.010	23.810	4.774	20.6%

Moreover, it is not only a good approximation of small populations, but also when the parameter k grows, dealing to 'relatively small' populations. Consider for example $k = 100$, in which $\chi_{SPN}(B_{new}) = 6.010$. The approximation given by the ρ-*semantics* ($\chi_\rho(B_{new}) = 4.774$) is much better than the value obtained by the continuous PN under ISS ($\chi_{ISS}(B_{new}) = 23.810$). See Table 3 for an overview of results for the case study.

8 Conclusions

Fluidization aims to reduce the analysis complexity of discrete PN models. Unfortunately, some logical and performance properties of the discrete PN may be lost by fluidization, specially when the system population is “relatively” small. In this paper, two sorts of techniques have been proposed to improve the fluid approximation of the discrete model.

The relaxation of the formalism entails the reachability of some *spurious* markings (i.e., non reachable in the discrete PN) in the continuous model. The removal of *spurious* markings, and in particular the removal of spurious deadlocks, in the fluidified model has been proposed. The technique is based on the addition of *cutting* places which are *implicit* in the discrete PN (they do not modify the behaviour of the original discrete system) but not in the continuous one. This yields a more faithful approximation to the original discrete system, for both the untimed and timed interpretations.

Moreover, we have studied the *Bound Reaching Problem* (BRP), which may appear in timed systems, and that causes the throughput of the original discrete PN system and the continuous approximation to be particularly different. A new method based on ISS, which is denoted ρ-*semantics*, has been proposed to tackle the BRP. Such novel semantics inherits basic properties from ISS such as *time homothecy*, not marking homothecy, and it is specially accurate when it is applied to transitions with a single input place. Moreover, the technique has been generalized to *join* transitions by means of *representative* places.

Acknowledgements

This work has been partially supported by CICYT - FEDER grant DPI2014-57252-R of the Spanish Government, the European Social Fund and the grant T27 to the GISED research group and the pre-doctoral grant B174/11, both of the Gobierno de Aragón.

References

1. M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. Wiley, 1995.
2. D. Avis and K. Fukuda. A pivoting algorithm for convex hulls and vertex enumeration of arrangements and polyhedra. *Discrete & Computational Geometry*, 8(1):295–313, 1992.
3. E. Balas, S. Ceria, G. Cornujols, and N. Natraj. Gomory cuts revisited. *Operations Research Letters*, 19(1):1 – 9, 1996.
4. G. Balbo and M. Silva, editors. *Performance Models for Discrete Event Systems with Synchronozations: Formalisms and Analysis Techniques, MATCH Performance Advanced School*, Jaca, Spain, 1998. Available on-line: http://webdiis.unizar.es/GISED/?q=news/matchbook.

5. M. Beccuti, E. Bibbona, A. Horváth, R. Sirovich, A. Angius, and G. Balbo. Analysis of Petri Net Models through Stochastic Differential Equations. In G. Ciardo and E. Kindler, editors, *Application and Theory of Petri Nets and Concurrency*, volume 8489 of *LNCS*, pages 273–293. Springer, 2014.
6. B. Bordbar, L. Giacomini, and D. J. Holding. Design of Distributed Manufacturing Systems using UML and Petri Nets. In *Proceedings of 6th International Federation of Automatic Control (IFAC)*, Workshop on Algorithms and Architectures for Real-Time Control, pages 182–196, Palma de Mallorca, Spain, May 2000.
7. J. L. Briz, J. Colom, and M. Silva. Simulation of Petri nets and linear enabling functions. In *IEEE Int. Conf. on Systems, Man, and Cybernetics*, volume 2, pages 1671–1676 vol.2, 1994.
8. 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.
9. G. Cornuejols. The ongoing story of Gomory cuts. *Documenta Mathematica - Optimization Stories*, pages 221–226, 2012.
10. R. David and H. Alla. *Discrete, Continuous and Hybrid Petri Nets*. Springer, Berlin, 2004. (2^{nd} edition, 2010).
11. F. DiCesare, G. Harhalakis, J. M. Proth, M. Silva, and F. Vernadat. *Practice of Petri Nets in Manufacturing*. Chapman & Hall, 1993.
12. S. Ethier and T. Kurtz. *Markov Processes: Characterization and Convergence*. John Wiley, 1986.
13. E. Fraca, J. Júlvez, and M. Silva. The Bound Reaching Problem on the fluidization of timed Petri nets. In *11th Int. Workshop on Discrete Event Systems (WODES'14)*, pages 142–148, Paris, France, 2014.
14. E. Fraca, J. Júlvez, and M. Silva. On the fluidization of Petri nets and marking homothecy. *Nonlinear Analysis: Hybrid Systems*, 12:3 – 19, 2014.
15. F. García-Vallés and J. Colom. Implicit Places in Net Systems. In *Petri Nets and Performance Models (PNPM'99)*, pages 104–113. IEEE Computer Society Press, 1999.
16. J. Jacod and A. Shiryaev. *Limit Theorems for Stochastic Processes*. Springer, 2002.
17. C. Mahulea, L. Recalde, and M. Silva. Basic server semantics and performance monotonicity of continuous Petri nets. *Discrete Event Dynamic Systems*, 19(2):189–212, 2009.
18. M. K. Molloy. Performance Analysis Using Stochastic Petri Nets. *IEEE Trans. on Computers*, 31(9):913–917, 1982.
19. L. Recalde, C. Mahulea, and M. Silva. Improving analysis and simulation of continuous Petri nets. In *2nd IEEE Conf. on Automation Science and Engineering*, pages 7–12, Shanghai, China, 2006.
20. M. Silva. Individuals, populations and fluid approximations: A Petri net based perspective. *Nonlinear Analysis: Hybrid Systems*, 22(11):72–97, 2016.

21. M. Silva, J. Júlvez, C. Mahulea, and C. R. Vázquez. On fluidization of discrete event models: observation and control of continuous Petri nets. *Discrete Event Dynamic Systems: Theory and Applications*, 21(4):427–497, 2011.
22. M. Silva and L. Recalde. Petri nets and integrality relaxations: A view of continuous Petri net models. *IEEE Trans. on Systems, Man, and Cybernetics*, 32(4):314–327, 2002.
23. M. Silva, E. Teruel, and J. M. Colom. Linear Algebraic and Linear Programming Techniques for the Analysis of Net Systems. *Lecture Notes in Computer Science*, 1491:309–373, 1998.
24. E. Teruel, J. M. Colom, and M. Silva. Linear Analysis of Deadlock-Freeness of Petri Net Models. In *Procs. of the* 2^{nd} *European Control Conf.*, volume 2, pages 513–518. North-Holland, 1993.
25. C. R. Vázquez and M. Silva. Stochastic continuous Petri nets: An approximation of Markovian net models. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 42(3):641–653, 2012.

A Checking if a marking is a vertex of the potential reachability set

A marking $\boldsymbol{m}_v \in LRS(\mathcal{N}, \boldsymbol{M}_0)$ is said to be a vertex if it is not a linear combination of markings in $LRS(\mathcal{N}, \boldsymbol{M}_0)$, i.e. $\nexists\ \boldsymbol{m}_1, \boldsymbol{m}_2 \in LRS(\mathcal{N}, \boldsymbol{M}_0)$ with $\boldsymbol{m}_1 \neq \boldsymbol{m}_2$ and $\alpha, \beta > 0$ such that $\boldsymbol{m}_v = \alpha \boldsymbol{m}_1 + \beta \boldsymbol{m}_2$. Hence, any vertex is a minimal support vector (although the reverse is not true).

A polynomial time method is presented below to check if a given solution of the state equation is a vertex. Let us define Ψ as a subset of linearly independent places in P such that $rank(\boldsymbol{C}[\Psi, T]) = rank(\boldsymbol{C}[P, T])$.

Proposition 3 *A solution $\boldsymbol{m}_v \in LRS(\mathcal{N}, \boldsymbol{m}_0)$ is a vertex iff $\forall p_i \in \Psi$, v_i is equal to 0, where v_i is defined by the following Linear Programming Problem (LPP):*

$$\begin{aligned} v_i = max\ & (\boldsymbol{m}_v[p_i] - \boldsymbol{m}_1[p_i]) \\ s.t.\ & \boldsymbol{m}_1 = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}_1, \boldsymbol{m}_1, \boldsymbol{\sigma}_1 \geq \boldsymbol{0} \\ & \boldsymbol{m}_2 = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}_2, \boldsymbol{m}_2, \boldsymbol{\sigma}_2 \geq \boldsymbol{0} \\ & \boldsymbol{m}_v = 0.5 \cdot \boldsymbol{m}_1 + 0.5 \cdot \boldsymbol{m}_2 \end{aligned} \tag{20}$$

Proof. ($\Rightarrow$) Assume $\boldsymbol{m}_v$ is a vertex. Thus, $\nexists \boldsymbol{m}_1, \boldsymbol{m}_2$ such that $\boldsymbol{m}_1 \neq \boldsymbol{m}_2$ and $\boldsymbol{m}_v = 0.5 \cdot \boldsymbol{m}_1 + 0.5 \cdot \boldsymbol{m}_2$. Therefore, $\forall p_j \in P$, $\boldsymbol{m}_1 = \boldsymbol{m}_2 = \boldsymbol{m}_v$ and $v_j = 0$. Hence,$\forall p_i \in \Psi$, $v_i = 0$. ($\Leftarrow$) Assume $\boldsymbol{m}_v$ is not a vertex. Then, there exist two interchangeable solutions $\boldsymbol{m}_1, \boldsymbol{m}_2 \in LRS(\mathcal{N}, \boldsymbol{m}_0)$ such that $\boldsymbol{m}_1 \neq \boldsymbol{m}_2$ and $\boldsymbol{m}_v = 0.5 \cdot \boldsymbol{m}_1 + 0.5 \cdot \boldsymbol{m}_2$. Hence, there exists at least a place $p_j \in P$ such that $\boldsymbol{m}_1[p_j] < \boldsymbol{m}_v[p_j] < \boldsymbol{m}_2[p_j]$. If $p_j \in \Psi$, directly $v_j > 0$. Otherwise, and because $rank(\boldsymbol{C}[\Psi, T]) = rank(\boldsymbol{C}[P, T])$, $\exists p_i \in \Psi$ which linearly depends on p_j, such that $\boldsymbol{m}_1[p_i] \neq \boldsymbol{m}_v[p_i] \neq \boldsymbol{m}_2[p_i]$. And hence, $v_i > 0$.

Proposition 3 can be checked in polynomial time. Because (A.1) is a LPP, it is of polynomial complexity. The decision procedure is based in the solution of $|\Psi|$ LPP, from which the first phase of the classical simplex approach is common (observe that only the objective function changes).

If $\mathcal{N}$ is consistent, the number of variables and constraints can be reduced, by replacing $\boldsymbol{m}_k = \boldsymbol{m}_0 + \boldsymbol{C} \cdot \boldsymbol{\sigma}_k$ by $B \cdot \boldsymbol{m}_k = B \cdot \boldsymbol{m}_0$, for $k = \{1, 2\}$, where B is a basis of *p-flows* of $\mathcal{N}$

[23]. Then, v_i is defined as:

$$\begin{aligned} v_i = max\ & (\boldsymbol{m}_v[p_i] - \boldsymbol{m}_1[p_i]) \\ s.t.\ \ & B \cdot \boldsymbol{m}_1 = B \cdot \boldsymbol{m}_0, \boldsymbol{m}_1 \geq \mathbf{0} \\ & B \cdot \boldsymbol{m}_2 = B \cdot \boldsymbol{m}_0, \boldsymbol{m}_2 \geq \mathbf{0} \\ & \boldsymbol{m}_v = 0.5 \cdot \boldsymbol{m}_1 + 0.5 \cdot \boldsymbol{m}_2 \end{aligned} \tag{21}$$