

Algebraic System Analysis of Timed Petri Nets

G. Cohen, S. Gaubert and J. P. Quadrat

Abstract

We show that Continuous Timed Petri Nets (CTPN) can be modeled by generalized polynomial recurrent equations in the $(\min, +)$ semiring. We establish a correspondence between CTPN and Markov decision processes. We survey the basic system theoretical results available: behavioral (input-output) properties, algebraic representations, asymptotic regime. A particular attention is paid to the subclass of stable systems (with asymptotic linear growth).

1 Introduction

The fact that a subclass of Discrete Event Systems equations write linearly in the $(\min, +)$ or in the $(\max, +)$ semiring is now almost classical [9, 2]. The $(\min, +)$ linearity allows the presence of synchronization and saturation features but unfortunately prohibits the modeling of many interesting phenomena such as “birth” and “death” processes (multiplication of tokens) and concurrency. The purpose of this paper is to show that after some simplifications, these additional features can be represented by polynomial recurrences in the $(\min, +)$ semiring.

We introduce a fluid analogue of general Timed Petri Nets (in which the quantities of tokens are real numbers), called Continuous Timed Petri Nets (CTPN). We show that, assuming a stationary routing policy, the counter variables of a CTPN satisfy recurrent equations involving the operators $\min, +, \times$. We interpret CTPN equations as dynamic programming equations of classical Markov Decision Problems: CTPN can be seen as the dedicated hardware executing the value iteration.

We set up a hierarchy of CTPN which mirrors the natural hierarchy of optimization problems (deterministic vs. stochastic, discounted vs. ergodic). For each level and sublevel of this hierarchy, we recall or introduce the required algebraic and analytic tools, we provide input-output characterizations and give asymptotic results.

The paper is organized as follows. In §2, we give the dynamic equations satisfied by general Petri Nets under the earliest firing rule. The counter

TPN by diagonal change of variable). 3. CTPN with fixed birth/death rate correspond to the well studied class of discounted Dynamic Programming recurrences.

2 Recurrent Equations of Timed Petri Nets

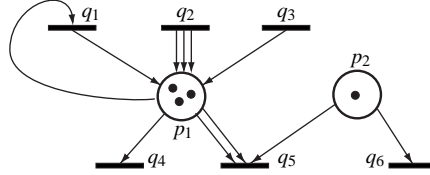


Figure 1: Notation for Petri Nets. $P = \{p_1, p_2\}$, $Q = \{q_1, \dots, q_6\}$, $p_1^{\text{out}} = \{q_1, q_4, q_5\}$, $p_1^{\text{in}} = \{q_1, q_2, q_3\}$, $p_2^{\text{out}} = \{q_5, q_6\}$, $M_{q_5 p_1} = 2$, $M_{p_1 q_2} = 3$, $m_{p_1} = 3$, $m_{p_2} = 1$.

Definition 2.1 (TPNM). A Timed Petri Net with Multipliers (TPNM) is a valued bipartite graph given by a 5-tuple $N = (P, Q, M, m, \cdot)$.

1. The finite set P is called the set of places. A place may contain tokens which travel from place to place according to a firing process described later on.
2. The finite set Q is called the set of transitions. A transition may fire. When it fires, it consumes and produces tokens.
3. $M = N^{P \times Q} \times Q \times P$. M_{pq} (resp. M_{qp}) gives the number of edges from transition q to place p (resp. from place p to transition q). In particular, the zero value for M corresponds to the absence of edge.
4. $m = N^P$: m_p denotes the number of tokens being initially in place p (initial marking).
5. $\cdot = N^P$: \cdot_p gives the minimal time a token must spend in place p before becoming available for consumption by downstream transitions¹. It will be called holding time of the place throughout this paper.

We denote by r^{out} the set of vertices (places or transitions) downstream a vertex r and r^{in} the set of vertices upstream r . Formally,

$$r^{\text{out}} = \{s \mid M_{sr} = 0\}, \quad r^{\text{in}} = \{s \mid M_{rs} = 0\}.$$

In order to specify a unique behavior of the system, we equip TPN with *routing policies*.

¹Without loss of modeling power, the firing of transitions is supposed to be instantaneous (i.e. it involves no delay in consuming and producing tokens).

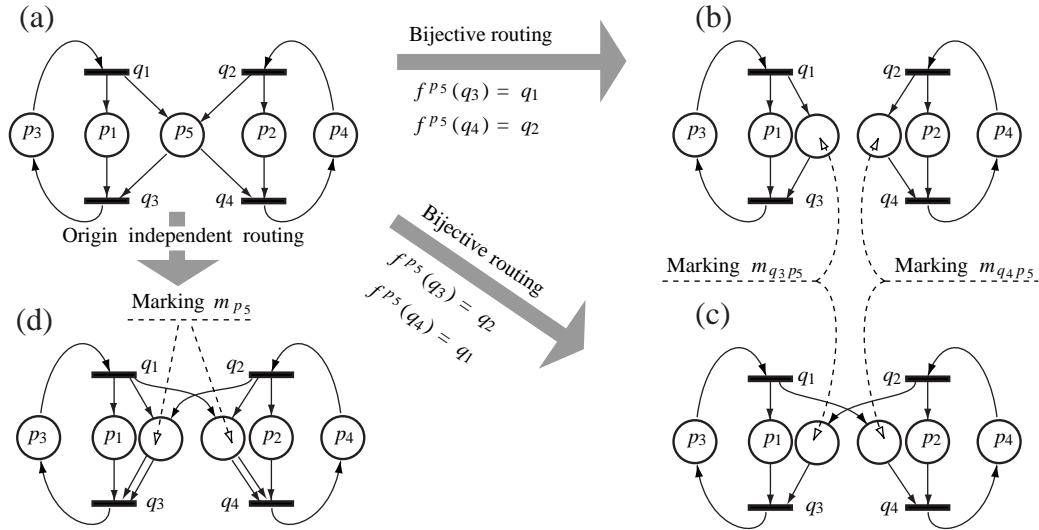


Figure 2: A Balanced Petri Net under various Routing Policies

Injective Routing We say that the routing function f^p at place p is *injective* if there is a map $f^p : p^{\text{out}} \rightarrow p^{\text{in}}$ such that

$$q, \quad \frac{p}{qq} = 0 \quad q = f^p(q) . \quad (3.8)$$

That is, all the tokens routed to q at place p come for a single transition $f(q)$. Such routings occur frequently when tokens correspond to *resources* (e.g. pallets) which follow some well defined physical routes. An injective routing exists if $|p^{\text{out}}| \leq |p^{\text{in}}|$.⁵ Indeed, the following stronger condition is often satisfied in practice (e.g. in Fig. 2a).

Definition 3.2 (Balanced TPN). A TPN is *balanced* if $|p|, |p|^{\text{out}} = |p|^{\text{in}}$.

In this particular case, we shall speak of *bijective* routing policies (since f^p becomes a bijection $p^{\text{out}} \rightarrow p^{\text{in}}$). We shall see later on that injective and bijective routing policies lead to tractable classes of systems.

4 Timed Event Graphs and (min,+) Linear Systems

4.1 Ordinary and Generalized Timed Event Graphs

Definition 4.1 (Timed Event Graphs). A *Continuous Timed Event Graph with Multipliers* (CTEGM) is a CTPN such that there is exactly one

⁵We denote by $|X|$ the cardinal of a set X .

