Dynamics and steady state analysis of controlled Generalized Batches Petri Nets*

Isabel Demongodin (♭), Alessandro Giua (♭,♯)

(♭) LSIS, Univ. of Aix-Marseille, France

(♯) DIEE, Univ. of Cagliari, Italy

(isabel.demongodin@lsis.org, giua@diee.unica.it)

Abstract

This paper is dedicated to an extended class of hybrid Petri nets called controlled Generalized Batches Petri Nets. The novel feature of these nets is that the firing flow of continuous and batch transitions and the transfer speed of batch places are control variables. First we propose a linear programming problem to compute the instantaneous firing flow vector and the instantaneous transfer speed vector solving an optimization problem, where the objective function depends on the control goal. Second we analyze and characterize the steady state of this model solving a programming problem that takes into account the net structure and the initial marking. This problem is linear if the transfer speeds are preassigned while it is nonlinear if the transfer speeds are control variables. In such a last case, a viable technique to compute a family of solutions by linear relaxation of the non linear problem is presented. The optimality of steady states for given linear objective functions is also addressed.

Published as:
I. Demongodin, A. Giua, "Dynamics and steady state analysis of controlled Generalized Batches Petri Nets,Ť *Nonlinear Analysis: Hybrid Systems*, Vol. 12, pp. 33Ű44, May 2014.
DOI: 10.1016/j.nahs.2013.11.010.

*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.

1 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 [6]. A recent survey of this area was presented by Silva *et al.* [22]. There are a few motivations behind this effort. First, 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 of 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 a viable means to bypass this issue and can also be used to speed up simulation [16]. Second, 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 [5] and references therein) and to Petri nets [3, 19, 27].

Many man-made systems contain continuous transfer elements that introduce variable delays depending on the 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 (BPNs), defined by Demongodin *et al.* [9], which 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 place* characterized by a transfer speed. Since its first definition, several extensions of batches Petri nets have been developed: Controlled BPN [2] where maximal transfer speeds and maximal flows may change during the dynamics; Colored BPN [4]; Generalized BPN [7]; Stochastic BPN [26]; Extended BPN [18] and BPN with controllable batch speed [8]. These models or variants of them are appropriate for describing high throughput production lines and conveyor belts [1, 4, 26, 25], transmission delay on a communication media with bus topology [23], interlocking system design for ERTMS/ETCS [13], multimodal systems [18] or transportation networks [8].

Here we consider an extended class of Generalized Batches Petri Nets (GBPNs), as defined by [7], which we call *controlled Generalized Batches Petri Nets* (cGBPNs) and devote the first part of the paper to formally describe this model and its dynamical behavior. The main novelty of this formalism is represented by a new semantics — inspired by First Order Hybrid Petri Nets [3] — that considers the instantaneous firing flow of continuous and batch transitions and the transfer speed of batch places as control inputs that can be used to drive the evolution of the net. This has three important consequences. First, we generalize previous semantics that consider a single possible autonomous evolution: this was the assumption in [6, 7] where it is assumed that a transition should always fire at its maximal admissible flow. Second, this allows us to consider problems of conflict that could not well be handled in the framework of non controlled evolutions. Third, in our framework a simple linear algebraic approach can be used to select a control input.

The notion of *admissible* control input, is discussed at length in the second part of the paper. To be admissible, i.e., physically realizable, a control input must satisfy appropriate constraints. These constraints are of different nature. Some of them are structural and impose that the flow produced by a transition or the transfer speed of a batch place cannot exceed a maximum value. Some of them are related to the physical interpretation of continuous and batch places, seen as deposits for stock of material whose quantity is nonnegative and should not exceed a given capacity. Some of them are related to the interconnection of a batch place, whose transfer element should be able to accept the flow arriving in the input and to sustain the flow required in the output. In our framework, the set of admissible control inputs can be characterized by the feasible solutions of a linear constraint set that depends on the net structure and on its current marking (state). By introducing an appropriate objective function, this framework allows one to select an "optimal" mode of operation of the net. Note that this optimization is myopic, i.e., the optimal solution may be applicable only for a finite time interval: as the state of the net changes, the linear constraint set that defines the admissible control inputs may change as well, and a new mode of operation may need to be reselected.

An important problem in many classes of physical systems than can described by cGBPNs is that of determining a *stationary* mode of operation that can be maintained for an indefinite time. Following [14] we may classify the stationary behavior of a cGBPN as *weakly stationary*, when the control input and the marking of the net are periodic, or *strongly stationary*, when the control input and the marking of the net are in a constant *steady state*. Related work on the steady state analysis of continuous nets can be found in [14, 15, 21, 20, 24].

In the last part of this paper we address the problem of characterizing the set of constant steady states of a cGBPN. Such states are described by a triple $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s, \boldsymbol{v}^s)$ where $\boldsymbol{m}^s$ is a constant marking, $\boldsymbol{\varphi}^s$ is a constant vector of instantaneous firing flows and $\boldsymbol{v}^s$ is a constant vector of transfer speeds associated to batch places. As a first result we consider the case in which the transfer speed of batch places is assigned: in this case the steady state reduces to a pair $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s)$ and it can be characterized by a linear constraint set that takes into account the net structure and the initial marking. As a second result, we show that when the speed of transfer places is a decision variable as well, a steady state is characterized by a constraints set that contains both linear and nonlinear constraints. 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 the optimal solution by means of a detailed numerical example. While a formal analysis of periodic steady states is out of the scope of the paper, such a case is discussed by means of an example.

As a final remark, we point out that this paper is concerned with the characterization of a steady state for the continuous and batch evolution of the net and not for its more general hybrid evolution. Thus a restricted class of nets, only composed by continuous and batch nodes, is considered for the steady state analysis. The same results, however, apply to any cGBPN during a period in which no discrete transition fires. In fact for each discrete marking of the net (assumed constant) there exists a set of possible steady states for its continuous and batch evolution that can be characterized with the approach here presented. However, between two discrete transition firings that occur in a short time a steady state may not be reachable.

The paper is structured as follows. In Section 2, 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 3 proposes a linear programming problem to compute the instantaneous firing flow and transfer speed vectors. Section 4 is dedicated to the computation of steady states for such a model and to the discussion of a detailed example.

Preliminary versions of these results have been presented in [11] and [12].

2 Background on batches Petri nets

2.1 Net structure

The following definition, taken from [12] slightly extends the notion of GBPN introduced in [7] and [11].

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 a 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 a 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 matrices, 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.

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

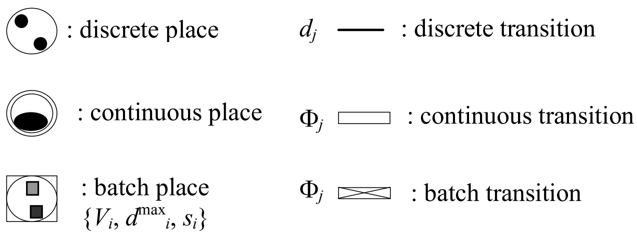

Figure 1: Nodes of a cGBPN.

- *$Time : T \to \mathbb{R}_{\geq 0}$ associates a nonnegative 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.* ■

It should be noted that *controlled GBPNs* have the same syntax of *GBPNs*. However — as the term "controlled" implies — we associate to cGBPNs a different semantics, assuming that the instantaneous firing flow of continuous and batch transitions and the transfer speed of batch places are control inputs that can be used to drive the evolution of the net. This will be discussed in detail in the following sections.

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 ${}^{(b)}p_i = {}^\bullet p_i \cap T^B$.

In this paper we will only consider well-formed nets, previously introduced in GBPNs [11] as follows.

Definition 2.2 *A cGBPN is said to be (*well-formed*) if the following conditions hold:*

- *discrete places can be connected to continuous and batch transitions only by self-loops, i.e., for all $p_i \in P^D$ and for all $t_j \in T^C \cup T^B$ it holds $Pre(p_i, t_j) = Post(p_i, t_j)$.*
- *the pre and post sets of batch places contain only batch transitions, i.e., for all $p_i \in P^B$ it holds ${}^\bullet p_i \cup p_i^\bullet \subset T^B$.* ■

The first condition, that is also commonplace in the framework of hybrid nets, is required to ensure that the marking of discrete places is not changed by the firing of continuous

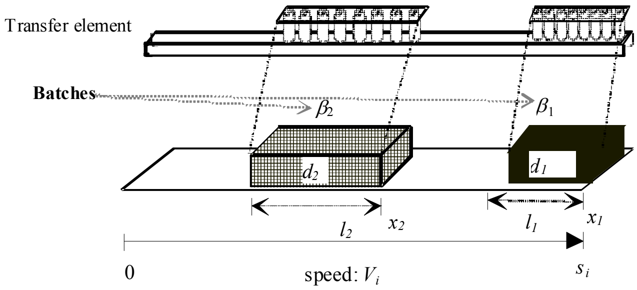

Figure 2: Batches.

and batch transitions. The second condition is due to the rules concerning the creation and destruction of batches.

Due to the partition of places and transitions into discrete, continuous and batch nodes (see Definition 2.1), the incidence matrix of a well-formed cGBPN, defined as $\boldsymbol{C} = \boldsymbol{Post} - \boldsymbol{Pre}$, can be partitioned as follows.

$$\begin{array}{cc} & \begin{array}{ccc} T^D & T^C & T^B \end{array} \\ \boldsymbol{C} = & \left[\begin{array}{ccc} \boldsymbol{C}^{DD} & \mathbf{0} & \mathbf{0} \\ \boldsymbol{C}^{CD} & \boldsymbol{C}^{CC} & \boldsymbol{C}^{CB} \\ \mathbf{0} & \mathbf{0} & \boldsymbol{C}^{BB} \end{array}\right] \begin{array}{c} P^D \\ P^C \\ P^B \end{array} \end{array}$$

2.2 Batches and markings

The main extension of batches Petri nets with respect to hybrid Petri nets is related to the notions of batch, i.e., a group of discrete entities characterized by three continuous variables (see Fig. 2).

Definition 2.3 *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 2.4 *The* marking *of a cGBPN 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 nonnegative 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 nonnegative 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.

Definition 2.5 *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 places, while for a batch place it represents the sum of the quantities of the batches it contains. ■

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 $\boldsymbol{q}$.

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

Let us now discuss some special type of batches.

Definition 2.6 *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* $\gamma(p_i) = (V_i, d_i^{\max}, s_i)$*.*

- *The batch* β_r *is called* dense *if its density is equal to the maximal density of batch place* p_i*,* $d_r(\tau) = d_i^{\max}$*.*
- *The batch* β_r *is called an* output batch *if its head position is equal to the length associated to the batch place, i.e.,* $x_r(\tau) = s_i$*.*
- *The* output density d_i^{out} *of a 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* $d_i^{\text{out}}(\tau) = 0$*.*

■

A place in a cGBPN can have at most one output batch. Note that the output density of place p_i at time τ depends on the marking $\boldsymbol{m}(\tau)$ and can also be denoted by $d_i^{\text{out}}(\boldsymbol{m})$.

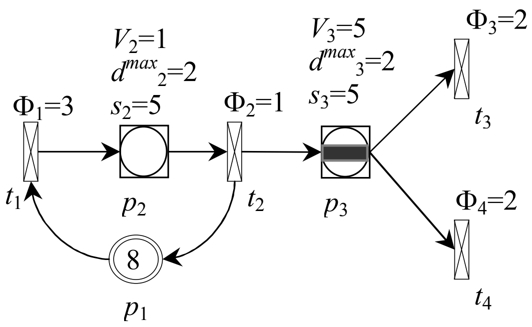

Figure 3: Net in Example 2.8.

Due to the bounded characteristics of a batch place, some constraints on batches characteristics have to be respected: $0 \leq l_r \leq x_r \leq s_i$ (position and length constraints) and $0 \leq d_r \leq d_i^{\max}$ (density constraint).

The notion of batch place function associated to a batch place implicitly assumes that the place capacity is finite. This is formalized in the following definition.

Definition 2.7 *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. ■

Example 2.8 *Consider a manufacturing system composed by 4 machines linked by 2 conveyor belts. The net in Fig. 3 represents such a system, where the four machines are represented by batch transitions t_1, t_2, t_3 and t_4, and their maximal throughputs are given by the maximal firing flows associated with batch transitions: $\Phi_1 = 3$, $\Phi_2 = 1$, $\Phi_3 = \Phi_4 = 2$. The two conveyor belts linking machine 1 to machine 2, and machine 2 to both machine 3 and machine 4, are respectively represented by batch places p_2 and p_3 with $\gamma(p_2) = (V_2, d_2^{\max}, s_2) = (1, 2, 5)$ and $\gamma(p_3) = (V_3, d_3^{\max}, s_3) = (5, 2, 5)$. The continuous place p_1 corresponding to the maximal capacity of conveyor 1, limits the capacity of batch place p_2 (see Def. 2.7).*

The initial marking is $\boldsymbol{m}_0 = [8 \ \emptyset \ \{\beta_1(0)\}]^T$ with $\beta_1(0) = (5, 2, 5)$. We remark that β_1 is the output batch of place p_3 and it 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 $q_3(0) = l_1(0) \cdot d_1(0) = 10 = Q_3$.

2.3 Net dynamics

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 following definition introduces the notion of transition firing flow.

Definition 2.9 *The* instantaneous firing flow *(IFF) $\varphi_j(\tau) \leq \Phi_j$, associated to a contin-*

uous or batch transition $t_j \in T^C \cup T^B$ represents the quantity of firing of transition t_j by time unit. The IFF vector at time τ is denoted by $\boldsymbol{\varphi}(\tau) \in \mathbb{R}^{n_C+n_B}$.

The input (*resp.,* output) flow *of a batch or continuous place p_i at time τ is the sum of all flows entering (resp., leaving) the place and can be written, respectively, as:*

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

■

The following definition introduces the notion of place speed.

Definition 2.10 *The* instantaneous transfer speed *(ITS) $v_i(\tau) \in [0, V_i]$ associated to a batch place $p_i \in P^B$ represents the transfer velocity within place p_i at time τ. The ITS vector at time τ is denoted by $\boldsymbol{v}(\tau) \in \mathbb{R}^{m_B}$.* ■

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 3 will show how to select admissible control inputs.

2.3.1 Enabling and firing conditions

The enabling and firing conditions of discrete transitions are those of classical transition timed discrete Petri nets.

Condition 2.11 *A discrete transition $t_j \in T^D$ is enabled at $\boldsymbol{m}$ if for all $p_i \in {}^\bullet t_j$, $m_i \geq Pre(p_i, t_j)$.*

A discrete transition $t_j \in T^D$ that is enabled at a marking $\boldsymbol{m}$ and has also been continuously enabled for a time equal to its firing delay, fires *yielding a new marking $\mathbf{m}' = \mathbf{m} + \boldsymbol{C}(\cdot, t_j)$.* □

The enabling conditions of continuous transitions are those of First Order Hybrid Petri nets [3] i.e., one distinguishes weakly and strongly enabled transitions.

Condition 2.12 *A continuous transition $t_j \in T^C$ is enabled at $\boldsymbol{m}$ if for all $p_i \in {}^{(d)}t_j$, $m_i \geq Pre(p_i, t_j)$. We say that the continuous transition is:*

- strongly enabled *if $\forall p_k \in {}^{(c)}t_j$, $m_k > 0$.*

- weakly enabled *if* $\exists p_r \in {}^{(c)}t_j,\ m_r = 0$.

□

We define similar conditions for batch transitions.

Condition 2.13 *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* $\exists p_r \in {}^{(c)}t_j,\ m_r = 0$.

□

The computation of the IFF of enabled continuous and batch transitions will be described in Section 3 and here we simply discuss the net evolution assuming that the IFF vector, $\boldsymbol{\varphi}(\tau)$, is given at time τ.

The evolution in time of the marking of a continuous place, $p_i \in P^C$ is described by:

$$\dot{m}_i(\tau) = \left[\ C^{CC}(p_i, \cdot) \quad C^{CB}(p_i, \cdot)\ \right] \boldsymbol{\varphi}(\tau).$$

Let us now focus on the hybrid dynamics of batch nodes.

2.3.2 Hybrid dynamics of batch places

To resume the hybrid dynamics of batches, let us first introduce some concepts necessary to the understanding of the evolution.

Definition 2.14 *At time τ, various static functions can be applied to the batches in a batch place p_i:*

- Create. *If the input flow of p_i is not null, i.e., $\phi_i^{\text{in}}(\tau) \neq 0$, a batch $\beta_r(\tau) = (0, d_r(\tau), 0)$ with $d_r(\tau) = \phi_i^{\text{in}}(\tau)/v_i$, is created and added to the marking of p_i, i.e., $m_i(\tau) = m_i(\tau) \cup \{\beta_r(\tau)\}$.*

- Destroy. *If the length of a batch, $\beta_k(\tau)$, is null, $l_k = 0$, and if it is not a created batch, $x_k \neq 0$, batch $\beta_k(\tau)$ is destroyed, noted $\beta_k(\tau) = \mathbf{0}$, and removed from the marking of p_i, i.e., $m_i(\tau) = m_i(\tau) \setminus \{\beta_k(\tau)\}$.*

- Merge. *If two batches with the same density are in contact, they can be merged. Let batches $\beta_r(\tau) = (l_r(\tau), d_r(\tau), x_r(\tau))$ and $\beta_k(\tau) = (l_k(\tau), d_k(\tau), x_k(\tau))$ in $m_i(\tau)$, such that $x_r(\tau) = x_k(\tau) + l_r(\tau)$ and $d_r(\tau) = d_k(\tau)$. In this case, batch $\beta_r(\tau)$ becomes $\beta_r(\tau) = (l_r(\tau)+l_k(\tau), d_r(\tau), x_r(\tau))$, batch $\beta_k(\tau)$ is destroyed, $\beta_k(\tau) = \mathbf{0}$, and $m_i(\tau) = m_i(\tau) \setminus \{\beta_k(\tau)\}$.*

- Split. *It is always possible to split a batch into two batches in contact with the same density.* ■

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 2.15 (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.*

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

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

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

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

- Exit. *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 2.16 (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, $d_r(\tau) \cdot v_i$, 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 [8]. Note that in these dynamics, we assume that the density of a batch in an accumulation behavior is equal to the maximal density of the batch place, i.e., it is dense. When a batch starts an accumulation, it is split into two batches in contact where the downstream batch is dense. The following example explains this behavior.

Example 2.17 *Consider a batch place, p_i with $\gamma(p_i) = (V_i, d_i^{\max}, s_i) = (1, 3, 5)$, connected to an output batch transition t_j where ${}^\bullet t_j = \{p_i\}$ and $p_i^\bullet = \{t_j\}$. Let $\beta_k(\tau_0) = (5, 2, 5)$ be an output batch in place p_i at time τ_0. According to Def. 2.6 the output density of p_i is: $d_i^{\text{out}}(\tau_0) = d_k(\tau_0) = 2$. Thus, the batch transition is strongly enabled at time τ_0 (see Condition 2.13).*

If we assume that the output flow of p_i is as follows, $\phi_i^{\text{out}}(\tau_0) < d_i^{\text{out}}(\tau_0) \cdot v_i(\tau_0)$, the batch place is in an accumulation behavior and the output batch starts an accumulation. This output batch is then split into two batches in contact, batch $\beta_{k2}(\tau_0) = (5, 2, 5)$ and batch $\beta_{k1}(\tau_0) = (0, 3, 5)$ which is dense. From this time on, the characteristics of both batches change as follows.

- *For batch β_{k1}: its length increases while its position and its density do not vary.*

$$\dot{d}_{k1} = \dot{x}_{k1} = 0; \quad \dot{l}_{k1} = \frac{v_i \cdot d_{k2} - \phi_i^{\text{out}}}{d_i^{max} - d_{k2}}$$

- *For batch β_{k2}: its length decreases, its position is reduced while its density does not vary.*

$$\dot{d}_{k2} = 0; \quad \dot{l}_{k2} = \frac{\phi_i^{\text{out}} - v_i \cdot d_i^{max}}{d_i^{max} - d_{k2}}; \quad \dot{x}_{k2} = -\frac{v_i \cdot d_{k2} - \phi_i^{\text{out}}}{d_i^{max} - d_{k2}}$$

Now, we assume that the maximal firing flow is null, $\Phi_j = 0$ (i.e., no product leaves the conveyor). Thus, the instantaneous firing flow of t_j and the output flow of p_i are both null: $\phi_i^{\text{out}}(\tau_0) = \varphi_j(\tau_0) = \Phi_j = 0$. Moreover, the transfer speed of p_i is supposed to be equal to its maximal value: $v_i(\tau_0) = V_i$. It holds from time $\tau \geq \tau_0$:

$$\dot{l}_{k1} = \frac{2}{3}; \quad \dot{l}_{k2} = -\frac{5}{3}; \quad \dot{x}_{k2} = -\frac{2}{3}$$

When the length of batch β_{k2} will be equal to 0, an event will appear corresponding to the total accumulation of batch β_k which will be then dense.

2.4 Net evolution

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 and the transfer speed of batch places are constants.

The IB-state changes if and only if one (or possibly several at the same time) of the following kind 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;
6. a batch place becomes full.

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);
- 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, 10]: $\boldsymbol{q}(\tau) = \boldsymbol{q}(\tau_0) + \boldsymbol{C} \cdot \boldsymbol{z}(\tau)$, 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*

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

3 Computation of instantaneous flows and speeds

In Section 2.3 it was mentioned that the instantaneous firing flows (IFF) vector $\boldsymbol{\varphi} \in \mathbb{R}^{(n^C+n^B)}_{\geq 0}$ of continuous and batch transitions and the instantaneous transfer speed (ITS) vector $\boldsymbol{v} \in \mathbb{R}^{m^B}_{\geq 0}$ of batch places should be considered as variables that can be suitably chosen by a supervisor to drive the evolution of a cGBPN. Thus we can define for a cGBPN a *control input* $\boldsymbol{u} = (\boldsymbol{\varphi}, \boldsymbol{v})$.

In this section, the admissibility of a control input is first defined and, later, the set of admissible control inputs is characterized using a linear programming approach.

Definition 3.1 (Admissible control input) *A control input is called* admissible *if it satisfies the following constraints that depend on the net structure and on its marking (i.e., state).*

- Structural constraints*: the flow of a continuous or batch transition should not exceed the maximal firing flow; the speed of a batch place should not exceed the maximal transfer speed.*
- Enabling constraints*: a continuous or batch transition that is not enabled should have a null instantaneous firing flow.*
- Nonnegativity of the marking quantity*: the marking quantity within a continuous place should be nonnegative.*
- Capacity constraint of batch places*: the marking quantity within a batch place should not exceed the place capacity.*
- Congruence constraints*: the flow arriving in a batch place should not exceed the flow that can be accepted by the transfer element; the flow leaving a batch place should not exceed the flow that can be sustained by the transfer element.*

■

It should be noted that the second type of congruence constraint implies that the marking quantity within a batch place should be nonnegative, because if a batch place is empty it can sustain no output flow, hence its marking cannot decrease.

Proposition 3.2 *Given a cGBPN* $\langle N, \boldsymbol{m} \rangle$ *with incidence matrix* $\boldsymbol{C}$*, let:*

- $T_{\mathcal{N}}(\boldsymbol{m}) \subset T^C \cup T^B$ *be the subset of continuous and batch transitions that are 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 batch places.*

A pair $\boldsymbol{u} = (\boldsymbol{\varphi}, \boldsymbol{v})$ *is an admissible input if and only if it is a feasible solution of the following linear set:*

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

The set of all feasible control inputs is denoted $\mathcal{U}(N, \boldsymbol{m})$*.*

Proof. Constraints of the form (a)-(b) correspond to the structural constraints, with the additional specification that the control input is nonnegative.

Constraints of the form (c) correspond to the enabling constraints.

Constraints of the form (d) require that the total algebraic flow $C(p_i, \cdot) \cdot \boldsymbol{\varphi}$ (positive if input, negative if output) of an empty continuous place p_i should be nonnegative and hence its marking cannot decrease, thus implying the nonnegativity of marking quantity.

Constraints of the form (e) require that the total algebraic flow $C(p_i, \cdot) \cdot \boldsymbol{\varphi}$ of a full batch place p_i should be non positive and hence its marking quantity cannot increase, thus implying the capacity constraint.

Constraints of the form (f) require that the input flow $Post(p_i, \cdot) \cdot \boldsymbol{\varphi}$ of a batch place p_i should be less than or equal to the flow $v_i \cdot d_i^{\max}$ that can be accepted by its transfer element, thus implying the first type of congruence constraints.

Constraints of the form (g) require that the output flow $Pre(p_i, \cdot) \cdot \boldsymbol{\varphi}$ of a batch place p_i should be less than or equal to the flow $v_i \cdot d_i^{\text{out}}(\boldsymbol{m})$ that can be sustained by its transfer element, thus implying the second type of congruence constraints. □

Constraints of the form (a), (c) and (d) are similar to those that describe the set of admissible IFF vectors in FOHPNs [3]. All other constraints are peculiar to cGBPNs. It can be noted that constraints (a), (b) and (f) do not depend on the current marking, while all other constraints are marking dependent (we have denoted here the output flow as $d_i^{\text{out}}(\boldsymbol{m})$ to make explicit its dependence on the marking).

It should also be noted that to impose a structural bound on a batch place p_i with capacity Q_i, one may add to the net a complementary continuous place $p_{i'}$ with $Pre(p_{i'},\cdot) = Post(p_i,\cdot)$, $Post(p_{i'},\cdot) = Pre(p_i,\cdot)$, and $m_{i'}(0) = Q_i - q_i(0)$. In such a case, constraint (d) can be removed from linear set (1) as it is imposed by the nonnegativity constraint for continuous place $p_{i'}$.

It is important to stress once again the assumption underlying Proposition 3.2: the firing flows of continuous and batch transitions and the transfer speed of batch places are control inputs whose value can be chosen a supervisor within the set $\mathcal{U}(N,\boldsymbol{m})$. To choose among the admissible control inputs that satisfy (1) the supervisor may use an objective function, or introduce additional constraints, as also discussed in the case of FOHPNs [3].

As a final remark, it is common to assume that the components of the control input are piecewise-constant signals. This means that the selected value of a control input will be used until the occurrence of an event (such as the firing of a discrete transition that may change the set of enabled transitions, the emptying of a continuous place, the filling of batch place, the change of the output density of a batch place) will require a new computation.

Example 3.3 *Consider the net of Example 2.8, represented in Fig. 3, with* $\beta_1(0) = (5,2,5)$ *and* $\boldsymbol{m}_0 = [8\ \emptyset\ \{\beta_1(0)\}]^T$.

- *At the initial time it holds:* $T_{\mathcal{N}}(\boldsymbol{m}_0) = \{t_2\}$, $P_\emptyset(\boldsymbol{m}_0) = \emptyset$ *and* $P_F(\boldsymbol{m}_0) = \{p_3\}$, *hence the control input* $\boldsymbol{u} = (\boldsymbol{\varphi},\boldsymbol{v})$ *must verify:*

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

  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 the objective function to maximize* $J = \varphi_3 + \varphi_4 + 0.1(\varphi_3 + \varphi_4) - 0.001(v_2 + v_3)$.

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

Incidentally, we note that if one also wants to impose a ratio $2 : 1$ *between the flows of* t_3 *and* t_4*, it is possible to add a constraint of the form* $\varphi_3 = 2\varphi_4$ *to get the solution* $\boldsymbol{\varphi} = (2, 0, 4/3, 2/3)$ *and* $\boldsymbol{v} = (v_2, v_3) = (1, 2)$.

From $\boldsymbol{\varphi} = (2, 0, 2, 2)$ *and* $\boldsymbol{v} = (1, 2)$*, as the input flow of* p_2 *is not null, a batch* $\beta_2(0) = (0, 2, 0)$ *(*$d_2(0) = 2/1 = 2$*) is created in* p_2*. The batches evolve as follows:* $\beta_1(\tau) = (5 - 2\tau, 2, 5)$ *and* $\beta_2(\tau) = (\tau, 2, \tau)$*, while the marking continuous evolution is described by* $\boldsymbol{m}_0 = [8 - 2\tau \ \{\beta_2(\tau)\} \ \{\beta_1(\tau)\}]^T$.

- *The next event will occur at time* $\tau_1 = 2.5$ *when the output batch* β_1 *will be completely out of place* p_3.

  At time τ_1, β_1 *is destroyed (see event 4 in Section 2.4) and a new marking is reached:* $\boldsymbol{m}_1 = [3 \ \{\beta_2(\tau_1)\} \ \emptyset]^T$ *with* $\beta_2(\tau_1) = (2.5, 2, 2.5)$*. It holds:* $T_{\mathcal{N}}(\boldsymbol{m}_1) = \{t_2, t_3, t_4\}$, $P_\emptyset(\boldsymbol{m}_1) = \emptyset$ *and* $P_F(\boldsymbol{m}_1) = \emptyset$.

  According to linear set (1), the control input must satisfy:

$$\begin{cases} (a) & 0 \leq \varphi_1 \leq 3 \\ (b) & 0 \leq v_2 \leq 1 \\ (b') & 0 \leq v_3 \leq 5 \\ (c) & \varphi_2 = 0 \\ (c') & \varphi_3 = 0 \\ (c'') & \varphi_4 = 0 \\ (f) & \varphi_1 \leq 2 \cdot v_2 \\ (f') & 0 \leq 2 \cdot v_3 \end{cases}$$

  Using the objective function previously described, one gets the optimal solution $\boldsymbol{\varphi} = (2, 0, 0, 0)$ *and* $\boldsymbol{v} = (1, 0)$.

  The batch evolves as before $\beta_2(\tau) = (\tau, 2, \tau)$*, while the marking continuous evolution is now described by* $\boldsymbol{m}_1 = [3 - 2(\tau - \tau_1) \ \{\beta_2(\tau)\} \ \emptyset]^T$.

- *The next event will occur when continuous place* p_1 *will be emptied at time* $\tau_2 = 4$*. By repeating the previous analysis, possibly changing the objective function depending on the IB-state, one can determine the complete evolution.*

Remark 3.4 *The procedure we have presented to determine an admissible control input fulfills two main purposes: (a) the set of constraints* (1) *defines within an IB-state all possible evolutions of the net that are legal, i.e., satisfy the behavioral constraints of a cGBPN; (b) the choice of a suitable objective function (as in Example 3.3) allows one to determine among all evolutions that are legal the one that optimizes a given criterion. Note, however, that this optimization is* myopic*: optimality holds within the current IB-state but in general not for the global evolution of the net as it evolves from one IB-state* *to another. This is a feature common with [3] and, more in general, with other greedy optimization schemes that may get stuck in a local minimum.*

4 Steady state computation of controlled GBPN

This section is devoted to the characterization, in linear algebraic terms, of the set of possible steady states of a cGBPN.

It should be noted that the steady state, as defined in the following, is only related to the continuous and batch evolution of the net and not to its hybrid evolution. Thus a restricted class of nets, only composed by continuous and batch nodes, is considered here ($P^D = T^D = \emptyset$).

The same results, however, apply to any cGBPN during a period in which no discrete transition fires. In fact for each discrete marking of the net (assumed constant) there exists a set of possible steady states for its continuous and batch evolution that can be characterized with the approach here presented. If the discrete dynamics is sufficiently slow, each time a discrete transition fires a new optimal steady state can be computed and reached.

4.1 Definition of steady state

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 $\tau \geq \tau_s$ 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 $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s, \boldsymbol{v}^s)$.* ■

Note that this definition also implies that the output density of batch places is constant at the steady state.

A first elementary but general result concerning an SS is the following.

Proposition 4.2 *Assume that a net $\langle N, \boldsymbol{m}_0 \rangle$ with $P^D = T^D = \emptyset$ is in a steady state $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s, \boldsymbol{v}^s)$. Then the marking quantity vector is such that: $\dot{\boldsymbol{q}}^s = \boldsymbol{C} \cdot \boldsymbol{\varphi}^s = \boldsymbol{0}$.*

Proof. The state equation of a cGBPN $\langle N, \boldsymbol{m}_0 \rangle$ with $P^D = T^D = \emptyset$ can be written:

$$\boldsymbol{q}(\tau) = \boldsymbol{q}(\tau_0) + \boldsymbol{C} \cdot \boldsymbol{z}(\tau) = \boldsymbol{q}(\tau_0) + \boldsymbol{C} \cdot \int_{\tau_0}^{\tau} \boldsymbol{\varphi}(\rho) d\rho$$

Thus the first equality follows from the state equation, and the second one from the fact that a constant marking implies a constant marking quantity in each place. □

4.2 Steady state of batch places

Two results that characterize the steady state of batch places are now presented. Obviously, for place p_i in a steady state $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s, \boldsymbol{v}^s)$ its input and output flows coincide, i.e.,

$$\phi_i^s = Post(p_i, \cdot) \cdot \boldsymbol{\varphi}^s = Pre(p_i, \cdot) \cdot \boldsymbol{\varphi}^s. \tag{2}$$

Furthermore we denote by $\delta_i^{\min,s}$ the *minimum delay* incurred traversing a batch place p_i moving at speed v_i^s, defined as follows:

$$\delta_i^{\min,s} = s_i / v_i^s \qquad \text{if } p_i \in P^B. \tag{3}$$

This represents the time spent in the place by an entity of a batch in free behavior. This transfer time will be greater if the place is in accumulation behavior.

Proposition 4.3 *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 quantity $q_i^s = \mu(m_i^s)$ of a batch place $p_i \in P^B$ satisfies*

$$Q_i \geq q_i^s \geq \phi_i^s \ \delta_i^{\min,s}. \tag{4}$$

Proof. The first inequality trivially follows from the boundedness of the place. The second inequality follows from Little's law for stationary behavior applied to each batch place p_i, which implies that the average quantity of marking it contains q_i equals the product of its average input flow and of its average delay δ_i. Since in our particular case these quantities are constants at the steady state, we write $q_i^s = \phi_i^s \ \delta_i \geq \phi_i^s \ \delta_i^{\min,s}$. □

Proposition 4.4 *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 $\beta_o = (l_o, d_i^{\max}, s_i)$ of 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 $\beta_o = (s_i, d_o, s_i)$ of length equal to the length of the place and with 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*
$$l_e = \frac{s_i d_i^{\max} v_i^s - q_i^s v_i^s}{d_i^{\max} v_i^s - \phi_i^s} \quad \text{and} \quad l_o = s_i - l_e. \tag{5}$$
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 of 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. In case (1) nothing can enter or leave the place: all the entities will move to the end of the place forming a single dense batch.

In case (2) the transfer time is equal to the minimum delay of the place. In such a case there exist an input batch $\beta_e = (l_e, d_e, l_e)$ and an output batch $\beta_o = (l_o, d_o, s_i)$ that are in free behavior (see Def. 2.15), have the same density $d_e = d_o = \phi_i^s / v_i^s$, and must be in contact (i.e., $l_e + l_o = s_i$) for the marking to remain constant. Hence they can be merged into a single batch $\beta_o = (s_i, d_o, s_i)$.

In case (3) the transfer time is greater than the minimum delay of the place. In such a case the output batch $\beta_o = (l_o, d_i^{\max}, s_i)$ must be in accumulation behavior. Consider now the input batch $\beta_e = (l_e, d_e, l_e)$ where $d_e = \phi_i^s / v_i^s$. The two batches must be in contact for the marking to remain constant. It holds:

$$\begin{array}{rcl} l_e + l_o & = & s_i \\ l_e d_e + l_o d_i^{\max} & = & q_i^s \end{array} \quad \Longrightarrow \quad \begin{array}{rcl} l_e + l_o & = & s_i \\ l_e \phi_i^s / v_i^s + l_o d_i^{\max} & = & q_i^s \end{array}$$

and it is easy to prove that if $s_i d_i^{\max} = Q_i > q_i^s > \phi_i^s \ \delta_i^{\min,s} = \phi_i^s \ s_i / v_i^s$ holds this system with unknown l_e and l_o admits a single solution given by (5).

In case (4) the place is full, hence it must contain a single dense batch with a length equal to the length of the place. Furthermore, since it is not in a free behavior, it must be in accumulation behavior. □

The previous result allows one to *abstract* the marking of batch places in an SS into a simple vector of marking quantities. This will be instrumental in characterizing the steady state of a cGBPN.

4.3 Steady state of batches nets with assigned transfer speeds

Let us first consider a result that applies to cGBPNs with assigned transfer speed in batch places, i.e., nets where $\boldsymbol{v} = \boldsymbol{V}$. In such a case the transfer speed vector $\boldsymbol{v}^s$ is not a decision variable and the steady state can be described by a pair $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s)$.

Proposition 4.5 *Given a cGBPN $\langle N, \boldsymbol{m}_0 \rangle$ with $P^D = T^D = \emptyset$, consider the following constraint set (CS):*

$$\left\{ \begin{array}{lll} (a) & \boldsymbol{0} \leq \boldsymbol{y} \leq \boldsymbol{\Phi} & \\ (b) & Q_i \geq q_i \geq Pre(p_i, \cdot) \cdot \boldsymbol{y} \cdot s_i / V_i & (\forall p_i \in P^B) \\ (c) & Post(p_i, \cdot) \cdot \boldsymbol{y} \leq V_i \cdot d_i^{\max} & (\forall p_i \in P^B) \\ (d) & \boldsymbol{C} \cdot \boldsymbol{y} = \boldsymbol{0} & \\ (e) & \boldsymbol{q} \in RQ(N, \boldsymbol{m}_0) & \end{array} \right. \qquad (6)$$

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. (6) *with $\boldsymbol{q} = \mu(\boldsymbol{m}^s)$ and $\boldsymbol{y} = \boldsymbol{\varphi}^s$.*

(B) If $(\boldsymbol{q}, \boldsymbol{y})$ is a solution of eq. (6) *then there exists a reachable steady state $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s)$ with $\boldsymbol{m}^s = \nu(\boldsymbol{q}, \boldsymbol{y})$ and $\boldsymbol{\varphi}^s = \boldsymbol{y}$.*

Proof. The two conditions are separately proved.

(Part A) Since $\boldsymbol{m}^s$ is reachable, then by definition, constraint (6).e is satisfied. Furthermore, $\boldsymbol{\varphi}^s$ is an admissible firing vector that satisfies eq. (1), hence eq. (6).a and eq. (6).c also hold. The assumption that $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s)$ is a steady state has two implications. By Proposition 4.2, eq. (6).d holds. Second, by Proposition 4.3 and eq. (2), eq. (6).b holds.

(Part B) Assume $(\boldsymbol{q}, \boldsymbol{y})$ is a solution of eq. (6). We first claim that if any marking $\boldsymbol{m} \in \mu^{-1}(\boldsymbol{q})$ is reachable, as implied by eq. (6).e, then the regular marking $\boldsymbol{m}^s = \nu(\boldsymbol{q}, \boldsymbol{y})$ is also reachable. This can be shown in two steps.

In a first step, from $\boldsymbol{m}$ blocking all transitions the marking quantity in each batch place accumulates in a single *output dense* batch reaching a marking $\boldsymbol{m}^{od} \in \mu^{-1}(\boldsymbol{q})$.

In a second step, from $\boldsymbol{m}^{od}$ it is possible to choose an IFF vector $\boldsymbol{\varphi}^s = \boldsymbol{y}$. In fact:

- eq. (1).a is equivalent to eq. (6).a;
- eq. (1).b is trivially satisfied, as $\boldsymbol{v} = \boldsymbol{V}$ by assumption;
- eq. (1).c is not relevant as we consider a net without discrete nodes;
- eq. (1).d and eq. (1).e are satisfied by (6).d;
- eq. (1).f is equivalent to eq. (6).c;
- eq. (1).g is also satisfied by eq. (6).c, that using eq. (6).d can be rewritten as $Pre(p_i, \cdot) \cdot \boldsymbol{y} \leq q_i V_i / s_i \leq V_i d_i^{\text{out}}$.

One can easily verify that from marking $\boldsymbol{m}^{od}$ the application of the IFF vector $\boldsymbol{\varphi}^s$ yields the regular marking $\boldsymbol{m}^s$ in a time $\tau \leq \max_{p_i \in P^B}\{\delta_i^{\min}\}$. Finally, once $\boldsymbol{m}^s$ is reached, the IFF vector $\boldsymbol{\varphi}^s$ can still be applied but will not change the marking thus $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s)$ is a reachable steady state. □

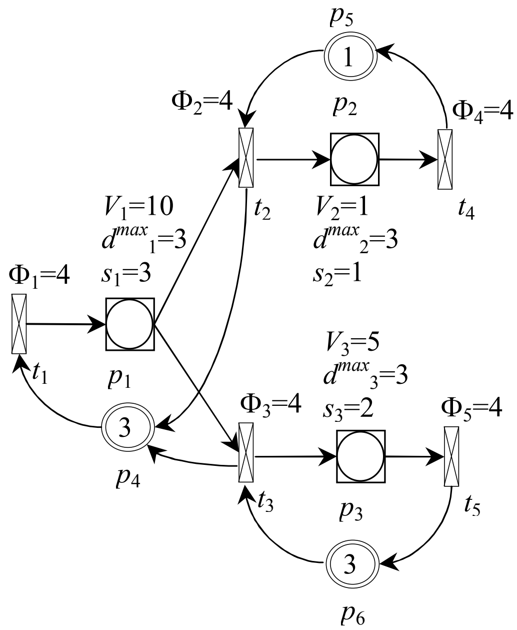

Figure 4: Net in Example 4.7.

Remark 4.6 *In CS* (6) *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 [17] 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 cGBPN where $P^D = T^D = \emptyset$. *Thus, we may usually write (see Section 2.4)*

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

■

Example 4.7 *Consider the net in Fig. 4 where the initial marking is* $\boldsymbol{m}_0 = [\emptyset\ \emptyset\ \emptyset\ 3\ 1\ 3]^T$, *the maximal capacities of batch places are* $Q_1 = 9$, $Q_2 = 3$ *and* $Q_3 = 6$, *the maximal flow vector is* $\boldsymbol{\Phi} = [4\ 4\ 4\ 4\ 4]^T$ *and the constant transfer speed vector is* $\boldsymbol{V} = [10\ 1\ 5]^T$.

[2] For details see the suggested reference.

According to Proposition 4.5, a steady state must verify CS (6)*:*

$$\begin{cases} (a) & 0 \leq y_j \leq 4 \qquad (j = 1, \ldots, 5) \\ (b) & 9 \geq q_1 \geq (y_2 + y_3) \cdot 3/10 \\ & 3 \geq q_2 \geq y_4 \\ & 6 \geq q_3 \geq y_5 \cdot 2/5 \\ (c) & y_2 + y_3 \leq 30 \\ & y_4 \leq 3 \\ & y_5 \leq 15 \\ (d) & y_1 = y_2 + y_3 \\ & y_2 = y_4 \\ & y_3 = y_5 \\ (e) & q_1 = z_1 - z_2 - z_3 \\ & q_2 = z_2 - z_4 \\ & q_3 = z_3 - z_5 \\ & q_4 = 3 - z_1 + z_2 + z_3 \\ & q_5 = 1 - z_2 + z_4 \\ & q_6 = 3 - z_3 + z_5 \end{cases} \tag{7}$$

Eqs. (e) derive from the state equation of the net:

$$\boldsymbol{q} = \mu(\boldsymbol{m}_0) + \boldsymbol{C} \cdot \boldsymbol{z}$$

where $\mu(\boldsymbol{m}_0) = [0\ 0\ 0\ 3\ 1\ 3]^T$*,* $\boldsymbol{z} = [z_1\ z_2\ z_3\ z_4\ z_5]^T$ *is the (unknown) firing vector of the sequence that reaches the steady state marking quantity* $\boldsymbol{q}$ *and the incidence matrix of the net is*

$$C = \begin{bmatrix} 1 & -1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 & 0 \\ 0 & 0 & 1 & 0 & -1 \\ -1 & 1 & 1 & 0 & 0 \\ 0 & -1 & 0 & -1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix}.$$

We select as objective function to maximize:

$$f = 10 \cdot y_4 + y_5 - 10^{-6} \cdot (z_1 + z_2 + z_3 + z_4 + z_5)$$

for the following reasons.

- *The flow* y_4 *(resp.,* y_5*) of transition* t_4 *(resp.,* t_5*) represents the production of a high (resp., low) quality product and the value of one item of the high quality product is 10 times higher than the value of one item of a low quality product.*
- *A very small negative weight has been given to all components of the firing vector* $\boldsymbol{z}$*. This is heuristically equivalent to search for the shortest firing sequence that reaches the steady state.*

The optimal solution $(\boldsymbol{q}, \boldsymbol{y})$ *of CS* (6) *and the optimal firing vector* $\boldsymbol{z}$ *are:*

$$\boldsymbol{q} = \begin{bmatrix} 1.2 \\ 1 \\ 1.2 \\ 1.8 \\ 0 \\ 1.8 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} 4 \\ 1 \\ 3 \\ 1 \\ 3 \end{bmatrix} \quad \textit{and} \quad \boldsymbol{z} = \begin{bmatrix} 3.4 \\ 1 \\ 1.2 \\ 0 \\ 0 \end{bmatrix}.$$

According to the value of the firing vector $\boldsymbol{z}$*, to reach the marking quantity that characterizes the steady state we need to fire transition* t_1 *(resp.,* t_2*,* t_3*) in a quantity equal to* 3.4 *(resp.,* 1, 1.2*) while other transitions need not fire. This can be done, as an example, as follows. First fire transition* t_1 *moving a quantity* 3 *in place* p_1*. Wait for the marking quantity in* p_1 *to generate a dense output batch and then fire transition* t_2 *(resp.,* t_3*,* t_1*) at their maximal speed moving a quantity* 1 *(resp.,* 1.2, 0.4*) in place* p_2 *(resp.,* p_3*,* p_1*). Wait for all these marking quantities to generate dense output batches. Now the stationary firing vector can be applied and in a short transient all places will reach a stationary marking in a regular form.* ■

4.4 Steady state of batches nets with variable transfer speeds

The previous result can be extended to cGBPNs where the transfer speed of a batch place may vary as follows.

Consider CS (6) and assume that batch places have variable transfer speed. In equation (b), $\delta_i^{\min,s}$ is not a constant any more and should be replaced by s_i/v_i. In equation (c), the assigned 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:

$$\begin{cases} (a) & \mathbf{0} \leq \boldsymbol{y} \leq \boldsymbol{\Phi} \\ (b') & Q_i \geq q_i & (\forall p_i \in P^B) \\ (b'') & q_i \geq Pre(p_i, \cdot) \cdot \boldsymbol{y} \cdot s_i/v_i & (\forall p_i \in P^B) \\ (c) & Post(p_i, \cdot) \cdot \boldsymbol{y} \leq v_i \cdot d_i^{\max} & (\forall p_i \in P^B) \\ (d) & \boldsymbol{C} \cdot \boldsymbol{y} = \mathbf{0} \\ (e) & \boldsymbol{q} \in RQ(N, \boldsymbol{m}_0) \\ (f) & 0 \leq v_i \leq V_i & (\forall p_i \in P^B) \end{cases} \tag{8}$$

We can thus state the following result.

Proposition 4.8 *Given a cGBPN $\langle N, \boldsymbol{m}_0 \rangle$ with $P^D = T^D = \emptyset$, consider the following constraint set:*

$$\begin{cases} (a) & \boldsymbol{0} \leq \boldsymbol{y} \leq \boldsymbol{\Phi} \\ (b') & Q_i \geq q_i & (\forall p_i \in P^B) \\ (c) & Post(p_i, \cdot) \cdot \boldsymbol{y} \leq v_i \cdot d_i^{\max} & (\forall p_i \in P^B) \\ (d) & \boldsymbol{C} \cdot \boldsymbol{y} = \boldsymbol{0} \\ (e) & \boldsymbol{q} \in RQ(N, \boldsymbol{m}_0) \\ (f) & 0 \leq v_i \leq V_i & (\forall p_i \in P^B) \end{cases} \tag{9}$$

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 the structure of the net, have previously been defined.

Given a solution of $(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$ of (9), 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. (9) 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. (9) 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.

While CS (8) can be used to characterize steady states following Proposition 4.5, it is non-linear in eq. (b'') and finding a solution is not practical. We relax constraint (b'') thus obtaining CS (9) whose admissible solutions are a superset of the solutions of CS (8). 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. If $(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$ is a solution of CS (8), resp., CS (9), then $(\boldsymbol{q}, \varrho\boldsymbol{y}, \boldsymbol{v})$ with $\varrho \in [0, 1]$ is also a solution of CS (8), resp., CS (9).

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 4.5 $(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$ satisfies CS (8), which in turn satisfies CS (9).

(Part B) Assume $(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$ is a solution of CS (9) 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 (8) for all $p_i \in P^B$, then $\varrho^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v}) = 1$. From the Fact 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 (8), and hence by Proposition 4.5 $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s, \boldsymbol{v}^s)$ is a steady state.

If, on the contrary, $\mathcal{I} \neq \emptyset$, then it is easy to show (recalling the definition of ϱ^* and using the Fact above) that $(\boldsymbol{q}, \varrho\boldsymbol{y}, \boldsymbol{v})$ satisfies constraint (b'') in CS (8) if and only if $\varrho \in [0, \varrho^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})]$. Hence by Proposition 4.5 $(\boldsymbol{m}^s, \boldsymbol{\varphi}^s, \boldsymbol{v}^s)$ is a steady state if and only if $\varrho \in [0, \varrho^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})]$. □

We now discuss the optimality of the solution characterized by the previous proposition.

Remark 4.9 *Although Proposition 4.8 allows one to characterize all possible steady states for a cGBPN, unlike Proposition 4.5 it may fail to determine an optimal steady state.*

As an example, assume the objective is that of finding a steady state that maximize the objective function $J = f(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$. *By solving CS* (9) *so as to optimize this objective function we obtain an optimal solution* $(\boldsymbol{q}', \boldsymbol{y}', \boldsymbol{v}')$ *and we compute, within this family, the solution of CS* (8) *that is obviously* $(\boldsymbol{q}', \varrho^*(\boldsymbol{q}', \boldsymbol{y}', \boldsymbol{v}') \cdot \boldsymbol{y}', \boldsymbol{v}')$.

However, a non-optimal solution $(\boldsymbol{q}'', \boldsymbol{y}'', \boldsymbol{v}'')$ *of CS* (9) *may lead to an even better solution of CS* (8) *that takes the form* $(\boldsymbol{q}'', \varrho^*(\boldsymbol{q}'', \boldsymbol{y}'', \boldsymbol{v}'') \cdot \boldsymbol{y}'', \boldsymbol{v}'')$ *and is such that*

$$J'' = f(\boldsymbol{q}'', \varrho^*(\boldsymbol{q}'', \boldsymbol{y}'', \boldsymbol{v}'') \cdot \boldsymbol{y}'', \boldsymbol{v}'') > f(\boldsymbol{q}', \varrho^*(\boldsymbol{q}', \boldsymbol{y}', \boldsymbol{v}') \cdot \boldsymbol{y}', \boldsymbol{v}') = J'$$

. ■

The following complete example shows how to apply the proposed procedure and in particular considers the case of a net with variable transfer speed where the computed solution fails to be optimal for CS (8), as mentioned in the previous remark.

Example 4.10 *Consider the same net discussed in Example 4.7 and represented in Fig. 4. We now assume transfer speeds* $\boldsymbol{v}$ *are variable and bounded by the maximal transfer speed vector* $\boldsymbol{V} = [10\ 1\ 5]^T$.

According to Proposition 4.8, a steady state must verify CS (9)*:*

$$
\begin{cases}
(a) & 0 \le y_j \le 4 \qquad (j = 1, \ldots, 5) \\
(b') & 9 \ge q_1 \\
 & 3 \ge q_2 \\
 & 6 \ge q_3 \\
(c) & y_2 + y_3 \le 3 \cdot v_1 \\
 & y_4 \le 3 \cdot v_2 \\
 & y_5 \le 3 \cdot v_3 \\
(d) & y_1 = y_2 + y_3 \\
 & y_2 = y_4 \\
 & y_3 = y_5 \\
(e) & q_1 = z_1 - z_2 - z_3 \\
 & q_2 = z_2 - z_4 \\
 & q_3 = z_3 - z_5 \\
 & q_4 = 3 - z_1 + z_2 + z_3 \\
 & q_5 = 1 - z_2 + z_4 \\
 & q_6 = 3 - z_3 + z_5 \\
(f) & 0 \le v_1 \le 10 \\
 & 0 \le v_2 \le 1 \\
 & 0 \le v_3 \le 5
\end{cases} \tag{10}
$$

We select the following objective function to maximize:

$$
f = 10 \cdot y_4 + y_5 - 0.01 \cdot (v_1 + v_2 + v_3) + 10^{-4} \cdot (q_1 + q_2 + q_3) - 10^{-6} \cdot (z_1 + z_2 + z_3 + z_4 + z_5) \tag{11}
$$

that is similar to the one discussed in Example 4.7 but contains two additional terms:

- *The small negative term given to the speed of batch places aims to reduce the cost of operating the transfer elements.*
- *A smaller positive weight is given to the marking quantities in the batch places in the steady state. This is heuristically equivalent to search, among the optimal solutions, for those that have a higher marking quantity in the batch places and thus have a better chance of satisfying constraint (b'').*

The optimal solution $(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$ *of CS* (9) *and the optimal firing vector* $\boldsymbol{z}$ *are:*

$$
\boldsymbol{q} = \begin{bmatrix} 3 \\ 1 \\ 3 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{y} = \begin{bmatrix} 4 \\ 3 \\ 1 \\ 3 \\ 1 \end{bmatrix}, \quad \boldsymbol{v} = \begin{bmatrix} 4/3 \\ 1 \\ 1/3 \end{bmatrix} \quad \text{and} \quad \boldsymbol{z} = \begin{bmatrix} 7 \\ 1 \\ 3 \\ 0 \\ 0 \end{bmatrix}.
$$

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

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

It holds:

$$q_1 = 3 < 9 = q_{\min,1}, \quad q_2 = 1 < 3 = q_{\min,2}, \quad q_3 = 3 < 6 = q_{\min,3}.$$

Since no batch place satisfies constraint (b'') it holds $\mathcal{I} = \{1, 2, 3\}$ and

$$\varrho^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v}) = \min_{i \in \mathcal{I}} q_i / q_{\min,i} = 1/3.$$

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 state. The best value of the objective function is given by $\varrho = \varrho^*(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v}) = 1/3$ that gives $f(\boldsymbol{q}, \boldsymbol{y}/3, \boldsymbol{v}) = 10.31$.

To prove that this solution is not optimal for CS (8), we now consider a different solution of CS (9):

$$\boldsymbol{q}' = \begin{bmatrix} 3 \\ 1 \\ 3 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad \boldsymbol{y}' = \begin{bmatrix} 4 \\ 1 \\ 3 \\ 1 \\ 3 \end{bmatrix}, \quad \boldsymbol{v}' = \begin{bmatrix} 4 \\ 1 \\ 2 \end{bmatrix}.$$

Since in this case all constraints (b'') are satisfied, $(\boldsymbol{q}', \boldsymbol{y}', \boldsymbol{v}')$ is also a solution of CS (8) with objective function $f(\boldsymbol{q}', \boldsymbol{y}', \boldsymbol{v}') = 12.93 > f(\boldsymbol{q}, \boldsymbol{y}/3, \boldsymbol{v}) = 10.31$.

4.5 Periodic stationary behaviors

The assumption that a stationary behavior can be characterized by constant marking, IFF vector and place speed vector is rather strong. Following [14], we can consider stationary evolutions where all these quantities change in time (e.g., periodically) but have a constant average value. Eq. (6) still gives necessary conditions for the existence of such a behavior, if the solution $(\boldsymbol{q}, \boldsymbol{y}, \boldsymbol{v})$ is understood as expressing the average values of: marking quantity $\boldsymbol{q} = E[\boldsymbol{q}^s(\tau)]$, IFF vector $\boldsymbol{y} = E[\boldsymbol{\varphi}^s(\tau)]$ and place speed vector $\boldsymbol{v} = E[\boldsymbol{v}^s(\tau)]$.

Example 4.11 *Consider a net obtained from the one shown in Fig. 3 removing place p_3 and transitions t_2 and t_3. A possible evolution of this net, assuming for sake of simplicity a constant transfer speed $v_2(\tau) = V_2 = 1$, is shown in Fig. 5.*

At time $\tau = 5$ the net reaches a periodic stationary behavior where $E[\varphi_1^s] = E[\varphi_2^s] = y_1 = y_2 = 1$, $E[q_1^s] = q_1 = 1.75$ and $E[q_2^s] = q_2 = 6.25$. In this case the k-th batch entering

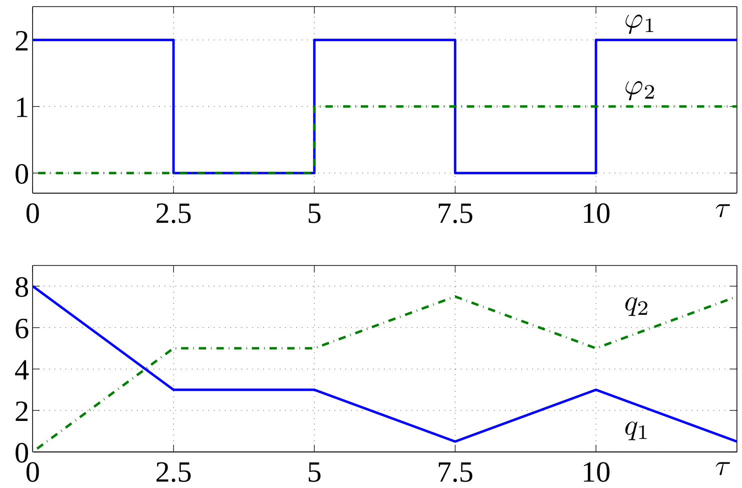

Figure 5: An evolution of the net in Example 4.11.

place p_2, denoted β_k, is created at time $\tau_k = 5(k-1)$, destroyed at time $\tau_k = 5(k+1)$ and evolves according to:

$$\beta_k(\tau) = \begin{cases} (\tau - \tau_k, 2, \tau - \tau_k) & \textit{if } \tau \in [\tau_k, \tau_k + 2.5] \\ (2.5, 2, \tau - \tau_k) & \textit{if } \tau \in [\tau_k + 2.5, \tau_k + 5] \\ (5 - 0.5(\tau - \tau_k), 2, 5) & \textit{if } \tau \in [\tau_k + 5, \tau_k + 10] \end{cases}$$

5 Conclusions and future work

In this paper we have presented a model called controlled Generalized Batches Petri nets and have developed new linear algebraic techniques for its analysis. Two main contributions have been presented.

The first contribution lies in the fact that although we consider the same GBPN model that has already been presented in the literature, we showed how to associate to this model a different semantics, considering that the firing flow of continuous and batch transitions and the transfer speed of batch places are control variables. We have proposed a linear programming problem to compute the instantaneous firing flow vector and the instantaneous transfer speed vector solving an optimization problem, where the objective function depends on the control goal.

The second contribution consists in the analysis of the steady state behavior of cGBPN. Computing a steady state requires solving a programming problem that is linear if the transfer speeds are preassigned, while it becomes nonlinear if the transfer speeds are control variables as well. A viable technique to compute a family of solutions by linear relaxation of the nonlinear problem has been presented. The optimality of steady states for given linear objective functions has also been addressed.

One line is open for future research, as we have discussed in a final example: we plan to characterize more general stationary evolutions where the marking, the firing flow vector and the transfer speed vector change in time (e.g., periodically) but have a constant average value. We also plan to apply to cGBPN optimal techniques based on perturbation analysis that have already been used [27] in the context of continuous Petri nets.

References

[1] N. Audry, I. Demongodin, and F. Prunet. Modelling of high throughput production lines by using generic models described in batches Petri nets. In *IEEE International Conference on Robotics and Automation*, pages 807–813, 1994.

[2] N. Audry and F. Prunet. Controlled batches Petri nets. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 1849–1854, 1994.

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

[4] M. Caradec and F. Prunet. Modelling of hybrid flexible production systems by coloured batches Petri nets. *European Journal of Automation*, 32(9&10):1255–1269, 1998.

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

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

[7] I. Demongodin. Generalised batches Petri net: hybrid model for high speed systems with variable delays. *Discrete Event Dynamic Systems*, 11(1&2):137–162, 2001.

[8] I. Demongodin. Modeling and analysis of transportation networks using batches Petri nets with controllable batch speed. In G. Franceschinis and Karsten W., editors, 30^{th} *International Conference on Application and Theory of Petri Nets*, volume 5606 of *Lecture Notes in Computer Science*, pages 204–222. Springer Berlin Heidelberg, 2009.

[9] I. Demongodin, N. Audry, and F. Prunet. Batches Petri nets. In *IEEE International Conference on Systems, Man and Cybernetics*, pages 607–617, 1993.

[10] I. Demongodin, M. Caradec, and F. Prunet. Fundamental concepts of analysis in batches Petri nets. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, pages 845–850, 1998.

[11] 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.

[12] I. Demongodin and A. Giua. Stationary behavior of controlled generalised batches Petri nets. In *Proc. WODES'12: 11th Work. on Discrete Event Systems*, Guadalajara, Mexico, October 2012.

[13] M. Seçkin Durmuş, U. Yıldırım, and M. Turan Söylemez. Interlocking system design for ERTMS / ETCS: An approach with batches Petri nets. In *Proc. WODES'12: 11th Work. on Discrete Event Systems*, Guadalajara, Mexico, October 2012.

[14] B. Gaujal and A. Giua. Optimal stationary behavior for a class of timed continuous Petri nets. *Automatica*, 40(9):1505–1516, 2004.

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

[16] J. Júlvez, C. Mahulea, and C.-R. Vázquez. SimHPN: A matlab toolbox for simulation, analysis and design with hybrid Petri nets. *Nonlinear Analysis: Hybrid Systems*, 6(2):806–817, 2012.

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

[18] F. Kaakai, S. Hayat, and A. El Moudni. Quantitative assessment of travellers' connection times into a multimodal hub owing to batches petri nets. In *IMACS'2005, the 17th IMACS World Congress on Scientific Computation, Applied Mathematics and Simulation*, Paris, France, july 2005.

[19] 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*, 1(4):451–465, 2007.

[20] C. Mahulea, A. Giua, L. Recalde, C. Seatzu, and M. Silva. Optimal model predictive control of timed continuous Petri nets. *IEEE Trans. on Automatic Control*, 53(7):1731–1735, 2008.

[21] 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.

[22] 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*, 21(4):427–497, 2011.

[23] M. Svádová and Z. Hanzálek. Modeling of systems with delays using hybrid Petri nets. In *IFAC Conference on Control Systems Design, Bratislava*, 2003.

[24] L. Wang, C. Mahulea, J. Júlvez, and M. Silva. Minimum-time decentralized control of choice-free continuous Petri nets. *Nonlinear Analysis: Hybrid Systems*, 7:39–53, 2013.

[25] Y. Wang. *High Volume Conveyor Sortation System Analysis*. PhD dissertation, Georgia Institute of Technology, 2006.

[26] Y. Wang and C. Zhou. Fluid-based simulation approach for high volume conveyor transportation systems. *Journal of Systems Science and Systems Engineering*, 13:297–317, 2004.

[27] Y. Wardi, A. Giua, and C. Seatzu. IPA for continuous stochastic marked graphs. *Automatica*, 49(5):1204–1215, 2013.