Stationary behavior of controlled Generalized Batches Petri Nets

Isabel Demongodin *, Alessandro Giua **
LSIS, Aix-Marseille University, France (isabel.demongodin@lsis.org)
DIEE, University of Cagliari, Italy (giua@diee.unica.it)

Abstract

In this paper we consider controlled Generalised Batches Petri Nets, a class of batches nets where the firing flow of continuous and batch transitions and the transfer speed of batch places are control variables to be selected within an admissible range. We show that the steady state behavior of this class of nets during a period in which no discrete transition fires can be characterized by solving a nonlinear programming problem that takes into account the net structure and the initial marking. A viable technique to compute a family of solutions by linear relaxation is also presented.

Published as:

I. Demongodin, A. Giua, "Stationary behavior of controlled Generalised Batches Petri Nets," WODES12: 11th Int. Work. on Discrete Event Systems (Guadalajara, Mexico), Oct 2012.

This work has been partially supported by: the European Community's Seventh Framework Programme under project DISC (Grant Agreement n. INFSO-ICT-224498); the *Visiting Professors 2011* program of the University of Cagliari sponsored by the Autonomous Region of Sardinia.

I. INTRODUCTION

In the last years several researchers have been working on the fluidization of Petri nets models, thus extending the Petri net formalism to encompass continuous and hybrid models (1). A recent survey of this area was presented by (2).

There are a few motivations behind this effort. Firstly, it is well known that discrete event models suffer from the state explosion problem, i.e., the number of states grows exponentially with the number of composed subsystems — in the case or modular systems — or with the number of individuals that compose the population — in the case of models that describe a population dynamics. Fluid models are viable means to bypass this issue. Secondly, in a fluid model gradient techniques can be applied to perform sensitivity or perturbation analysis. These techniques have been successfully applied to fluid-queueing networks (see (3) and references therein) and to Petri nets (4; 5; 6).

Many man-made systems contain continuous transfer elements that introduce variable delays depending on traffic intensity. These elements cannot be represented in detail by standard fluid and hybrid Petri net models. This led to the definition of batches Petri nets (7), that extend the hybrid Petri net model with primitives to represent a *batch*, i.e., a group of entities moving through a transfer zone at a certain speed and the corresponding notion of *batch node*.

In a previous paper (8) we considered Generalised Batches Petri Nets (GBPNs) as defined by (7) and developed new linear algebraic techniques for the computation of instantaneous firing flows of transitions and the analysis of the steady state behaviors of this model.

In (8) we associated to GBPNs a semantics inspired by First Order Hybrid Petri Nets (4) and assumed that the instantaneous firing flow of continuous and batch transitions are control variables that can take an arbitrary value provided appropriate constraints are satisfied. These constraints may be structural, e.g., a transition flow cannot exceed the maximal firing flow, or behavioral, e.g., the total flow exiting an empty place cannot be greater than the input flow. In this framework the steady state (SS) behavior of GBPNs is described by a pair $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s)$ where $\boldsymbol{m}^s$ is a constant marking and $\boldsymbol{\varphi}^s$ is a constant vector of instantaneous firing flows. We showed that for nets with only continuous and batch nodes an SS can be characterized solving a linear programming problem that takes into account the net structure and the initial marking. Related work on the steady state analysis of continuous nets can be found in (9; 10; 11; 12).

In this paper we further extend this analysis considering a model where also the transfer speed of batch places — that describe the evolution of transfer elements such as conveyor belts — is a control variable. This adds an additional degree of freedom when modeling the delays incurred in traversing such elements. The resulting model is called controlled Generalised Batches Petri Nets (cGBPN). The steady state behavior of cGBPNs is described by a triple $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s, \boldsymbol{v}^s)$ where the additional element $\boldsymbol{v}^s$ is a constant vector of transfer speeds associated to batch places. Such a steady state can be characterized as the solution of a programming problem that contains nonlinear constraints. We shows that by relaxing the nonlinear constraints a relaxed solution can be obtained and from it a family of solutions of the original nonlinear program. We also discuss the notion of optimal solution by means of a detailed numerical example.

As a final remark, we point out that this paper is concerned with the characterization of an SS for a net with only continuous and batch nodes, but these results also apply to nets with discrete nodes between two firings of discrete transitions. However, if the time interval between the two firings is too short, an SS may not be reachable.

The paper is structured as follows. In Section II, the basic definitions of controlled Generalized Batches Petri Nets are presented, including the enabling and firing rules for transitions and the description of the hybrid dynamics of batches. Section III proposes a linear programming problem to compute the instantaneous firing flow and transfer speed vectors. Section IV is dedicated to the computation of steady states for such a model and to the discussion of a detailed example.

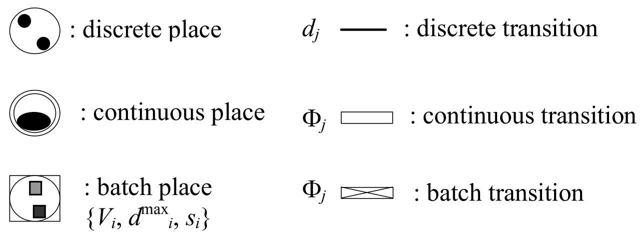

Fig. 1. Nodes of a cGBPN.

II. Background on batches Petri nets

A. Basic definitions

The following definition slightly extends the notion of GBPN introduced in (7) and (8).

***Definition** 2.1:* A *controlled Generalized Batches Petri net* (cGBPN) is a 6-tuple $N = (P, T, Pre, Post, \gamma, Time)$ where:

- $P = P^D \cup P^C \cup P^B$ is finite set of places partitioned into the three classes of *discrete*, *continuous* and *batch* places.
- $T = T^D \cup T^C \cup T^B$ is finite set of transitions partitioned into the three classes of *discrete*, *continuous* and *batch* transitions.
- $Pre, Post : (P^D \times T \to \mathbb{N}) \cup ((P^C \cup P^B) \times T \to \mathbb{R}_{\geq 0})$ are[1], respectively, the pre-incidence and post-incidence matrixes, denoting the weight of the arcs from places to transitions and transitions to places.
- $\gamma : P^B \to \mathbb{R}^3_{>0}$ is the *batch place function*. It associates to each batch place $p_i \in P^B$ the triple $\gamma(p_i) = (V_i, d_i^{\max}, s_i)$ that represents, respectively, *maximal transfer speed*, *maximal density* and *length* of p_i.
- $Time : T \to \mathbb{R}_{\geq 0}$ associates a non negative number to every transition:
  - if $t_j \in T^D$, then $Time(t_j) = d_j$ denotes the *firing delay* associated to the discrete transition;
  - if $t_j \in T^C \cup T^B$, then $Time(t_j) = \Phi_j$ denotes the *maximal firing flow* associated to the continuous or batch transition. ■

We denote the number of places and transitions, resp., $m = |P|$ and $n = |T|$. We also use the following notations: $m^X = |P^X|$ and $n^X = |T^X|$ for $X \in \{D, C, B\}$. The *preset* and *postset* of transition t_j are: ${}^\bullet t_j = \{p_i \in P \mid Pre(p_i, t_j) > 0\}$ and $t_j^\bullet = \{p_i \in P \mid Post(p_i, t_j) > 0\}$. Similar notations may be used for pre and post transition sets of places and its restriction to discrete, continuous or batch transitions is denoted as ${}^{(d)}p_i = {}^\bullet p_i \cap T^D$, ${}^{(c)}p_i = {}^\bullet p_i \cap T^C$, and ${}^{(b)}p_i = {}^\bullet p_i \cap T^B$. The *incidence matrix* of the net is defined as $\mathbf{C} = Post - Pre$.

***Definition** 2.2:* A *batch* β_r at time τ, is defined by a triple, $\beta_r(\tau) = (l_r(\tau), d_r(\tau), x_r(\tau))$, where $l_r(\tau) \in \mathbb{R}_{\geq 0}$ is the *length*, $d_r(\tau) \in \mathbb{R}_{\geq 0}$ is the *density* and $x_r(\tau) \in \mathbb{R}_{\geq 0}$ is the *head position*. ■

A batch place contains a series of batches, ordered by their head positions and moving forward at the same speed.

The state of a cGBPN is represented by its marking.

***Definition** 2.3:* The *marking* of a GBPN at time τ is defined as $\boldsymbol{m}(\tau) = [m_1(\tau) ... m_i(\tau) ... m_n(\tau)]^T$, where:

- if $p_i \in P^D$ then $m_i \in \mathbb{N}$, i.e., the marking of a discrete place is a non negative integer.
- if $p_i \in P^C$ then $m_i \in \mathbb{R}_{\geq 0}$, i.e., the marking of a continuous place is a non negative real.
- if $p_i \in P^B$ then $m_i = \{\beta_h, ..., \beta_r\}$, i.e., the marking of a batch place is a series of batches.

■

A similar definition is the following.

[1] We denote $\mathbb{R}_{\geq 0}$ (resp., $\mathbb{R}_{>0}$) the set of non negative (resp., positive) real numbers.

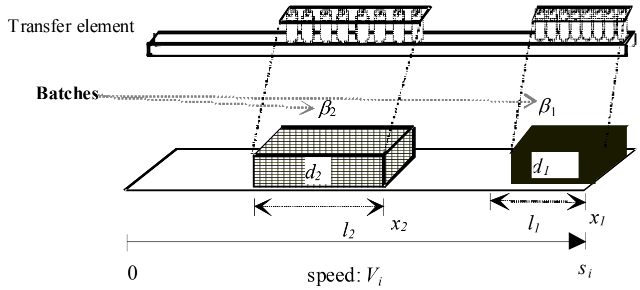

Fig. 2. Batches.

***Definition** 2.4:* The *marking quantity vector* $\boldsymbol{q} = \mu(\boldsymbol{m}) \in \mathbb{R}^m$ associated to a marking $\boldsymbol{m}$ is defined as follows:

$$q_i = \begin{cases} m_i & \text{if } p_i \in P^D \cup P^C \\ \sum_{\beta_r \in m_i} l_r \cdot d_r & \text{if } p_i \in P^B \end{cases},$$

i.e., the marking quantity coincides with the marking for discrete and continuous place, while for a batch place it represents the sum of the quantities of the batches it contains. ■

We denote $\boldsymbol{m}_0 = \boldsymbol{m}(\tau_0)$ the initial marking. When time can be omitted, we denote the marking as $\boldsymbol{m}$.

***Definition** 2.5:* Let $\beta_r(\tau) = (l_r(\tau), d_r(\tau), x_r(\tau)) \in m_i(\tau)$ be a batch in place $p_i \in P^B$, with $\gamma(p_i) = (V_i, d_i^{\max}, s_i)$.
β_r is called an *output batch* if its head position is equal to the length associated to the batch place, i.e., $x_r(\tau) = s_i$.
A batch is said to be *dense* if its density is equal to the maximal density of batch place p_i, $d_r(\tau) = d_i^{\max}$.
The *output density* d_i^{out} of batch place p_i is defined as follows. If at time τ, place p_i has an output batch $\beta_r(\tau)$, then $d_i^{\text{out}}(\tau) = d_r(\tau)$, else $d_i^{\text{out}}(\tau) = 0$. ■

***Definition** 2.6:* The *maximal capacity* of batch place $p_i \in P^B$, with $\gamma(p_i) = (V_i, d_i^{\max}, s_i)$, is $Q_i = s_i \cdot d_i^{\max}$. A place such that $q_i(\tau) = Q_i$ is called a *full batch place*. ■

To every continuous and batch transition $t_j \in T^C \cup T^B$ is associated an *instantaneous firing flow* (IFF) $\varphi_j(\tau) \leq \Phi_j$, representing the quantity of markings by time unit that fires transition t_j. An input flow and an output flow are defined for batch place p_i at time τ such as:

- $\phi_i^{\text{in}}(\tau) = \sum_{t_j \in {}^\bullet p_i} Post(p_i, t_j) \cdot \varphi_j(\tau)$.
- $\phi_i^{\text{out}}(\tau) = \sum_{t_j \in p_i^\bullet} Pre(p_i, t_j) \cdot \varphi_j(\tau)$.

To every batch place $p_i \in P^B$ is associated an *instantaneous transfer speed* (ITS) $v_i(\tau) \leq V_i$, representing the speed of place p_i at time τ. Note that here we are extending the interpretation of cGBPN used in (7) where the transfer speed V_i associated to a batch place p_i by function γ represents the constant speed of the place.

The instantaneous flows and transfer speeds should be considered as control inputs that drive the evolution of the system; Section III will show how to determine admissible control inputs.

The dynamics of a cGBPN is ruled by the firing of its enabled transitions and by a hybrid dynamics inside batch places, that produce a change in the marking. The enabling and firing conditions of discrete transitions are those of classical transition-timed discrete Petri nets. The enabling conditions of continuous transitions are those of First Order Hybrid Petri nets (4), i.e., one distinguishes weakly and strongly enabled transitions. Let us now focus on the hybrid dynamics of batch nodes.

B. Hybrid dynamics of batch places

We define for batch transitions similar conditions of continuous transitions, i.e., weakly and strongly enabled transitions.

***Condition** 2.7:* A batch transition $t_j \in T^B$ is enabled at $\boldsymbol{m}$ if:

- $\forall p_i \in {}^{(d)}t_j$, $m_i \geq Pre(p_i, t_j)$.
- $\forall p_s \in {}^{(b)}t_j$, $d_s^{\text{out}} > 0$.

We say that the batch transition is:

- *strongly enabled* if $\forall p_k \in {}^{(c)}t_j$, $m_k > 0$.
- *weakly enabled* if $\exists p_r \in {}^{(c)}t_j$, $m_r = 0$. ■

Batch places describe the transfer of batches according to a switching dynamics between two behaviors: the free behavior and the accumulation behavior. Both dynamics of a batch place are governed by the state of the batches composing it and various equations govern the evolution of batches: inputting, moving and exiting.

***Definition** 2.8 (Free behavior):* Batch $\beta_r(\tau)$ of p_i is in a *free behavior* if it moves freely at the transfer speed v_i. Three different dynamics can occur.

- *Inputting*. A created batch, $\beta_r(\tau) = (0, d_r(\tau), 0)$ freely enters in place p_i according to:

$$\dot{l}_r = v_i; \quad \dot{d}_r = 0; \quad \dot{x}_r = v_i$$

- *Moving*. A batch, $\beta_r(\tau) = (l_r(\tau), d_r(\tau), x_r(\tau))$ freely moves inside place p_i according to:

$$\dot{l}_r = 0; \quad \dot{d}_r = 0; \quad \dot{x}_r = v_i$$

- *Exiting*. An output batch, $\beta_r(\tau) = (l_r(\tau), d_r(\tau), s_i)$ freely exits from place p_i according to:

$$\dot{l}_r = -v_i; \quad \dot{d}_r = 0; \quad \dot{x}_r = 0$$

Batch place p_i is in a *free behavior* if its output batch is in a free behavior, i.e., $\phi_i^{\text{out}}(\tau) = d_i^{\text{out}}(\tau) \cdot v_i$. ■

***Definition** 2.9 (Accumulation behavior):* Batch $\beta_r(\tau)$ of p_i is in an *accumulation behavior* if it is not moving at the transfer speed of p_i. Two situations can cause this behavior.

- Let $\beta_r(\tau)$ be an output batch of p_i. If the output flow of p_i is lower than the free batch flow $d_r(\tau)/v_i$ then batch $\beta_r(\tau)$ accumulates while it exits the place.
- Let $\beta_r(\tau)$ be a batch in contact with a downstream output batch in an accumulation behavior. In this case, batch $\beta_r(\tau)$ cannot move freely at transfer speed v_i, but starts an accumulation that will be merged with the downstream dense output batch.

Batch place p_i is in an *accumulation behavior* if its output batch is in an accumulation behavior, i.e., $\phi_i^{\text{out}}(\tau) < d_i^{\text{out}}(\tau) \cdot v_i$. ■

A complete and general description of the equations that govern this behavior can be found in (13).

C. Net dynamics

The behavior algorithm of a cGBPN is based on a discrete event approach with linear or constant continuous evolutions between timed events. Between two timed events, the state of the net has an invariant behavior state (IB-state), which corresponds to a period of time such that: the marking in discrete places is constant; the instantaneous firing flow of continuous and batch transitions is constant; the output density of batch places is constant.

The IB-state changes if and only if one (or possibly several at the same time) of the following kinds of events occurs:

1) a discrete transition fires;
2) a continuous place becomes empty;
3) a discrete transition becomes enabled;
4) a batch arrives at the end of a batch place thus becoming an output batch;
5) an output batch of a batch place is destroyed.

Inside a batch place, several timed events have to be taken into account in the dynamic evolution of batches:

- a batch becomes an output batch (i.e., event 4 above);

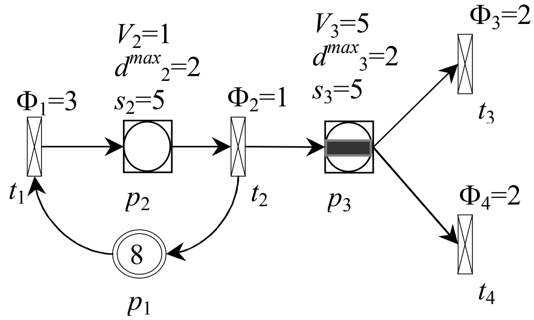

Fig. 3. Net in Example 3.1.

- two batches meet;
- a batch in accumulated behavior becomes dense;
- a batch is destroyed (for an output batch, this corresponds to event 5 above).

Finally, the state equation that governs the dynamic behavior of a cGBPN in terms of marking quantity vector is (7): $\boldsymbol{q}(\tau) = \boldsymbol{q}(\tau_0) + \boldsymbol{C} \cdot \boldsymbol{z}(\tau)$, where $\boldsymbol{z}(\tau) \in \mathbb{R}^n_{\geq 0}$, called *characteristic vector*, denotes how many times a discrete transition has fired and the quantity fired for continuous and batch transitions during $[\tau_0, \tau]$.

We denote $R(N, \boldsymbol{m}_0)$ the set of reachable markings of a cGBPN and define the *reachable marking quantity set* as

$$RQ(N, \boldsymbol{m}_0) = \{\boldsymbol{q} \mid \exists \boldsymbol{m} \in R(N, \boldsymbol{m}_0) : \ \boldsymbol{q} = \mu(\boldsymbol{m})\}.$$

A larger approximation of the *reachable marking quantity set* is the *potentially reachable marking quantity set*

$$\begin{aligned} PRQ(N, \boldsymbol{m}_0) &= \{\boldsymbol{q} \mid \exists \boldsymbol{z} \in \mathbb{R}^n_{\geq 0} : \ \boldsymbol{q} = \mu(\boldsymbol{m}_0) + \boldsymbol{C} \cdot \boldsymbol{z}\} \\ &\supseteq RQ(N, \boldsymbol{m}_0). \end{aligned}$$

Note that while $\mu(\boldsymbol{m})$ is an injective mapping, its inverse $\mu^{-1}(\boldsymbol{q})$ is not, i.e., more than one marking $\boldsymbol{m}$ may correspond to a given marking quantity vector $\boldsymbol{q}$.

III. Instantaneous firing flow and transfer speed vectors

In a previous paper (8) we have considered as control variable the instantaneous firing flow (IFF) of continuous and batch transitions. In the framework of this work, the instantaneous transfer speed (ITS) of batch places can be controlled as well. In this section we show how an admissible control input can be computed.

Let $\langle N, \boldsymbol{m} \rangle$ be a cGBPN with incidence matrix $\boldsymbol{C}$. Let

- $T_{\mathcal{E}}(\boldsymbol{m}) \subset T^C \cup T^B$ ($T_{\mathcal{N}}(\boldsymbol{m}) \subset T^C \cup T^B$) be the subset of continuous and batch transitions enabled (not enabled) at $\boldsymbol{m}$.
- $P_{\emptyset}(\boldsymbol{m}) = \{p_i \in P^C \mid m_i = 0\}$ be the subset of empty continuous places.
- $P_F(\boldsymbol{m}) = \{p_i \in P^B \mid q_i = Q_i\}$ be the subset of full bach places.

Any admissible evolution of the net at a marking $\boldsymbol{m}$, is characterized by a pair $(\boldsymbol{\varphi}, \boldsymbol{v})$ where $\boldsymbol{\varphi} : T^C \cup T^B \to \mathbb{R}_{\geq 0}$ is an *IFF vector* and $\boldsymbol{v} : P^B \to \mathbb{R}_{\geq 0}$ is an *ITS vector*. Such a pair is a feasible solution of the following linear set:

$$\begin{cases} (a) & 0 \le \varphi_j \le \Phi_j & \forall t_j \in T_{\mathcal{E}}(\boldsymbol{m}) \\ (b) & \varphi_j = 0 & \forall t_j \in T_{\mathcal{N}}(\boldsymbol{m}) \\ (c) & \sum_{t_j \in T_{\mathcal{E}}} C(p_i, t_j) \cdot \varphi_j \ge 0 & \forall p_i \in P_{\emptyset}(\boldsymbol{m}) \\ (d) & \sum_{t_j \in T_{\mathcal{E}}} C(p_i, t_j) \cdot \varphi_j \le 0 & \forall p_i \in P_F(\boldsymbol{m}) \\ (e) & \sum_{t_j \in T_{\mathcal{E}}} Post(p_i, t_j) \cdot \varphi_j \le v_i \cdot d_i^{\max} & \forall p_i \in P^B \\ (f) & \sum_{t_j \in T_{\mathcal{E}}} Pre(p_i, t_j) \cdot \varphi_j \le v_i \cdot d_i^{\text{out}} & \forall p_i \in P^B \\ (g) & 0 \le v_i \le V_i & \forall p_i \in P^B \end{cases} \tag{1}$$

The set of all feasible solutions is denoted $\mathcal{S}(N, \boldsymbol{m})$. ■

***Example** 3.1:* Consider the net in Fig. 3, where $\Phi_1 = 3$, $\Phi_2 = 1$, $\Phi_3 = \Phi_4 = 2$, $\gamma(p_2) = (V_2, d_2^{\max}, s_2) = (1, 2, 5)$ and $\gamma(p_3) = (V_3, d_3^{\max}, s_3) = (5, 2, 5)$. The initial marking is $\boldsymbol{m}_0 = [8 \ \emptyset \ \{\beta_1(0)\}]^T$ with $\beta_1(0) = (5, 2, 5)$. We remark that the output batch of place p_3 is dense. Thus, $d_3^{\text{out}}(0) = d_1(0) = d_3^{\max} = 2$ and $d_2^{\text{out}}(0) = 0$. Moreover, place p_3 is initially full as $q_3(0) = l_1(0) \cdot d_1(0) = 10 = Q_3$.

At the initial time it holds: $T_{\mathcal{E}}(\boldsymbol{m}_0) = \{t_1, t_3, t_4\}$, $T_{\mathcal{N}}(\boldsymbol{m}_0) = \{t_2\}$, $P_{\emptyset}(\boldsymbol{m}_0) = \emptyset$ and $P_F(\boldsymbol{m}_0) = \{p_3\}$, hence the IFF and ITS vectors must verify:

$$\begin{cases} (a) & 0 \le \varphi_1 \le \Phi_1 \\ (a') & 0 \le \varphi_3 \le \Phi_3 \\ (a'') & 0 \le \varphi_4 \le \Phi_4 \\ (b) & \varphi_2 = 0 \\ (d) & -\varphi_3 - \varphi_4 \le 0 \\ (e) & \varphi_1 \le v_2 \cdot d_2^{\max} \\ (e') & 0 \le v_3 \cdot d_3^{\max} \\ (f) & \varphi_3 + \varphi_4 \le v_3 \cdot d_3^{\text{out}} \\ (g) & 0 \le v_2 \le V_2 \\ (g') & 0 \le v_3 \le V_3 \end{cases}$$

Assume the priority is that of maximizing the output flow of the net ($\varphi_3 + \varphi_4$) while also requiring all other transitions to have a flow as large as possible with an additional lower priority objective: minimize the speed of batch places. This can be enforced defining as objective function to maximize $J = \varphi_3 + \varphi_4 + 0.1(\varphi_3 + \varphi_4) - 0.001(v_2 + v_3)$.

One gets solution $\boldsymbol{\varphi} = (2, 0, 2, 2)$ and $\boldsymbol{v} = (v_2, v_3) = (1, 2)$.

IV. Steady state computation of controlled GBPN

We now define the notion of steady state for cGBPN only composed by continuous and batch nodes; the same definition, however, apply to arbitrary cGBPN during a period in which no discrete transition fires.

***Definition** 4.1 (Steady State):* Let $\langle N, \boldsymbol{m}_0 \rangle$ be a cGBPN with $P^D = T^D = \emptyset$. The net is in a *steady state* (SS) at time τ_s if for $\tau \ge \tau_s$ the marking $\boldsymbol{m}^s$, the instantaneous firing flow vector $\boldsymbol{\varphi}^s$ and the instantaneous transfer speed vector $\boldsymbol{v}^s$ remain constant. Thus a steady state is defined by the triple $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s, \boldsymbol{v}^s)$.

For a GBPN with constant batch place speeds, we simply denote a steady state by the pair $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s)$, omitting the constant vector of batch place speeds. ■

One important feature of an SS is that the marking of batch places can only take a regular form, whose exact value only depends on the quantity of marking it contains. This is shown, following (8), by next proposition.

***Proposition** 4.2:* Assume that a net $\langle N, \boldsymbol{m}_0 \rangle$ with $P^D = T^D = \emptyset$ is in a steady state $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s, \boldsymbol{v}^s)$. The marking m_i^s of a batch place $p_i \in P^B$ — with input/output flow ϕ_i^s and marking quantity $q_i^s = \mu(m_i^s)$ — takes the following *regular form*:

1) *If* $\phi_i^s = 0$, marking $m_i^s = \{\beta_o\}$ contains a single dense output batch $\beta_o = (l_o, d_i^{\max}, s_i)$ with a length $l_o = q_i^s / d_i^{\max}$.
2) *If* $q_i^s = \phi_i^s s_i / v_i^s > 0$, marking $m_i^s = \{\beta_o\}$ contains a single output batch $\beta_o = (s_i, d_o, s_i)$ with a length equal to the length of the place and with a density $d_o = \phi_i^s / v_i^s$.
3) *If* $Q_i > q_i^s > \phi_i^s s_i / v_i^s > 0$, marking $m_i^s = \{\beta_e, \beta_o\}$ contains a dense output batch $\beta_o = (l_o, d_i^{\max}, s_i)$ in contact with one input batch $\beta_e = (l_e, d_e, l_e)$ such that $d_e = \phi_i^s / v_i^s$ and
$$l_e = \frac{s_i d_i^{\max} v_i^s - q_i^s v_i^s}{d_i^{\max} v_i^s - \phi_i^s} \quad \text{and} \quad l_o = s_i - l_e. \tag{2}$$
4) *If* $Q_i = q_i^s > \phi_i^s s_i / v_i^s > 0$, marking $m_i^s = \{\beta_o\}$ contains a single dense output batch $\beta_o = (s_i, d_i^{\max}, s_i)$ in accumulation behavior with a length equal to the length of the place.

Thus from $\boldsymbol{q}^s$, $\boldsymbol{\varphi}^s$ and $\boldsymbol{v}^s$ the regular marking $\boldsymbol{m}^s$ can be uniquely reconstructed; we denote this $\boldsymbol{m}^s = \nu(\boldsymbol{q}^s, \boldsymbol{\varphi}^s, \boldsymbol{v}^s)$.

Proof. Proposition 4.4 in (8) presented an equivalent result for GBPNs with constant batch place speeds; in such a case the minimum delay introduced by batch place p_i, denoted $\delta_i^{min} = s_i / V_i$, is constant. In the case of variable transfer speeds the minimum delay introduced by batch place p_i in SS depends on the place speed and is denoted s_i / v_i^s. With this minor modification, the same proof used in (8) can be used to prove the more general result stated above. □

The previous result allows one to *abstract* the marking of batch places into a simple vector of marking quantities. This was instrumental in characterizing the steady state of a GBPN with constant batch place speeds as shown in next proposition.

Proposition 4.3 ((8)): Given a GBPN $\langle N, \boldsymbol{m}_0 \rangle$ with $P^D = T^D = \emptyset$, consider the following constraint set (CS):
$$\begin{cases} (a) & \mathbf{0} \leq \boldsymbol{y} \leq \boldsymbol{\Phi} \\ (b) & Q_i \geq q_i \geq Pre(p_i, \cdot) \cdot \boldsymbol{y} \cdot \delta_i^{\min} & (\forall p_i \in P^B) \\ (c) & Post(p_i, \cdot) \cdot \boldsymbol{y} \leq V_i \cdot d_i^{\max} & (\forall p_i \in P^B) \\ (d) & \boldsymbol{C} \cdot \boldsymbol{y} = \mathbf{0} \\ (e) & \boldsymbol{q} \in RQ(N, \boldsymbol{m}_0) \end{cases} \tag{3}$$
where $\boldsymbol{q} \in \mathbb{R}^m$ and $\boldsymbol{y} \in \mathbb{R}^n$ are unknown, and all other parameters, that depend on the structure on the net, have previously been defined.

(A) If $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s)$ is a reachable steady state then $(\boldsymbol{q}, \boldsymbol{y})$ satisfies eq. (3) with $\boldsymbol{q} = \mu(\boldsymbol{m}^s)$ and $\boldsymbol{y} = \boldsymbol{\varphi}^s$.
(B) If $(\boldsymbol{q}, \boldsymbol{y})$ is a solution of eq. (3) then there exists a reachable steady state $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s)$ with $\boldsymbol{m}^s = \nu(\boldsymbol{q}, \boldsymbol{y})$ and $\boldsymbol{\varphi}^s = \boldsymbol{y}$.

■

In CS 3 this is the meaning of the different constraints. Constrains (a) follows from the definition of admissible IFF; constraint (b) follows from Little's law applied to each batch place; constraint (c) implies that the output flow of a batch place can be sustained by an output batch; constraint (d) implies that the IFF must be a T-semiflow, since the marking of the net remains constant; constraint (e) implies that steady state is reachable from the initial marking.

Here we extend this result to cGBPNs where the transfer speed of a batch place may vary.

Proposition 4.4: Given a cGBPN $\langle N, \boldsymbol{m}_0 \rangle$ with $P^D = T^D = \emptyset$, consider the following constraint set:
$$\begin{cases} (a) & \mathbf{0} \leq \boldsymbol{y} \leq \boldsymbol{\Phi} \\ (b') & Q_i \geq q_i & (\forall p_i \in P^B) \\ (c) & Post(p_i, \cdot) \cdot \boldsymbol{y} \leq v_i \cdot d_i^{\max} & (\forall p_i \in P^B) \\ (d) & \boldsymbol{C} \cdot \boldsymbol{y} = \mathbf{0} \\ (e) & \boldsymbol{q} \in RQ(N, \boldsymbol{m}_0) \\ (f) & 0 \leq v_i \leq V_i & (\forall p_i \in P^B) \end{cases} \tag{4}$$
where $\boldsymbol{q} \in \mathbb{R}^m$, $\boldsymbol{y} \in \mathbb{R}^n$ and $\boldsymbol{v} \in \mathbb{R}^{m^B}$ are unknown, and all other parameters, that depend on structure on the net, have previously been defined.

Given a solution of $(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$ of (4) let

$$\mathcal{I} = \{i \mid p_i \in P^B, q_i < Pre(p_i, \cdot) \cdot \boldsymbol{y} \cdot s_i/v_i\},$$

and for all $i \in \mathcal{I}$ denote

$$\varrho_i^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v}) = \frac{q_i}{Pre(p_i, \cdot) \cdot \boldsymbol{y} \cdot s_i/v_i}$$

and define

$$\varrho^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v}) = \begin{cases} 1 & \text{if } \mathcal{I} = \emptyset; \\ \min_{i \in \mathcal{I}} \{\varrho_i^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})\} & \text{otherwise.} \end{cases}$$

(A) If $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s, \boldsymbol{v}^s)$ is a reachable steady state then $(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$ satisfies eq. (4) with $\boldsymbol{q} = \mu(\boldsymbol{m}^s)$, $\boldsymbol{y} = \boldsymbol{\varphi}^s$, $\boldsymbol{v} = \boldsymbol{v}^s$.

(B) If $(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$ is a solution of eq. (4) then there exists a family of steady states $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s, \boldsymbol{v}^s)$ with $\boldsymbol{m}^s = \nu(\boldsymbol{q}, \varrho\boldsymbol{y}, \boldsymbol{v})$, $\boldsymbol{\varphi}^s = \varrho\boldsymbol{y}$, $\boldsymbol{v}^s = \boldsymbol{v}$ for all $\varrho \in [0, \varrho^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})]$.

Proof. We first show how Proposition 4.3 can be rewritten for cGBPN. Consider CS (3) and assume that batch places have variable transfer speed. In equation (b), $\delta_i^{\min}$ is not a constant any more and should be replaced by s_i/v_i. In equation (c), the constant transfer speed V_i of batch place p_i should be replaced by the instantaneous transfer speed v_i. Finally, appropriate constraints should be introduced to limit the admissible instantaneous transfer speed of each batch place p_i in the interval $[0, V_i]$. This leads to the following system

$$\begin{cases} (a) & \boldsymbol{0} \leq \boldsymbol{y} \leq \boldsymbol{\Phi} & \\ (b') & Q_i \geq q_i & (\forall p_i \in P^B) \\ (b'') & q_i \geq Pre(p_i, \cdot) \cdot \boldsymbol{y} \cdot s_i/v_i & (\forall p_i \in P^B) \\ (c) & Post(p_i, \cdot) \cdot \boldsymbol{y} \leq v_i \cdot d_i^{\max} & (\forall p_i \in P^B) \\ (d) & \boldsymbol{C} \cdot \boldsymbol{y} = \boldsymbol{0} & \\ (e) & \boldsymbol{q} \in RQ(N, \boldsymbol{m}_0) & \\ (f) & 0 \leq v_i \leq V_i & (\forall p_i \in P^B) \end{cases} \tag{5}$$

While CS (5) can be used to characterize steady states following Proposition 4.3, it is non-linear in eq. (b'') and finding a solution is not practical. We relax constraint (b'') thus obtaining CS (4) whose admissible solutions are a superset of the solutions of CS (5). We will discuss how the solutions of the two sets are related, thus proving the proposition.

Let us first point out an obvious fact.

Fact 4.5: If $(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$ is a solution of CS (4), resp., CS (5), then $(\boldsymbol{q}, \varrho\boldsymbol{y}, \boldsymbol{v})$ with $\varrho \in [0, 1]$ is also a solution of CS (4), resp., CS (5). ■

This can be immediately seen from eqs. (a), (b''), (c) and (d).

We can now prove separately the two parts of the proposition.

(Part A) Assume $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s, \boldsymbol{v}^s)$ is a reachable steady state and let $\boldsymbol{q} = \mu(\boldsymbol{m}^s)$, $\boldsymbol{y} = \boldsymbol{\varphi}^s$, $\boldsymbol{v} = \boldsymbol{v}^s$. By Proposition 4.3 $(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$ satisfies CS (5), which in turn satisfies CS (4).

(Part B) Assume $(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$ is a solution of CS (4) and let $\boldsymbol{m}^s = \nu(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$, $\boldsymbol{\varphi}^s = \boldsymbol{y}$, $\boldsymbol{v}^s = \boldsymbol{v}$. We consider two cases.

If $\mathcal{I} = \emptyset$, i.e., $(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$ satisfies constraint (b'') in CS (5) for all $p_i \in P^B$, then $\varrho^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v}) = 1$. From Fact 1 above, it follows that for $0 \leq \varrho \leq 1 = \varrho^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$ the triple $(\boldsymbol{q}, \varrho\boldsymbol{y}, \boldsymbol{v})$ is a solution of CS (5), and hence by Proposition 4.3 $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s, \boldsymbol{v}^s)$ is an SS.

If, on the contrary, $\mathcal{I} \neq \emptyset$, then it is easy to show (recalling the definition of ϱ^* and using Fact 1) that $(\boldsymbol{q}, \varrho\boldsymbol{y}, \boldsymbol{v})$ satisfies constraint (b'') in CS (5) if and only if $\varrho \in [0, \varrho^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})]$. Hence by Proposition 4.3 $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s, \boldsymbol{v}^s)$ is an SS if and only if $\varrho \in [0, \varrho^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})]$. □

Two issues warrant comment.

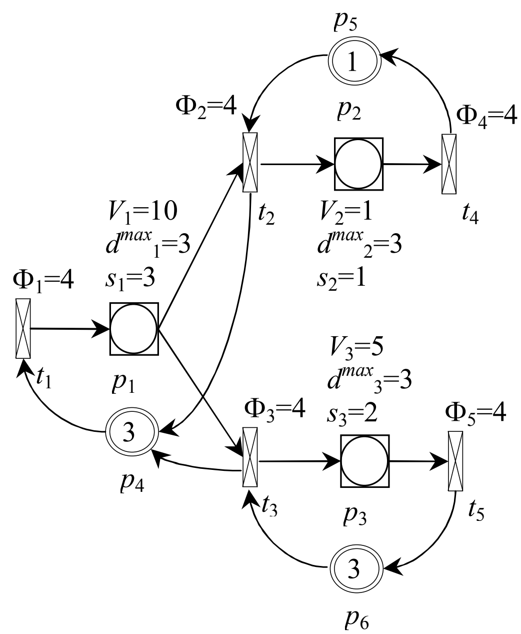

Fig. 4. Net in Example 4.8.

***Remark** 4.6:* In CS (3) and CS (4) all constraints are linear but for the reachability constraint $\boldsymbol{q} \in RQ(N, \boldsymbol{m}_0)$ that ensures that a marking $\boldsymbol{m}$ with marking quantity $\boldsymbol{q} = \mu(\boldsymbol{m})$ is reachable from the initial marking $\boldsymbol{m}_0$.

While a linear algebraic characterization of the reachability set is not always possible for discrete nets, for continuous nets such a characterization is *usually*[2] possible (14) in terms of the state equation or, for special classes of nets, in even simpler terms using P-semiflows.

We point out that the same characterization applies to the reachable marking quantity set of a GBPN where $P^D = T^D = \emptyset$. Thus, we may usually write (see Section II-C)

$$\begin{aligned} RQ(N, \boldsymbol{m}_0) &= PRQ(N, \boldsymbol{m}_0) \\ &= \{\boldsymbol{q} \mid \exists \boldsymbol{z} \in \mathbb{R}^n_{\geq 0} : \ \boldsymbol{q} = \mu(\boldsymbol{m}_0) + \boldsymbol{C} \cdot \boldsymbol{z}\}. \end{aligned}$$

■

***Remark** 4.7:* Although Proposition 4.4 allows one to characterize all possible SS for a cGBPN, unlike Proposition 4.3 it may fail to determine an optimal SS.

As an example, assume the objective is that of finding an SS that maximize the objective function $J = \boldsymbol{c}^T \cdot \boldsymbol{y}$. By solving CS 4 so as to optimize this objective function we obtain a solution $(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$ and we compute, within this family, the optimal solution of CS 4 that is obviously $(\boldsymbol{q}, \varrho^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})\boldsymbol{y}, \boldsymbol{v})$.

However, a non-optimal solution $(\boldsymbol{q}', \boldsymbol{y}', \boldsymbol{v}')$ of CS 4 may lead to an even better solution of CS 4 that takes the form $(\boldsymbol{q}', \varrho^*(\boldsymbol{q}', \boldsymbol{y}', \boldsymbol{v}')\boldsymbol{y}', \boldsymbol{v}')$ and is such that

$$\boldsymbol{c}^T \cdot \varrho^*(\boldsymbol{q}', \boldsymbol{y}', \boldsymbol{v}')\boldsymbol{y}' > \boldsymbol{c}^T \cdot \varrho^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})\boldsymbol{y}.$$

■

This will be shown by means of the following example.

***Example** 4.8:* Consider the net in Fig. 4 where the initial marking is $\boldsymbol{M}_0 = [\emptyset\ \emptyset\ \emptyset\ 3\ 1\ 3]^T$, the maximal flow vector is $\boldsymbol{\Phi} = [4\ 4\ 4\ 4\ 4]^T$ and the maximal transfer speed vector is $\boldsymbol{V} = [10\ 1\ 5]^T$.

We select the following objective function to maximize: $f = 0.01 \cdot q_1 + 0.01 \cdot q_2 + 0.01 \cdot q_3 + 10 \cdot y_4 + y_5$ for the following reasons.

[2]For details see the suggested reference.

- The flow y_4 of transition t_4 represents the production of a high quality product.
- The flow y_5 of transition t_5 represents the production of a low quality product.
- The value of one item of the high quality product is 10 time higher than the value of one item of low quality product.
- A small weight was given to the marking quantities in the batch places in the steady state. This is heuristically equivalent to search for, among the optimal solutions with performance index $f = 10 \cdot y_4 + y_5$, those that have a higher marking quantity in the batch places. These solutions have a better chance of satisfying constraint (b'').

The optimal solution $(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$ of CS (4) is: $\boldsymbol{q} = [3\ 1\ 3\ 0\ 0\ 0]^T$, $\boldsymbol{y} = [4\ 3\ 1\ 3\ 1]^T$, $\boldsymbol{v} = [8.2\ 1\ 4.3]^T$, and the optimal value of the objective function of CS (4) is: $f_4^{opt} = 31.07$.

We now check for all batch places if constraint (b'') is satisfied, i.e., if

$$q_i \geq Pre(p_i, \cdot) \cdot \boldsymbol{y} \cdot s_i / v_i = q_{\min,i}.$$

It holds: $q_1 = 3 > \frac{12}{8.2} = q_{\min,1}$, $q_2 = 1 < \frac{3}{1} = q_{\min,2}$, $q_3 = 3 > \frac{2}{4.3} = q_{\min,3}$.

Consequently, $\mathcal{I} = \{2\}$ and

$$\varrho^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v}) = \varrho_2^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v}) = q_{\min,2}/q_2 = 0.3333.$$

For all $\varrho \in [0, \varrho^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})]$ it holds that $\boldsymbol{m}^s = \nu(\boldsymbol{q}, \varrho\boldsymbol{y}, \boldsymbol{v})$, $\boldsymbol{\varphi}^s = \varrho\boldsymbol{y}$ and $\boldsymbol{v}^s = \boldsymbol{v}$ characterize a steady states. The best value of the objective function is given by $\varrho = \varrho^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$ that gives a solution of CS (5) with $f_5 = 10.36$.

To prove that this solution is not optimal, we now consider a different solution of CS (4) reducing the maximal flow of t_4: in fact, the only constraint (b'') that was not satisfied was that of p_2, whose output transition is t_4. We now set Φ_4 equal to the previous optimal value $\varphi_4^s = y_4 \cdot \varrho^* = 1$. This time the optimal solution $(\boldsymbol{q}', \boldsymbol{y}', \boldsymbol{v}')$ of CS (4) is: $\boldsymbol{q}' = [3\ 1\ 3\ 0\ 0\ 0]^T$, $\boldsymbol{y}' = [4\ 1\ 3\ 1\ 3]^T$, $\boldsymbol{v}' = [7.89\ 0.82\ 4.18]^T$, and the optimal value of the objective function of CS (4) is: $f_4^{opt'} = 13.07$. One can verify that constraint (b'') for place p_2 is still not satisfied and now $\varrho^*(\boldsymbol{q}', \boldsymbol{y}', \boldsymbol{v}') = 0.82$. We obtain a family of solutions of CS (4) among which the best gives an objective function $f_5 = 10.78$, better than the one previously computed.

We iterate once more reducing the maximal flow of t_4 to $\Phi_4 = 0.8$ roughly equal φ_4^s. This time the optimal solution $(\boldsymbol{q}'', \boldsymbol{y}'', \boldsymbol{v}'')$ of CS (4) is: $\boldsymbol{q}'' = [3\ 1\ 3\ 0\ 0\ 0]^T$, $\boldsymbol{y}'' = [4\ 0.8\ 3.2\ 0.8\ 3.2]^T$, $\boldsymbol{v}'' = [7.87\ 0.8\ 4.13]^T$, and the optimal value of the objective function of CS (4) is: $f_4^{opt''} = 11.25$. Since in this case all constraints (b'') are satisfied, this is also the solution we have found of CS (5).

V. Conclusions and future work

In this paper we considered controlled Generalised Batches Petri nets and characterized their steady state behavior. Computing a steady state requires solving a nonlinear programming problem that takes into account the net structure and the initial marking. A viable technique to compute a family of solutions by linear relaxation is also presented. The issue of computing optimal steady states for cGBPN for given linear objective functions will be addressed in future work.

References

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

[2] 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*, vol. 21, no. 4, pp. 427–497, 2011.

[3] C. Cassandras, *Stochastic flow systems: Modeling and sensitivity analysis*. CRC Press, New York, 2006.

[4] F. Balduzzi, A. Giua, and G. Menga, “First–order hybrid Petri nets: a model for optimization and control,” *IEEE Trans. on Robotics & Automation*, vol. 16, no. 4, pp. 382–399, 2000.

[5] D. Lefebvre, D. Catherine, F. Leclerq, and F. Druaux, "Some contributions with Petri nets for the modelling, analysis and control of HDS," *Nonlinear Analysis: Hybrid Systems*, vol. 1, no. 4, pp. 451–465, 2007.

[6] A. Giua, C. Seatzu, and Y. Wardi, "Ipa for continuous Petri nets with threshold-based flow control," in *Proc. 18th IFAC World Congress*, Milan, Italy, August/September 2011.

[7] I. Demongodin, "Generalised batches Petri net: Hybrid model for high speed systems with variable delays," *Discrete Event Dynamic Systems*, vol. 11, no. 1&2, pp. 137–162, 2001.

[8] I. Demongodin and A. Giua, "Linear programming techniques for analysis and control of batches Petri nets," in *Proc. WODES'10: 10th Work. on Discrete Event Systems*, Berlin, Germany, August/September 2010.

[9] B. Gaujal and A. Giua, "Optimal stationary behavior for a class of timed continuous Petri nets," *Automatica*, vol. 40, no. 9, pp. 1505–1516, 2004.

[10] J. Júlvez, L. Recalde, and M. Silva, "Steady-state performance evaluation of continuous mono-t-semiflow Petri nets," *Automatica*, vol. 41, no. 4, pp. 605 – 616, 2005.

[11] C. Mahulea, L. Recalde, and M. Silva, "Basic server semantics and performance monotonicity of continuous Petri nets," *Discrete Event Dynamic Systems*, vol. 19, no. 2, pp. 189–212, 2009.

[12] A. Giua, C. Mahulea, L. Recalde, C. Seatzu, and M. Silva, "Optimal control of timed continuous Petri nets via explicit MPC," in *POSTA 2006, LNCS 341*, 2006, pp. 383–390.

[13] I. Demongodin, "Modeling and analysis of transportation networks using batches Petri nets with controllable batch speed," in *ICATPN 2009, LNCS 5606*, 2009, pp. 204–222.

[14] J. Júlvez, L. Recalde, and M. Silva, "On reachability in autonomous continuous Petri net systems," in 24^{th} *International Conference on Application and Theory of Petri Nets*, ser. Lecture Notes in Computer Science, W. van der Aalst and E. Best, Eds. Eindhoven, The Netherlands: Springer, 2003, vol. 2679, pp. 221–240.