Unbounded Product-Form Petri Nets[*][†]

Patricia Bouyer[1], Serge Haddad[2], and Vincent Jugé[3]

1 **LSV, CNRS, ENS Paris-Saclay, Université Paris-Saclay, France**
2 **LSV, CNRS, ENS Paris-Saclay, Inria, Université Paris-Saclay, France**
3 **LSV, CNRS, ENS Paris-Saclay, Université Paris-Saclay, France**

Abstract

Computing steady-state distributions in infinite-state stochastic systems is in general a very difficult task. Product-form Petri nets are those Petri nets for which the steady-state distribution can be described as a natural product corresponding, up to a normalising constant, to an exponentiation of the markings. However, even though some classes of nets are known to have a product-form distribution, computing the normalising constant can be hard. The class of (closed) Π^3-nets has been proposed in an earlier work, for which it is shown that one can compute the steady-state distribution efficiently. However these nets are bounded. In this paper, we generalise queuing Markovian networks and closed Π^3-nets to obtain the class of open Π^3-nets, that generate infinite-state systems. We show interesting properties of these nets: (1) we prove that liveness can be decided in polynomial time, and that reachability in live Π^3-nets can be decided in polynomial time; (2) we show that we can decide ergodicity of such nets in polynomial time as well; (3) we provide a pseudo-polynomial time algorithm to compute the normalising constant.

1998 ACM Subject Classification D.2.2 Petri Nets

Keywords and phrases Performance evaluation, infinite-state systems, Petri nets, steady-state distribution

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2017.31

1 Introduction

Quantitative analysis of stochastic infinite-state systems. Performance measures of stochastic systems can be roughly classified in two categories: those related to the transient behaviour as expressed by a temporal logic formula and those related to the long-run behaviour whose main measure is the steady-state distribution (when it exists).

There are different relevant questions concerning the steady-state distribution of infinite-state systems: (1) given a state and a threshold, one can ask whether the steady-state probability of this state is above (or below) the threshold; (2) given a state, one can compute the steady-state probability of this state, either in an exact way, or in an accurate approximate way; and (3) one can give a symbolic representation of both the set of reachable states and its associated distribution (or an accurate approximation thereof).

Clearly the last question is the most difficult one, and the first breakthrough in that direction has been obtained in the framework of open queuing Markovian networks: in those systems, the measure of a state is obtained as the product of terms, where each term is related to the parameters of a queue (service and visit rate) and the number of clients in the queue [13]. In order to get a probability distribution over the set of reachable states,

[*] A full version of the paper is available at `https://arxiv.org/abs/1708.05847`.
[†] This work has been supported by ERC project EQualIS (FP7-308087).

28th International Conference on Concurrency Theory (CONCUR 2017).
Editors: Roland Meyer and Uwe Nestmann; Article No. 31; pp. 31:1–31:16
Leibniz International Proceedings in Informatics
LIPICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

this product is normalised by a constant, whose computation is easy when the service rates of the queues do not depend on the number of clients. This work has been adapted to closed networks, and the main contribution in [9] consists in computing the normalising constant without enumerating the (finite) reachability set, leading to an algorithm which runs in polynomial-time w.r.t. the size of the network and the number of clients. Later, Markov chains generated by a stochastic Petri net with a single unbounded place (that is, quasi-birth death processes) have been investigated [8], and an algorithm which approximates up to arbitrary precision the steady-state distribution has been proposed; however the complexity of the algorithm is very high, since it requires the computation of the finite reachability sets of some subnets, whose size may be non primitive recursive. More recently, the abstract framework of (infinite-state) Markov chains with a finite eager attractor (e.g. probabilistic lossy channel systems) has been used to develop an algorithm which approximates up to arbitrary precision the steady-state distribution as well [1], but there is no complexity bound for the algorithm.

Product-form Petri nets. While queuing networks are very interesting since they allow for an explicit representation of the steady-state distribution, they lack two important features available in Petri nets [19, 20], which are very relevant for modelling concurrent systems: resource competition and process synchronisation. So very soon researchers have tried to get the best of the two formalisms and they have defined subclasses of Petri nets for which one can establish product-form steady-state distributions. Historically, solutions have been based on purely behavioural properties (i.e. by an analysis of the reachability graph) like in [15], and then progressively have moved to more and more structural characterisations [16, 6]. Building on the work of [6], [11] has established the first purely structural condition for which a product-form steady-state distribution exists, and designed a polynomial-time algorithm to check for the condition (see also [17] for an alternative characterisation). These nets are called Π^2-nets. However the computation of the normalising constant remains a difficult open issue since a naive approach in the case of finite-state Π^2-nets would require to enumerate the potentially huge reachability state-space. Furthermore, the lower bounds shown in [10] for behavioural properties of Π^2-nets strongly suggest that the computation of the normalising constant can probably not be done in an efficient way. In [6, 23], the authors introduce semantical classes of product-form Petri nets for which this constant is computed in pseudo-polynomial time. However their approach suffers two important drawbacks: (1) checking whether a net fufills this condition as least as hard as the reachability problem, and (2) the only syntactical class for which this condition is fulfilled boils down to queuing networks.

To overcome this problem, the model of Π^3-nets is defined in [10] as a subclass of Π^2-nets obtained by structuring the synchronisation between concurrent activity flows in layers. This model strictly generalises closed product-form queuing networks (in which there is a single activity flow). Two interesting properties of those nets is that liveness for Π^3-nets and reachability for live Π^3-nets can both be checked in polynomial time. Furthermore, from a quantitative point-of-view, the normalising constant of the steady-state distribution can be efficiently computed using an elaborated dynamic programming algorithm.

Product-form Petri nets have been applied for the specification and analysis of complex systems. From a modelling point-of-view, compositional approaches have been proposed [18, 2] as well as hierarchical ones [12]. Application fields have also been identified, for instance, hardware design and more particularly RAID storage [12], or software architectures [3].

Our contributions. Unfortunately Π^3-nets generate finite-state systems. Here we address this problem by introducing and studying open Π^3-nets. Informally, an open Π^3-net has a main activity flow, which roughly corresponds to an open queuing network, and has other activity flows which are structured as in (standard, or closed) Π^3-nets. More precisely, in the case of a single activity flow this model is exactly equivalent to an open queuing network, but the general model is enriched with other activity flows, raising difficult computation issues. In particular, several places may in general be unbounded. Open Π^3-nets are particularly appropriate when modelling, in an open environment, protocols and softwares designed in layers. In adddition, they allow to specify dynamical management of resources where processes may produce and consume them with no *a priori* upper bound on their number. Our results on open Π^3-nets can be summarised as follows:

- We first establish that the liveness problem can be solved in polynomial time, and that the boundedness as well as the reachability problem in live nets can also be solved in polynomial time. On the other side, we show that the unboundedness, the reachability and even the covering problem become NP-hard without the liveness assumption.
- Contrary to the case of closed Π^3-nets, open Π^3-nets may not be ergodic (that is, there may not exist a steady-state distribution). We design a polynomial-time algorithm to decide ergodicity of an open Π^3-net.
- Our main contribution is the computation of the normalising constant for ergodic live Π^3-nets. Our procedure combines symbolic computations and dynamic programming. The time complexity of the algorithm is polynomial w.r.t. the size of the structure of the net and the maximal value of integers occuring in the description of the net (thus pseudo-polynomial). As a side result, we improve the complexity for computing the normalising constant of closed Π^3-nets that was given in [10] (the complexity was the same, but was assuming that the number of activity flows is a constant).

In Section 2, we introduce and illustrate product-form nets, and recall previous results. In Section 3, we focus on qualitative behavioural properties, while quantitative analysis is developed in Section 4. All proofs are postponed to the full version of this article.

2 Product-form Petri nets

Notations. Let A be a matrix over $I \times J$, one denotes $A(i,j)$ the item whose row index is i and column index is j. When I and J are disjoint, $W(k)$ denotes the row (resp. column) vector indexed by k when $k \in I$ (resp. $k \in J$). Given a real vector $\mathbf{v}$ indexed by I its norm, denoted $\|\mathbf{v}\|$, is defined by $\|\mathbf{v}\| = \sum_{i \in I} |\mathbf{v}(i)|$. Sometimes, one writes $\mathbf{v}_i$ for $\mathbf{v}(i)$. Given two vectors $\mathbf{v}$, $\mathbf{w}$ indexed by I their scalar product denoted $\mathbf{v} \cdot \mathbf{w}$ is defined by $\sum_{i \in I} \mathbf{v}_i \mathbf{w}_i$. Finally, if $\mathbf{v}$ is a vector over I, we define its support as $\mathsf{Supp}(\mathbf{v}) = \{i \in I \mid \mathbf{v}(i) \neq 0\}$.

We briefly recall Petri nets and stochastic Petri nets. The state of a Petri net, called a *marking* is defined by the number of *tokens* contained in every *place*. A Petri net models concurrent activities by *transitions* whose enabling requires tokens to be consumed in some places and then tokens to be produced in some places.

▶ **Definition 1** (Petri net). A *Petri net* is a tuple $\mathcal{N} = (P, T, W^-, W^+)$ where:

- P is a finite set of *places*;
- T is a finite set of *transitions*, disjoint from P;
- W^- and W^+ are $P \times T$ matrices with coefficients in $\mathbb{N}$.

W^- (resp. W^+) is called the backward (resp. forward) incidence matrix, $W^-(p,t)$ (resp. $W^+(p,t)$) specifies the number of tokens consumed (resp. produced) in place p by the *firing* of transition t, and $W^-(t)$ (resp. $W^+(t)$) is the t-th column of W^- (resp. W^+). One

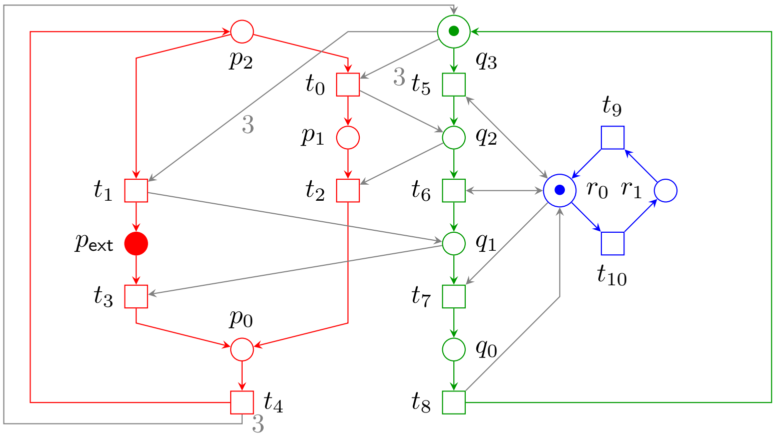


Figure 1 A marked Petri net (initial marking: $q_3 + r_0$).

assumes that for all $t \in T$, $W^-(t) \neq W^+(t)$ (i.e. no useless transition) and for all $t' \neq t$, either $W^-(t) \neq W^-(t')$ or $W^+(t) \neq W^+(t')$ (i.e. no duplicated transition); this will not affect our results.

A *marking* of $\mathcal{N}$ is a vector of $\mathbb{N}^P$; in the sequel we will often see m as a multiset ($m(p)$ is then the number of occurrences of p), or as a symbolic sum $\sum_{p \in P \mid m(p)>0} m(p)\,p$. The symbolic sum $\sum_{p \in P'} p$ will be more concisely written P'. Transition t is *enabled* by marking $m \in \mathbb{N}^P$ if for all $p \in P$, $m(p) \geqslant W^-(p,t)$. When enabled, its firing leads to the marking m' defined by: for all $p \in P$, $m'(p) = m(p) - W^-(p,t) + W^+(p,t)$. This firing is denoted by $m \xrightarrow{t} m'$. The *incidence matrix* $W = W^+ - W^-$ allows one to rewrite the marking evolution as $m' = m + W(t)$ if $m \geqslant W^-(t)$. Given an initial marking $m_0 \in \mathbb{N}^P$, the *reachability set* $\mathcal{R}_{\mathcal{N}}(m_0)$ is the smallest set containing m_0 and closed under the firing relation. When no confusion is possible one denotes it more concisely by $\mathcal{R}(m_0)$. Later, if $m \in \mathcal{R}(m_0)$, we may also write $m_0 \to^* m$. We will call $(\mathcal{N}, m_0)$ a *marked* Petri net

From a qualitative point of view, one is interested by several standard relevant properties including reachability. *Liveness* means that the modelled system never loses its capacity: for all $t \in T$ and $m \in \mathcal{R}(m_0)$, there exists $m' \in \mathcal{R}(m)$ such that t is enabled in m'. *Boundedness* means that the modelled system is a finite-state system: there exists $B \in \mathbb{N}$ such that for all $m \in \mathcal{R}(m_0)$, $\|m\| \leqslant B$. While decidable, these properties are costly to check: (1) Reachability is EXPSPACE-hard in general and PSPACE-complete for 1-bounded nets [7], (2) using results of [21] liveness has the same complexity, and (3) boundedness is EXPSPACE-complete [22]. Furthermore there is a family of bounded nets $\{\mathcal{N}_n\}_{n \in \mathbb{N}}$ whose size is polynomial in n such that the size of their reachability set is lower bounded by some Ackermann function [14].

▶ **Example 2.** An example of marked Petri net is given on Figure 1. Petri nets are represented as bipartite graphs where places are circles containing their initial number of tokens and transitions are rectangles. When $W^-(p,t) > 0$ (resp. $W^+(p,t) > 0$) there is an edge from p (resp. t) to t (resp. p) labelled by $W^-(p,t)$ (resp. $W^+(p,t)$). This label is called the *weight* of this edge, and is omitted when its value is equal to one. For sake of readability, one merges edges $p \xrightarrow{W^-(p,t)} t$ and $t \xrightarrow{W^+(p,t)} p$ when $W^-(p,t) = W^+(p,t)$ leading to a pseudo-edge with two arrows, as in the case of (r_0, t_5).

The net of Figure 1 is not live. Indeed t_0, t_1, t_2, t_3 and t_4 will never be enabled due to the absence of tokens in p_0, p_1, p_2, p_{ext}. Suppose that one deletes the place p_{ext} and its input and output edges. Consider the firing sequence $q_3 + r_0 \xrightarrow{t_5} q_2 + r_0 \xrightarrow{t_6} q_1 + r_0 \xrightarrow{t_3} p_0 + r_0 \xrightarrow{t_4} p_2 + 3q_3 + r_0$. Since the marking $p_2 + 3q_3 + r_0$ is (componentwise) larger than the initial marking $q_3 + r_0$, we can iterate this sequence and generate markings with an arbitrarily large number of tokens in p_2 and q_3; this new net is unbounded. Applying technics that we will develop in this paper (Section 3), we will realize that this new net is actually live.

▶ **Definition 3** (Stochastic Petri net)**.** A *stochastic Petri net* (SPN) is a pair $(\mathcal{N}, \lambda)$ where:

- $\mathcal{N} = (P, T, W^-, W^+)$ is a Petri net;
- λ is a mapping from T to $\mathbb{R}_{>0}$.

A marked stochastic Petri net $(\mathcal{N}, \lambda, m_0)$ is a stochastic Petri net equipped with an initial marking. In a marked stochastic Petri net, when becoming enabled a transition triggers a random delay according to an exponential distribution with (firing) rate $\lambda(t)$. When several transitions are enabled, a *race* between them occurs. Accordingly, given some initial marking m_0, the stochastic process underlying a SPN is a continuous time Markov chain (CTMC) whose (possibly infinite) set of states is $\mathcal{R}(m_0)$ and such that the rate $\mathbf{Q}(m, m')$ of a transition from some m to some $m' \neq m$ is equal to $\sum_{t: m \xrightarrow{t} m'} \lambda(t)$ (as usual $\mathbf{Q}(m, m) = -\sum_{m' \neq m} \mathbf{Q}(m, m')$). Matrix $\mathbf{Q}$ is called the *infinitesimal generator* of the CTMC (see [5] for more details).

From a quantitative point of view, one may be interested in studying the long-run behaviour of the net and in particular in deciding whether there exists a steady-state distribution and in computing it in the positive case. When the underlying graph of the CTMC is strongly connected (i.e. an *irreducible* Markov chain) it amounts to deciding whether there exists a non-zero distribution π over $\mathcal{R}(m_0)$ such that $\pi \cdot \mathbf{Q} = 0$.

It is in general non-trivial to decide whether there exists a steady-state distribution, and even when such a distribution exists, given some state it is hard to compute its steady-state probability (see the introduction). Furthermore, even when the net is bounded, the size of the reachability set may prevent any feasible computation of π.

Thus one looks for subclasses of nets where the steady-state distribution π can be computed more easily and in particular when π has a *product-form*, that is: there exist a constant vector $\mu \in \mathbb{R}^P_{\geq 0}$ only depending on $\mathcal{N}$ such that for all $m \in \mathcal{R}(m_0)$, $\pi(m) = G \cdot \prod_{p \in P} \mu_p^{m(p)}$ where $G = \left(\sum_{m \in \mathcal{R}(m_0)} \prod_{p \in P} \mu_p^{m(p)}\right)^{-1}$ is the so-called *normalising constant* [9].

The most general known class of nets admitting a structural product-form distribution is the class of Π^2-nets [11]. It is based on two key ingredients: *bags* and *witnesses*. A bag is a multiset of tokens that is consumed or produced by some transition. Considering a bag as a whole, one defines the bag graph whose vertices are bags and, given a transition t, an edge goes from the bag consumed by t to the bag produced by t. Observe that there are at most $2|T|$ vertices and exactly $|T|$ edges. This alternative representation of a net via the bag graph does not lose any information: from the bag graph, one can recover the original net. Formally:

▶ **Definition 4** (Bag graph of a Petri net)**.** Let $\mathcal{N} = (P, T, W^-, W^+)$ be a Petri net. Then its bag graph is a labelled graph $G_{\mathcal{N}} = (V_{\mathcal{N}}, E_{\mathcal{N}})$ defined by:

- $V_{\mathcal{N}} = \{W^-(t), W^+(t) \mid t \in T\}$ the finite set of *bags*;
- $E_{\mathcal{N}} = \{W^-(t) \xrightarrow{t} W^+(t) \mid t \in T\}$.

▶ **Example 5.** The bag graph of the net of Figure 1 is described in Figure 2. The bag is written inside the vertex (the external label of the vertices will be explained later). Observe that this graph has three connected components, both of them being strongly connected.

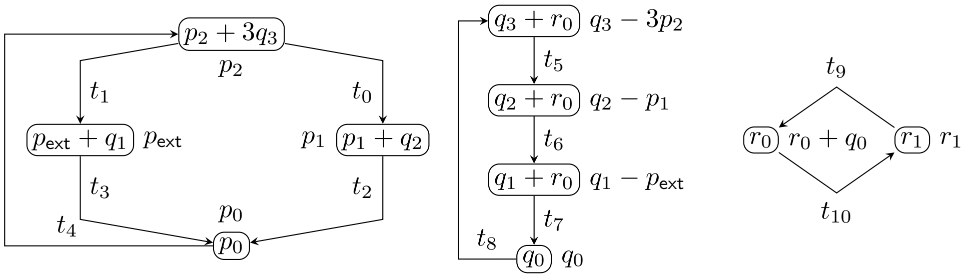


Figure 2 Bags and witnesses.

We now turn to the notion of witness. A queuing network models a single activity flow where activities are modelled by queues and clients leave their current queue when served and enter a new one depending on a routing probability. In Π^2-nets there are several activity flows, one per component of the bag graph. So one wants to witness production and consumption of every bag b by the transition firings. In order to witness it, one looks for a linear combination of the places *wit* such that for every firing of a transition t that produces (resp. consumes) the bag b, for every marking m, $m \cdot wit$ is increased (resp. decreased) by one unit, and such that all other transition firings let $m \cdot wit$ invariant.

▶ **Definition 6** (Witness of a bag)**.** Let $\mathcal{N} = (P, T, W^-, W^+)$ be a a Petri net, $b \in V_{\mathcal{N}}$ and $wit \in \mathbb{Q}^P$. Then wit is a witness of b if:

$$\begin{cases} wit \cdot W(t) = -1 & \text{if } W^-(t) = b; \\ wit \cdot W(t) = 1 & \text{if } W^+(t) = b; \\ wit \cdot W(t) = 0 & \text{otherwise.} \end{cases}$$

▶ **Example 7.** All bags of the net of Figure 1 have (non unique) witnesses. We have depicted them close to their vertices in Figure 2. For instance, consider the bag $b = q_1 + r_0$: transition t_6 produces b while transition t_7 consumes it. Let us check that $w = q_1 - p_{\mathsf{ext}}$ is a witness of b. t_6 produces a token in q_1 and the marking of p_{ext} is unchanged. t_7 consumes a token in q_1 and the marking of p_{ext} is unchanged. The other transitions that change the marking of q_1 and p_{ext} are t_1 and t_3. However since they simultaneously produce or consume a token in both places, $m \cdot w$ is unchanged (m is the current marking).

The definition of Π^2-nets relies on structural properties of the net and on the existence of witnesses. Every connected component of the graph bag will represent an activity flow of some set of processes where every activity (i.e. a bag) has a witness.

▶ **Definition 8** (Π^2-net)**.** Let $\mathcal{N}$ be a Petri net. Then $\mathcal{N}$ is a Π^2-net if:

- every connected component of $G_{\mathcal{N}}$ is strongly connected;
- every bag b of $V_{\mathcal{N}}$ admits a witness (denoted wit_b).

Observe that the first condition called *weak reversibility* ensures that the reachability graph is strongly connected since the firing of any transition t can be "withdrawn" by the firing of transitions occurring along a path from $W^+(t)$ to $W^-(t)$ in the bag graph. The complexity of reachability in weakly reversible nets is still high: EXPSPACE-complete [4].

The next theorem shows the interest of Π^2-nets. Let us define $\lambda(b)$ the *firing rate* of a bag b by $\lambda(b) = \sum_{t|W^-(t)=b} \lambda(t)$ and the choice probability pr_t of transition t by $pr_t = \frac{\lambda(t)}{\lambda(W^-(t))}$.

The routing matrix $\mathbf{P}$ of bags is the stochastic matrix indexed by bags such that for all t, $\mathbf{P}(W^-(t), W^+(t)) = pr_t$ and $\mathbf{P}(b, b') = 0$ otherwise. Consider $\mathbf{vis}$ some positive solution of $\mathbf{vis} \cdot \mathbf{P} = \mathbf{vis}$. Since $\mathbf{P}$ is a stochastic matrix such a vector always exists but is not unique in general; however given b, b' two bags of the same connected component, $\frac{\mathbf{vis}(b')}{\mathbf{vis}(b)}$ measures the ratio between visits of b' and b in the discrete time Markov chain induced by $\mathbf{P}$.

▶ **Theorem 9** ([11]). *Let $(\mathcal{N}, \lambda, m_0)$ be a marked stochastic Π^2-net. Then defining for all $m \in \mathcal{R}(m_0)$, $\mathbf{v}(m) = \prod_{b \in V_{\mathcal{N}}} \left(\frac{\mathbf{vis}(b)}{\lambda(b)}\right)^{m \cdot wit_b}$, the next assertions hold:*

- $\mathbf{v} \cdot \mathbf{Q} = 0$*;*
- $(\mathcal{N}, \lambda, m_0)$ *is ergodic iff* $\|\mathbf{v}\| < \infty$
- *when ergodic, the associated Markov chain admits* $\|\mathbf{v}\|^{-1}\mathbf{v}$ *as steady-state distribution.*

Let us discuss the computational complexity of the product-form of the previous theorem. First deciding whether a net is a Π^2-net is straightforwardly performed in polynomial time. The computation of the visit ratios, the witnesses and the rate of bags can also be done in polynomial time. So computing an item of vector $\mathbf{v}$ is easy. However without additional restriction on the nets the normalising constant $\|\mathbf{v}\|^{-1}$ requires to enumerate all items of $\mathcal{R}_{\mathcal{N}}(m_0)$, which can be prohibitive.

So in [10], the authors introduce Π^3-net, a subclass of Π^2-net which still strictly generalises closed queuing networks, obtained by structuring the activity flows of the net represented by the components of the bag graph. First there is a bijection between places and bags such that the input (resp. output) transitions of the bag produce (consume) one token of this place. The other places occurring in the bag may be viewed as resources associated with the bag and thus the *potential* of the bag is its total number of resources. Second the components of the graph may be ordered as N layers such that the resources of a bag occurring in layer i correspond to places associated with bags of layer $i-1$ (for $i > 1$) and more precisely to those with maximal potential. Informally a token in such a place means that it is a resource available for the upper layer.

▶ **Definition 10** (Π^3-net)**.** Let $\mathcal{N}$ be a net. Then $\mathcal{N}$ is an N-closed Π^3-net if:

- There is a bijection between P and $V_{\mathcal{N}}$. Denoting b_p the bag associated with place p (and p_b the place associated with bag b), we have $b_p(p) = 1$.
  The potential of a place $\mathsf{pot}(p)$ is equal to $\|b_p\| - 1$.
- $V_{\mathcal{N}}$ is partitioned into N strongly connected components $V_1, \ldots, V_N$. One denotes: $P_i = \{p_b \mid b \in V_i\}$ and $P_i^{\max} = \mathrm{argmax}(\mathsf{pot}(p) \mid p \in P_i)$. By convention, $P_0^{\max} = P_0 = \emptyset$.
- For all $b \in V_i$ and $p \in P \setminus \{p_b\}$, $b(p) > 0$ implies $p \in P_{i-1}^{\max}$.

A net is an N-open Π^3-net if it is obtained by deleting some place $p_{\mathsf{ext}} \in P_N$ (and its input/output edges) from an N-closed Π^3-net.

Given an open Π^3-net $\mathcal{N}$, $\overline{\mathcal{N}}$ denotes the closed net based on which $\mathcal{N}$ has been defined. For every $1 \leq i \leq N$, we will later write $P_i^{\neg\max}$ for the set $P_i \setminus P_i^{\max}$, and T_i for the set of transitions t such that $W^-(t) \in V_i$. The next proposition establishes that Π^3-nets are product-form Petri nets.

▶ **Proposition 11.** *Let $\mathcal{N}$ be a (closed or open) Π^3-net. Then $\mathcal{N}$ is a Π^2-net.*

▶ **Example 12.** The net of Figure 1 is a 3-closed Π^3-net. We have used different colors for the layers: $P_3 = \{p_0, p_1, p_2, p_{\mathsf{ext}}\}$ (in red), $P_2 = \{q_0, q_1, q_2, q_3\}$ (in green) and $P_1 = \{r_0, r_1\}$ (in blue). The place q_0 (resp. q_1, q_2, q_3) is associated with the bag q_0 (resp. $q_1{+}r_0, q_2{+}r_0, q_3{+}r_0$) and has potential 0 (resp. $1, 1, 1$). Thus $P_2^{\max} = \{q_1, q_2, q_3\}$ and indeed q_0 does not occur in bags of layer 3. In order to highlight the use of resources, edges between places of P_i and transitions of T_{i+1} are depicted in gray.

The next theorem shows the interest of closed Π^3-nets.

▶ **Theorem 13** ([10]). *Let $(\mathcal{N}, \lambda, m_0)$ be a N-closed Π^3-net. Then:*

- *$(\mathcal{N}, m_0)$ is bounded.*
- *One can decide whether $(\mathcal{N}, m_0)$ is live in polynomial time.*
- *When $(\mathcal{N}, m_0)$ is live, one can decide in polynomial time, given m, whether $m \in \mathcal{R}_{\mathcal{N}}(m_0)$.*
- *For any fixed N, when $(\mathcal{N}, m_0)$ is live, one can compute the normalising constant of the steady-state distribution (i.e. $\|\mathbf{v}\|^{-1}$ in Theorem 9) in polynomial time with respect to $|P|$, $|T|$, the maximal weight of the net's edges, and $\|m_0\|$ (thus in pseudo-polynomial time w.r.t. the size of m_0).*

The above results do not apply to infinite-state systems and in particular to the systems generated by open Π^3-nets. In addition, the polynomial-time complexity for computing the normalising constant requires to fix N. We address these issues in the next sections.

3 Qualitative analysis

In this section we first give a simple characterisation of the liveness property in a marked Π^3-net. We then fully characterise the set of reachable markings in a live marked Π^3-net. These characterisations give polynomial-time algorithms for deciding liveness of a marked Π^3-net, and the boundedness property of a live marked Π^3-net. We end the section with a coNP-hardness result for the boundedness property of a marked Π^3-net, when it is not live. For the rest of this section, we assume $\mathcal{N} = (P, T, W^-, W^+)$ is an open or closed Π^3-net with N layers. We further use the notations of Definition 10. In particular, if $\mathcal{N}$ is open, then we write p_{ext} for the place which has been removed (and we call it virtual). We therefore set $P_N^\star = P_N \cup \{p_{\mathsf{ext}}\}$ if the net is open and $P_N^\star = P_N$ otherwise; For every $1 \leq i \leq N-1$ we define $P_i^\star = P_i$; And we set $P^\star = \bigcup_{i=1}^{N} P_i^\star$.

3.1 Liveness analysis

We give a simple characterisation of the liveness property through a dependence between the number of tokens at some layer and potentials of places activated on the next layer. More precisely, for every $1 \leq i \leq N-1$, Live_i is defined as the set of markings m such that:

$$m \cdot P_i \geq \min\{\mathsf{pot}(p) \mid p \in P_{i+1}^\star \text{ and } (m(p) > 0 \text{ or } p = p_{\mathsf{ext}})\}.$$

Note that $p = p_{\mathsf{ext}}$ can only happen when $\mathcal{N}$ is open and $i = N-1$. We additionally define Live_N as the set of markings m such that $m \cdot P_N > 0$ if $\mathcal{N}$ is closed, and as the set of all markings if $\mathcal{N}$ is open.

For every $1 \leq i \leq N$ (resp. $1 \leq i \leq N-1$), we define $\mathsf{POT}_i = \max\{\mathsf{pot}(p) \mid p \in P_i\}$ when $\mathcal{N}$ is closed (resp. open). When $\mathcal{N}$ is open, we write $\mathsf{POT}_N = \mathsf{pot}(p_{\mathsf{ext}})$. Given a marking m, when no place p fulfills $p \in P_{i+1}^\star$ and $(m(p) > 0$ or $p = p_{\mathsf{ext}})$, the minimum is equal to POT_{i+1}. Thus given a marking m, the condition $m \in \mathsf{Live}_i$ for $i < n$ only depends on the values of $m(p)$ for $p \in P_i \cup P_{i+1}^{\neg\max}$.

The intuition behind condition Live_i is the following: transitions in $\bigcup_{j \leq i} T_j$ cannot create new tokens on layer i (layer i behaves like a state machine, and smaller layers do not change the number of tokens in that layer); therefore, to activate a transition of T_{i+1} out of some marked place $p \in P_{i+1}$, it must be the case that enough tokens are already present on layer i; hence there should be at least as many tokens in layer i as the minimal potential of a marked place in layer $i+1$. When $\mathcal{N}$ is open, the virtual place p_{ext} behaves like a source of tokens, hence it is somehow always “marked”; this is why it is taken into account in the right part of Live_i. The following characterisation was already stated in [10] in the restricted case of closed nets.

▶ **Theorem 14.** *A marking m is live if and only if for every $1 \le i \le N$, $m \in \mathsf{Live}_i$.*

▶ **Example 15.** Building on the marked Petri net of Figure 1, the marking $q_3 + r_0$ is live when the net is open but not live if the net is closed. Indeed, transitions of the two first layers can trivially be activated from q_3+r_0 (hence by weak-reversibility from every reachable marking). We see that in the case of the closed net, transitions of layer 3 cannot be activated (no fresh token can be produced on that layer). On the contrary, in the case of the open net, the token in q_3 can be moved to q_1, which will activate transition t_3; from there, all transitions of layer 3 will be eventually activated.

As a consequence of the characterisation of Theorem 14, we get:

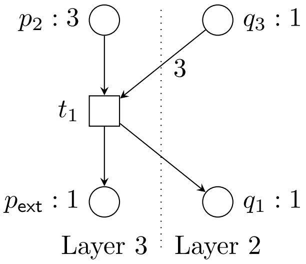
▶ **Corollary 16.** *We can decide the liveness of a marked Π^3-net in polynomial time.*

3.2 Reachable markings

We will now give a characterisation of the set of reachable markings $\mathcal{R}_{\mathcal{N}}(m_0)$ when m_0 is live. We will first give linear invariants of the net: those are vectors in the left kernel of W (or P-flows). The name “invariants” comes from the fact that they will allow to infer real invariants satisfied by the reachable markings. Furthermore, for every $1 \le i \le N$, for every $p \in P_i$, we define $\mathsf{cin}(p) = \mathsf{POT}_i - \mathsf{pot}(p)$. Except when $\mathcal{N}$ is open and $i = N$, the cin value of a place is nonnegative.

▶ **Proposition 17.** *The following vectors are linear invariants of $\mathcal{N}$:*

- *for every $1 \leqslant i \leqslant N-1$, $v^{(i)} = \sum_{p\in P_i} p + \sum_{p \in P_{i+1}} \mathsf{cin}(p)\ p$;*
- *if $\mathcal{N}$ is closed, $v^{(N)} = \sum_{p \in P_N} p$.*



First observe that for $i < N-1$, $\mathsf{Supp}(v^{(i)}) = P_i \cup P_{i+1}^{\neg\max}$. Thus given a marking m, only firing of transitions $t \in T_i \cup T_{i+1}$ could change $m \cdot v^{(i)}$. This is not the case of a transition $t \in T_i$ since it moves a token from a place of P_i to another one. To give an intuition why transitions in T_{i+1} do not change $m \cdot v^{(i)}$, we consider part of the closed net (that is, p_{ext} is a real place) of Figure 1 depicted on the right, where numbers close to place names are potential values. We focus on transition t_1 and explain why $m \cdot v^{(2)}$ is unchanged by its firing. The impact of transition t_1 is to decrease the sum $\sum_{p \in P_2} p$ by 2; due to the weights of places of P_3 in $v^{(2)}$, place p_2 counts as 0 and place p_{ext} counts as +2. This intuition extends into a formal proof.

For every $1 \le i \le N-1$, we set $\mathbf{C}_i^{m_0} = m_0 \cdot v^{(i)}$, and if $\mathcal{N}$ is closed, we set $\mathbf{C}_N^{m_0} = m_0 \cdot v^{(N)}$. For every $1 \le i \le N-1$, we define $\mathsf{Inv}_i(m_0)$ as the set of markings m such that $m \cdot v^{(i)} = \mathbf{C}_i^{m_0}$, and if $\mathcal{N}$ is closed, we define $\mathsf{Inv}_N(m_0)$ as the set of markings m such that $m \cdot v^{(N)} = \mathbf{C}_N^{m_0}$. For uniformity, if $\mathcal{N}$ is open, we define $\mathsf{Inv}_N(m_0)$ as the set of all markings. As a consequence of Proposition 17, we get:

▶ **Corollary 18.** $\mathcal{R}_{\mathcal{N}}(m_0) \subseteq \bigcap_{i=1}^{N} \mathsf{Inv}_i(m_0)$.

More complex invariants were given in [10] for closed Π^3-nets. The advantage of the above invariants is that each of them only involves two neighbouring layers. This will have a huge impact on various complexities, and will allow the development of our methods for quantitative analysis.

▶ **Example 19.** Going back to the Petri net of Figure 1, with initial marking $m_0 = q_3 + r_0$. We first consider the closed net. Then: $\mathsf{POT}_3 = 2$ and $\mathsf{POT}_2 = 1$. Therefore:

$$\begin{cases} \mathsf{Inv}_3(m_0) = \{m \mid \sum_{i=0}^{2} m(p_i) + m(p_{\mathsf{ext}}) = 0\} \\ \mathsf{Inv}_2(m_0) = \{m \mid \sum_{i=0}^{3} m(q_i) + 3m(p_0) + 2m(p_1) + 2m(p_{\mathsf{ext}}) = m_0(q_3) = 1\} \\ \mathsf{Inv}_1(m_0) = \{m \mid m(r_0) + m(r_1) + m(q_0) = m_0(r_0) = 1\} \end{cases}$$

We now turn to the open net, obtained by deleting p_{ext}. Definition of POT_3 differs from the previous case: $\mathsf{POT}_3 = \mathsf{pot}(p_{\mathsf{ext}}) = 1$ and $\mathsf{POT}_2 = 1$. Therefore:

$$\begin{cases} \mathsf{Inv}_3(m_0) = \mathbb{N}^P \\ \mathsf{Inv}_2(m_0) = \{m \mid \sum_{i=0}^{3} m(q_i) + m(p_0) - 2m(p_2) = m_0(q_3) = 1\} \\ \mathsf{Inv}_1(m_0) = \{m \mid m(r_0) + m(r_1) + m(q_0) = m_0(r_0) = 1\} \end{cases}$$

The invariants of Corollary 18 do not fully characterise the set of reachable markings, since they do not take into account the enabling conditions of the transitions. However, they will be very helpful for characterising the reachable markings when m_0 is live.

▶ **Theorem 20.** *Suppose that $(\mathcal{N}, m_0)$ is a live Π^3-net. Then:*

$$\mathcal{R}_{\mathcal{N}}(m_0) = \bigcap_{i=1}^{N} \mathsf{Inv}_i(m_0) \ \cap \ \bigcap_{i=1}^{N} \mathsf{Live}_i$$

Thus reachability in live Π^3-nets can be checked in polynomial time.

▶ **Example 21.** In the open Petri net of Figure 1, with initial marking $m_0 = q_3 + r_0$, the sets Live_i are $\mathsf{Live}_1 = \{m \mid m(r_0) + m(r_1) + m(q_0) \geqslant 1\}$, $\mathsf{Live}_2 = \{m \mid \sum_{i=0}^{3} m(q_i) + m(p_0) \geqslant 1\}$ and $\mathsf{Live}_3 = \mathbb{N}^P$. Observing that $\mathsf{Inv}_i(m_0) \subseteq \mathsf{Live}_i$ for $1 \leqslant i \leqslant 3$, the net has reachability set

$$\mathcal{R}_{\mathcal{N}}(m_0) = \{m \mid m(r_0) + m(r_1) + m(q_0) = 1 \text{ and } \textstyle\sum_{i=0}^{3} m(q_i) + m(p_0) - 2m(p_2) = 1\}.$$

The idea of the proof when the net is closed is to show that, from every marking m satisfying the right handside condition in the theorem, one can reach a specific marking m_0^* (where, for every $1 \leq i \leq N$, $\mathbf{C}_i^{m_0}$ tokens are in one arbitrary place of $P_i^{\max}$). Hence, given two markings m and m' satisfying the conditions, $m \to^* m_0^*$ and $m' \to^* m_0^*$, which implies by weak reversibility of the net: $m \to^* m_0^* \to^* m'$. This is in particular the case from m_0: every marking satisfying the conditions is reachable from m_0. In the case of an open net, this is a bit more tricky, and a joint marking to every pair (m, m') of markings satisfying the conditions has to be chosen.

3.3 Boundedness analysis

As a consequence of the characterisation given in Theorem 20, we get:

▶ **Corollary 22.** *We can decide the boundedness of a live marked Π^3-net in polynomial time.*

Indeed, it can be shown that, if $\mathcal{N}$ is closed, then $\mathcal{N}$ is bounded, and that if $\mathcal{N}$ is open, then it is bounded if and only if $\mathsf{cin}(q) > 0$ for all $q \in P_N$ (that is, p_{ext} has maximal potential, and no other place of P_N has maximal potential). Furthermore, if $\mathcal{N}$ is bounded, then the overall number of tokens in the net is bounded by $\sum_{i=1}^{N-1} \mathbf{C}_i^{m_0}$ (resp. $\sum_{i=1}^{N} \mathbf{C}_i^{m_0}$) if the net is open (resp. closed).

The polynomial-time complexity of Corollary 22 is in contrast with the following hardness result, which can be obtained by a reduction from the independent-set problem.

▶ **Proposition 23.** *Deciding the boundedness of a marked Π^3-net which is not live is coNP-hard. The reachability (and even the coverability) problem is NP-hard.*

4 Quantitative analysis

Contrary to closed Π^3-nets, open Π^3-nets may not be ergodic. In this section, we first give a simple characterisation of the ergodicity property for open Π^3-nets, which gives us a polynomial-time algorithm for deciding ergodicity. We then provide a polynomial-time algorithm for computing the steady-state distribution of ergodic (open and closed) Π^3-nets.

For the rest of this section, we assume that $(\mathcal{N}, \lambda, m_0)$ is a stochastic Π^3-net with N layers, and that m_0 is live. Let $\mathbf{W}$ be the maximal weight of the edges of $\mathcal{N}$. Then we assume that the constants $\mu_p = \prod_{b \in \mathcal{V}_\mathcal{N}} (\mathbf{vis}(b)/\lambda_b)^{wit_b \cdot p}$ have already been precomputed (in polynomial time with respect to $|P|$, $|T|$ and $\log(1 + \mathbf{W})$). [1]

In what follows, and for every vector $\delta \in \mathbb{N}^P$, we denote by $\hat{v}(\delta)$ the product $\prod_{p \in P} \mu_p^{\delta(p)}$. Consequently, the vector $\mathbf{v}$ mentioned in Theorem 9 is then defined by $\mathbf{v}(m) = \hat{v}(m)$ for all markings $m \in \mathcal{R}(m_0)$, and its norm is $\|\mathbf{v}\| = \sum_{m \in \mathcal{R}(m_0)} \hat{v}(m)$. Note that $\mathbf{v}$ and $\hat{v}$ only differ by their domain. In addition, in what follows, and for every set $Z \subseteq P$, we simply denote by $\mathsf{cin}(Z)$ the formal sum $\sum_{p \in Z} \mathsf{cin}(p)p$.

4.1 Ergodicity analysis

We assume here that $\mathcal{N}$ is open. We give a simple characterisation of the ergodicity property through a comparison of the constants μ_p for a limited number of places p. Those constraints express congestion situations that may arise; we show that they are sufficient. These places are the elements of the subset Y of places that is defined by $Y = P_N \cup P_{N-1}^{\max}$. In particular, as soon as the initial marking m_0 is live, then the ergodicity of the stochastic net $(\mathcal{N}, \lambda, m_0)$ does *not* depend on m_0.

According to Theorem 9, the net is ergodic if and only if the norm $\|\mathbf{v}\| = \sum_{m \in \mathcal{R}(m_0)} \hat{v}(m)$ is finite. Hence, deciding ergodicity amounts to deciding the convergence of a sum. The following characterisation holds.

▶ **Theorem 24.** *Let $(\mathcal{N}, \lambda, m_0)$ be a live open stochastic Π^3-net with N layers. This net is ergodic if and only if all of the following inequalities hold:*

- *for all places $p \in P_N$, if $\mathsf{cin}(p) = 0$, then $\mu_p < 1$;*
- *for all places $p, q \in P_N$, if $\mathsf{cin}(p) > 0 > \mathsf{cin}(q)$, then $\mu_p^{|\mathsf{cin}(q)|} \mu_q^{\mathsf{cin}(p)} < 1$;*
- *for all places $p \in P_{N-1}^{\max}$ and $q \in P_N$, if $0 > \mathsf{cin}(q)$, then $\mu_p^{|\mathsf{cin}(q)|} \mu_q < 1$.*

Proof (sketch). Let $\mathcal{F}$ be the family formed of the vectors p (for $p \in P_N$ such that $\mathsf{cin}(p) = 0$), $\mathsf{cin}(p)q - \mathsf{cin}(q)p$ (for $p, q \in P_N$ such that $\mathsf{cin}(p) > 0 > \mathsf{cin}(q)$) and $q - \mathsf{cin}(q)p$ (for

[1] In the rest of this section, we will design polynomial-time procedures w.r.t. $\mathbf{W}$, hence polynomial if it is encoded in unary and pseudo-polynomial if it is encoded in binary.

$p \in P_{N-1}^{\max}$ and $q \in P_N$ such that $0 > \mathsf{cin}(q)$). Let also $\mathcal{L}$ be the sublattice of $\mathbb{N}^P$ generated by the vectors in $\mathcal{F}$, and let $\mathcal{G}$ be the finite subset of $\mathbb{N}^P$ formed of those vectors whose entries are not greater than some adequately chosen constant $\mathbf{G}$.

Since m_0 is live, Theorem 20 applies, which allows us to prove the inclusions $\{m_0\} + \mathcal{L} \subseteq \mathcal{R}(m_0) \subseteq \mathcal{G} + \mathcal{L}$. Hence, the sum $\sum_{m \in \mathcal{R}(m_0)} \hat{v}(m)$ is finite iff the sum $\sum_{m \in \mathcal{L}} \hat{v}(m)$ is finite, i.e. iff each constant $\hat{v}(\delta)$ is (strictly) smaller than 1, for $\delta \in \mathcal{F}$. ◀

▶ **Example 25.** Going back to the open Π^3-net of Figure 1, with any live initial marking m_0, we obtain the following necessary and sufficient conditions for being ergodic:

$$\mu_{p_1} < 1,\ \mu_{p_0}^2 \mu_{p_2} < 1,\ \mu_{q_1}^2 \mu_{p_2} < 1,\ \mu_{q_2}^2 \mu_{p_2} < 1 \text{ and } \mu_{q_3}^2 \mu_{p_2} < 1.$$

As a consequence of the characterisation of Theorem 24, we get:

▶ **Corollary 26.** *We can decide the ergodicity of a marked, live and open stochastic Π^3-net in polynomial time.*

4.2 Computing the steady-state distribution

In case the Π^3-net is ergodic, it remains to compute its steady-state distribution, given by $\pi(m) = \|\mathbf{v}\|^{-1}\mathbf{v}(m)$ for all $m \in \mathcal{R}(m_0)$. Since we already computed $\mathcal{R}(m_0)$ and $\mathbf{v}(m)$ for all markings $m \in \mathcal{R}(m_0)$, it remains to compute the normalising constant $\|\mathbf{v}\|$.

This section is devoted to proving the following result.

▶ **Theorem 27.** *Let $(\mathcal{N}, \lambda, m_0)$ be a live ergodic stochastic Π^3-net. There exists an algorithm for computing the normalising constant $\|\mathbf{v}\|$ in polynomial time with respect to $|P|$, $|T|$, $\mathbf{W}$, and $\|m_0\|$ (thus in pseudo-polynomial time).*

This theorem applies to both closed and open Π^3-nets. For closed nets, it provides a similar yet stronger result than Theorem 13, where the polynomial-time complexity was obtained only for a fixed value of N (the number of layers of the net).

We prove below Theorem 27 in the case of open nets. The case of closed nets is arguably easier: one can transform a closed net into an equivalent open net by adding a layer $N+1$ with one place (and one virtual place p_{ext}), and set a firing rate $\lambda_t = 0$ for all transitions t of the layer $N+1$. We therefore assume for the rest of this section that $\mathcal{N}$ is open.

We first describe a naive approach. The normalisation constant $\|\mathbf{v}\|$ can be computed as follows. Recall the family $\mathcal{F}$ introduced in the proof of Theorem 24. We may prove that the set $\mathcal{R}(m_0)$ is a union of (exponentially many) translated copies of the lattice $\mathcal{L}$ generated by $\mathcal{F}$. These copies may intersect each other, yet their intersections themselves are translated copies of $\mathcal{L}$. Hence, using an inclusion-exclusion formula and a doubly exponential computation step, computing the sum $\|\mathbf{v}\| = \sum_{\mathfrak{m} \in \mathcal{R}(m_0)} \hat{v}(\mathfrak{m})$ reduces to computing the sum $\sum_{\ell \in \mathcal{L}} \hat{v}(\ell)$.

The family $\mathcal{F}$ is not free a priori, hence computing this latter sum is not itself immediate. Using again inclusion-exclusion formulæ, we may write $\mathcal{L}$ as a finite, disjoint union of exponentially many lattices generated by free subfamilies of $\mathcal{F}$. This last step allows us to compute $\sum_{\ell \in \mathcal{L}} \hat{v}(\ell)$, and therefore $\|\mathbf{v}\|$.

Such an approach suffers from a prohibitive computational cost. Yet it is conceptually simple, and it allows proving rather easily that $\|\mathbf{v}\|$ is a rational fraction in the constants μ_p, whose denominator is the product $\prod_{\ell \in \mathcal{F}}(1 - \hat{v}(\ell))$.

▶ **Example 28.** Going back to the open Π^3-net of Figure 1, with initial marking $m_0 = q_3 + r_0$, this algorithm allows us to compute

$$\|\mathbf{v}\| = \frac{\mu_{q_0} a + (\mu_{r_0} + \mu_{r_1}) b}{c}, \text{ with}$$
$$a = \mu_{p_2}^2 \mu_{p_0} \mu_{q_1} \mu_{q_2} \mu_{q_3} + \mu_{p_2}(\mu_{p_0}\mu_{q_1} + \mu_{p_0}\mu_{q_2} + \mu_{p_0}\mu_{q_3} + \mu_{q_1}\mu_{q_2} + \mu_{q_1}\mu_{q_3} + \mu_{q_2}\mu_{q_3}) + 1,$$
$$b = \mu_{p_2}(\mu_{p_0}\mu_{q_1}\mu_{q_2} + \mu_{p_0}\mu_{q_1}\mu_{q_3} + \mu_{p_0}\mu_{q_2}\mu_{q_3} + \mu_{p_1}\mu_{q_2}\mu_{q_3}) + \mu_{p_0} + \mu_{q_1} + \mu_{q_2} + \mu_{q_3} \text{ and}$$
$$c = (1 - \mu_{p_0}^2 \mu_{p_2})(1 - \mu_{p_1})(1 - \mu_{q_1}^2 \mu_{p_2})(1 - \mu_{q_2}^2 \mu_{p_2})(1 - \mu_{q_3}^2 \mu_{p_2}).$$

Recall Example 25, which states that $\|\mathbf{v}\|$ is finite if and only if all of $1 - \mu_{p_1}$, $1 - \mu_{p_0}^2 \mu_{p_2}$, $1 - \mu_{q_1}^2 \mu_{p_2}$, $1 - \mu_{q_2}^2 \mu_{p_2}$ and $1 - \mu_{q_3}^2 \mu_{p_2}$ are positive. We observe that, as suggested above, the denominator c is precisely the product of these five factors.

Our approach for computing $\|\mathbf{v}\|$ will involve first computing variants of $\|\mathbf{v}\|$. More precisely, for all subsets Z of P, we consider a congruence relation $\sim_Z$ on markings, such that $m \sim_Z m'$ iff m and m' coincide on all places $p \in Z$. Then, we denote by $\mathfrak{M}_Z$ the quotient set $\mathbb{N}^P / \sim_Z$ and, for every element $\mathfrak{m}$ of $\mathfrak{M}_Z$, we denote by $\hat{v}(\mathfrak{m})$ the product $\prod_{p \in Z} \mu_p^{\mathfrak{m}(p)}$. Two remarkable subsets Z are the set $X = \bigcup_{i \leqslant N-2} P_i \cup P_{N-1}^{\neg\max}$ and its complement $Y = P_{N-1}^{\max} \cup P_N = P \setminus X$, which was already mentioned in Section 4.1. Indeed, places in X are necessarily bounded while, if the net is unbounded, then so are the places in Y.

Based on these objects, and for all integers $c \geqslant 0$, we define two sets $\mathcal{C}_{m_0}(c)$ and $\mathcal{D}_{m_0}(c)$, which are respective subsets of $\mathfrak{M}_X$ and of $\mathfrak{M}_Y$. In some sense, the sets $\mathcal{C}_{m_0}(c)$ are meant to describe the "bounded" part of the markings in $\mathcal{R}(m_0)$, whose "unbounded" part is described by the sets $\mathcal{D}_{m_0}(c)$.

▶ **Definition 29.** The set $\mathcal{C}_{m_0}(c)$ is the set of those classes $\mathfrak{m}_X$ in $\mathfrak{M}_X$ such that $\mathfrak{m}_X \cdot P_{N-1}^{\neg\max} = c$, and such that $\mathfrak{m}_X$ contains some marking in $\mathcal{R}(m_0)$. The set $\mathcal{D}_{m_0}(c)$ is the set of those classes $\mathfrak{m}_Y$ in $\mathfrak{M}_Y$ such that $c + \mathfrak{m}_Y \cdot P_{N-1}^{\max} + \mathfrak{m}_Y \cdot \mathsf{cin}(P_N) = \mathbf{C}_{N-1}^{m_0}$, and such that $\mathfrak{m}_Y$ contains some marking in $\mathcal{R}(m_0)$.

These two sets of classes allow us to split nicely the huge sum $\|\mathbf{v}\|$ into smaller independent sums, as stated in the result below. This decomposition result has the flavour of convolution algorithms, yet it requires a specific treatment since its terms are infinite sums.

▶ **Lemma 30.** *The normalisation constant $\|\mathbf{v}\|$ is equal to the following finite sum:*

$$\|\mathbf{v}\| = \sum_{c=0}^{|P|\mathbf{W}\|m_0\|} \left(\sum_{\mathfrak{m}_X \in \mathcal{C}_{m_0}(c)} \hat{v}(\mathfrak{m}_X) \right) \left(\sum_{\mathfrak{m}_Y \in \mathcal{D}_{m_0}(c)} \hat{v}(\mathfrak{m}_Y) \right).$$

It remains to compute sums $\sum_{\mathfrak{m}_X \in \mathcal{C}_{m_0}(c)} \hat{v}(\mathfrak{m}_X)$ and $\sum_{\mathfrak{m}_Y \in \mathcal{D}_{m_0}(c)} \hat{v}(\mathfrak{m}_Y)$ for polynomially many values of c. It turns out that such computations can be performed in polynomial time, as mentioned in Propositions 31 and 32.

▶ **Proposition 31.** *Let $(\mathcal{N}, \lambda, m_0)$ be a live ergodic stochastic open Π^3-net, and let $c \geqslant 0$ be an integer. There exists an algorithm for computing the sum $\sum_{\mathfrak{m}_X \in \mathcal{C}_{m_0}(c)} \hat{v}(\mathfrak{m}_X)$ in polynomial time with respect to $|P|$, $|T|$, $\mathbf{W}$, c and $\|m_0\|$.*

Proposition 31 is similar to the results of [10], and the algorithm can be adapted to closed Π^3-nets. The only major improvement here is that, instead of obtaining an algorithm that is polynomial-time for fixed values of N only, our choice of invariants leads to a polynomial-time algorithm independently of N.

▶ **Proposition 32.** *Let $(\mathcal{N}, \lambda, m_0)$ be a live ergodic stochastic open Π^3-net, and let $c \geqslant 0$ be an integer. There exists an algorithm for computing the sum $\sum_{\mathfrak{m}_Y \in \mathcal{D}_{m_0}(c)} \hat{v}(\mathfrak{m}_Y)$ in polynomial time with respect to $|P|$, $|T|$, $\mathbf{W}$, c and $\|m_0\|$.*

Proof (sketch). We compute the sum $\sum_{\mathfrak{m}_Y \in \mathcal{D}_{m_0}(c)} \hat{v}(\mathfrak{m}_Y)$ by using a dynamic-programming approach. Let $a = |P_{N-1}^{\max}|$, $b = |P_N|$, and define t and u to be integers such that (i) $\mathsf{cin}(p_i^N) > 0$ iff $1 \leqslant i \leqslant t$, (ii) $\mathsf{cin}(p_i^N) = 0$ iff $t < i < u$ and (iii) $\mathsf{cin}(p_i^N) < 0$ iff $u \leqslant i \leqslant b$. In addition, for all $i \leqslant b$, let $\Delta_i = \max\{1, |\mathsf{cin}(p_i^N)|\}$. We consider below the lattice $\mathbb{L} = \mathbb{Z}^{a-1} \times \prod_{i=1}^{b}(\Delta_i \mathbb{Z})$.

Now, consider integers $A, B \in \mathbb{Z}$ and $1 \leqslant \alpha \leqslant a$, $1 \leqslant \gamma \leqslant \beta \leqslant b$, as well as a vector $\mathbf{w}$ in the quotient set $\mathbb{Z}^{a+b-1}/\mathbb{L}$. We define auxiliary sets of vectors of the form

$$\overline{\mathcal{D}}(A, B, \mathbf{w}, \alpha, \beta, \gamma) = \left\{ \mathbf{xy} \in \mathbb{N}^{a+b-1} \;\middle|\; \begin{array}{l} A + \sum_{j=u}^{\beta} y_j \geqslant \sum_{i=1}^{\alpha} x_i + \sum_{j=\gamma}^{t} y_j \\ B + \sum_{j=u}^{\beta} y_j \geqslant \sum_{j=\gamma}^{t} y_j \\ \mathbf{xy} - \mathbf{w} \in \mathbb{L} \\ x_i = 0 \text{ for all } i > \alpha \\ y_j = 0 \text{ for all } j < \gamma \text{ and all } j > \beta \end{array} \right\},$$

where $\mathbf{xy}$ denotes the vector $(x_1, \ldots, x_{a-1}, y_1, \ldots, y_b)$.

First, we exhibit a bijection $\mathfrak{m} \mapsto \overline{\mathfrak{m}}$, from the set $\mathcal{D}_{m_0}(c)$ to a finite union of sets $\overline{\mathcal{D}}(\mathbf{A}, \mathbf{B}, \mathbf{0}, a-1, b, \mathbf{s})$, where the integers $\mathbf{A}$, $\mathbf{B}$ and $\mathbf{s}$ can be computed efficiently. We can also prove a relation of the form $\hat{v}(\mathfrak{m}) = \hat{\hat{v}}(\overline{\mathfrak{m}})$ for all markings $\mathfrak{m} \in \mathcal{D}_{m_0}(c)$, where $\hat{\hat{v}}$ is a product-form function $\hat{\hat{v}} : \mathbf{xy} \mapsto \mathbf{P} \prod_{i=1}^{a-1} \nu_{x,i}^{x_i} \prod_{i=1}^{b} \nu_{y,i}^{y_i}$ such that the constants $\mathbf{P}$, $\nu_{x,i}$ and $\nu_{y,j}$ can be computed efficiently. It remains to compute sums of the form

$$\mathcal{U}(A, B, \mathbf{w}, \alpha, \beta, \gamma) = \sum_{\mathbf{xy} \in \overline{\mathcal{D}}(A, B, \mathbf{w}, \alpha, \beta, \gamma)} \hat{\hat{v}}(\mathbf{xy}).$$

We do it by using recursive decompositions of the sets $\overline{\mathcal{D}}(A, B, \mathbf{w}, \alpha, \beta, \gamma)$, where the integers A and B belong to the finite set $\{-|\mathbf{A}| - |\mathbf{B}|, \ldots, |\mathbf{A}| + |\mathbf{B}|\}$, and where the vector $\mathbf{w}$ is constrained to have at most one non-zero coordinate, chosen from some finite domain. These decompositions involve computing polynomially many auxiliary sums, which will prove Proposition 32. ◀

We illustrate the last part of this sketch of proof in the case where $\gamma \leqslant t$, $u \leqslant \beta$, and $\mathbf{w}$ is of the form $\lambda\ \mathbf{1}_{y,\beta}$, for some $\lambda \in \mathbb{Z}/\Delta_\beta\mathbb{Z}$ (where we denote by $\mathbf{1}_{y,j}$ the vector of the canonical basis whose unique non-zero entry is y_j). In this case, we show how the sum $\mathcal{U}(A, B, \mathbf{w}, \alpha, \beta, \gamma)$ can be expressed in terms of "smaller" sums $\mathcal{U}(A', B', \mathbf{w}', \alpha', \beta', \gamma')$.

Let us split the set $\overline{\mathcal{D}}(A, B, \mathbf{w}, \alpha, \beta, \gamma)$ into two subsets:

$\overline{\mathcal{D}}_{\oplus}(A, B, \mathbf{w}, \alpha, \beta, \gamma) = \{\mathbf{xy} \in \overline{\mathcal{D}}(A, B, \mathbf{w}, \alpha, \beta, \gamma) \mid y_\beta \leqslant y_\gamma\}$ and

$\overline{\mathcal{D}}_{\ominus}(A, B, \mathbf{w}, \alpha, \beta, \gamma) = \{\mathbf{xy} \in \overline{\mathcal{D}}(A, B, \mathbf{w}, \alpha, \beta, \gamma) \mid y_\beta \geqslant y_\gamma\}$, whose intersection is

$\overline{\mathcal{D}}_{\odot}(A, B, \mathbf{w}, \alpha, \beta, \gamma) = \{\mathbf{xy} \in \overline{\mathcal{D}}(A, B, \mathbf{w}, \alpha, \beta, \gamma) \mid y_\beta = y_\gamma\}$.

Computing $\mathcal{U}(A, B, \mathbf{w}, \alpha, \beta, \gamma)$ amounts to computing the three sums $\sum_{\mathbf{xy}} \hat{\hat{v}}(\mathbf{xy})$, where $\mathbf{xy}$ ranges respectively on $\overline{\mathcal{D}}_{\oplus}(A, B, \mathbf{w}, \alpha, \beta, \gamma)$, $\overline{\mathcal{D}}_{\ominus}(A, B, \mathbf{w}, \alpha, \beta, \gamma)$ and $\overline{\mathcal{D}}_{\odot}(A, B, \mathbf{w}, \alpha, \beta, \gamma)$. We show here how to compute the first sum, which we simply denote by $\mathcal{U}_{\oplus}$. The two latter sums are computed similarly.

Splitting every vector $\mathbf{xy}$ into one vector $y_\beta(\mathbf{1}_{y,\gamma} + \mathbf{1}_{y,\beta})$ and one vector $\mathbf{xy}' = \mathbf{xy} - y_\beta(\mathbf{1}_{y,\gamma} + \mathbf{1}_{y,\beta})$, we observe that $\mathbf{xy} - \lambda\ \mathbf{1}_{y,\beta} \in \mathbb{L}$ if and only if (i) $y_\beta \equiv \lambda \mod \Delta_\beta$, and (ii)

$\mathbf{xy}' - y_\beta \mathbf{1}_{y,\gamma} \in \mathbb{L}$. It follows that

$$\begin{aligned}\overline{\mathcal{D}}_\oplus(A,B,\lambda\ \mathbf{1}_{y,\beta},\alpha,\beta,\gamma) &= \bigsqcup_{j\geqslant 0}\{\mathbf{xy}\in\overline{\mathcal{D}}_\oplus(A,B,\lambda\ \mathbf{1}_{y,\beta},\alpha,\beta,\gamma) \mid y_\beta = j\} \\ &= \bigsqcup_{j\geqslant 0, j\equiv\lambda \mod \Delta_\beta} \left(\overline{\mathcal{D}}(A,B,j\ \mathbf{1}_{y,\gamma},\alpha,\beta-1,\gamma) + j(\mathbf{1}_{y,\gamma}+\mathbf{1}_{y,\beta})\right).\end{aligned}$$

Then, using the change of variable $j = k + \Delta_\gamma\Delta_\beta\ell$ (with $0 \leqslant k < \Delta_\gamma\Delta_\beta$), observe that $j \equiv k \bmod \Delta_\beta$ and that $j\ \mathbf{1}_{y,\gamma} \equiv k\ \mathbf{1}_{y,\gamma} \mod \mathbb{L}$, whence

$$\begin{aligned}\mathcal{U}_\oplus &= \sum_{j\geqslant 0} \mathbf{1}_{j\equiv\lambda \mod \Delta_\beta}\ \nu_{y,\gamma}^j\ \nu_{y,\beta}^j\ \mathcal{U}(A,B,j\ \mathbf{1}_{y,\gamma},\alpha,\beta-1,\gamma) \\ &= \sum_{\ell\geqslant 0}\sum_{k=0}^{\Delta_\gamma\Delta_\beta-1} \mathbf{1}_{k\equiv\lambda \mod \Delta_\beta}\ (\nu_{y,\gamma}\nu_{y,\beta})^{k+\Delta_\gamma\Delta_\beta\ell}\mathcal{U}(A,B,k\ \mathbf{1}_{y,\gamma},\alpha,\beta-1,\gamma) \\ &= \frac{1}{1-(\nu_{y,\gamma}\nu_{y,\beta})^{\Delta_\gamma\Delta_\beta}}\sum_{k=0}^{\Delta_\gamma\Delta_\beta-1} \mathbf{1}_{k\equiv\lambda \mod \Delta_\beta}\ (\nu_{y,\gamma}\nu_{y,\beta})^{k}\mathcal{U}(A,B,k\ \mathbf{1}_{y,\gamma},\alpha,\beta-1,\gamma).\end{aligned}$$

This shows, in this specific case, that computing $\mathcal{U}_\oplus$ reduces to computing finitely many sums of the form $\mathcal{U}(A,B,\mathbf{w}',\alpha,\beta-1,\gamma)$. Similar constructions are successfully used in all other cases.

5 Conclusion

Performance analysis of infinite-state stochastic systems is a very difficult task. Already checking the ergodicity is difficult in general, and even for systems which are known to be ergodic and which have product-form steady-state distributions, computing the normalising constant can be hard. In this work, we have proposed the model of open Π^3-nets; this model generalises queuing networks and closed produc-form Petri nets and generates a potentially infinite state-space. We have shown that we can efficiently decide (in polynomial time!) many behavioural properties, like the boundedness, the reachability in live nets, and the most important quantitative property: ergodicity. Furthermore, using dynamic programming algorithms managing infinite sums, we have shown that we can compute the normalising constant of the steady-state distribution in pseudo-polynomial time.

We believe our approach can be extended to Π^3-nets in which one place is removed in every layer, without affecting too much the complexity. This setting would allow to model production of resources by the environment while in the current version resources may grow in an unbounded way but only when the number of processes of the main layer also grows. We leave this as future work.

References

1 Parosh Aziz Abdulla, Noomene Ben Henda, Richard Mayr, and Sven Sandberg. Limiting behavior of markov chains with eager attractors. In *Proc. 3rd International Conference on the Quantitative Evaluation of Systems (QEST'06)*, pages 253–264, 2006.

2 Simonetta Balsamo and Andrea Marin. Determining product-form steady-state solutions of generalized stochastic Petri nets by the analysis of the reversed process. In *Proc. 7th IEEE/ACS International Conference on Computer Systems and Applications (AICCSA'09)*, pages 808–815. IEEE Computer Society, 2009.

3 Simonetta Balsamo and Andrea Marin. Performance engineering with product-form models: efficient solutions and applications. In *Proc. 2nd Joint WOSP/SIPEW International Conference on Performance Engineering (ICPE'11)*, pages 437–448. ACM Press, 2011.

4 E. Cardoza, Richard J. Lipton, and Albert R. Meyer. Exponential space complete problems for Petri nets and commutative semigroups: Preliminary report. In *Proc. 8th Annual ACM Symposium on Theory of Computing (STOC'76)*, pages 50–54. ACM Press, 1976.

5 E. Cinlar. *Introduction to Stochastic Processes.* Prentice Hall, 1975.

6 J. L. Coleman, William Henderson, and Peter G. Taylor. Product form equilibrium distributions and a convolution algorithm for stochastic Petri nets. *Performance Evaluation*, 26(3):159–180, 1996.

7 Javier Esparza and Mogens Nielsen. Decidability issues for Petri nets – A survey. *Bulletin of the EATCS*, 52:244–262, 1994.

8 Gerard Florin and Stéphane Natkin. One-place unbounded stochastic Petri nets: Ergodic criteria and steady-state solutions. *Journal of Systems and Software*, 6(1):103 – 115, 1986.

9 William J. Gordon and Gordon F. Newell. Closed queuing systems with exponential servers. *Operations Research*, 15(2):254–265, 1967.

10 Serge Haddad, Jean Mairesse, and Hoang-Thach Nguyen. Synthesis and analysis of product-form Petri nets. *Fundamenta Informaticae*, 122(1-2):147–172, 2013.

11 Serge Haddad, Patrice Moreaux, Matteo Sereno, and Manuel Silva. Product-form and stochastic Petri nets: A structural approach. *Performance Evaluation*, 59(4):313–336, 2005.

12 Peter G. Harrison and Catalina M. Lladó. Hierarchically constructed Petri-nets and product-forms. In *Proc. 5th International ICST Conference on Performance Evaluation Methodologies and Tools Communications (VALUETOOLS'11)*, pages 101–110. ICST/ACM Press, 2011.

13 James R. Jackson. Jobshop-like queueing systems. *Management Science*, 10(1):131–142, 1963.

14 Matthias Jantzen. Complexity of place/transition nets. In *Proc. Advances in Petri Nets (1986)*, volume 254 of *Lecture Notes in Computer Science*, pages 413–434. Springer, 1987.

15 Aurel A. Lazar and Thomas G. Robertazzi. Markovian Petri net protocols with product form solution. *Performance Evaluation*, 12(1):67 – 77, 1991.

16 Man Li and Nicolas D.Georganas. Exact parametric analysis of stochastic Petri nets. *IEEE Transactions on Computers*, 41:1176–1180, 1992.

17 Jean Mairesse and Hoang-Thach Nguyen. Deficiency zero Petri nets and product form. *Fundamenta Informaticae*, 105(3):237–261, 2010.

18 Andrea Marin, Simonetta Balsamo, and Peter G. Harrison. Analysis of stochastic Petri nets with signals. *Performance Evaluation*, 69(11):551–572, 2012.

19 Marco Ajmone Marsan, Gianfranco Balbo, Gianni Conte, Susanna Donatelli, and Giuliana Franceschinis. *Modelling with Generalized Stochastic Petri Nets.* John Wiley & Sons, Inc., 1995.

20 Michael K. Molloy. Performance analysis using stochastic Petri nets. *IEEE Transactions on Computers*, 31(9):913–927, 1982.

21 J.L. Peterson. *Petri Net Theory and the Modeling of Systems.* Prentice Hall, 1981.

22 Charles Rackoff. The covering and boundedness problems for vector addition systems. *Theoretical Computer Science*, 6:223–231, 1978.

23 Matteo Sereno and Gianfranco Balbo. Mean value analysis of stochastic Petri nets. *Performance Evaluation*, 29(1):35–62, 1997.