On the complexity of some state-counting problems for bounded Petri nets

Spyros Reveliotis

Abstract— **Motivated by an emerging need for pertinent sizing and indexing of various data structures that are used for the efficient storage and processing of the reachability graph of certain bounded PN subclasses, this work investigates the complexity of the cardinality assessment of various marking sets that have been proposed as reasonable (over-)approximations for the set of reachable markings. Along these lines, our main results establish the #*P*-hardness of the aforementioned estimation for the most prominent of these marking sets. To the best of our knowledge, this is also a first attempt to provide formal #*P*-hardness results for counting problems that arise in the PN (and the broader DES) modeling framework.**

I. Introduction

As suggested by its title, this paper deals with the complexity of certain state-counting problems for bounded Petri nets (PNs). At a more immediate level, the presented results can be perceived as an attempt to provide systematic and thorough responses to some complexity questions that were recently raised in [1] and stand open in the relevant literature.[1] At the same time, the overall positioning of the considered problems, the presented analysis, and the derived conclusions are developed at a much greater generality than the motivating themes of [1], and they contribute to the development of a formal counting theory adjusted to the dynamics and the needs of the PN modeling framework. Hence, when viewed from this broader standpoint, the presented developments parallel, in spirit, and complement some fairly recent developments presented in [2]. But while the work of [2] is interested in developing pertinent bounds for the targeted state sets, the work presented in this paper investigates the computational complexity of the counting task itself, and develops the necessary perspectives and tools for establishing intractability results; in fact, to the best of the author's knowledge, this is the first attempt to establish formal intractability results for counting problems that arise in the PN modeling framework.[2]

In more concrete terms, the basic counting problem addressed in this paper is the estimation of the cardinality of the reachable-state set of any bounded PN. This problem is important for the sizing of the data structures that might be necessary for a succinct representation of the underlying net dynamics [4], while, more recently, the same problem has been related to the design of efficient indexing schemes for these data structures [1]. But due to the complexity of the underlying net dynamics, the development of an exact estimate of the target quantity is deemed practically impossible. Hence, the relevant community usually strives for the computation of some reasonable / fairly tight upper bound of the target value. The ratio of the computed bound to the actual target value is characterized as the "inflation ratio", and it quantifies the "waste" that is incurred by the employed storage scheme. For many bounded PNs, some reasonable (over-)estimates of the size of the underlying state space are provided by the sets of states that satisfy (a relaxed version of) the state equation or certain invariants that are present in the net dynamics [5].[3] Indeed, for many practical applications, the cardinality of these supersets of the net state space is of the same order of magnitude as the cardinality of the state space itself. Hence, sizing the employed data structures based on the size of these surrogate sets leads to fairly efficient storage schemes; we refer to [4], [1] for some more concrete examples regarding these claims.

In view of the above remarks, it is pertinent to ask how easy it is to evaluate the cardinality of the surrogate state sets that were described in the previous paragraph, for any bounded PN. The main contribution of this work is to formally show that the considered counting problems constitute hard problems within the class of counting problems; in more formal terms, they are #*P*-hard problems [6]. We establish this result by analyzing the restriction of the considered counting problems to a particular PN sub-class that is known as the class of Gadara PNs [7], and has received particular attention in the recent years since it models the allocation of mutex locks in the multithreaded software used in the emergent multi-core computer architectures. Hence, the rest of the paper is organized as follows: In Section II we introduce the PN concepts that are needed for the developments of this work, and the particular class of the Gadara PNs that will be the focus of attention in the subsequent developments. In Section III we provide the necessary background on the class #*P* of counting problems, the notions of #*P*-hardness and #*P*-completeness, and some particular #*P*-completeness results that will be used in the establishment of the main results of this paper. These results is the subject of Section IV. Finally, Section V concludes the paper by summarizing the presented results, and pointing directions for future work.

II. Petri nets and the Gadara sub-class

A. Petri nets: basic concepts and definitions

A formal definition of the basic Petri net model that is considered in this work, is as follows:

Definition 1: [8] A *Petri net (system)* is defined by a quadruple $\mathcal{N} = (P,T,W,M_0)$, where

- P is the set of *places*,
- T is the set of *transitions*,
- $W : (P\times T)\cup(T\times P) \rightarrow \mathbb{Z}_0^+$ is the *flow relation*, and
- $M_0 : P \rightarrow \mathbb{Z}_0^+$ is the net *initial marking*, assigning to each place $p \in P$, $M_0(p)$ *tokens*.

The first three items in Definition 1 essentially define a weighted bipartite digraph representing the system structure that governs its underlying dynamics. The last item defines the system initial state.

PN structure-related concepts and properties Given a transition $t \in T$, the set of places p for which $W(p,t) > 0$ (resp., $W(t,p) > 0$) is known as the set of *input* (resp., *output*) places of t. Similarly, given a place $p \in P$, the set of transitions t for which $W(t,p) > 0$ (resp., $W(p,t) > 0$) is known as the set of *input* (resp., *output*) transitions of p. It is customary in the PN literature to denote the set of input (resp., output) transitions of

S. Reveliotis is with the School of Industrial & Systems Engineering, Georgia Institute of Technology, email: spyros@isye.gatech.edu

[1] We detail these questions and their relationship to the presented developments in the more technical parts of this paper.

[2] In fact, the validity of this statement also extends to the broader modeling framework of Discrete Event Systems (DES) [3].

[3] The PN modeling framework and all the necessary concepts that are needed for the systematic development of the presented results, are introduced in the subsequent sections of this work.

a place p by $\bullet p$ (resp., $p\bullet$). Similarly, the set of input (resp., output) places of a transition t is denoted by $\bullet t$ (resp., $t\bullet$). This notation is also generalized to any set of places or transitions, X, e.g. $\bullet X = \bigcup_{x \in X} \bullet x$.

The ordered set $X = < x_1, \ldots, x_n > \in (P \cup T)^*$ is a *path*, if and only if (*iff*) $x_{i+1} \in x_i\bullet, i = 1, \ldots, n-1$. Furthermore, a path X is characterized as a *circuit iff* $x_1 = x_n$.

A PN with a flow relation W mapping onto $\{0,1\}$ is said to be *ordinary*. An ordinary PN such that (s.t.) $\forall t \in T$, $|t\bullet| = |\bullet t| = 1$, is characterized as a *state machine*.

A PN is said to be *pure* if $\forall (x,y) \in (P \times T) \cup (T \times P)$, $W(x,y) > 0 \Rightarrow W(y,x) = 0$. The flow relation of pure PNs can be represented by the $|P| \times |T|$-dimensional *flow matrix* $\Theta(\mathcal{N})$; in particular, $\Theta(\mathcal{N}) = \Theta^+(\mathcal{N}) - \Theta^-(\mathcal{N})$ where $\Theta^+(p,t;\mathcal{N}) = W(t,p)$ and $\Theta^-(p,t;\mathcal{N}) = W(p,t)$. Also, a non-pure PN can be easily converted to a pure one in a way that the dynamics of the converted net express unambiguously the dynamics of the original system [8]; hence, in the following we shall focus on the class of pure PNs.

PN dynamics-related concepts and properties In the PN modeling framework, the system state is represented by the net *marking* M, i.e., a function from P to $\mathbb{Z}_0^+$ that assigns a *token* content to the various places of the net. The net marking M is initialized to marking M_0, introduced in Definition 1, and it subsequently evolves through a set of rules summarized in the concept of *transition firing*. A concise characterization of this concept is as follows: Given a marking M, a transition t is *enabled iff* for every place $p \in \bullet t$, $M(p) \geq W(p,t)$, and this is denoted by $M[t\rangle$. On the other hand, $t \in T$ is said to be *disabled* by a place $p \in \bullet t$ at M *iff* $M(p) < W(p,t)$. Given a marking M, a transition t can be *fired* only if it is enabled in M, and firing such an enabled transition t results in a new marking M', which is obtained from M by removing $W(p,t)$ tokens from each place $p \in \bullet t$, and placing $W(t,p')$ tokens in each place $p' \in t\bullet$.

The set of markings reachable from the initial marking M_0 through any *fireable* sequence of transitions is denoted by $\mathcal{R}(\mathcal{N})$ and it is referred to as the net *reachability space*. In the following, we also use the notation $\mathcal{R}(\mathcal{N}, M)$ to denote the reachability space of net $\mathcal{N}$ when it is initialized at a marking M not necessarily equal to M_0.[4] A PN $\mathcal{N} = (P,T,W,M_0)$ is said to be *bounded iff* there exists some constant k such that $M(p) \leq k$, for all places $p \in P$ and all markings $M \in \mathcal{R}(\mathcal{N})$. $\mathcal{N}$ is said to be *structurally bounded iff* it is bounded for any initial marking M_0. Obviously, bounded PNs have a finite reachability space.

The state equation and the corresponding marking sets For pure PNs, the marking evolution incurred by the firing of a transition t can be concisely expressed by the *state equation*:

$$M' = M + \Theta(\mathcal{N}) \cdot \mathbf{1}_t \tag{1}$$

where $\mathbf{1}_t$ denotes the unit vector of dimensionality $|T|$ and with the unit element located at the component corresponding to transition t. Hence, in the case of pure PNs, a necessary condition for any vector $M \in (\mathbb{Z}_0^+)^{|P|}$ to constitute a reachable marking, is that the following system of equations is feasible in z:

$$M = M_0 + \Theta(\mathcal{N}) \cdot z\,; \;\; z \in (\mathbb{Z}_0^+)^{|T|} \tag{2}$$

[4]Under this more general notation, $\mathcal{R}(\mathcal{N}) \equiv \mathcal{R}(\mathcal{N}, M_0)$.

For a given (pure) PN $\mathcal{N} = (P,T,W,M_0)$, let

$$\mathcal{SE}(\mathcal{N}) \equiv \{M \in (\mathbb{Z}_0^+)^{|P|} :\; \exists z \in (\mathbb{Z}_0^+)^{|T|} \text{ s.t. } M = M_0 + \Theta(\mathcal{N}) \cdot z\} \tag{3}$$

We shall refer to $\mathcal{SE}(\mathcal{N})$ as the set of markings satisfying the state equation of net $\mathcal{N}$. We shall also consider an expanded version of the set $\mathcal{SE}(\mathcal{N})$, to be denoted by $\widehat{\mathcal{SE}}(\mathcal{N})$, that is obtained from $\mathcal{SE}(\mathcal{N})$ by relaxing the integrality requirement for the elements of the "counting" vector z:

$$\widehat{\mathcal{SE}}(\mathcal{N}) \equiv \{M \in (\mathbb{Z}_0^+)^{|P|} :\; \exists z \in (\mathbb{R}_0^+)^{|T|} \text{ s.t. } M = M_0 + \Theta(\mathcal{N}) \cdot z\} \tag{4}$$

$\widehat{\mathcal{SE}}(\mathcal{N})$ is characterized as the set of markings satisfying the "relaxed" state equation of net $\mathcal{N}$. Furthermore it is evident from the above definitions of $\mathcal{R}(\mathcal{N})$, $\mathcal{SE}(\mathcal{N})$ and $\widehat{\mathcal{SE}}(\mathcal{N})$, that

$$\mathcal{R}(\mathcal{N}) \;\subseteq\; \mathcal{SE}(\mathcal{N}) \;\subseteq\; \widehat{\mathcal{SE}}(\mathcal{N}) \tag{5}$$

In [5] it is shown that the two inclusions appearing in Equation 5 can be strict.

PN invariants and the induced marking sets The PN modeling framework recognizes two notions of invariance: p and t-invariants. For the purposes of the developments pursued in this work, we define as a p-invariant of a PN $\mathcal{N}$, any non-zero $|P|$-dimensional vector y satisfying

$$y^T \cdot \Theta(\mathcal{N}) = 0 \tag{6}$$

In the light of Equation 1, the invariance property in the net dynamics that is represented by a p-invariant y, refers to the fact that

$$\forall M \in \widehat{\mathcal{SE}}(\mathcal{N}),\;\; y^T \cdot M = y^T \cdot M_0 \tag{7}$$

Equation 6 further implies that the p-invariants of a PN $\mathcal{N}$ constitute a vector space, and therefore, they can be collectively represented by any basis B of this space. In particular, the set of markings M that satisfy all the p-invariants of net $\mathcal{N}$, in the sense of Equation 7, can be expressed as follows:

$$\mathcal{PI}(\mathcal{N}) \equiv \{M \in (\mathbb{Z}_0^+)^{|P|} :\; B^T \cdot M = B^T \cdot M_0\} \tag{8}$$

Equations 4, 6 and 8 also imply that

$$\widehat{\mathcal{SE}}(\mathcal{N}) \;\subseteq\; \mathcal{PI}(\mathcal{N}) \tag{9}$$

while [5] demonstrates that the above inclusion can be strict.

A p-invariant y is said to be a p-semiflow if it also holds that $y \geq 0$. Given a p-semiflow y, its *support* is defined as $\|y\| = \{p \in P \mid y(p) > 0\}$. A p-semiflow y is said to be of *minimal support iff* there is no p-semiflow y' s.t. $\|y'\| \subset \|y\|$. On the other hand, a p-semiflow y is said to be *minimal* if it is of minimal support and its non-zero elements are relatively prime. The set of all minimal p-semiflows of a PN $\mathcal{N}$ is unique and it is called the *fundamental set of p-semiflows* [5]. Let the fundamental set of p-semiflows of a PN $\mathcal{N}$ be collectively represented by matrix $\Phi(\mathcal{N})$; i.e., each column of $\Phi(\mathcal{N})$ is a minimal p-semiflow of $\mathcal{N}$. Then, the set of markings satisfying all the minimal p-semiflows of $\mathcal{N}$ can be compactly expressed by:

$$\mathcal{PSF}(\mathcal{N}) \equiv \{M \in (\mathbb{Z}_0^+)^{|P|} :\; \Phi(\mathcal{N})^T \cdot M = \Phi(\mathcal{N})^T \cdot M_0\} \tag{10}$$

In [5] it is also shown that every p-semiflow of $\mathcal{N}$ can be obtained as a non-negative linear combination of its minimal p-semiflows. Hence, the set $\mathcal{PSF}(\mathcal{N})$ contains all the markings M of $\mathcal{N}$ that satisfy every p-semiflow of this net. Furthermore, since the set of p-semiflows is only a subset of the set of the p-invariants of any given PN $\mathcal{N}$, it also holds that

$$\mathcal{PI}(\mathcal{N}) \subseteq \mathcal{PSF}(\mathcal{N}) \tag{11}$$

Once again, [5] also demonstrates that the above inclusion can be strict.

We close this section with some remarks that relate to the well-posedness of the counting problems that are defined by the sets $\widehat{\mathcal{SE}}(\mathcal{N})$, $\mathcal{PI}(\mathcal{N})$ and $\mathcal{PSF}(\mathcal{N})$, and some relationships that exist among these problems. It is evident from the definition of the set $\mathcal{PSF}(\mathcal{N})$ in Eq. 10, and the non-negative and integral nature of the elements of $\Phi(\mathcal{N})$ and M, that $\mathcal{PSF}(\mathcal{N})$ will have a finite cardinality *iff* the matrix $\Phi(\mathcal{N})$ has no zero rows. When combined with the discussion that follows the definition of the set $\mathcal{PSF}(\mathcal{N})$, this last remark further implies that $\mathcal{PSF}(\mathcal{N})$ will have a finite cardinality *iff* the net $\mathcal{N}$ has a strictly positive p-semiflow; such a PN $\mathcal{N}$ is characterized as *conservative*.

Furthermore, we define as a t-invariant of a PN $\mathcal{N} = (P,T,W,M_0)$ any $|T|$-dimensional vector x such that[5]

$$\Theta(\mathcal{N}) \cdot x = 0 \tag{12}$$

Similar to the case of p-invariants, we define as a t-semiflow any t-invariant with nonnegative elements. In the particular case where a PN $\mathcal{N}$ possesses a t-semiflow x with strictly positive elements, it is said to be *consistent*. The following proposition reveals some further implications of the notions of PN conservatism and consistency in the context of our work, and it is formally established in [5].

Proposition 1: Consider a PN $\mathcal{N}$. Then, the following holds true:
1. If $\mathcal{N}$ is conservative, $\mathcal{PI}(\mathcal{N}) = \mathcal{PSF}(\mathcal{N})$.
2. If $\mathcal{N}$ is consistent, $\widehat{\mathcal{SE}}(\mathcal{N}) = \mathcal{PI}(\mathcal{N})$.

B. Gadara Petri nets

Gadara PNs were introduced in [7] as a particular PN class that models the allocation dynamics of the mutex locks among the various threads of multithreaded software. A formal definition of a Gadara net is as follows:

Definition 2: [7] Let $I_{\mathcal{N}} = \{1,2,\ldots,m\}$ be a finite set of indices. A Gadara PN is an ordinary, pure PN $\mathcal{N}_G = (P,T,W,M_0)$ where:
1. $P = P_0 \cup P_S \cup P_R$ is a partition of the net places such that: (a) $P_S = \bigcup_{i\in I_{\mathcal{N}}} P_{S_i}$, $P_{S_i} \neq \emptyset$, and $P_{S_i} \cap P_{S_j} = \emptyset$ for all $i \neq j$; (b) $P_0 = \bigcup_{i\in I_{\mathcal{N}}} P_{0_i}$, where $P_{0_i} = \{p_{0_i}\}$; and (c) $P_R = \{r_1, r_2, \ldots, r_k\}$, $k > 0$.
2. $T = \bigcup_{i\in I_{\mathcal{N}}} T_i$; $T_i \neq \emptyset$; and $T_i \cap T_j = \emptyset$, for all $i \neq j$.
3. For all $i \in I_{\mathcal{N}}$, the subnet $\mathcal{N}_i$ generated by $P_{S_i} \cup \{p_{0_i}\} \cup T_i$ is a strongly connected state machine. There are no direct connections between the elements of $P_{S_i} \cup \{p_{0_i}\}$ and T_j for any pair $\{i, j\}$ with $i \neq j$.
4. $\forall p \in P_S$, if $|p\bullet| > 1$, then $\forall t \in p\bullet$, $\bullet t \cap P_R = \emptyset$.
5. For each $r \in P_R$, there exists a unique minimal-support p-semiflow, y_r, such that (a) $\{r\} = ||y_r|| \cap P_R$; (b) $\forall p \in ||y_r||$, $y_r(p) = 1$; (c) $P_0 \cap ||y_r|| = \emptyset$; and (d) $P_S \cap ||y_r|| \neq \emptyset$.
6. $\forall r \in P_R$, $M_0(r) = 1$; $\forall p \in P_S$, $M_0(p) = 0$; and $\forall p_0 \in P_0$, $M_0(p_0) \geq 1$.
7. $P_S = \bigcup_{r\in P_R}(||y_r|| \setminus \{r\})$.

Items 1-3 of Definition 2 imply that a Gadara PN essentially consists of m subnets $\mathcal{N}_i$ that have the structure of a strongly connected state machine and interact through the places in the set P_R. Each subnet $\mathcal{N}_i$, $i \in I_{\mathcal{N}}$, models the sequential logic of a thread executing in one of its critical regions. In particular, tokens in places of P_{S_i} model instances of this thread type executing various stages in the corresponding critical region, while place p_{0_i} models instances waiting to enter (or exiting) this critical region. We shall refer to subnets $\mathcal{N}_i$, $i \in I_{\mathcal{N}}$, as the *process* subnets, and to the particular places p_{0_i} as the corresponding *idle places*. Item 6 of Definition 2 implies that in the initial marking M_0 of a Gadara PN, no process instance has entered any of the critical regions modeled by its process subnets, but for each subnet $\mathcal{N}_i$, there are some process instances waiting in the idle place p_{0_i}. On the other hand, at M_0, each place in P_R contains a single token modeling the availability of the corresponding mutex lock. Item 5 of Definition 2 implies that each of these locks r_i, $i \in \{1,\ldots,k\}$, is allocated in an exclusive and reusable manner to the process instances that execute some of the net processing stages, more specifically, the processing stages in $||y_r||$. Item 7 of Definition 2 further stipulates that each processing stage modeled by a place $p \in P_S$ requests at least one mutex lock for its execution (since it is a stage executing in a critical region). Finally, item 4 establishes a separation of the lock allocation function from the routing decisions that are effected by a process instance while in its critical region; this element is important for the pertinent modeling of the dynamics that are induced by the considered resource allocation, but it is not relevant to the subsequent developments.

As revealed by the above discussion, an important class of minimal p-semiflows for Gadara PNs is defined by the limited and reusable nature of the allocated locks. In particular, according to items 5 and 6 of Definition 2, the allocation of the lock corresponding to each place r_i, $i \in \{1,\ldots,k\}$, implies the following invariant for the marking of a Gadara net $\mathcal{N}_G$:

$$\forall M \in R(\mathcal{N}_G), \quad \sum_{p\in||y_{r_i}||\setminus\{r_i\}} M(p) + M(r_i) = 1 \tag{13}$$

An additional class of minimal p-semiflows is induced by the structure of the process subnets $\mathcal{N}_i$, $i \in I_{\mathcal{N}}$, that is stipulated by item 3 of Definition 2. More specifically, the strongly-connected-state-machine type of each subset $\mathcal{N}_i$ implies the minimal p-semiflow:

$$\forall M \in R(\mathcal{N}_G), \quad \sum_{p\in P_{S_i}} M(p) + M(p_{0_i}) = M_0(p_{0_i}) \tag{14}$$

It turns out that the p-semiflows defined by Equations 13 and 14 constitute the entire set of the minimal p-semiflows of any Gadara PN $\mathcal{N}_G$. Hence, according to the discussion of Section II-A, any marking M that satisfies Equations 13 and 14 will satisfy any other p-semiflow of $\mathcal{N}_G$. An additional implication of Equations 13 and 14 is that the availability of the

[5] The invariance property corresponding to the t-invariant x is expressed by the fact that $M = M_0 + \Theta(\mathcal{N}) \cdot x = M_0$.

sub-marking $M_S \equiv \{M(p) : p \in P_S\}$ for any marking M that satisfies these equations, specifies completely the entire marking M. Indeed, each marking $M(r_i)$, for $i = 1, \ldots, k$, is specified by the corresponding instantiation of Equation 13, while the instantiations of Equation 14 specify accordingly the corresponding markings $M(p_{0_i})$. Hence, for a Gadara net $\mathcal{N}_G$, we can express the set $\mathcal{PSF}(\mathcal{N}_G)$, containing the markings M that satisfy the p-semiflows of $\mathcal{N}_G$, by the following system of inequalities:

$$\forall r_i \in P_R, \quad \sum_{p \in ||y_{r_i}|| \setminus \{r_i\}} M(p) \leq 1 \tag{15}$$

$$\forall i \in I_{\mathcal{N}}, \quad \sum_{p \in P_{S_i}} M(p) \leq M_0(p_{0_i}) \tag{16}$$

Equation 15, when combined with item 7 of Definition 2 and the nonnegative and integral nature of the PN marking, further implies that the sub-marking M_S of any marking M that satisfies the p-semiflows of any Gadara net $\mathcal{N}_G$, is a *binary* vector.[6] Furthermore, in most practical instantiations of Gadara nets, the inequalities of Equation 16 are rendered implicit by setting[7]

$$\forall i \in I_{\mathcal{N}}, \ M_0(p_{0_i}) \geq |P_{S_i}| \tag{17}$$

Hence, in the following, we assume that the condition of Equation 17 replaces the condition '$\forall i \in I_{\mathcal{N}}, \ M_0(p_{0_i}) \geq 1$' in item 6 of Definition 2, and thus, the corresponding sets $\mathcal{PSF}(\mathcal{N}_G)$ are characterized by the sets $\mathcal{M}_S(\mathcal{N}_G)$ containing the sub-markings M_S that satisfy Equation 15.

The work of [1] proposes an algorithm for evaluating $|\mathcal{M}_S(\mathcal{N}_G)|$ (and therefore, $|\mathcal{PSF}(\mathcal{N}_G)|$). However, the algorithm presented in [1] is of exponential computational complexity with respect to (w.r.t.) $|P_R|$, a fact that raises the question of the existence of any algorithms of polynomial complexity w.r.t. the size of (the constituent elements of) $\mathcal{N}_G$ for the execution of this counting task. In Section IV we shall show that computing $|\mathcal{M}_S(\mathcal{N}_G)|$ is #P-complete; hence, the existence of such an efficient counting algorithm is highly unlikely. Furthermore, this negative result extends to the estimation of the cardinality of the sets $\widehat{\mathcal{SE}}(\mathcal{N}_G)$ and $\mathcal{PI}(\mathcal{N}_G)$. Instrumental for this extension is the following proposition:

Proposition 2: Any Gadara PN $\mathcal{N}_G$ is, both, conservative and consistent.

The result of Proposition 2 is well known to the relevant community and, therefore, its proof is omitted; a relevant formal argument can be found, for instance, in [9]. On the other hand, the significance of Proposition 2 for the needs of the undertaken analysis is revealed by the next corollary; this corollary results immediately when combining Proposition 2 with Proposition 1 and the discussion that precedes this second proposition in Section II-A.

Corollary 1: For any Gadara net $\mathcal{N}_G$,

$$\widehat{\mathcal{SE}}(\mathcal{N}_G) = \mathcal{PI}(\mathcal{N}_G) = \mathcal{PSF}(\mathcal{N}_G) \tag{18}$$

[6] In more natural terms, this result expresses the fact that no processing stage can be executed simultaneously by two or more process instances, since these process instances would have to share the mutex locks that are required by this processing stage.

[7] Since the marking $M(p)$ of each place $p \in P_S$ in a Gadara PN is a binary variable, under the selections for $M_0(p_{0_i})$, $i \in I_{\mathcal{N}}$, proposed in Equation 17, the restrictions of Equation 16 are implied by those of Equation 15. This effect is motivated naturally by the need to avoid situations where the dynamics of the considered Gadara net are shaped by artificially/externally imposed restrictions on the number of the processes that might execute concurrently in the various critical regions, and not by the allocation patterns of the various locks that are contested by these processes.

Furthermore, all these sets are of finite cardinality, and the corresponding counting problems are well-defined.

III. The class #P of counting problems and the notion of #P-completeness

In this section we provide a set of results that define a framework for analyzing the complexity of counting problems. This framework was originally introduced by Valiant in [6], [10], but the exposition that we adopt here is based on the coverage of this material in [11]. Furthermore, in the following, we assume that the reader is familiar with the basic concepts relating to the problem class NP and to NP-completeness.

Definition 3: Let $R \subseteq \Sigma^* \times \Sigma^*$ be a binary relation on the strings that are generated by some alphabet Σ. Also, for any string $x \in \Sigma^*$, let $|x|$ denote its length. Then:
1. R is called *polynomially decidable* if there is a deterministic Turing machine deciding the language $L = \{x;y : (x,y) \in R\}$ in polynomial time.
2. R is called *polynomially balanced* if there is some $k \geq 1$ such that for all $(x,y) \in R$, $|y| \leq |x|^k$.

Definition 4: Let R be a polynomially balanced and polynomially decidable binary relation defined on the strings of some alphabet Σ. The *counting problem* associated with R is defined as follows: *"Given $x \in \Sigma^*$, how many $y \in \Sigma^*$ are there such that $(x,y) \in R$?"*. Furthermore, the class of all counting problems associated with polynomially balanced and polynomially decidable relations is denoted by #P.

It is evident from the above that a counting problem Q in #P is defined by a triplet (Σ, R, x), where Σ is an alphabet set, R is a polynomially balanced and polynomially decidable binary relation defined on Σ^*, and x is an element of Σ^*. Such a triplet also defines the *decision problem* $\hat{Q}$ of whether there exists *any* $y \in \Sigma^*$ such that $(x,y) \in R$. Furthermore, Definition 3 implies that problem $\hat{Q}$ is in NP.

For more general counting problems Q (not necessarily in #P), we can still consider the decision problem $\hat{Q}$ of whether the set of counted entities is non-empty. Also, a reduction from a decision problem $\hat{Q}_1$ to a decision problem $\hat{Q}_2$ is called *parsimonious* if it preserves the number of solutions for the two problems. Then, in view of all the above definitions, the notions of #P-hardness and #P-completeness can be defined as follows:

Definition 5: A counting problem Q is called #P-hard *iff*, for any other counting problem U in #P, there is a polynomial-time parsimonious reduction of the decision problem $\hat{U}$ to $\hat{Q}$. A #P-hard counting problem Q will be called #P-complete *iff* Q also belongs in #P.

In [10] it was established that counting all the satisfying truth assignments of a monotone 2-SAT problem is #P-complete.[8] Next, we use this result to establish the #P-completeness of counting certain objects in graphs. These objects are formally introduced in the following definition, and the corresponding counting problems will be used in Section IV for the establishment of the main result of this paper.

Definition 6: Consider an (undirected) graph $G = (V,E)$, where V denotes the node (or vertex) set, and E denotes the

[8] A 2-SAT problem is a SAT problem in Conjunctive Normal Form (CNF) where every clause contains only two literals in its disjunction. On the other hand, a SAT problem in CNF is monotone if its defining Boolean expression contains no negations.

edge set; every edge $e \in E$ corresponds to an (unordered) pair of nodes $\{v_1, v_2\}$.
1. A subset V^{co} of V is a *(vertex) cover* of G *iff* every edge has at least one node in V^{co}.
2. A subset V^{is} of V is an *independent set* of G *iff* there is no pair of nodes in V^{is} connected by an edge in G.

It can be easily checked that the following proposition is also true; a formal proof can be found in any introductory text of graph theory (e.g., [12]).

Proposition 3: Consider a graph $G = (V, E)$. Then, V^{co} is a vertex cover of G *iff* $V \setminus V^{co}$ is an independent set of G.

Now we are ready to state the key result that will provide the reduction employed in the main result of this paper. A formal proof of this result is provided, for completeness, in the Appendix.

Proposition 4: Counting the covers and the independent sets of any graph $G = (V, E)$, with finite vertex set V, are #P-complete problems.

IV. The main results

In this section we provide the main results of the paper, i.e., we show that assessing the cardinality of the marking sets $\widehat{\mathcal{SE}}(\mathcal{N}_G)$, $\mathcal{PI}(\mathcal{N}_G)$ and $\mathcal{PSF}(\mathcal{N}_G)$ of a Gadara net $\mathcal{N}_G$ is a #P-complete problem, which further renders the cardinality assessment of the corresponding sets for any (conservative) bounded PN $\mathcal{N}$, a #P-hard problem. We also indicate some conditions under which the cardinality assessment of the aforementioned sets becomes #P-complete. The next theorem is instrumental for establishing the aforementioned results.

Theorem 1: The problem of assessing the cardinality of the set $\mathcal{M}_S(\mathcal{N}_G)$, containing the sub-markings M_S that satisfy Equation 15 for a Gadara net $\mathcal{N}_G$, is #P-complete.

Proof: First we notice that the considered counting problem belongs in #P since (i) the sub-markings in $\mathcal{M}_S(\mathcal{N}_G)$ are binary vectors for any Gadara net $\mathcal{N}_G$ (and thus, the relation R that is implied by Definition 4 for the considered counting problem is polynomially balanced), and (ii) the verification of Equation 15 for any given marking M is a task of polynomial complexity w.r.t. the size of $\mathcal{N}_G$ (and thus, the aforementioned relation R is polynomially decidable).

Next, we prove the #P-hardness of the considered problem by providing a polynomial and parsimonious reduction to this problem of the #P-complete problem of counting the independent sets of any given graph G. Hence, let $G = (V, E)$ with finite vertex set $V = \{v_1, \ldots, v_m\}$ and edge set $E = \{e_1, \ldots, e_n\}$, and construct a Gadara net $\mathcal{N}_G$ as follows: The place set P_S of $\mathcal{N}_G$ is in one-to-one correspondence with the node set V of G; let $p(v_i)$ denote the place in P_S corresponding to node $v_i \in V$. All places $p \in P_S$ belong to a single process subnet with an additional idle place p_0 and the structure of a *circuit* that is defined by the place ordering $\langle p_0, p(v_1), \ldots, p(v_m)\rangle$, after the introduction of the necessary transitions. On the other hand, each edge $e_j \in E$ induces a resource place $r(e_j)$ in the place set P_R of the proposed net $\mathcal{N}_G$. More specifically, for every edge $e_j \in E$, let $e_j = \{v_{1(j)}, v_{2(j)}\}$, $1(j), 2(j) \in \{1, \ldots, m\}$, and further assume that $1(j) \neq 2(j)$; a close examination of the proof of Proposition 4 will reveal that the imposition of the restriction $1(j) \neq 2(j)$ for the considered graphs G retains the result of that proposition, and therefore, the pursued reduction is still valid. The p-semiflow $y_r(e_j)$ of item 5 in Definition 2, that is associated with place $r(e_j)$ in P_R, is completely determined by specifying the corresponding support $||y_r(e_j)||$ according to the following rule:

$$||y_r(e_j)|| = \{r(e_j), p(v_{1(j)}), p(v_{2(j)})\} \tag{19}$$

The specification of the p-semiflow $y_r(e_j)$, for every $e_j \in E$, also determines completely the connectivity of the corresponding place $r(e_j)$ to the transitions of net $\mathcal{N}_G$. However, in the case that the considered graph G contains isolated vertices (i.e., vertices with no incident edges[9]), the above construction would violate Condition 7 of Definition 2. To deal with this issue, we also associate with each place $p(v_i)$ corresponding to a zero-degree vertex v_i, a resource place $r(v_i)$ that is used exclusively by the process place $p(v_i)$; i.e., for each such resource place $r(v_i)$,

$$||y_r(v_i)|| = \{r(v_i), p(v_i)\} \tag{20}$$

Hence, eventually the resource-place set P_R consists of all places $r(e_j)$, for each edge $e_j \in E$, and the places $r(v_i)$ corresponding to zero-degree vertices v_i. Finally, the specification of the net $\mathcal{N}_G$ is completed by specifying its initial marking M_0 by setting $M_0(p) = 0$, for every $p \in P_S$, $M_0(r) = 1$, for every $r \in P_R$, and $M_0(p_0) = |V|$.

It can be easily checked that, for any given graph G, the net $\mathcal{N}_G$ that is constructed as described in the previous paragraph, satisfies all the conditions of Definition 2, and it is, therefore, a Gadara net. The constructed net $\mathcal{N}_G$ also observes the condition of Equation 17 in Section II-B. Finally, it should be clear that the above construction is a task of polynomial computational complexity w.r.t. the size of the input graph G. Next we show that a binary vector M_S of dimensionality equal to $|P_S|$ will satisfy Equation 15 for the constructed net $\mathcal{N}_G$ *iff* the nodes v_i corresponding to the places $p(v_i) \in P_S$ with $M_S(p(v_i)) = 1$ constitute an independent set of G. Hence, counting the independent sets of G is equivalent to assessing the cardinality of the set $\mathcal{M}_S(\mathcal{N}_G)$, which renders the latter a #P-hard problem.

To establish the aforestated equivalence, first assume that vector M_S satisfies Equation 15. Hence, there is no pair of places p_1, p_2 of P_S with $M_S(p_1) = M_S(p_2) = 1$ that belong to the same support $||y_r||$ for some $r \in P_R$. But then, Equation 19 implies that there is no edge connecting nodes of V corresponding to any pair of places p_1, p_2 of P_S with $M_S(p_1) = M_S(p_2) = 1$. Hence, the nodes v_i corresponding to the places $p(v_i) \in P_S$ with $M_S(p(v_i)) = 1$ are, indeed, an independent set of G. Next, consider an independent set V^{is} of G. Then, each support $||y_r||$ that is defined by Equation 19 contains at most one place $p \in P_S$ corresponding to an element of V^{is}. Hence, the binary vector M_S of dimensionality equal to $|P_S|$ and with $M_S(p) = 1$ *iff* the corresponding node v belongs in V^{is}, is a feasible solution for the inequalities of Equation 15. □

The following corollary results immediately from the combination of Theorem 1 with the definition of the set $\mathcal{M}_S(\mathcal{N}_G)$ in Section II-B and Corollary 1 in the same section.

Corollary 2: The counting problems of assessing the cardinality of the sets $\widehat{\mathcal{SE}}(\mathcal{N}_G)$, $\mathcal{PI}(\mathcal{N}_G)$ and $\mathcal{PSF}(\mathcal{N}_G)$, for a Gadara net $\mathcal{N}_G$, are #P-complete.

The next corollary generalizes the result of Corollary 2 to the broader class of bounded PNs.

[9] Such vertices are characterized as "zero-degree".

Corollary 3: The counting problems of assessing the cardinality of the sets $\widehat{\mathcal{SE}}(\mathcal{N})$, $\mathcal{PI}(\mathcal{N})$ and $\mathcal{PSF}(\mathcal{N})$, for a bounded PN $\mathcal{N}$, are #*P*-hard.

Proof: Corollary 3 results immediately from Corollary 2 upon noticing that the counting problems addressed in Corollary 2 are special cases of the corresponding counting problems that are considered in this new corollary. □

Since, in general, the sets $\widehat{\mathcal{SE}}(\mathcal{N})$, $\mathcal{PI}(\mathcal{N})$ and $\mathcal{PSF}(\mathcal{N})$ can be infinite, the corresponding counting problems cannot be claimed to be in class #*P*. To obtain such a stronger result for a particular PN sub-class, as in the case of the Gadara nets considered in this work, we must also be able to show that the binary relation R that is induced by the corresponding counting problem(s) according to Definition 4, is polynomially balanced and polynomially decidable. In the case of bounded PNs, the polynomial decidability of the binary relation R corresponding to the cardinality assessment of the sets $\widehat{\mathcal{SE}}(\mathcal{N})$ and $\mathcal{PI}(\mathcal{N})$ can be easily established by means of the respective Equations 4 and 8, that define the relation R in each of the two cases. Furthermore, in the case of conservative PNs, the last remark also implies the polynomial decidability of the binary relation R that corresponds to the cardinality assessment of the set $\mathcal{PSF}(\mathcal{N})$ (since, in that case, Proposition 1 implies that membership in $\mathcal{PSF}(\mathcal{N})$ is equivalent to membership in $\mathcal{PI}(\mathcal{N})$). On the other hand, R will be polynomially balanced only in PN classes where the marking vector M of any given net $\mathcal{N}$ from this class admits a concise representation that is polynomially bounded w.r.t. any concise representation of the net $\mathcal{N}$ itself; a typical such case is the class of *safe* PNs, i.e., those PNs for which all the components of the net marking M are guaranteed to be binary.

V. Conclusion

Motivated by an emerging need for pertinent sizing and indexing of various data structures that are used for the efficient storage and processing of the reachability graph of certain bounded PN subclasses, this work has investigated the complexity of assessing the cardinality of various marking sets that, in many cases, have been proposed as reasonable (over-)approximations for the set of reachable markings. Along these lines, our main results establish the #*P*-hardness of the aforementioned estimation for the most prominent of these marking sets. These results also resolve some relevant questions that are raised in [1], and, to the best of our knowledge, they constitute a first attempt to provide formal #*P*-hardness results for counting problems that arise in the PN and the broader DES modeling frameworks.

Future work will seek a more systematic assessment of the implications of the results that were derived in this work for the sizing and indexing problems that motivated this analysis in the first place. It will also consider the development of efficient approximating algorithms and effective practical solutions for the considered counting problems. This last task can be further guided by the study of approximating strategies and any available heuristics for the counting problems that were discussed in Section III and facilitated the reduction that was eventually employed in Section IV; such approximating results and some further leads to the relevant literature can be found, for instance, in [13].

Appendix

A proof for Proposition 4

Since the verification of a vertex cover and of an independent set can be performed in polynomial complexity w.r.t. the size of the corresponding graph, Definition 4 implies that both counting problems of Proposition 4 belong in #*P*. To establish their #*P*-hardness, we shall provide a polynomial parsimonious reduction of the monotone 2-SAT problem to the vertex cover problem; once the #*P*-hardness of the vertex-cover counting problem has been established, the #*P*-hardness of the independent-set counting problem will result immediately from Proposition 3.

Hence, consider a monotone 2-SAT problem $F = C_1 \wedge \ldots \wedge C_m$ that is defined over the set of Boolean variables $\{x_1, \ldots, x_n\}$. We construct the graph $G = (V, E)$ with node set $V = \{x_1, \ldots, x_n\}$ and edge set E defined by the clauses $C_1, \ldots, C_m$ as follows: For $i = 1, \ldots, m$, clause $C_i = y_{i1} \vee y_{i2}$ defines the edge $e_i = \{y_{i1}, y_{i2}\}$. Clearly, the construction of graph G is a task of polynomial complexity w.r.t. the size of the defining elements of the aforementioned 2-SAT problem. Furthermore, the monotonicity of the considered SAT problem implies that a truth assignment for $\{x_1, \ldots, x_n\}$ will be satisfying for F *iff* the subset of the variables that is set to TRUE is a cover for G. Hence, counting the satisfying truth assignments for F boils down to counting the covers of G, and the proof is completed.

Acknowledgment

The author would like to thank the anonymous reviewers for some very pertinent comments that have led to the enhancement of the presentational quality and the integrity of the manuscript.

References

[1] Y. Wang, J. Stanley, and S. Lafortune, "Explicit storage and analysis of billions of states using commodity computers," in *WODES 2012*, 2012, pp. 364–371.

[2] Y. Ru and C. N. Hadjicostis, "Bounds on the number of markings consistent with label observations in Petri nets," *IEEE Trans. on Automation Science and Engineering*, vol. 6, pp. 334–344, 2009.

[3] C. G. Cassandras and S. Lafortune, *Introduction to Discrete Event Systems (2nd ed.)*. NY,NY: Springer, 2008.

[4] A. Nazeem and S. Reveliotis, "A practical approach for maximally permissive liveness-enforcing supervision of complex resource allocation systems," *IEEE Trans. on Automation Science and Engineering*, vol. 8, pp. 766–779, 2011.

[5] M. Silva, E. Teruel, and J. M. Colom, "Linear algebraic and linear programming techniques for the analysis of place/transition net systems," in *Lecture Notes in Computer Science, Vol. 1491*, W. Reisig and G. Rozenberg, Eds. Springer-Verlag, 1998, pp. 309–373.

[6] L. G. Valiant, "The complexity of computing the permanent," *Theoretical Computer Science*, vol. 8, pp. 189–201, 1979.

[7] H. Liao, Y. Wang, H. K. Cho, J. Stanley, T. Kelly, S. Lafortune, S. Mahlke, and S. Reveliotis, "Concurrency bugs in multithreaded software: Modeling and analysis using Petri nets," *Discrete Event Systems: Theory and Applications*, vol. 23, pp. 157–195, 2013.

[8] T. Murata, "Petri nets: Properties, analysis and applications," *Proceedings of the IEEE*, vol. 77, pp. 541–580, 1989.

[9] J. P. Lopez Grao and J. M. Colom, "On the deadlock analysis of multithreaded control software," in *Proceedings of ETFA 2011*. IEEE, 2011.

[10] L. G. Valiant, "The complexity of enumeration and reliability problems," *SIAM Journal on Computing*, vol. 8, pp. 410–421, 1979.

[11] C. H. Papadimitriou, *Computational Complexity*. Reading, MA: Addison Wesley, 1994.

[12] D. B. West, *Introduction to Graph Theory (2nd ed)*. Upper Saddle River, NJ: Prentice Hall, 2001.

[13] L. E. Rasmussen, "Approximately counting cliques," *Random Struct. Alg.*, vol. 11, pp. 395–411, 1997.