

Steady State Analysis of Flexible Nets

Jorge Júlvez*

Stephen G Oliver†

Abstract—The modeling and analysis of complex dynamic systems, such as those in manufacturing, logistics and biology, require powerful analysis methods for their study and optimization. A significant modeling and analysis challenge posed by both, artificial and natural systems, is the existence of uncertain parameters. Flexible Nets is a novel modeling formalism, inspired by Petri nets, that can handle different types of uncertain parameters in a natural way. This paper develops an efficient method to analyse the evolution of a system modeled by a Flexible Net in the long run. More precisely, the method focuses on the computation of steady state bounds for an objective function of interest. The method makes use of a set of constraints, expressed as linear inequalities, that the state variables must satisfy in the steady state. In order to account for systems that do not reach a constant steady state, the developed constraints allow the system state to switch among different values, i.e. the steady state variables are not forced to be constant.

I. INTRODUCTION

Formal models play a critical role in the modeling, analysis, optimization and control of dynamic systems in different application domains [31], [6], [32], [28]. Despite their success, the increasing complexity of artificial systems, and the inherent complexity of natural systems, make the modelling and analysis tasks challenging. A major modeling difficulty lies in the accurate determination of the system parameters. This can be due to the lack of precise measurement devices or methods, to the impossibility of repeating experiments under identical conditions, or to the inability of building or implementing systems with absolute precision. This lack of accuracy leads to uncertain system parameters that need to be properly incorporated in the model of the system for its analysis.

Flexible Nets (FNs) is a novel modeling formalism inspired by Petri nets [24] that is able to accommodate several types of uncertain and unknown parameters. FNs were introduced in [20] to model the evolution of Wilson disease under different treatment regimes and the glucose consumption by a microbial population.

* J. Júlvez is with the Cambridge Systems Biology Centre, University of Cambridge; the Department of Biochemistry, University of Cambridge, Cambridge, UK ; and the Department of Computer Science and Systems Engineering, University of Zaragoza, Zaragoza, Spain. E-mail: julvez@unizar.es; Tel: +34 976762336; Fax: +34 976761914

† Stephen G Oliver is with the Cambridge Systems Biology Centre, University of Cambridge; and the Department of Biochemistry, University of Cambridge, Cambridge, UK. E-mail: sgo24@cam.ac.uk

This work was supported by the European Commission 7th through a Framework Program BIOLEDGE Contract No: 289126 to SGO, and a Marie Curie Intra European Fellowship to JJ (FormalBio Contract No: 623995, Call reference: FP7-PEOPLE-2013-IEF). Further support came from the Biotechnology & Biological Sciences Research Council (UK) grant no. BB/N02348X/1 as part of the iBiotech Program, and by the Industrial Biotechnology Catalyst (Innovate UK, BBSRC, EPSRC) to support the translation, development and commercialisation of innovative Industrial Biotechnology processes.

Similarly to Petri nets, the *transitions* of an FN model processes (or events) of the system, and the *places* model state variables. The value of the state variables associated with the places is known as *marking*. In a dynamic system, the basic relationships between the processes and the marking are: a) the execution of processes produces changes in the marking; b) the marking can modulate the speed of the processes. FNs aim to model these relationships by means of two nets: the *event net* and the *intensity net*. While the *event net* models relationship a), the *intensity net* models relationship b).

The event net is a tripartite graph whose vertices are places, transitions, and *event handlers*. Places are connected to event handlers by directed edges (or simply *arcs*), and transitions are connected to event handlers by undirected edges (or simply *edges*). Each event handler is associated with a set of linear inequalities that captures the potential marking changes that the execution of a transition can produce. Thus, in contrast to Petri nets, the marking change produced by the execution of a transition is allowed to be nondeterministic. The marking of an event net is a vector or real nonnegative markings. Thus, a continuous Petri net [30] can be interpreted as a subclass of event nets in which the marking changes produced by the execution of transitions is deterministic. From a structural point of view, event nets can be seen as a generalization of classical Petri nets.

In a similar way, the intensity net is a tripartite graph whose vertices are places, transitions, and *intensity handlers*. Transitions are connected to intensity handlers by arcs, and places are connected to intensity handlers by edges. Each intensity handler is associated with a set of linear inequalities that captures the potential intensities (or speeds) that the marking can produce on the transitions. This way, the actual speed of the transitions is allowed to be nondeterministic and is constrained to a range of values that depends on the current marking of the net.

An FN is composed of an event net and an intensity net that have the same set of places and the same set of transitions.

Table I reports some popular modeling formalisms for dynamic systems, that are somehow related to FNs, together with commonly desired modeling features. For the sake of brevity, we focus on the way uncertain parameters and nonlinear dynamics are handled by the formalisms.

Although the reported formalisms do not consider the notion of stochasticity in their original definition, many of them include it in well-known extensions. For instance, in the Petri nets [24] arena, the firing of a transition in stochastic Petri nets [3] follows a probability distribution. Similarly, stochastic extensions [37], [38] exist of continuous [30], [2] and hybrid Petri nets [15], [16].

The introduction of stochasticity in hybrid automata [23],

	Discrete states	Continuous states	Stochasticity	Missing information	Nonlinear dynamics	Graphical
Petri nets	✓		*			✓
Continuous Petri nets		✓	*			✓
Hybrid Petri nets	✓	✓	*			✓
Hybrid automata	✓	✓	*		✓	✓
Boolean networks	✓		*			✓
Process algebras	✓	✓	*		✓	
Constraint-based models		✓		✓		✓
Differential equations		✓	*		✓	
Flexible nets	✓	✓		✓	✓	✓

TABLE I: Popular modeling formalisms and some modeling features. The symbol * means that particular classes or extensions of the basic formalism exist that support stochasticity.

[19] results in stochastic hybrid automata [11] which can account for stochastic processes. In a similar way, probabilistic Boolean networks [33], stochastic process algebras [13] and, stochastic differential equations [25] are the stochastic extension of Boolean networks [40], process algebras [29], [12], [14] and, differential equations [34], [35] respectively.

Stochasticity is often used, not only to model inherently stochastic parts of the system, but also to account for uncertain parameters. While stochasticity enriches the modeling power of the formalisms, it must be noted that stochastic models require the introduction of appropriate probability distribution functions together with the parameters associated with them. The acquisition or estimation of these functions can be a challenging task. Moreover, the study of stochastic models usually relies on exhaustive simulations or costly numerical analysis methods that can limit their application to relatively small systems.

In FNs, uncertain parameters are incorporated in the model by means of linear inequalities that determine the region where the value of the parameter is known to lie. This is rather similar to the intervals that determine the firing times in time Petri nets [7] and the instantaneous firing speeds in first-order hybrid Petri nets [4]. In FNs, the parameters whose value can be specified by linear inequalities are: initial marking, default speeds, marking change produced by the firing of transitions and, speeds produced by the marking. Moreover, FNs can also accommodate missing information and parameters that are completely unknown. For instance, a process whose speed is unknown can be modeled by an event handler that is not connected to any transition. This implies that the process can proceed at any speed.

Some models can incorporate nonlinear dynamics by means of rather general nonlinear functions, e.g. process algebra with nonlinear dynamics [41]. This generality usually comes at the price of computational difficulties in the analysis tasks. FNs can also accommodate nonlinear dynamics by means of *guards* associated with the arcs of the intensity net. A guard is defined as a region in the state space, and is associated with a set of linear inequalities. The dynamics of an FN is determined by the sets of linear inequalities associated with active guards (a guard is active when the state of the FN is contained in it). FNs with guards allow the modeler to account for discrete states and can be used, in particular, to model hybrid systems [21].

The computational cost of considering guards is that boolean variables are introduced in the optimization methods, thus leading to mixed-integer linear problems instead of the linear ones associated with unguarded FNs.

In [20] computational methods were developed to study the potential trajectories of FNs over a given time interval. These methods are, therefore, useful for assessing the transient state but not for studying the system in the long run, i.e. in the steady state. In this paper, we develop an efficient computational method to analyse the steady state of FNs (both with and without guards).

In the area of biological systems, constraint-based models [36], [27], [9] are widely used because of the ease with which speed bounds on reaction rates can be imposed and also because steady state fluxes of reactions can be efficiently computed by means of flux balance analysis [26] techniques. A drawback of constraint-based models is that metabolite concentrations are not represented in the model; hence, complex reaction rates that depend on the concentrations cannot be modeled. This shortcoming can be alleviated in part by Petri nets [39] in which transitions can model fluxes and places can model concentrations. Several approaches have been proposed to study the steady state of Petri net subclasses, e.g. free choice nets [10], continuous mono-T-semiflow nets [22], generalized stochastic Petri nets [8]. Most of the existing approaches focus on particular net subclasses and aim at computing steady state performance bounds.

A particular strength of FNs is that they can help bridge the gap between differential equations and constraint-based models in biological systems by accounting for all reaction types, i.e. with known and unknown dynamics. In fact, in the particular case of FNs in which the transition speeds do not depend on the marking, the steady state method proposed here is equivalent to flux balance analysis in constraint-based models.

The computational method proposed in this paper for analyzing the steady state of FNs is based on the development of a set of constraints, expressed as mathematical inequalities, that involve the average state variables of the system. Such inequalities represent conditions that must necessarily hold in the steady state.

Once the set of inequalities is obtained, steady state bounds can be straightforwardly computed by adding an objective

function of interest and solving the resulting programming problem. In addition to the bound given by the objective function, the solution of the programming problem contains the values of the steady state variables that produce that bound. Notice that, if the uncertainties of the FN are used to model the effect of constrained input actions, the result of the problem would contain the value of the actions that optimize the given objective function. The proposed approach is not constrained to a particular net subclass, and any state variable can be included in the objective function, i.e. the objective function is not restricted to performance measures.

Both artificial and natural systems can exhibit complex dynamics over longer timescales. In particular, they do not necessarily reach a steady state in which all their variables remain constant. Instead, some of these systems can oscillate or switch among different states indefinitely. The constraints obtained here can account for such systems with a "non-constant steady state". Interestingly, such constraints can be used to assess the fraction of time that the system spends in each of the states among which it is switching.

The paper is organized as follows: Section II introduces the FNs formalism, Section III develops the set of constraints that must be satisfied in the steady state, Section IV makes use of the developed constraints to compute steady state bounds on several FNs, and Section V concludes the paper.

II. FLEXIBLE NETS

This section introduces Flexible Nets (FNs) which are composed of an event net and an intensity net. While the event net determines how the system processes change the state of the system, the intensity net establishes the speed of such processes. For the sake of brevity, some concepts of FNs are presented through examples. A detailed definition of FNs can be found in [20]. The reader is assumed to be familiar with Petri nets (see [24] for a gentle introduction).

A. Event nets

Definition 1 (Event net): An event net is a tuple $\mathcal{N}_V = (P, T, V, E_V, A, B)$ where (P, T, V, E_V) is a tripartite graph determining the net structure and (A, B) are matrices determining the potential evolutions of the marking.

The vertices of the net are P , T and V , where P is a set of $|P|$ places, T is a set of $|T|$ transitions and V is a set of $|V|$ event handlers.

Similarly to Petri nets, places are depicted as circles and model the types of components in the system. Transitions are depicted as rectangles and model the system processes. Such processes need time to be performed and have the potential to change the marking. Event handlers are depicted as dots and model the different ways in which the transitions can change the marking.

The vertices of the net are connected by the edges in E_V . Each pair of vertices can be connected by at most one edge. The set E_V is partitioned into two sets E_V^P and E_V^T . The edges in E_V^P are directed and are referred as *event arcs*, the edges in E_V^T are undirected and are referred as *event edges*:

- Every $e \in E_V^P$ is either an arc $e = (p_i, v_k)$ from a place p_i to a handler v_k , or an arc $e = (v_k, p_i)$ from a handler v_k to a place p_i .
- Every $e \in E_V^T$ is an edge $e = \{t_j, v_k\}$ connecting a transition t_j and a handler v_k .

Direct connections among places and transitions are not allowed.

Example 1: The event net in Figure 1(a) has three places, $P = \{p_a, p_b, p_c\}$, two transitions, $T = \{t_1, t_2\}$, and two event handlers $V = \{v_1, v_2\}$. The set of arcs is $E_V^P = \{(v_1, p_a), (p_a, v_2), (v_2, p_b), (p_b, v_1)\}$, and the set of edges is $E_V^T = \{\{t_1, v_1\}, \{t_2, v_2\}\}$.

In an event net, each place p_i contains a number of tokens (or marking), $m[p_i]$, and each transition t_j contains a number of actions $\sigma[t_j]$ that represent the potential of the system to carry out the associated process. In contrast to tokens, actions require time to be produced (the production rate of actions is determined by the intensity net, see Section II-B). The actions $\sigma[t_j]$ of a transition t_j can be executed by any event handler v_k connected to t_j . The number of actions of t_j executed by v_k is denoted $a_E[\{t_j, v_k\}]$. The number of actions available in t_j , denoted $a_T[t_j]$, is the number of actions produced minus the number of actions executed. For instance, in the net in Figure 1(a), it holds that $\sigma[t_1] = a_T[t_1] + a_E[\{t_1, v_1\}]$ and $\sigma[t_2] = a_T[t_2] + a_E[\{t_2, v_2\}]$. This can be expressed in matrix form as:

$$\sigma = a_T + Y_\sigma a_E \quad (1)$$

where Y_σ is a matrix with rows indexed by T , columns indexed by E_V^T , and such that $Y_\sigma[t_j, \{t_j, v_k\}] = 1 \forall \{t_j, v_k\} \in E_V^T$ and the rest of the elements in Y_σ are 0,

The execution of actions by an event handler v_k produces marking changes in the places connected to v_k . The number of tokens in p_i produced by v_k is denoted $\Delta m[(v_k, p_i)]$, and the number of tokens in p_i consumed by v_k is denoted $\Delta m[(p_i, v_k)]$. The relation between the number of actions executed by v_k and the marking changes is given by a set of linear inequalities associated with each event handler.

In the net in Figure 1(a), the equation $a = b = v$ associated with v_1 makes use of the labels a , b and c associated with the connected edge and arcs that are shorthands for $\Delta m[(v_1, p_a)]$, $\Delta m[(p_b, v_1)]$ and $a_E[\{t_1, v_1\}]$ respectively. Such an equation implies that the execution of one action in t_1 by v_1 produces one token in p_a and consumes one token in p_b . The sets of inequalities associated with all the event handlers can be expressed in matrix form as:

$$A \Delta m \leq B a_E \quad (2)$$

In the net in Figure 1(a), inequality (2) becomes: $\Delta m[(v_1, p_a)] = \Delta m[(p_b, v_1)] = a_E[\{t_1, v_1\}]$ and $\Delta m[(v_2, p_b)] = \Delta m[(p_a, v_2)] = a_E[\{t_2, v_2\}]$.

The marking of places is the result of accounting for all marking changes produced by the event handlers, e.g. $m[p_a] = m_0[p_a] + \Delta m[(v_1, p_a)] - \Delta m[(p_a, v_2)]$, $m[p_b] = m_0[p_b] + \Delta m[(v_2, p_b)] - \Delta m[(p_b, v_1)]$, $m[p_c] = m_0[p_c]$ where $m_0[p_i]$ is the initial marking of p_i . This can be expressed in matrix form as:

$$m = m_0 + Z_m \Delta m \quad (3)$$

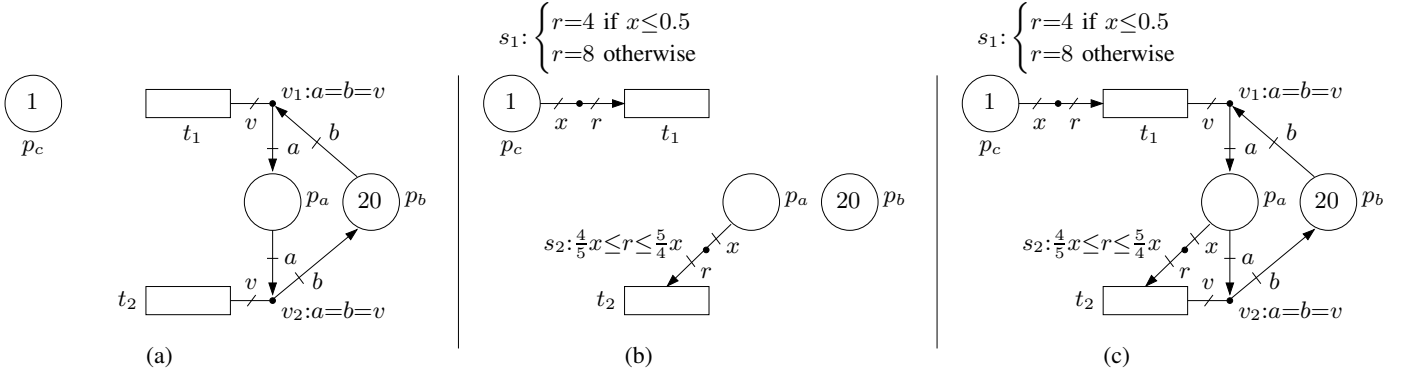


Fig. 1: (a) Event net. (b) Intensity net. (c) Flexible net resulting of combining the event and the intensity net.

where Z_m is a matrix with rows indexed by P , columns indexed by E_V^P , and such that $Z_m[p_i, (p_i, v_k)] = -1 \forall (p_i, v_k) \in E_V^P$, $Z_m[p_i, (v_k, p_i)] = 1 \forall (v_k, p_i) \in E_V^P$ and the rest of the elements in Z_m are 0.

Definition 2 (Event net state): The state of an event net \mathcal{N}_V is given by the tuple $(\sigma, a_T, a_E, \Delta m, m)$ where $\sigma[t_j]$ is the number of actions produced in t_j , $a_T[t_j]$ is the number of actions available in t_j , $a_E[\{t_j, v_k\}]$ is the number of actions of t_j executed by v_k , $\Delta m[(p_i, v_k)]$ is the number of tokens in p_i consumed by v_k , $\Delta m[(v_k, p_i)]$ is the number of tokens in p_i produced by v_k , and $m[p_i]$ is the number of tokens in p_i .

Notice that actions need time to be produced; hence, at the initial state, it holds that $\sigma = 0$, $a_T = 0$ and $a_E = 0$. Equations (1), (2) and (3) can be combined to express necessary reachability conditions for the state of an event net.

Proposition 1 (Event net state equations): Let \mathcal{N}_V be an event net with initial marking m_0 and actions in the transitions σ , every state $(\sigma, a_T, a_E, \Delta m, m)$ reachable from $(\sigma, \sigma, 0, 0, m_0)$ belongs to $SE_{\mathcal{N}_V}(\sigma, m_0)$ where:

$$\begin{aligned} SE_{\mathcal{N}_V}(\sigma, m_0) &= \{(\sigma, a_T, a_E, \Delta m, m) \\ &\quad \sigma = a_T + Y_\sigma a_E \\ &\quad A\Delta m \leq B a_E \\ &\quad m = m_0 + Z_m \Delta m\} \end{aligned} \quad (4)$$

where all variables are nonnegative real numbers.

B. Intensity nets

Definition 3 (Intensity net): An intensity net is a tuple $\mathcal{N}_S = (P, T, S, E_S, C, D, \mathcal{P}, \varphi)$ where (P, T, S, E_S) is a tripartite graph determining the net structure, (C, D) are matrices determining the potential intensity changes produced by the marking, \mathcal{P} is a set of partitions, and φ is a function that associates regions with intensity arcs.

The vertices of the net are P , T and S , where P is a set of $|P|$ places, T is a set of $|T|$ transitions and S is a set of $|S|$ intensity handlers. Places and transitions model the same system features as in the event net. The intensity handlers are depicted as dots and model the different ways in which the tokens can generate intensities, or speeds, in the transitions.

The vertices of the net are connected by the edges in E_S . Each pair of vertices can be connected by at most one edge. The set E_S is partitioned into two sets E_S^T and E_S^P . The edges

in E_S^T are directed and are referred as *intensity arcs*, the edges in E_S^P are undirected and are referred as *intensity edges*:

- Every $e \in E_S^T$ is either an arc $e = (t_j, s_l)$ from a transition t_j to a handler s_l , or an arc $e = (s_l, t_j)$ from a handler s_l to a transition t_j .
- Every $e \in E_S^P$ is an edge $e = \{p_i, s_l\}$ connecting a place p_i and a handler s_l .

As in the event net, connections among places and transitions are not allowed.

Example 2: The intensity net in Figure 1(b) has three places, $P = \{p_a, p_b, p_c\}$, two transitions, $T = \{t_1, t_2\}$, and two intensity handlers $S = \{s_1, s_2\}$. The set of arcs is $E_S^T = \{(s_1, t_1), (s_2, t_2)\}$, and the set of edges is $E_S^P = \{\{p_c, s_1\}, \{p_a, s_2\}\}$.

The tokens in places can be used by the intensity handlers to produce intensities. A token is *active* if it is being used by an intensity handler, otherwise it is *idle*. While idle tokens are associated with places, active tokens are associated with edges. The number of idle tokens in a place p_i is denoted $\mu_P[p_i]$, and the number of active tokens in an edge $\{p_i, s_l\}$ is denoted $\mu_E[\{p_i, s_l\}]$. The number of tokens of a place is equal to its number of idle tokens plus the number of active tokens in its connected edges. In the net in Figure 1(b), it holds that $m[p_a] = \mu_P[p_a] + \mu_E[\{p_a, s_2\}]$, $m[p_b] = \mu_P[p_b]$ and $m[p_c] = \mu_P[p_c] + \mu_E[\{p_c, s_1\}]$. This can be expressed in matrix form as:

$$m = \mu_P + Y_m \mu_E \quad (5)$$

where Y_m is a matrix with rows indexed by P , columns indexed by E_S^P , and such that $Y_m[p_i, \{p_i, s_l\}] = 1 \forall \{p_i, s_l\} \in E_S^P$ and the rest of the elements in Y_m are 0.

An intensity handler determines how much intensity is produced in its arcs as a function of the number of active tokens in its edges. The intensity produced in an arc (s_l, t_j) is denoted $\Delta \lambda[(s_l, t_j)]$. Several sets of linear inequalities can be associated with an intensity arc to define the relation between the number of active tokens and the intensity produced in the arc. If there is only one set of linear inequalities, the arc is *unguarded*, otherwise it is *guarded*. For instance, $\{s_1, t_1\}$ in Figure 1(b) is associated with two sets of linear inequalities: a) $r = 4$; and b) $r = 8$ (r and x are shorthands for the intensity in (s_1, t_1) and the number of active tokens in $\{p_c, s_1\}$). The arc $\{s_2, t_2\}$ is associated with only one set of linear inequalities: $0.8x \leq r, r \leq 1.25x$.

Each set of linear inequalities is associated with a *region*, \mathcal{R}_r , of the state space. It is assumed that the regions are defined on the state space given by the variables m , μ_P and μ_E . More precisely, each region \mathcal{R}_r is a convex polytope of the form:

$$\mathcal{R}_r = \{v \mid S_r v \leq Q_r\} \quad (6)$$

where v is a vector containing m , μ_P and μ_E , i.e. $v = (m, \mu_P, \mu_E)$. The set of regions is denoted \mathcal{R} .

The state space can be partitioned in different ways to facilitate the modeling, the set of partitions is denoted \mathcal{P} . In Figure 1(b), there is only one partition, i.e. $\mathcal{P} = \{\mathcal{P}_1\}$, that consists of two regions which will be denoted \mathcal{R}_{low} and \mathcal{R}_{up} , i.e. $\mathcal{P}_1 = \{\mathcal{R}_{low}, \mathcal{R}_{up}\}$, where \mathcal{R}_{low} is defined as $x \leq 0.5$, and \mathcal{R}_{up} is defined as $x > 0.5$ (recall that x is shorthand for $\mu_E[\{p_c, s_1\}]$).

Regions, and in turn their associated set of linear inequalities, are assigned to intensity arcs through the function $\varphi: E_S^T \rightarrow 2^{\mathcal{R}}$, e.g. $\varphi(\{s_1, t_1\}) = \{\mathcal{R}_{low}, \mathcal{R}_{up}\}$. Each region $\mathcal{R}_r \in \varphi(e)$ is a *guard* of e that is denoted g_r . All the regions in $\varphi(e)$ are assumed to be disjoint. The guard g_r is active when the state is in \mathcal{R}_r . In the following, unguarded arcs will be considered as guarded arcs with a single associated region \mathcal{R}_{true} that is equal to the whole state space. See [20] for more details on the definition of regions and partitions.

The intensity defined in an arc (s_l, t_j) by a given set with guard g_r is denoted $\Delta\lambda_U[(s_l, t_j), g_r]$. For instance, in Figure 1(b), the values of $\Delta\lambda_U$ are $\Delta\lambda_U[(s_1, t_1), g_{low}] = 4$, $\Delta\lambda_U[(s_1, t_1), g_{up}] = 8$, $0.8\mu_E[\{p_a, s_2\}] \leq \Delta\lambda_U[(s_2, t_2), g_{true}] \leq 1.25\mu_E[\{p_a, s_2\}]$, which will be expressed in matrix form as:

$$C\Delta\lambda_U \leq D\mu_E \quad (7)$$

The value of $\Delta\lambda[(s_l, t_j)]$ is equal to $\Delta\lambda_U[(s_l, t_j), g_r]$ where g_r is the guard that is active, e.g. $\Delta\lambda[(s_1, t_1)] = \delta_{low}\Delta\lambda_U[(s_1, t_1), g_{low}] + \delta_{up}\Delta\lambda_U[(s_1, t_1), g_{up}]$, $\Delta\lambda[(s_2, t_2)] = \Delta\lambda_U[(s_2, t_2), g_{true}]$ where $\delta_{low}, \delta_{up} \in \{0, 1\}$, $\delta_{low} = 1$ iff $\mu_E[\{p_c, s_1\}] \leq 0.5$ and $\delta_{low} + \delta_{up} = 1$. In matrix form, $\Delta\lambda$ is expressed as:

$$\Delta\lambda = \delta\Delta\lambda_U \quad (8)$$

where $\delta[e, (e, g_r)] = \delta_r$ (the pair (e, g_r) is the index of the column associated with the guard g_r of e) for every $e \in E_S^T$ and every guard g_r of e , and the rest of the elements of δ are 0. Each δ_r is a binary variable in $\{0, 1\}$ such that $\delta_r = 1$ iff g_r is active.

Similarly to the initial marking of places, an initial, or default intensity, $\lambda_0[t_j]$, can be assigned to each transition t_j . The actual intensity, $\lambda[t_j]$, of t_j is obtained by adding the intensity of its input arcs and subtracting the intensity of its output arcs, e.g. $\lambda[t_1] = \lambda_0[t_1] + \Delta\lambda[(s_1, t_1)]$ and $\lambda[t_2] = \lambda_0[t_2] + \Delta\lambda[(s_2, t_2)]$ (in this example there are no output intensity arcs and $\lambda_0[t_1] = \lambda_0[t_2] = 0$). Thus, the intensity, i.e. the rate at which actions are produced, in t_1 is equal to 4 per time unit if the number of active tokens in $\{p_c, s_1\}$ is lower or equal to 0.5, and 8 otherwise. The intensity in t_2 is any value in the interval $[0.8\mu_E[\{p_a, s_2\}], 1.25\mu_E[\{p_a, s_2\}]]$.

The vector λ can be expressed in matrix form as:

$$\lambda = \lambda_0 + Z_\lambda\Delta\lambda \quad (9)$$

where Z_λ is a matrix with rows indexed by T , columns indexed by E_S^T , and such that $Z_\lambda[t_j, (t_j, s_l)] = -1 \forall (t_j, s_l) \in E_S^T$, $Z_\lambda[t_j, (s_l, t_j)] = 1 \forall (s_l, t_j) \in E_S^T$ and the rest of the elements in Z_λ are 0,

Definition 4 (Intensity net state): The state of an intensity net \mathcal{N}_S is given by the tuple $(m, \mu_P, \mu_E, \Delta\lambda_U, \Delta\lambda, \lambda)$, where $m[p_i]$ is the number of tokens in p_i , $\mu_P[p_i]$ is the number of idle tokens in p_i , $\mu_E[\{p_i, s_l\}]$ is the number of active tokens of p_i being used by s_l , $\Delta\lambda_U[(s_l, t_j), g_r]$ is the intensity defined in (s_l, t_j) by guard g_r , $\Delta\lambda[(t_j, s_l)]$ is a decrease of intensity in t_j produced by s_l , $\Delta\lambda[(s_l, t_j)]$ is an increase of intensity in t_j produced by s_l , and $\lambda[t_j]$ is the intensity in t_j .

Similarly to (4), Equations (5), (7), (8), (9) can be combined to express necessary reachability conditions for the state of an intensity net with a given marking m and default intensities λ_0 .

Proposition 2 (Intensity net state equations): Let the state of an intensity net \mathcal{N}_S be $(m, m, 0, 0, 0, \lambda_0)$, i.e. m idle tokens are available and no intensity handler is working. Every state $(m, \mu_P, \mu_E, \Delta\lambda_U, \Delta\lambda, \lambda)$ reachable from $(m, m, 0, 0, 0, \lambda_0)$ belongs to $SE_{\mathcal{N}_S}(m, \lambda_0)$ where:

$$\begin{aligned} SE_{\mathcal{N}_S}(m, \lambda_0) = \{ & (m, \mu_P, \mu_E, \Delta\lambda_U, \Delta\lambda, \lambda) \\ & m = \mu_P + Y_m\mu_E \\ & C\Delta\lambda_U \leq D\mu_E \\ & \Delta\lambda = \delta\Delta\lambda_U \\ & \lambda = \lambda_0 + Z_\lambda\Delta\lambda \} \end{aligned} \quad (10)$$

where δ is a matrix of binary variables, $\Delta\lambda_U$ is a vector of real variables, and the rest of variables are nonnegative real numbers.

C. Flexible nets

A Flexible Net (FN) consists of an event net and an intensity net that have the same set of places and the same set of transitions. For instance, the FN in Figure 1(c) is composed of the event net in Figure 1(a) and the intensity net in Figure 1(b).

Definition 5 (Flexible net): A Flexible Net (FN) is a tuple $\mathcal{N} = (P, T, V, E_V, A, B, S, E_S, C, D, \mathcal{P}, \varphi)$ where (P, T, V, E_V, A, B) is an event net and $(P, T, S, E_S, C, D, \mathcal{P}, \varphi)$ is an intensity net.

While the event net of an FN determines the way actions produce marking changes, the intensity net determines the way tokens produce intensity changes.

In order to express the number of actions produced in the transitions, let us first define the number of actions produced in the intensity arcs. Let $\Delta\sigma(\tau)$ denote the number of actions produced in the intensity arcs until time τ ($\Delta\sigma[e](\tau)$ with $e \in E_S^T$ denotes the number of actions produced in intensity arc e). The value of $\Delta\sigma(\tau)$ is defined as the integral of $\Delta\lambda$ over time:

$$\Delta\sigma(\tau) = \int_0^\tau \Delta\lambda(s) ds \quad (11)$$

The overall number of actions, $\sigma[t_j](\tau)$, produced in a transition t_j until time τ can be computed by making use of Z_λ , see (9):

$$\sigma(\tau) = \lambda_0\tau + Z_\lambda\Delta\sigma(\tau) \quad (12)$$

The state of an FN is composed of the state variables of the event net, the state variables of the intensity net and $\Delta\sigma$.

Definition 6 (Flexible net state): The state \mathbf{x} of an FN is given by the tuple $\mathbf{x} = (m, \mu_P, \mu_E, \Delta\lambda_U, \Delta\lambda, \lambda, \Delta\sigma, \sigma, a_T, a_E, \Delta m)$.

All the state variables are time dependent. For the sake of clarity, the time dependency will be omitted when it is clear from the context, e.g. $m(\tau)$ is shortened to m . At time 0 it holds that $\Delta\sigma = 0$, $\sigma = 0$, $a_T = 0$, $a_E = 0$, $\Delta m = 0$, i.e. the initial state can be written as: $(m, \mu_P, \mu_E, \Delta\lambda_U, \Delta\lambda, \lambda, 0, 0, 0, 0, 0)$.

In order to account for uncertain initial markings, m_0 is assumed to be constrained by:

$$J_m m_0 \leq K_m \quad (13)$$

where J_m and K_m are real matrices of appropriate size. Uncertain default intensities can be accounted for similarly, i.e. λ_0 is assumed to be constrained by:

$$J_\lambda \lambda_0 \leq K_\lambda \quad (14)$$

where J_λ and K_λ are real matrices of appropriate size.

By making use of (4), (10), (11), (12), (13) and (14), it is possible to write a set of equations that any potential state at time τ must satisfy.

Proposition 3 (Flexible net state equations): Let \mathcal{N} be an FN with initial marking m_0 satisfying $J_m m_0 \leq K_m$, and default intensities λ_0 satisfying $J_\lambda \lambda_0 \leq K_\lambda$. Every state $(m, \mu_P, \mu_E, \Delta\lambda_U, \Delta\lambda, \lambda, \Delta\sigma, \sigma, a_T, a_E, \Delta m)$ reachable at time τ belongs to $SE_{\mathcal{N}}(\tau, J_m, K_m, J_\lambda, K_\lambda)$ where:

$$\begin{aligned} SE_{\mathcal{N}}(\tau, J_m, K_m, J_\lambda, K_\lambda) = & \\ \{ & (m, \mu_P, \mu_E, \Delta\lambda_U, \Delta\lambda, \lambda, \Delta\sigma, \sigma, a_T, a_E, \Delta m) \mid \\ & m = \mu_P + Y_m \mu_E; C \Delta\lambda_U \leq D \mu_E; \Delta\lambda = \delta \Delta\lambda_U \\ & \lambda = \lambda_0 + Z_\lambda \Delta\lambda; J_\lambda \lambda_0 \leq K_\lambda \\ & \Delta\sigma = \int_0^\tau \Delta\lambda(s) ds; \sigma = \lambda_0 \tau + Z_\lambda \Delta\sigma \\ & \sigma = a_T + Y_\sigma a_E; A \Delta m \leq B a_E \\ & m = m_0 + Z_m \Delta m; J_m m_0 \leq K_m \} \end{aligned} \quad (15)$$

where δ is a matrix of binary variables, $\Delta\lambda_U$ is a vector of real variables, and the rest of variables are nonnegative real numbers.

III. STEADY STATE BOUNDS

This section develops a set of constraints that the state variables must satisfy in the steady state, i.e. the constraints are necessary reachability conditions. Such a set can be combined with an objective function in order to compute steady state bounds. First, the relevant steady state variables are defined, then state equations for such variables are obtained. These equations are the basis for the constraints developed.

A. State variables

When analyzing the steady state, some quantities (e.g. the number of produced actions σ) are meaningless as they might tend to infinity as time, τ , tends to infinity. However, the rate at which these quantities change, e.g. the average number of

actions produced per time unit, is meaningful and can be used to determine conditions that the net must fulfill in the steady state.

Thus, in addition to the already defined state variables, the following ones are defined for every time $\tau > 0$ to account for averages per time unit:

$$\Delta\sigma_\tau = \frac{\Delta\sigma}{\tau}; \sigma_\tau = \frac{\sigma}{\tau}; a_{T\tau} = \frac{a_T}{\tau}; a_{E\tau} = \frac{a_E}{\tau}; \Delta m_\tau = \frac{\Delta m}{\tau} \quad (16)$$

Let \mathbf{z} be a vector defined as the following concatenation of variables: $\mathbf{z} = (m, \mu_P, \mu_E, \Delta\lambda_U, \Delta\lambda, \lambda, \Delta\sigma_\tau, \sigma_\tau, a_{T\tau}, a_{E\tau}, \Delta m_\tau)$. It will be assumed that none of these variables can tend to infinity.

B. Average state

The average state $\bar{\mathbf{z}}$ as time tends to infinity is defined as:

$$\bar{\mathbf{z}} = \lim_{\theta \rightarrow \infty} \frac{1}{\theta} \int_0^\theta \mathbf{z}(\tau) d\tau \quad (17)$$

We say that the net reaches a steady state if the limit defining $\bar{\mathbf{z}}$ exists. In the following, it is assumed that such a limit exists.

Recall that $\delta_r(\tau)$ is a binary variable associated with region \mathcal{R}_r that indicates whether the state is in \mathcal{R}_r ¹:

$$\delta_r(\tau) = \begin{cases} 1 & \text{if } \mathbf{x}(\tau) \in \mathcal{R}_r \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

Thus, the time ratio spent by the net in \mathcal{R}_r is:

$$\bar{\delta}_r = \lim_{\theta \rightarrow \infty} \frac{1}{\theta} \int_0^\theta \delta_r(\tau) d\tau \quad (19)$$

Given that each \mathcal{P}_n is a partition of the state space, the state is always in one, and only one, region of \mathcal{P}_n . Hence, the sum of the ratios of the regions in a partition must be equal to 1:

$$\sum_{\mathcal{R}_r \in \mathcal{P}_n} \bar{\delta}_r = 1 \quad \forall \mathcal{P}_n \in \mathcal{P} \quad (20)$$

The average state in region \mathcal{R}_r , $\bar{\mathbf{z}}_r$, is defined as:

$$\bar{\mathbf{z}}_r = \lim_{\theta \rightarrow \infty} \frac{1}{\int_0^\theta \delta_r(\tau) d\tau} \int_0^\theta \delta_r(\tau) \mathbf{z}(\tau) d\tau \quad (21)$$

Notice that, if the system never visits a region \mathcal{R}_r , then $\delta_r(\tau) = 0$ for every $\tau \geq 0$ and $\bar{\mathbf{z}}_r$ is indeterminate. If \mathcal{R}_r is visited, then by (21) it holds that $\bar{\mathbf{z}}_r \in \mathcal{R}_r$, i.e. $\bar{\mathbf{z}}_r$ satisfies:

$$S_r \bar{\mathbf{z}}_r \leq Q_r \quad (22)$$

where (22) expresses the same inequalities as (6).

Inequality (22) can be slightly modified as follows to take into account regions that are not visited or are not reachable:

$$S_r \bar{\mathbf{z}}_r \leq Q_r + W(1 - \alpha_r) \quad (23)$$

where W is a vector such that $W \geq S_r \mathbf{x} - Q_r$ for every reachable state \mathbf{x} (see Appendix B for methods to compute W), and α_r is a binary variable $\alpha_r \in \{0, 1\}$ that indicates if

¹In (18) and (22), \mathbf{x} and $\bar{\mathbf{z}}_r$ are used instead of v in (6) in order to reduce the number of variables and, hence, the columns of S_r that correspond to variables other than m , μ_P and μ_E are set to 0.

the region \mathcal{R}_r is visited in the steady state. More precisely, variable α_r is defined as:

$$\alpha_r = 0 \leftrightarrow \bar{\delta}_r = 0 \quad (24)$$

Expression (24) is difficult to translate and handle as a linear expression in an exact way. Thus, it will be relaxed to $\alpha_r = 0 \leftrightarrow \bar{\delta}_r \leq \epsilon$ where $\epsilon > 0$ is an arbitrarily small quantity, e.g. the computer's precision. This expression is equivalent to:

$$\begin{aligned} \epsilon - \bar{\delta}_r &\leq 2(1 - \alpha_r) \\ \epsilon - \bar{\delta}_r &\geq -2\alpha_r \end{aligned} \quad (25)$$

By making use of (17) and (21), the average state \bar{z} can be expressed in terms of \bar{z}_r as:

$$\begin{aligned} \bar{z} &= \lim_{\theta \rightarrow \infty} \frac{1}{\theta} \int_0^\theta z(\tau) d\tau = \lim_{\theta \rightarrow \infty} \frac{1}{\theta} \int_0^\theta \sum_{\mathcal{R}_r \in \mathcal{P}_n} \delta_r(\tau) z(\tau) d\tau \\ &= \sum_{\mathcal{R}_r \in \mathcal{P}_n} \lim_{\theta \rightarrow \infty} \frac{1}{\theta} \int_0^\theta \delta_r(\tau) z(\tau) d\tau = \sum_{\mathcal{R}_r \in \mathcal{P}_n} \bar{\delta}_r \bar{z}_r \quad \forall \mathcal{P}_n \in \mathcal{P} \end{aligned} \quad (26)$$

C. Average state equations

According to (15), the variables in z at time $\tau > 0$ satisfy:

$$\begin{aligned} m &= \mu_P + Y_m \mu_E; \quad C \Delta \lambda_U \leq D \mu_E; \quad \Delta \lambda = \delta \Delta \lambda_U \\ \lambda &= \lambda_0 + Z_\lambda \Delta \lambda; \quad J_\lambda \lambda_0 \leq K_\lambda \\ \Delta \sigma_\tau &= \frac{1}{\tau} \int_0^\tau \Delta \lambda(s) ds; \quad \sigma_\tau = \lambda_0 + Z_\lambda \Delta \sigma_\tau \\ \sigma_\tau &= a_{T\tau} + Y_\sigma a_{E\tau}; \quad A \Delta m_\tau \leq B a_{E\tau} \\ m &= m_0 + Z_m \Delta m_\tau; \quad J_m m_0 \leq K_m \end{aligned} \quad (27)$$

In (27), every equation, except $\Delta \lambda = \delta \Delta \lambda_U$, $\Delta \sigma = \int_0^\tau \Delta \lambda(s) ds$ and $m = m_0 + Z_m \Delta m_\tau$, is linear. Let us integrate $\Delta \lambda = \delta \Delta \lambda_U$ over time to compute the average intensities $\Delta \bar{\lambda}$ in arcs.

By (17) and $\Delta \lambda = \delta \Delta \lambda_U$:

$$\Delta \bar{\lambda} = \lim_{\theta \rightarrow \infty} \frac{1}{\theta} \int_0^\theta \delta(\tau) \Delta \lambda_U(\tau) d\tau \quad (28)$$

Let us consider each component $\Delta \bar{\lambda}[e]$ of $\Delta \bar{\lambda}$ separately. Then, as in (26), by the definition of δ and the fact that the regions in $\varphi(e)$ are disjoint, $\Delta \bar{\lambda}[e]$ in (28) can be expressed as:

$$\begin{aligned} \Delta \bar{\lambda}[e] &= \lim_{\theta \rightarrow \infty} \frac{1}{\theta} \int_0^\theta \sum_{\mathcal{R}_r \in \varphi(e)} \delta_r(\tau) \Delta \lambda_U[(e, g_r)](\tau) d\tau \\ &= \sum_{\mathcal{R}_r \in \varphi(e)} \lim_{\theta \rightarrow \infty} \frac{1}{\theta} \int_0^\theta \delta_r(\tau) \Delta \lambda_U[(e, g_r)](\tau) d\tau \\ &= \sum_{\mathcal{R}_r \in \varphi(e)} \bar{\delta}_r \Delta \bar{\lambda}_{U_r}[(e, g_r)] \end{aligned} \quad (29)$$

where $\bar{\lambda}_{U_r}[(e, g_r)]$ denotes the average intensity in e when the system is in region \mathcal{R}_r , and the pair (e, g_r) indexes the element associated with the guard g_r of e . This can be expressed in matrix form as:

$$\Delta \bar{\lambda} = \bar{\delta} \Delta \bar{\lambda}_G \quad (30)$$

where $\bar{\delta}$ has the same size as δ , and it is defined as $\bar{\delta}[e, (e, g_r)] = \bar{\delta}_r$ for every $e \in E_S^T$ and every $\mathcal{R}_r \in \varphi(e)$,

and the rest of elements of $\bar{\delta}$ are 0; and $\Delta \bar{\lambda}_G$ is indexed by (e, g_r) and it is defined as:

$$\Delta \bar{\lambda}_G[(e, g_r)] = \Delta \bar{\lambda}_{U_r}[(e, g_r)] \quad \forall e \in E_S^T, \quad \forall \mathcal{R}_r \in \varphi(e) \quad (31)$$

If region \mathcal{R}_r is not reachable (which would imply $\alpha_r = 0$), then $\Delta \bar{\lambda}_{U_r}[(e, g_r)]$ can be negative. In order to avoid assigning negative values to $\Delta \bar{\lambda}_G[(e, g_r)]$, (31) can be replaced by:

$$\Delta \bar{\lambda}_G[(e, g_r)] = \alpha_r \Delta \bar{\lambda}_{U_r}[(e, g_r)] \quad \forall e \in E_S^T, \quad \forall \mathcal{R}_r \in \varphi(e) \quad (32)$$

Average state equations can be obtained by computing the integral and limit of the equations in (27) as in (17) what results in:

$$\begin{aligned} \bar{m} &= \bar{\mu}_P + Y_m \bar{\mu}_E; \quad C \Delta \bar{\lambda}_U \leq D \bar{\mu}_E; \quad \Delta \bar{\lambda} = \bar{\delta} \Delta \bar{\lambda}_G \\ \bar{\lambda} &= \lambda_0 + Z_\lambda \Delta \bar{\lambda}; \quad J_\lambda \lambda_0 \leq K_\lambda \\ \Delta \bar{\sigma}_\tau &= \lim_{\theta \rightarrow \infty} \frac{1}{\theta} \int_0^\theta \frac{1}{\tau} \int_0^\tau \Delta \lambda(s) ds d\tau = \Delta \bar{\lambda}; \quad \bar{\sigma}_\tau = \bar{\lambda} \\ \bar{\sigma}_\tau &= \bar{a}_{T\tau} + Y_\sigma \bar{a}_{E\tau}; \quad A \Delta \bar{m}_\tau \leq B \bar{a}_{E\tau} \\ \bar{m} &= m_0 + \lim_{\theta \rightarrow \infty} \frac{1}{\theta} \int_0^\theta Z_m \Delta m_\tau(\tau) d\tau; \quad J_m m_0 \leq K_m \end{aligned} \quad (33)$$

Average state equations in regions can be obtained by computing the integral and limit of (27) as in (21):

$$\begin{aligned} \bar{m}_r &= \bar{\mu}_{P_r} + Y_m \bar{\mu}_{E_r}; \quad C \Delta \bar{\lambda}_{U_r} \leq D \bar{\mu}_{E_r}; \quad \bar{\lambda}_r = \lambda_0 + Z_\lambda \Delta \bar{\lambda}_r \\ \bar{\sigma}_{r\tau} &= \bar{a}_{T_r\tau} + Y_\sigma \bar{a}_{E_r\tau}; \quad A \Delta \bar{m}_{r\tau} \leq B \bar{a}_{E_r\tau}; \\ \bar{m}_r &= m_0 + \lim_{\theta \rightarrow \infty} \frac{1}{\theta} \int_0^\theta \delta_r(\tau) Z_m \Delta m_\tau(\tau) d\tau \end{aligned} \quad (34)$$

Notice that if $\mathcal{R}_r \in \varphi(e)$ then $\Delta \bar{\lambda}_r[e] = \Delta \bar{\lambda}_{U_r}[(e, g_r)]$. That is, the value of $\Delta \bar{\lambda}_r$ can be easily related for those arcs that have \mathcal{R}_r as guard:

$$\Delta \bar{\lambda}_r[e] = \Delta \bar{\lambda}_{U_r}[(e, g_r)] \quad \forall e \in E_S^T, \quad \forall \mathcal{R}_r \in \varphi(e) \quad (35)$$

As in (32), in order to avoid assigning a negative value to $\Delta \bar{\lambda}_r[e]$, (35) can be replaced by:

$$\Delta \bar{\lambda}_r[e] = \alpha_r \Delta \bar{\lambda}_{U_r}[(e, g_r)] \quad \forall e \in E_S^T, \quad \forall \mathcal{R}_r \in \varphi(e) \quad (36)$$

Equations (33) and (34) do not provide straightforward expressions for the average markings \bar{m} and \bar{m}_r . However, a condition that \bar{m} must necessarily satisfy can be obtained from (15). Equation (15) states that for every reachable marking there exists $\sigma, a_T, a_E, \Delta m \geq 0$ such that $\sigma = a_T + Y_\sigma a_E$, $A \Delta m \leq B a_E$, $m = m_0 + Z_m \Delta m$ and $J_m m_0 \leq K_m$.

Since \bar{m} and \bar{m}_r are computed as the average of a number of reachable markings, there must exist $\sigma, a_T, a_E, \Delta m, m_0 \geq 0$ and $\sigma_r, a_{T_r}, a_{E_r}, \Delta m_r \geq 0$ for each region \mathcal{R}_r such that:

$$\begin{aligned} \sigma &= a_T + Y_\sigma a_E; \quad A \Delta m \leq B a_E; \quad \bar{m} = m_0 + Z_m \Delta m \\ \sigma_r &= a_{T_r} + Y_\sigma a_{E_r}; \quad A \Delta m_r \leq B a_{E_r}; \quad \bar{m}_r = m_0 + Z_m \Delta m_r \end{aligned} \quad (37)$$

where m_0 satisfies $J_m m_0 \leq K_m$, and variables $\sigma, a_T, a_E, \Delta m$ and $\sigma_r, a_{T_r}, a_{E_r}, \Delta m_r$ are just used to express a necessary condition for the reachability of average markings. That is, the value of these variables is not informative about the transient dynamics, i.e. about the evolution of the net from the initial marking to the average marking in the steady state, e.g. a_E

in (37) must not be interpreted as the number of executed actions from the initial state to the average steady state.

D. Reachability conditions

In the steady state, the rate of tokens produced in each place p_i and the rate of tokens consumed from p_i must necessarily be the same. This yields the following necessary condition:

$$Z_m \Delta \bar{m}_\tau = 0 \quad (39)$$

Notice that if a given region \mathcal{R}_r is contained in a region \mathcal{R}_s , i.e. $\mathcal{R}_r \subseteq \mathcal{R}_s$, then it necessarily holds that $\bar{\delta}_r \leq \bar{\delta}_s$. This idea can be extended to sets of regions. Let \mathcal{V} and \mathcal{W} be the spaces defined as the union of a set of disjoint regions, e.g. $\mathcal{V} = \dots \cup \mathcal{R}_r \cup \dots$, and $\mathcal{W} = \dots \cup \mathcal{R}_s \cup \dots$. Then, the following conditions must hold in the steady state:

$$\sum_{\mathcal{R}_r \in \mathcal{V}} \bar{\delta}_r \leq \sum_{\mathcal{R}_s \in \mathcal{W}} \bar{\delta}_s \quad \text{if } \mathcal{V} \subseteq \mathcal{W} \quad (40)$$

$$\sum_{\mathcal{R}_r \in \mathcal{V}'} \bar{\delta}_r \bar{z}_r = \sum_{\mathcal{R}_s \in \mathcal{W}'} \bar{\delta}_s \bar{z}_s \quad \text{if } \mathcal{V}' = \mathcal{W}' \quad (41)$$

In order to enhance the modelling possibilities, the average intensities in arcs can be related by means of matrices E_c , F_c , E and F such that the following conditions must hold in the steady state:

$$E_c \Delta \bar{\lambda} = F_c \quad (42)$$

$$E \Delta \bar{\lambda} \leq F \quad (43)$$

Recall that event handlers are not forced to fire. However, in some cases, it is useful to consider only the states that have executed all the actions of some transitions. Let $T_{Av} \subseteq T$ be the set of transitions whose actions must have been executed, i.e. the number of available actions of $t_j \in T_{Av}$ must be 0. Then, the following constraints must hold:

$$\bar{a}_{T\tau}[t_j] = 0 \quad \forall t_j \in T_{Av} \quad (44)$$

Similarly, it is possible to consider only the states in which all the tokens of some places are active. Let $P_{Av} \subseteq P$ be the set of places whose tokens must be active, i.e. the number of idle tokens of $p_i \in P_{Av}$ must be 0. This can be modeled by:

$$\bar{\mu}_P[p_i] = 0 \quad \forall p_i \in P_{Av} \quad (45)$$

The obtained conditions can be used to define a set of constraints such that every potential steady state satisfies them. Namely, these necessary conditions are: a) equations that relate the overall state variables: (33), (39) and (37); b) equations that hold for each region: (23), (34), (38), (25), (32), (36), and the equations used to define $\bar{\delta}$; c) equations that hold for each partition: (20), (26); d) equations that hold for sets of regions: (40), (41); e) equations enforced by the modeller: (42), (43), (44), (45).

Proposition 4: Let \mathcal{N} be an FN with initial marking m_0 satisfying $J_m m_0 \leq K_m$, and default intensities λ_0 satisfying $J_\lambda \lambda_0 \leq K_\lambda$. In the steady state, every average state $\bar{z} = (\bar{m}, \bar{\mu}_P, \bar{\mu}_E, \Delta \bar{\lambda}_U, \Delta \bar{\lambda}, \bar{\lambda}, \Delta \bar{\sigma}_\tau, \bar{\sigma}_\tau, \bar{a}_{T\tau}, \bar{a}_{E\tau}, \Delta \bar{m}_\tau)$ belongs to $SC_{\mathcal{N}}(J_m, K_m, J_\lambda, K_\lambda)$ where:

$$\begin{aligned} SC_{\mathcal{N}}(J_m, K_m, J_\lambda, K_\lambda) = & \{ \bar{z} = (\bar{m}, \bar{\mu}_P, \bar{\mu}_E, \Delta \bar{\lambda}_U, \Delta \bar{\lambda}, \bar{\lambda}, \Delta \bar{\sigma}_\tau, \bar{\sigma}_\tau, \bar{a}_{T\tau}, \bar{a}_{E\tau}, \Delta \bar{m}_\tau) \mid \\ & \bar{m} = \bar{\mu}_P + Y_m \bar{\mu}_E; \quad C \Delta \bar{\lambda}_U \leq D \bar{\mu}_E; \quad \Delta \bar{\lambda} = \bar{\delta} \Delta \bar{\lambda}_G \\ & \bar{\lambda} = \lambda_0 + Z_\lambda \Delta \bar{\lambda}; \quad J_\lambda \lambda_0 \leq K_\lambda \\ & \Delta \bar{\sigma}_\tau = \Delta \bar{\lambda}; \quad \bar{\sigma}_\tau = \bar{\lambda}; \quad \bar{\sigma}_\tau = \bar{a}_{T\tau} + Y_\sigma \bar{a}_{E\tau} \\ & A \Delta \bar{m}_\tau \leq B \bar{a}_{E\tau}; \quad J_m m_0 \leq K_m \\ & Z_m \Delta \bar{m}_\tau = 0 \\ & \sigma = a_T + Y_\sigma a_E; \quad A \Delta m \leq B a_E; \quad \bar{m} = m_0 + Z_m \Delta m \\ & \text{for every } \mathcal{R}_r \in \mathcal{R} \\ & S_r \bar{z}_r \leq Q_r + W(1 - \alpha_r) \\ & \bar{m}_r = \bar{\mu}_{Pr} + Y_m \bar{\mu}_{Er}; \quad C \Delta \bar{\lambda}_{Ur} \leq D \bar{\mu}_{Er} \\ & \bar{\lambda}_r = \lambda_0 + Z_\lambda \Delta \bar{\lambda}_r \\ & \bar{\sigma}_{\tau r} = \bar{a}_{T\tau r} + Y_\sigma \bar{a}_{E\tau r}; \quad A \Delta \bar{m}_{\tau r} \leq B \bar{a}_{E\tau r} \\ & \sigma_r = a_{T_r} + Y_\sigma a_{E_r}; \quad A \Delta m_r \leq B a_{E_r} \\ & \bar{m}_r = m_0 + Z_m \Delta m_r \\ & \epsilon - \bar{\delta}_r \leq 2(1 - \alpha_r); \quad \epsilon - \bar{\delta}_r \geq -2\alpha_r \\ & \text{for every } e \in E_S^T \text{ such that } \mathcal{R}_r \in \varphi(e) \\ & \Delta \bar{\lambda}_G[(e, g_r)] = \alpha_r \Delta \bar{\lambda}_{Ur}[(e, g_r)] \\ & \Delta \bar{\lambda}_r[e] = \alpha_r \Delta \bar{\lambda}_{Ur}[(e, g_r)] \\ & \bar{\delta}[e, (e, g_r)] = \bar{\delta}_r \\ & \text{for every } e' \in E_S^T \text{ such that } e' \neq e \\ & \bar{\delta}[e', (e, g_r)] = 0 \\ & \text{for every } \mathcal{P}_n \in \mathcal{P} \\ & \sum_{\mathcal{R}_r \in \mathcal{P}_n} \bar{\delta}_r = 1 \\ & \bar{z} = \sum_{\mathcal{R}_r \in \mathcal{P}_n} \bar{\delta}_r \bar{z}_r \\ & \text{for every } \mathcal{V}, \mathcal{W} \text{ such that } \mathcal{V} \subseteq \mathcal{W} \\ & \sum_{\mathcal{R}_r \in \mathcal{V}} \bar{\delta}_r \leq \sum_{\mathcal{R}_s \in \mathcal{W}} \bar{\delta}_s \\ & \text{for every } \mathcal{V}', \mathcal{W}' \text{ such that } \mathcal{V}' = \mathcal{W}' \\ & \sum_{\mathcal{R}_r \in \mathcal{V}'} \bar{\delta}_r = \sum_{\mathcal{R}_s \in \mathcal{W}'} \bar{\delta}_s \\ & \sum_{\mathcal{R}_r \in \mathcal{V}'} \bar{\delta}_r \bar{z}_r = \sum_{\mathcal{R}_s \in \mathcal{W}'} \bar{\delta}_s \bar{z}_s \\ & E_c \Delta \bar{\lambda} = F_c; \quad E \Delta \bar{\lambda} \leq F \\ & \bar{a}_{T\tau}[t_j] = 0 \quad \forall t_j \in T_{Av} \\ & \bar{\mu}_P[p_i] = 0 \quad \forall p_i \in P_{Av} \end{aligned} \quad (46)$$

where every variable, except $\Delta \bar{\lambda}_U$ and $\Delta \bar{\lambda}_{Ur}$, is nonnegative, and every $\alpha_r \in \{0, 1\}$.

The constraints (46) allow the system to visit different regions during steady state. In other words, the system can switch over the regions while keeping the overall average

marking constant. These constraints can be slightly modified to forbid the switch between regions, hence focusing on steady states that keep the same constant marking over time. If the state of the system is forced to be only in one region during the steady state, then $\bar{\delta}_r = \alpha_r$ for every $\mathcal{R}_r \in \mathcal{R}$, and (46) can be reformulated as:

$$\begin{aligned}
& SC_{\mathcal{N}}(J_m, K_m, J_\lambda, K_\lambda) = \\
& \{ \bar{\mathbf{z}} = (\bar{m}, \bar{\mu}_P, \bar{\mu}_E, \Delta\bar{\lambda}_U, \Delta\bar{\lambda}, \bar{\lambda}, \Delta\bar{\sigma}_\tau, \bar{\sigma}_\tau, \bar{a}_{T\tau}, \bar{a}_{E\tau}, \Delta\bar{m}_\tau) | \\
& \quad \bar{m} = \bar{\mu}_P + Y_m \bar{\mu}_E; C\Delta\bar{\lambda}_U \leq D\bar{\mu}_E; \Delta\bar{\lambda} = \alpha\Delta\bar{\lambda}_G \\
& \quad \bar{\lambda} = \lambda_0 + Z_\lambda \Delta\bar{\lambda}; J_\lambda \lambda_0 \leq K_\lambda \\
& \quad \Delta\bar{\sigma}_\tau = \Delta\bar{\lambda}; \bar{\sigma}_\tau = \bar{\lambda}; \bar{\sigma}_\tau = \bar{a}_{T\tau} + Y_\sigma \bar{a}_{E\tau} \\
& \quad A\Delta\bar{m}_\tau \leq B\bar{a}_{E\tau}; J_m m_0 \leq K_m \\
& \quad Z_m \Delta\bar{m}_\tau = 0 \\
& \quad \sigma = a_T + Y_\sigma a_E; A\Delta m \leq B a_E; \bar{m} = m_0 + Z_m \Delta m \\
& \text{for every } \mathcal{R}_r \in \mathcal{R}
\end{aligned}$$

$$\begin{aligned}
& S_r \bar{z}_r \leq Q_r + W(1 - \alpha_r) \\
& \bar{m}_r = \bar{\mu}_{Pr} + Y_m \bar{\mu}_{Er}; C\Delta\bar{\lambda}_{Ur} \leq D\bar{\mu}_{Er} \\
& \bar{\lambda}_r = \lambda_0 + Z_\lambda \Delta\bar{\lambda}_r \\
& \bar{\sigma}_{rr} = \bar{a}_{Trr} + Y_\sigma \bar{a}_{Err}; A\Delta\bar{m}_{rr} \leq B\bar{a}_{Err} \\
& \sigma_r = a_{Tr} + Y_\sigma a_{Er}; A\Delta m_r \leq B a_{Er} \\
& \bar{m}_r = m_0 + Z_m \Delta m_r
\end{aligned}$$

$$\begin{aligned}
& \text{for every } e \in E_S^T \text{ such that } \mathcal{R}_r \in \varphi(e) \\
& \quad \Delta\bar{\lambda}_G[(e, g_r)] = \alpha_r \Delta\bar{\lambda}_{Ur}[(e, g_r)] \\
& \quad \Delta\bar{\lambda}_r[e] = \alpha_r \Delta\bar{\lambda}_{Ur}[(e, g_r)] \\
& \quad \alpha[e, (e, g_r)] = \alpha_r \\
& \quad \text{for every } e' \in E_S^T \text{ such that } e' \neq e \\
& \quad \quad \alpha[e', (e, g_r)] = 0
\end{aligned}$$

for every $\mathcal{P}_n \in \mathcal{P}$

$$\begin{aligned}
& \sum_{\mathcal{R}_r \in \mathcal{P}_n} \alpha_r = 1 \\
& \bar{\mathbf{z}} = \sum_{\mathcal{R}_r \in \mathcal{P}_n} \alpha_r \bar{\mathbf{z}}_r
\end{aligned}$$

for every \mathcal{V}, \mathcal{W} such that $\mathcal{V} \subseteq \mathcal{W}$

$$\sum_{\mathcal{R}_r \in \mathcal{V}} \alpha_r \leq \sum_{\mathcal{R}_s \in \mathcal{W}} \alpha_s$$

for every $\mathcal{V}', \mathcal{W}'$ such that $\mathcal{V}' = \mathcal{W}'$

$$\begin{aligned}
& \sum_{\mathcal{R}_r \in \mathcal{V}'} \alpha_r = \sum_{\mathcal{R}_s \in \mathcal{W}'} \alpha_s \\
& \sum_{\mathcal{R}_r \in \mathcal{V}'} \alpha_r \bar{\mathbf{z}}_r = \sum_{\mathcal{R}_s \in \mathcal{W}'} \alpha_s \bar{\mathbf{z}}_s
\end{aligned}$$

$$E_c \Delta\bar{\lambda} = F_c; E\Delta\bar{\lambda} \leq F$$

$$\bar{a}_{T\tau}[t_j] = 0 \quad \forall t_j \in T_{Av}$$

$$\bar{\mu}_P[p_i] = 0 \quad \forall p_i \in P_{Av}$$

(47)

where $\alpha[e, (e, g_r)] = \alpha_r$ for every $e \in E_S^T$ and every guard g_r of e , and the rest of the elements of α are 0.

unguarded, the above conditions reduce to:

$$\begin{aligned}
& SC_{\mathcal{N}}(J_m, K_m, J_\lambda, K_\lambda) = \\
& \{ \bar{\mathbf{z}} = (\bar{m}, \bar{\mu}_P, \bar{\mu}_E, \Delta\bar{\lambda}, \bar{\lambda}, \Delta\bar{\sigma}_\tau, \bar{\sigma}_\tau, \bar{a}_{T\tau}, \bar{a}_{E\tau}, \Delta\bar{m}_\tau) | \\
& \quad \bar{m} = \bar{\mu}_P + Y_m \bar{\mu}_E; C\Delta\bar{\lambda} \leq D\bar{\mu}_E \\
& \quad \bar{\lambda} = \lambda_0 + Z_\lambda \Delta\bar{\lambda}; J_\lambda \lambda_0 \leq K_\lambda \\
& \quad \Delta\bar{\sigma}_\tau = \Delta\bar{\lambda}; \bar{\sigma}_\tau = \bar{\lambda}; \bar{\sigma}_\tau = \bar{a}_{T\tau} + Y_\sigma \bar{a}_{E\tau} \\
& \quad A\Delta\bar{m}_\tau \leq B\bar{a}_{E\tau}; J_m m_0 \leq K_m \\
& \quad Z_m \Delta\bar{m}_\tau = 0 \\
& \quad \sigma = a_T + Y_\sigma a_E; A\Delta m \leq B a_E; \bar{m} = m_0 + Z_m \Delta m \\
& \quad E_c \Delta\bar{\lambda} = F_c; E\Delta\bar{\lambda} \leq F \\
& \quad \bar{a}_{T\tau}[t_j] = 0 \quad \forall t_j \in T_{Av} \\
& \quad \bar{\mu}_P[p_i] = 0 \quad \forall p_i \in P_{Av} \}
\end{aligned} \tag{48}$$

where C is indexed by the set of intensity arcs, and $C\Delta\bar{\lambda} \leq D\bar{\mu}_E$ captures how active tokens produce intensity in the intensity arcs.

IV. STEADY STATE ANALYSIS

This section demonstrates the usefulness of the bounds derived in Section III by analyzing the potential steady states of three FNs with different features.

A. Assessing constant steady state

Let us consider the FN in Figure 1(c) with $T_{Av} = \{t_1, t_2\}$ and $P_{Av} = \{p_a\}$, i.e. the actions of both transitions are forced to be executed at the same rate at which they are produced, and the tokens in p_a are forced to be always active. The initial marking of p_a and the default intensities of the transitions are 0 and are not written in the figure. Thus, the steady state of the system depends on: a) the number of active tokens in $\{p_c, s_1\}$ and b) the intensity that s_2 produces in (s_2, t_2) by using the tokens in $\{p_a, s_2\}$. Both a) and b) can be interpreted either as uncertain parameters, i.e. any steady state that satisfies the system constraints is feasible, or as control actions that can be applied on the system to achieve a given goal, e.g. the number of active tokens in $\{p_c, s_1\}$ is interpreted as a control action that determines the intensity in t_1 . In any case, the constraints obtained in Section III-D can be used to bound all the potential steady states arising from a) and b). These constraints will be combined with an objective function (this results in a programming problem) in order to compute steady state bounds of interest.

Recall that, in Section III-D, two sets of constraints for guarded nets were developed. While the set (46) allows the steady state to switch among the different regions, i.e. the average state is the result of the average state in the regions that are visited, the set (47) does not allow the system to switch among regions. The net in Figure 1(c) will be used to assess both (46) and (47). The different objective functions considered and some of the steady state values are summarized in Table II. The columns below ‘‘Steady state’’ correspond to (46), and the columns below ‘‘Constant steady state’’ correspond to (47).

The objective function in the first row in Table II is to maximize the average marking of p_a , $\bar{m}[p_a]$. Notice that in the

If the net is not guarded, i.e. all the intensity arcs are

Objective function and constraints	Steady state				Constant steady state			
	$\bar{m}[p_a]$	$\lambda[t_1]$	$\lambda[t_2]$	δ_{low}	$\bar{m}[p_a]$	$\lambda[t_1]$	$\lambda[t_2]$	α_{low}
$max \bar{m}[p_a]$	10.1	8.08	8.08	0.01	10.0	8.0	8.0	0
$min \bar{m}[p_a]$	3.136	3.92	3.92	0.99	3.2	4.0	4.0	1
$min (\bar{m}[p_a] - 6.0)^2$	6.0	5.56	5.56	0.59	6.4	8.0	8.0	0
$max \delta_{low}$ subject to $\bar{m}[p_a]=6.0$	6.0	4.8	4.8	0.83	–	–	–	–
$min \delta_{low}$ subject to $\bar{m}[p_a]=6.0$	6.0	7.48	7.48	0.1	–	–	–	–

TABLE II: Steady state values of the FN in 1(c) for different objective functions.

steady state it necessarily holds that $\bar{\lambda}[t_1] = \bar{\lambda}[t_2]$ (otherwise the markings would either increase or decrease indefinitely). Then, given that $\bar{\lambda}[t_2]$ depends on $\bar{m}[p_a]$, the maximum value of $\bar{m}[p_a]$ is achieved when $\bar{\lambda}[t_2]$ takes its maximum value (i.e. 8.0) which is obtained when the system is in region \mathcal{R}_{up} (the number of active tokens in $\{p_c, s_1\}$ is higher than 0.5).

At first glance, it is surprising that the value of $\bar{\lambda}[t_1]$ under “Steady state” is 8.08. This is due to the linear approximation of $\bar{\delta}\Delta\bar{\lambda}_G$ and $\bar{\delta}_r\bar{z}_r$ by the constraints in Appendix A (the values of d_j in (50) were set to $j/100$, this implies that a value of $\bar{\delta}_{low}$ equal to 0.01 allows the solver to assign 8.08 to $\bar{\lambda}[t_1]$). This numerical approximation error does not happen under “Constant steady state” as only one region is allowed to be visited ($\alpha_{low} = 0$ implies $\alpha_{up} = \bar{\delta}_{up} = 1$) and no linear approximation of a multiplication is required. Notice that the maximum $\bar{m}[p_a]$ is obtained when s_2 produces the minimum possible intensity in (s_2, t_2) per active token in $\{p_a, s_2\}$.

The objective function in the second row in Table II is to minimize the average marking of p_a . This is achieved by keeping the system in region \mathcal{R}_{low} and by producing the maximum possible intensity in (s_2, t_2) . As in the previous case, numerical errors happen under “Steady state” but not under “Constant steady state”.

From a control perspective, the maximum(minimum) average marking in p_a is obtained by allocating more(less) than 0.5 tokens in $\{p_c, s_1\}$ and producing the minimum(maximum) possible intensity in (s_2, t_2) per active token in $\{p_a, s_2\}$.

The quadratic objective function in the third row in Table II implies that an average marking of p_a as close as possible to 6 is desired. Notice that, according to the inequalities associated with s_2 , if $m[p_a] = 6$ then $\lambda[t_2] \in [4.8, 7.5]$. Given that $\lambda[t_1] = 4$ when the system is in \mathcal{R}_{low} and $\lambda[t_1] = 8$ when it is in \mathcal{R}_{up} , a steady state that visits just one region and satisfies $\bar{m}[p_a] = 6.0$ is not possible (remember that, in the steady state, $\bar{\lambda}[t_1] = \bar{\lambda}[t_2]$ must hold). In fact, if only one region is allowed to be visited in the steady state, then the closest value of $\bar{m}[p_a]$ to 6 is 6.4 (see columns below “Constant steady state”). This marking can be reached when the system is exclusively in region \mathcal{R}_{up} , i.e. $\alpha_{low} = 0$.

If the system is allowed to switch between regions in the steady state, then a state with $\bar{m}[p_a] = 6.0$ can be achieved (see columns below “Steady state”). The value $\bar{\delta}_{low} = 0.59$ is the time ratio spent by the system in region \mathcal{R}_{low} during the steady state. Notice, however, that given the inequalities associated with s_2 , the average marking $\bar{m}[p_a] = 6.0$ can be obtained with other time ratios.

The objective functions in the last two rows in Table II are used to compute the interval of time ratios, $\bar{\delta}_{low}$, that allow

a steady state with $\bar{m}[p_a] = 6.0$. This interval is computed by maximizing and minimizing $\bar{\delta}_{low}$ subject to the constraint $\bar{m}[p_a] = 6.0$. The solutions of the resulting programming problems state that $\bar{\delta}_{low}$ must be in the interval $[0.1, 0.83]$ so that $\bar{m}[p_a] = 6.0$ is achieved. Notice that the variable $\bar{\delta}_{low}$ does not exist in the set (47), and hence, the proposed objective functions in the last two rows do not make sense under “Constant steady state”.

B. Handling uncertainties

The FN in Figure 2(a) consists of 5 places, 3 transitions, 5 event handlers, and 2 intensity handlers. For clarity, the inequalities of handlers that make the values of all their arcs and edges equal are omitted, e.g. the equations $a = b = v$ of v_2 is omitted. Thus, the only equations written in the Figure are the ones associated with s_1 (see below s_1). The initial markings and default intensities that are 0 are also omitted. The initial marking of p_1 is uncertain but constrained to the interval $[1, 2]$. Similarly, the default intensity of t_2 is uncertain but constrained to $[1, 2]$. All the tokens in places p_2, p_4 and p_5 are assumed to be active, i.e. $P_{Av} = \{p_2, p_4, p_5\}$, and all the actions in the three transitions are forced to be executed instantaneously, i.e. $T_{Av} = \{t_1, t_2, t_3\}$.

Notice that v_1 is not connected to any transition, this means that v_1 can fire as soon as there are tokens in p_1 and p_2 . This is, however, not equivalent to an immediate transition since v_1 is not forced to fire as soon as there are tokens.

Intensity handler s_1 models the inhibition of t_1 by p_4 . More precisely, if the marking of p_4 is less than or equal to 1.0, then the rate at which actions are produced in t_1 (and in turn, the rate at which tokens from p_2 are consumed) is equal to the marking in p_2 . If the marking of p_4 is greater than 1.0, then no actions are produced in t_1 , and hence no token is consumed from p_2 .

Transition t_2 models a nondeterministic choice, i.e. the actions produced in t_2 can be used either to consume tokens from p_3 and produce in p_1 and p_5 , or to consume tokens from p_4 and produce in p_5 .

The constraints in (47) will be used to account for all the possible steady states that arise from the existing uncertainties in the FN (recall that (47) focuses on steady states that do not switch between regions). The goal is to compute upper bounds for $\bar{\lambda}[t_1]$, i.e. the objective function is $max \bar{\lambda}[t_1]$.

Figure 2(b) shows the obtained upper bounds of $\bar{\lambda}[t_1]$ for different initial markings of p_2 . Several trends can be appreciated in the Figure. Notice that, in the steady state, the rate at which tokens are produced in p_5 (event handlers v_3 and v_4) must be equal to the rate at which tokens are consumed

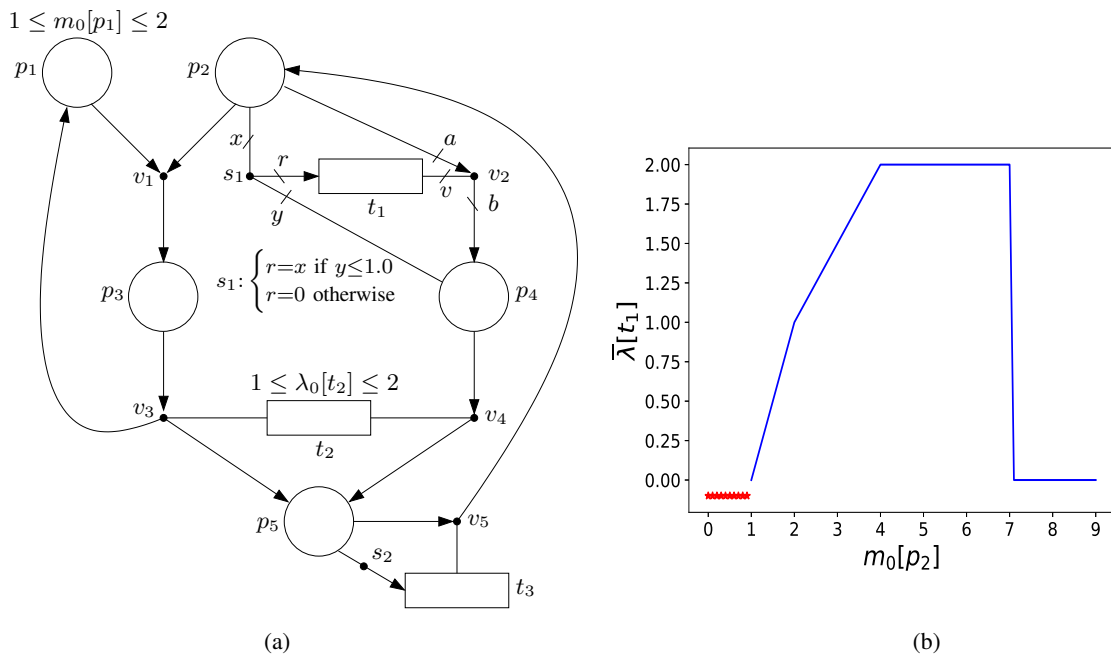


Fig. 2: (a) FN with uncertain initial marking, uncertain default intensity and choice in the process modeled by t_2 . (b) Steady state bounds of $\bar{\lambda}[t_1]$ for different values of $m_0[p_2]$.

from p_5 (event handler v_5). Thus, in the steady state, it must hold that $\bar{\lambda}[t_2] = \bar{\lambda}[t_3]$.

Given that $\bar{\lambda}[t_3] = \bar{m}[p_5]$ (see s_2) and $\bar{m}[p_5]$ is at most equal to $m_0[p_2]$ (see net structure), the value of $\bar{\lambda}[t_3]$ is upper bounded by $m_0[p_2]$. Since $\lambda_0[t_2]$ is required to be at least 1, the set of constraints (47) cannot be satisfied for low values of $m_0[p_2]$. More precisely, if $m_0[p_2]$ is in the interval $[0, 1)$, the constraints (47) cannot be satisfied and the programming problem becomes infeasible (infeasibility is depicted as red stars in Figure 2(b)).

If $m_0[p_2] = 1$, the initial token of p_2 can be moved to p_5 in order to produce one intensity unit in t_3 , this results in a feasible steady state at which $\bar{\lambda}[t_1] = 0$ and $\bar{\lambda}[t_2] = \bar{\lambda}[t_3] = 1$. In general, for $m_0[p_2] \in [1, 2]$, one token of p_2 can be moved to p_5 , and $m_0[p_2] - 1$ can be left in p_2 to produce intensity in t_1 . This leads to the linear increase of $\bar{\lambda}[t_1]$ in Figure 2(b) in the interval $m_0[p_2] \in [1, 2]$.

If $m_0[p_2] = 2$, the steady state reached satisfies $\bar{\lambda}[t_1] = \bar{\lambda}[t_2] = \bar{\lambda}[t_3] = \bar{m}[p_2] = \bar{m}[p_5] = 1$ and all the actions produced in t_2 are used by v_4 . If $m_0[p_2]$ is increased further, its tokens must be used to increase both the intensity of t_1 and t_3 . This results in the linear increase with slope 0.5 in Figure 2(b) in the interval $m_0[p_2] \in [2, 4]$.

If $m_0[p_2] = 4$ then in the steady state $\bar{\lambda}[t_1] = \bar{\lambda}[t_2] = \bar{\lambda}[t_3] = \bar{m}[p_2] = \bar{m}[p_5] = 2$ holds, i.e. the intensity of t_2 is at its maximum. Hence, no steady state is possible with $\bar{\lambda}[t_1] > 2$. If $m_0[p_2]$ is increased further, 2 tokens can be moved to p_5 , 2 tokens can be left in p_2 and $m_0[p_2] - 4$ can be moved either to p_3 or p_4 . Thus, in the interval $m_0[p_2] \in [4, 7]$, the upper bound of $\bar{\lambda}[t_1]$ keeps constant at 2.

If $m_0[p_2] = 7$ then the steady state reached satisfies $\bar{m}[p_1] = 0$, $\bar{m}[p_2] = 2$, $\bar{m}[p_3] = 2$, $\bar{m}[p_4] = 1$ and $\bar{m}[p_5] = 2$. Notice, that at this point the average marking of p_3 is at its

maximum (recall that $1 \leq m_0[p_1] \leq 2$), thus if $m_0[p_2]$ is increased further, more than one token needs to be moved to p_4 leading to the inhibition of t_1 by s_1 . This results in $\bar{\lambda}[t_1] = 0$ for $m_0[p_2] > 7$.

C. Shared resources

The FN in Figure 3 models a system in which shared resources can be used to activate the system processes. More precisely, the tokens in X can be used to produce intensity in t_1 or t_2 , and the tokens in Y can be used to produce intensity in t_2 or to synchronize with tokens in C to produce intensity in t_4 . Thus, while the intensity in t_2 is the result of adding the intensities in $\{s_2, t_2\}$ and $\{s_3, t_2\}$, the intensity in t_4 is the minimum of the active tokens in $\{Y, s_5\}$ and $\{C, s_5\}$ (such a minimum can be easily expressed by means of two regions).

If the FN is interpreted as a biochemical network, X and Y would model enzymes that catalyze reactions: enzyme X can catalyze the reactions modeled by v_1 (i.e. $E \rightarrow A$) and v_2 (i.e. $E \rightarrow C$); enzyme Y can catalyze the reactions modeled by v_2 and v_4 (i.e. $C \rightarrow D$), being the rate of this last reaction equal to the minimum concentration of Y and C . Transition t_5 together with v_5 and s_6 would model a reaction $B + 2D \rightarrow 3E$ with a rate equal to $\max(\min(b, d) - 4, 0)$ where b and d are the concentrations of B and D involved in the reaction, i.e. both b and d must be at least 4 for the reaction to happen, and the rate is given by the minimum of the concentrations minus 4.

Notice that v_5 imposes a synchronization between the branch composed of A and B , and the branch composed of C and D . In particular, the equations associated with v_5 imply that, in the steady state, the number of tokens consumed from D per time unit has to be twice as much the number of tokens

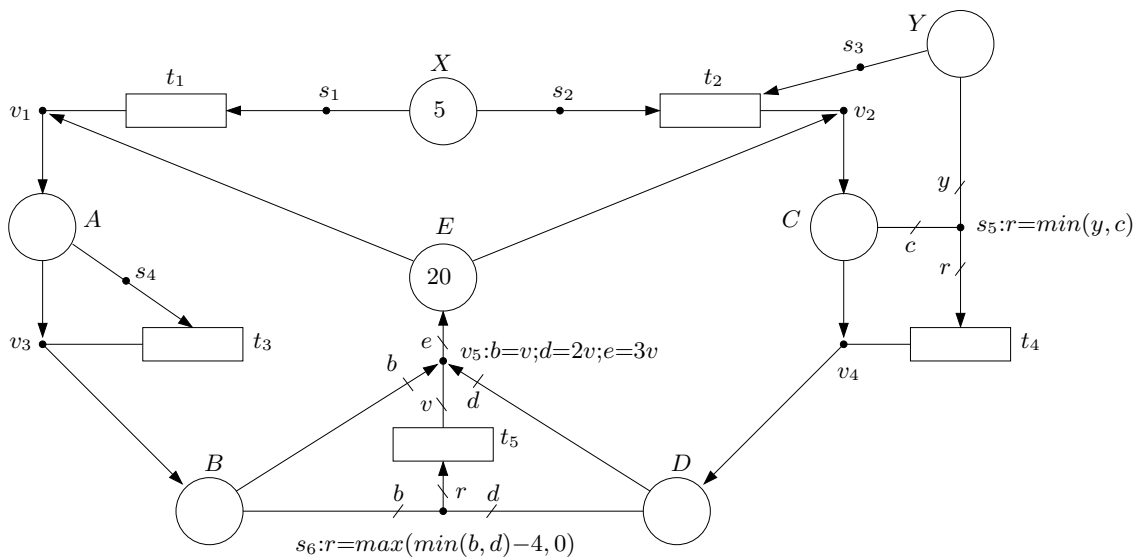


Fig. 3: FN with shared resources X and Y regulating the speeds of transitions t_1 , t_2 and t_4 .

consumed from B per time unit. In other words, the intensity of t_4 in the steady state must be twice the intensity of t_3 .

The actions of all the transitions in Figure 3 are forced to fire, i.e. $T_{Av} = T$, and no token is forced to be active, i.e. $P_{Av} = \{\}$. The focus will be on constant steady states (see (47)) that maximize the objective function $\bar{\lambda}[t_5]$, i.e. the rate or flux of the reaction modeled by t_5 .

Figure 4(a) shows the value of the objective function, $\bar{\lambda}[t_5]$, for initial markings of Y in the range $[0, 10]$. Three different *modes* of operation can be distinguished in the figure that depend on $m_0[Y]$:

- $m_0[Y] \in [0, 10/3]$: The intensities in t_1 and t_2 are produced exclusively by the tokens in X (see Figure 4(b) and (c) showing the number of active places in intensity edges). Given the synchronization forced by v_5 , twice as many active tokens are used in $\{X, s_2\}$ than in $\{X, s_1\}$. All the tokens in Y are used to produce intensity in t_4 .
- $m_0[Y] \in (10/3, 7]$: All the tokens in X are active. The ratio of active tokens between $\{X, s_2\}$ and $\{X, s_1\}$ is now less than 2 because Y is also using some of its tokens to produce intensity in t_2 .
- $m_0[Y] \in (7, 10]$: If $m_0[Y] = 7$, all the tokens in X , Y , and E are used to produce intensities, more precisely, in the steady state it holds that $\bar{m}[A] = 2.4$, $\bar{m}[B] = 6.4$, $\bar{m}[C] = 4.8$, $\bar{m}[D] = 6.4$, $\bar{m}[E] = 0.0$. Thus, if $m_0[Y] > 7$, the tokens in $m_0[E]$ will be limiting the intensities (namely in t_3 , t_4 and t_5) and, hence, the objective function cannot be increased further.

All the methods to analyse the steady state of FNs have been implemented in the software tool `fnyzer` (<https://bitbucket.org/Julvez/fnyzer.git>) which makes use of Pyomo [18], Gurobi [17] and CPLEX [1] to build and solve the associated programming problems. The CPU time to solve the most CPU-demanding problem together with the computation of the required auxiliary bounds was 0.86 seconds (Intel i7, 2.00 GHz, 8 GiB, Ubuntu 14.04 LTS).

V. CONCLUSIONS

FNs is a powerful modeling formalism that allows the incorporation of different types of uncertainties in a single model, e.g. uncertainties in the initial markings, initial intensities, marking changes produced by the processes and intensities produced by the marking. Moreover, FNs can handle nonlinear dynamics in a natural way by means of guards associated with the intensity arcs.

This paper has devised and implemented a mathematical method to analyse the steady state behavior of FNs. The method is based on the development of a set of constraints that the FN necessarily satisfies in the steady state. In particular, two different sets of constraints have been developed, one of them assumes that the system can switch among guards during the steady state, and the other assumes that a constant steady state is reached. These constraints, together with an objective function on the state variables, result in a programming problem whose solution represents a bound of the system in the steady state. Under a control perspective, uncertainties can be used to model input constraints, and hence, the solution of the programming problem would contain the values of the input actions that optimize the given objective function.

APPENDIX A

RELAXING NONLINEAR CONSTRAINTS

The equations $\alpha_r \Delta \bar{\lambda}_{U_r}[(e, g_r)]$, $\bar{\delta} \Delta \bar{\lambda}_G$, $\bar{\delta}_r \bar{z}_r$ and $\alpha_r \bar{z}_r$ in (46) and (47), are not linear and hence could be difficult to handle by a solver. Such constraints can be relaxed and approximated by inequalities containing real and binary variables [5]. This Appendix shows how such an approximation can be carried out.

A. Linear approximations

Notice that, while $\alpha_r \Delta \bar{\lambda}_{U_r}[(e, g_r)]$ and $\alpha_r \bar{z}_r$ involve the multiplication of a binary variable and a real variable, $\bar{\delta} \Delta \bar{\lambda}_G$

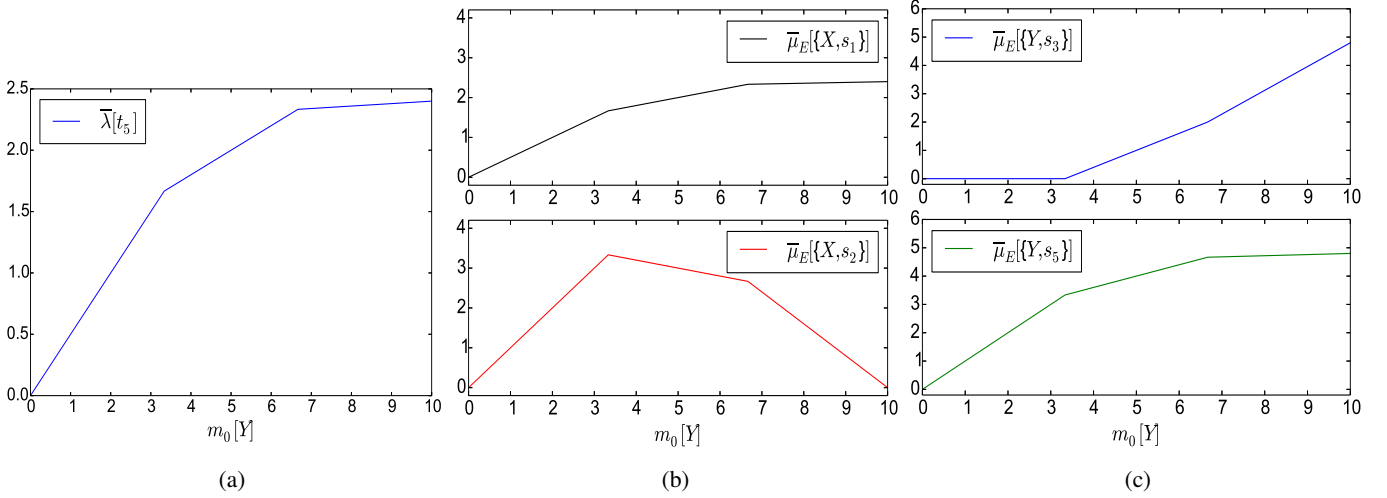


Fig. 4: (a) Steady state bounds of $\bar{\lambda}[t_5]$ in Figure 3. (b), (c) Allocation of resources X and Y .

and $\bar{\delta}_r \bar{z}_r$ involve the multiplication of two real variables, one of them in the interval $[0, 1]$.

The product $y = \alpha w$ where $\alpha \in \{0, 1\}$ and $w \in [w_l, w_u]$ can be linearized as:

$$\begin{aligned} y &\leq \alpha w_u \\ y &\geq \alpha w_l \\ y &\leq w - w_l(1 - \alpha) \\ y &\geq w - w_u(1 - \alpha) \end{aligned} \quad (49)$$

This linearization can be used directly on the multiplications containing binary variables and real variables. The values of w_l and w_u do not need to be tight, and they can be computed by using the methods in Appendix B. As only $\Delta\lambda_U$, $\Delta\bar{\lambda}_U$ and $\Delta\bar{\lambda}_{U_r}$ can get negative values, w_l in (49) can be taken as 0 for the multiplications not involving any of these three variables.

Let us now focus on equations of the form $v = \phi u$ where both ϕ and u are nonnegative real variables and $\phi \in [0, 1]$. Let us partition the interval $[0, 1]$ in q intervals $[0, d_1]$, $[d_1, d_2]$, ..., $[d_{q-1}, 1]$ and let us define $q-1$ binary variables $\beta_j \in \{0, 1\}$ as follows:

$$\beta_j = 1 \leftrightarrow \phi \geq d_j \quad \forall j \in \{1, q-1\} \quad (50)$$

As in (25), such a necessary and sufficient condition can be modeled by including in the set of constraints the following inequalities:

$$d_j - \phi \leq 2(1 - \beta_j) \quad \forall j \in \{1, q-1\} \quad (51)$$

$$d_j - \phi \geq -2\beta_j \quad \forall j \in \{1, q-1\} \quad (52)$$

In order to avoid that $\phi = d_j$ can occur for several values of ϕ and d_j (this can imply a loss of accuracy because β_j can be either 0 or 1, see first objective function in Subsection IV-A) uneven intervals $[d_{j-1}, d_j]$ can be considered.

The product $v = \phi u$, where $\phi \in [0, 1]$ and $u \geq 0$, can be

replaced and bounded by the following inequalities:

$$\begin{aligned} \left(\sum_{j=1}^{q-1} \beta_j (d_j - d_{j-1}) \right) u &\leq v \quad \forall j \in \{1, q-1\} \\ v &\leq \left(1 - \sum_{j=1}^{q-1} (1 - \beta_j) (d_{j+1} - d_j) \right) u \quad \forall j \in \{1, q-1\} \end{aligned} \quad (53)$$

where d_0 and d_q are defined as $d_0 = 0$ and $d_q = 1$.

The products of binary variables with real variables in (53) can be linearized by using the inequalities in (49).

Note that, $\Delta\bar{\lambda}_{U_r}$ can be negative and hence the above approach cannot be used to compute the multiplication $\bar{\delta}_r \bar{z}_r$ corresponding to the components $\Delta\bar{\lambda}_{U_r}$. However, as the systems dynamics is driven by $\Delta\bar{\lambda}_{U_r}$ (and in turn by $\Delta\bar{\lambda}_G$) and not by $\Delta\bar{\lambda}_U = \sum_{\mathcal{R}_r \in \mathcal{P}_n} \bar{\delta}_r \Delta\bar{\lambda}_{U_r}$, such multiplications corresponding to $\Delta\bar{\lambda}_{U_r}$ can be ignored. Similarly, $\Delta\bar{\lambda}_r$ and $\bar{\lambda}_r$ do not play a role in the system dynamics either, and their multiplications in $\bar{\delta}_r \bar{z}_r$ can be ignored.

B. Average variables

The relaxation $\bar{\delta}_r \bar{z}_r$ in (46) can be improved in the following cases: a) a given variable in \bar{z}_r is known to be 0; b) a given region \mathcal{R}_r is not visited and hence $\bar{\delta}_r$ is 0; c) only one region \mathcal{R}_r of a given partition is visited.

Case a): Variables $\bar{a}_{T\tau}$ and $\bar{\mu}_P$ are 0 for those transitions that are required to be executed instantaneously, and those places whose tokens must always be active. Thus, equations

$$\bar{\mu}_{P_r}[p_i] = 0 \quad \forall p_i \in P_{Av}, \forall \mathcal{R}_r \in \mathcal{R}$$

$$\bar{a}_{T\tau_r}[t_j] = 0 \quad \forall t_j \in T_{Av}, \forall \mathcal{R}_r \in \mathcal{R}$$

can be added to (46) and (47).

Case b): If \mathcal{R}_r is not visited then α_r is 0. Hence the constraint:

$$\bar{\delta}_r \bar{z}_r \leq \alpha_r w_u \quad \forall \mathcal{R}_r \in \mathcal{R} \quad (54)$$

can be used to improve the product $\bar{\delta}_r \bar{z}_r$ for every nonnegative \bar{z}_r , i.e. $\Delta \bar{\lambda}_{U_r}$ is excluded from $\bar{\delta}_r \bar{z}_r$, where w_u is an upper bound for \bar{z}_r .

Case c): If only one region \mathcal{R}_r of a given partition is visited, then \bar{z} should be equal to \bar{z}_r , and $\Delta \bar{\lambda}[e]$ should be equal to $\Delta \bar{\lambda}_G[(e, g_r)]$. This can be achieved by defining an auxiliary binary variable γ_r such that $\gamma_r = 1$ iff \mathcal{R}_r is the only region visited of its partition. The value of γ_r can be computed by the following constraints:

$$1 - \epsilon \leq \bar{\delta}_r + 2(1 - \gamma_r) \quad \forall \mathcal{R}_r \in \mathcal{R} \quad (55)$$

$$1 - \epsilon \geq \bar{\delta}_r - 2\gamma_r \quad \forall \mathcal{R}_r \in \mathcal{R} \quad (56)$$

In order to avoid numerical issues, $\bar{\delta}_r$ can be set exactly to 0 for those regions that are not visited with the following constraint:

$$\bar{\delta}_r \leq \alpha_r \quad \forall \mathcal{R}_r \in \mathcal{R} \quad (57)$$

Then, the constraints below can be added to set $\bar{z} = \bar{z}_r$ where \mathcal{R}_r is the only visited region of its partition:

$$\bar{z} \leq \bar{z}_r + w_u(1 - \gamma_r) \quad \forall \mathcal{R}_r \in \mathcal{P}_n \quad (58)$$

$$\bar{z} \geq \bar{z}_r - 2w_u(1 - \gamma_r) \quad \forall \mathcal{R}_r \in \mathcal{P}_n \quad (59)$$

where w_u is an upper bound for \bar{z} .

Similarly, $\Delta \bar{\lambda}[e] = \Delta \bar{\lambda}_G[(e, g_r)]$ can be obtained by:

$$\Delta \bar{\lambda}[e] \leq \Delta \bar{\lambda}_G[(e, g_r)] + w_u(1 - \gamma_r) \quad \forall e \in E_S^T, \forall \mathcal{R}_r \in \varphi(e) \quad (60)$$

$$\Delta \bar{\lambda}[e] \geq \Delta \bar{\lambda}_G[(e, g_r)] - 2w_u(1 - \gamma_r) \quad \forall e \in E_S^T, \forall \mathcal{R}_r \in \varphi(e) \quad (61)$$

where w_u is an upper bound for $\Delta \lambda$.

APPENDIX B AUXILIARY BOUNDS

This Appendix proposes methods to compute bounds to linearize $\alpha_r \Delta \bar{\lambda}_{U_r}[(e, g_r)]$, $\bar{\delta} \Delta \bar{\lambda}_G$, $\bar{\delta}_r \bar{z}_r$ and $\alpha_r \bar{z}_r$ (see Appendix A) and components of W in (46) and (47). For simplicity, it will be assumed that $\Delta \lambda_U$ is bounded by the constraint $C \Delta \lambda_U \leq D \mu_E$.

Let us consider the following set of constraints based on (15):

$$\begin{aligned} C_{un} = \{ & m = \mu_P + Y_m \mu_E; C \Delta \lambda_U \leq D \mu_E \\ & \sigma = a_T + Y_\sigma a_E; A \Delta m \leq B a_E \\ & m = m_0 + Z_m \Delta m; J_m m_0 \leq K_m \} \end{aligned} \quad (62)$$

The constraints in (62) can be used to computed auxiliary bounds for the steady state. Namely, w_l and w_u for the linearization of $\bar{\delta} \Delta \bar{\lambda}_G$ and $\alpha_r \Delta \bar{\lambda}_{U_r}[(e, g_r)]$ can be computed with:

$$\begin{aligned} w_l &= \min_{\substack{(e, g_r) \\ e \in E_S^T, \mathcal{R}_r \in \varphi(e)}} \{ \min \{ \Delta \lambda_U[(e, g_r)] | C_{un} \} \} \\ w_u &= \max_{\substack{(e, g_r) \\ e \in E_S^T, \mathcal{R}_r \in \varphi(e)}} \{ \max \{ \Delta \lambda_U[(e, g_r)] | C_{un} \} \} \end{aligned} \quad (63)$$

Consider the following constraints:

$$\begin{aligned} C_{st} = \{ & \bar{m} = \bar{\mu}_P + Y_m \bar{\mu}_E; C \Delta \bar{\lambda}_U \leq D \bar{\mu}_E \\ & \bar{\lambda} = \lambda_0 + Z_\lambda \Delta \bar{\lambda}; J_\lambda \lambda_0 \leq K_\lambda \\ & \Delta \bar{\lambda}[e] \leq w_u \quad \forall e \in E_S^T \\ & \bar{\sigma}_\tau = \bar{\lambda}; \bar{\sigma}_\tau = \bar{a}_{T\tau} + Y_\sigma \bar{a}_{E\tau} \\ & A \Delta \bar{m}_\tau \leq B \bar{a}_{E\tau}; J_m m_0 \leq K_m \\ & Z_m \Delta \bar{m}_\tau = 0 \\ & \sigma = a_T + Y_\sigma a_E; A \Delta m \leq B a_E \\ & \bar{m} = m_0 + Z_m \Delta m \} \end{aligned} \quad (64)$$

where w_u is computed according to (63).

The value w'_u for the linearization of $\bar{\delta}_r \bar{z}'_r$ in (46) and for the multiplication $\alpha_r \bar{z}'_r$ in (47) by (49), where \bar{z}'_r is the result of removing the components $\Delta \bar{\lambda}_{U_r}$, $\Delta \bar{\lambda}_r$ and $\bar{\lambda}_r$ from \bar{z}_r , can be computed by:

$$w'_u = \max \{ \mathbf{1} \bar{m} + \mathbf{1} \bar{\lambda} + \mathbf{1} \Delta \bar{m}_\tau | C_{st} \} \quad (65)$$

In the objective function of (65), the terms $\mathbf{1} \bar{m}$; $\mathbf{1} \bar{\lambda}$; and $\mathbf{1} \Delta \bar{m}_\tau$ guarantee and upper bound for \bar{m} , $\bar{\mu}_P$ and $\bar{\mu}_E$; $\bar{\sigma}_\tau$, $\bar{a}_{T\tau}$ and $\bar{a}_{E\tau}$; and $\Delta \bar{m}_\tau$ respectively. A trivial value for w'_i for the linearization of $\bar{\delta}_r \bar{z}'_r$ is 0.

Finally, all the components of W in (46) and (47) can be computed by the following linear program:

$$w = \max \{ \sum_{\mathcal{R}_r \in \mathcal{R}} \sum_k (\sum_l |S_r[k, l]| \mathbf{x}[l] + |Q_r[k]|) | C_{un} \} \quad (66)$$

where k and l are the indices for the rows and columns of S_r . Notice that in (66), the constraint $C \Delta \lambda_U \leq D \mu_E$ in C_{un} can be removed as the columns of S_r that correspond to $\Delta \lambda_U$, $\Delta \lambda$ and λ are assumed to be 0.

REFERENCES

- [1] IBM ILOG CPLEX Optimizer, 2010.
- [2] A. Ahangarani and A. Dideban. Continuous-Time Delay-Petri Nets as a new tool to Design State Space Controller. *Information Technology And Control*, 45, 01 2017.
- [3] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis. *Modelling with Generalized Stochastic Petri Nets*. Wiley, 1995.
- [4] F. Balduzzi, A. Giua, and G. Menga. First-order hybrid Petri nets: a model for optimization and control. *IEEE Transactions on Robotics and Automation*, 16(4):382–399, Aug 2000.
- [5] A. Bemporad and M. Morari. Control of systems integrating logic, dynamics, and constraints. *Automatica*, 35(3):407–427, March 1999.
- [6] J. A. Bergstra, A. Ponse, and S. A. Smolka, editors. *Handbook of Process Algebra*. Elsevier Science Inc., New York, NY, USA, 2001.
- [7] B. Berthomieu and M. Diaz. Modeling and Verification of Time Dependent Systems Using Time Petri Nets. *IEEE Trans. on Software Engineering*, 17(3):259–273, 1991.
- [8] J. Billington and G. E. Gallasch. On Parametric Steady State Analysis of a Generalized Stochastic Petri Net with a Fork-join Subnet. In *Proceedings of the 32Nd International Conference on Applications and Theory of Petri Nets*, pages 268–287, Berlin, Heidelberg, 2011. Springer-Verlag.
- [9] A. Bordbar, J. M. Monk, Z. A. King, and B. Ø. Palsson. Constraint-based models predict metabolic and associated cellular functions. *Nature Reviews Genetics*, 15:107–120, 2014.
- [10] J. Campos, G. Chiola, and M. Silva. Properties and Performance Bounds for Closed Free Choice Synchronized Monoclass Queuing Networks. *IEEE Transactions on Automatic Control*, 36(12):1368–1382, December 1991.
- [11] G. Caravagna, A. d’Onofrio, M. Antonioti, and G. Mauri. Stochastic hybrid automata with delayed transitions to model biochemical systems with delays. *Information and Computation*, 236:19 – 34, 2014. Special Issue on Hybrid Systems and Biology.

- [12] F. Ciocchetta and J. Hillston. Bio-PEPA: An Extension of the Process Algebra PEPA for Biochemical Networks. *Electron. Notes Theor. Comput. Sci.*, 194(3):103–117, January 2008.
- [13] A. Clark, S. Gilmore, J. Hillston, and M. Tribastone. *Stochastic Process Algebras*, pages 132–179. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [14] P.J.L. Cuijpers and M.A. Reniers. Hybrid process algebra. *The Journal of Logic and Algebraic Programming*, 62(2):191 – 245, 2005.
- [15] R. David and H. Alla. *Discrete, Continuous and Hybrid Petri Nets*. Springer, Berlin, 2004. (2nd edition, 2010).
- [16] M. Dotoli, M.P. Fanti, A. Giua, and C. Seatzu. First-order hybrid Petri nets. An application to distributed manufacturing systems. *Nonlinear Analysis: Hybrid Systems*, 2(2):408 – 430, 2008. Proceedings of the International Conference on Hybrid Systems and Applications, Lafayette, LA, USA, May 2006: Part II.
- [17] Gurobi Optimization, Inc. Gurobi Optimizer Reference Manual, 2015.
- [18] W. E. Hart, C. Laird, J.-P. Watson, and D. L. Woodruff. *Pyomo—Optimization Modeling in Python*, volume 67. Springer Science & Business Media, 2012.
- [19] T. A. Henzinger. The Theory of Hybrid Automata. In *Proceedings of the 11th Annual IEEE Symposium on Logic in Computer Science, LICS '96*, pages 278–, Washington, DC, USA, 1996. IEEE Computer Society.
- [20] J. Júlvez, D. Dikicioglu, and S. G. Oliver. Handling variability and incompleteness of biological data by flexible nets: a case study for Wilson disease. *npj Systems Biology and Applications*, 4(1):7, 1 2018.
- [21] J. Júlvez and S. G. Oliver. Modeling, analyzing and controlling hybrid systems by Guarded Flexible Nets. *Nonlinear Analysis: Hybrid Systems*, 32:131 – 146, 2019.
- [22] J. Júlvez, L. Recalde, and M. Silva. Steady-state performance evaluation of continuous mono-T-semiflow Petri nets. *Automatica*, 41(4):605 – 616, 2005.
- [23] N. Lynch, R. Segala, and F. Vaandrager. Hybrid i/o automata. *Information and Computation*, 185(1):105 – 157, 2003.
- [24] T. Murata. Petri Nets: Properties, Analysis and Applications. *Procs. of the IEEE*, 77(4):541–580, 1989.
- [25] B. Øksendal. *Stochastic Differential Equations: An Introduction with Applications (Universitext)*. Springer, Berlin, Heidelberg, New York, 6th edition, sep 2010.
- [26] J. Orth, I. Thiele, and B.Ø. Palsson. What is flux balance analysis? *Nature biotechnology*, 28:245–248, 03 2010.
- [27] J. D. Orth, T. M Conrad, J. Na, J. A Lerman, H. Nam, A. M Feist, and B. Ø. Palsson. A comprehensive genome-scale reconstruction of Escherichia coli metabolism–2011. *Molecular Systems Biology*, 7(1), 2011.
- [28] O. Pourret, P. Naim, and B. Marcot. *Bayesian Networks: A Practical Guide to Applications*. Statistics in Practice. Wiley, 2008.
- [29] C. Priami, A. Regev, E. Shapiro, and W. Silverman. Application of a stochastic name-passing calculus to representation and simulation of molecular processes. *Information Processing Letters*, 80(1):25 – 31, 2001. Process Algebra.
- [30] L. Recalde and M. Silva. Petri Nets Fluidification revisited: Semantics and Steady state. *APII-JESA*, 35(4):435–449, 2001.
- [31] W. Reisig. *Understanding Petri Nets: Modeling Techniques, Analysis Methods, Case Studies*. Springer Publishing Company, Incorporated, 2013.
- [32] I. Shmulevich and E. R. Dougherty. *Probabilistic Boolean Networks: The Modeling and Control of Gene Regulatory Networks*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2009.
- [33] I. Shmulevich, E. R. Dougherty, S. Kim, and W. Zhang. Probabilistic Boolean networks: a rule-based uncertainty model for gene regulatory networks. *Bioinformatics*, 18(2):261–274, 2002.
- [34] J. J. Tyson, W. T. Baumann, C. Chen, A. Verdugo, I. Tavassoly, Y. Wang, L. M. Weiner, and R. Clarke. Dynamic modelling of oestrogen signalling and cell fate in breast cancer cells. *Nature reviews. Cancer*, 11 7:523–32, 2011.
- [35] R. van den Berg, E. Lefeber, and K. Rooda. Modeling and control of a manufacturing flow line using partial differential equations. *Control Systems Technology, IEEE Transactions on*, 16:130 – 136, 02 2008.
- [36] A. Varma and B. Ø. Palsson. Metabolic Flux Balancing: Basic Concepts, Scientific and Practical Use. *Nature Biotechnology*, 12(10):994–998, October 1994.
- [37] C. R. Vázquez and M. Silva. Stochastic Continuous Petri Nets: An Approximation of Markovian Net Models. *IEEE Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans*, 42(3):641–653, May 2012.
- [38] C. R. Vázquez and M. Silva. Stochastic Hybrid Approximations of Markovian Petri nets. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 45(9):1231–1244, Sep. 2015.
- [39] K. Voss, M. Heiner, and I. Koch. Petri nets for steady state analysis of metabolic systems. *Studies in health technology and informatics*, 162:56–76, 01 2011.
- [40] R. Sheng Wang, A. Saadatpour, and R. Albert. Boolean modeling in systems biology: an overview of methodology and applications. *Physical Biology*, 9(5):055001, 2012.
- [41] T. Wright and I. Stark. The bond-calculus: A process algebra for complex biological interaction dynamics. *CoRR*, abs/1804.07603, 2018.



Jorge Júlvez is associate professor at the Department of Computer Science and Systems Engineering in the University of Zaragoza. Jorge received his M.S. and Ph.D. degrees in computer science engineering in 1998 and 2005 from the University of Zaragoza. His Ph.D. was related to the study of qualitative and quantitative properties of continuous Petri nets. In 2005 he joined, as a PostDoc researcher, the Department of Software in the Technical University of Catalonia, where he spent three years. As a researcher, he has visited the Department of Electrical and Electronic Engineering in the University of Cagliari, the Department of Information Engineering in the University of Siena, the SYSTeMS Research group in the University of Ghent, and the Computing Laboratory in the University of Oxford. In 2014, Jorge moved to the Cambridge Systems Biology Centre where he spent four years, two of them as a Marie Curie fellow. Jorge's research interests are in the development of modeling formalisms and computational methods to model, analyse and optimize biological systems.



Steve Oliver is Emeritus Professor of Systems Biology at the University of Cambridge and was Director of the Cambridge Centre for Systems Biology. His research involves both experimental and bioinformatics approaches to understanding the workings of the eukaryotic cell. Steve Oliver led the European team that sequenced the first chromosome, from any organism, yeast chromosome III. He continued to play a major role in the Yeast Genome Sequencing Project, and went on to become Scientific Coordinator of EUROFAN, which pioneered a wide range of approaches to the systematic analysis of gene function, using *S. cerevisiae*. His current work employs a range of comprehensive, high-throughput analytical techniques (transcriptomics, proteomics, metabolomics, and rapid phenotyping). He exploited genome-wide metabolic models to identify functional modules within the yeast metabolic network and predict epistatic interactions between genes. Steve collaborated with Ross King to develop the Robot Scientist closed-loop machine-learning system for functional genomic hypothesis generation and experimentation, and re-engineered the genome configuration of yeast to provide a direct test of the chromosomal theory of evolution.

Steve Oliver is a member of EMBO, and a Fellow of: the American Association for the Advancement of Science, the American Academy of Microbiology, the Academy of Medical Sciences, and the Royal Society of Biology. He is an Honorary Member of both the Hungarian Academy of Sciences and the British Mycological Society. Prof. Oliver was the Kathleen Barton-Wright Memorial Lecturer of the Institute of Biology & Society for General Microbiology in 1996, won the AstraZeneca Award of the Biochemical Society in 2001, the Marjory Stephenson Prize of the Microbiology Society in 2016, and was the 2018 recipient of the Genetics Society of America Yeast Genetics Meetings Lifetime Achievement Award.