This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

A quantitative approach to input generation in real-time testing of stochastic systems

Laura Carnevali, Lorenzo Ridi, Enrico Vicario
Dipartimento di Sistemi e Informatica - Università degli Studi di Firenze
{ laura.carnevali, lorenzo.ridi, enrico.vicario }@unifi.it

***Abstract*—In the process of testing of concurrent timed systems, input generation identifies values of temporal parameters that let the Implementation Under Test (IUT) execute selected cases. However, when some parameters are not under control of the driver, test execution may diverge from the selected input and produce an inconclusive behavior.**

We formulate the problem on the basis of an abstraction of the IUT which we call partially stochastic Time Petri Net (psTPN), where controllable parameters are modeled as non-deterministic values and non-controllable parameters as random variables with general (GEN) distribution. With reference to this abstraction, we derive the analytical form of the probability that the IUT runs along a selected behavior as a function of choices taken on controllable parameters. In the applicative perspective of real-time testing, this identifies a theoretical upper limit on the probability of a conclusive result, thus providing a means to plan the number of test repetitions that are necessary to guarantee a given probability of test-case coverage. It also provides a constructive technique for an optimal or sub-optimal approach to input generation and a way to characterize the probability of conclusive testing under other suboptimal strategies.

Keywords: *Real-time testing, input generation, Time Petri Nets, non-Markovian Stochastic Petri Nets, stochastic processes, Difference Bound Matrix.*

I. Introduction

In the testing process, an implementation is exercised under controlled conditions with the intent of observing deviations with respect to the expected behavior [30], [2]. In particular, in the development of reactive and real-time systems, this often relies on the execution of test-suites derived from abstractions that focus on finite-state [17], [37], [22], [29] and timed behavior [36], [16], [19], [34], [27], [23], [24], with the aim of revealing defects related to concurrency, communication and timeliness [33]. A rich survey on timed and untimed model-based approaches to testing can be found in [21], focusing on the problems of test-suites generation and execution. In [37] a model-based theory for conformance testing of Labelled Transition Systems (LTS) and Input/Output Transition Systems (IOTS) is presented; the theory defines the *input-output conformance* (ioco) implementation relation, which in turn allows the definition of test-suite generation algorithms [22]. In a more applicative perspective, [29] derives test-suites as coverage of a bisimulation reduction of the LTS modeling system dynamics. Coverage of an untimed and deterministic Finite State Machine (FSM) is also addressed in the partial-WpMethod [17], [13], where state-verification and full fault-coverage are achieved under the assumption of an upper bound on the number of states in the implementation. The approach is extended to timed models in [16], by considering a finite sampling of the Region Graph [1] of a Timed Input Output Automaton (TIOA) [36]. In [28] a comprehensive framework based on FSMs is presented for the generation of tests on systems whose temporal parameters may also include time intervals or random variables. In [34], [27], [23] and [24], timed extensions of the ioco relation of [37] are presented, and algorithms for the generation of timed test suites are defined [24]. In [19] test cases are selected as deterministically timed event sequences of a deterministic and output-urgent Timed Automaton, either as witnesses of real-time logic expressions capturing specific *testing purposes* or as elements of a test-suite covering locations of the specification automaton [20].

The actual execution of tests poses the further problem of *generation of inputs* that let the Implementation Under Test (IUT) run along each test case. Various works have addressed the problem in the real-time context, with different assumptions about *feasibility* and *controllability* of timings. If every timing accepted by the specification is also feasible and controllable in the IUT, input generation becomes a matter of identifying and applying the values of temporal parameters accepted by the specification model under the restriction of the selected case [19]. However, implementation usually involves a *partial determinization* that rules out runs allowed by the specification but not exercised by the IUT. This issue was addressed in [23], where the IUT is assumed to allow non-deterministic choices and delays. The problem was also addressed and practically experimented on real-time SW components in [12], where test-cases are specified as symbolic runs that prescribe the order of events but not their timing.

In most practical cases, the generation of timed inputs faces the further complexity of non-controllable temporal parameters: for instance, in the exercise of a real-time task-set, a test-driver can reasonably set the release time of asynchronous tasks, but it is unlikely that it can control with adequate precision the Execution Time of computations. Test execution is thus afflicted by the infamous problem of *inconclusive behaviors*, where the IUT diverges from a selected test-case. In [23], controllable temporal parameters are adapted on-line so as to keep the overall timing in the range that makes a

Digital Object Indentifier 10.1109/TSE.2012.42

conclusive end possible, and to terminate the execution as soon as this turns out to be impossible. However, the approach does not rely on a measure of probability for the choice among different acceptable inputs. Moreover, when the IUT runs on limited resources or under severe real-time constraints, run-time adaptation is not a viable option, and timings must be decided off-line [12]. In this case, a characterization of the probability of successful execution as a function of choices taken on controllable timers enables optimization on the expected number of repetitions needed to conclude each test-case. This characterization is largely hurdled by the complexity of the stochastic process that underlies the behavior of the IUT, which normally includes multiple concurrently enabled timers associated with general (GEN) distributions, often supported over finite domains.

In [43], this problem is faced on the basis of a specification model represented as a Stochastic Input Output Automaton (STIOA), where input actions are events that can be forced to occur at any desired time, and output actions are non-controllable events with uniform distribution over finite intervals; under these assumptions, for each selected timing of input actions the probability of conclusive execution of a symbolic run is proven to be proportional to the volume of a multi-variate Difference Bounds Matrix (DBM) domain [15] collecting the timings that make the run feasible. In [9], preliminary results were reported on the evaluation of the probability of execution of symbolic runs of a more general model that combines controllable transitions bounded to fire within a (possibly infinite) Firing Interval and non-controllable transitions with general (GEN) distribution over non-pointlike (possibly finite) supports.

In this paper, we reformulate and extend the theory of [9], and we apply it to the stochastic characterization of the problem of test input generation in partially controllable concurrent timed systems. To this end: in Sect.II, we formalize the model of partially stochastic Time Petri Nets, which we assume as abstraction of the concurrent and timed behavior of an IUT with controllable and non-controllable transitions distributed according to any GEN probability density function; in Sect.III, we recall the salient concepts of symbolic state space enumeration of non-deterministic timed models based on the abstraction of state classes, we motivate the assumption of test-cases specified as symbolic runs, and we characterize the probability of conclusive execution of a test-case as a function of values given to controllable temporal parameters; in Sect.IV, we report on an implementation of the proposed theory and we then show how the probability function can be applied to reduce the number of inconclusive tests, evaluating the gain that can be attained through an optimization approach or through sub-optimal sampling techniques. Conclusions are finally drawn in Sect.V.

II. Combining non-deterministic and stochastic behavior in psTPN models

A *partially stochastic Time Petri Net* (psTPN) is a *Time Petri Net* (TPN) [38][5] with a measure of probability for the times-to-fire sampled by transitions accounting for non-controllable events. In a complementary perspective, it can also be regarded as a *stochastic Time Petri Net* (STPN) [39][8] that leaves undefined the probability distribution of times-to-fire of transitions accounting for controllable events. In both perspectives, the salient traits of psTPNs are the combination of non-deterministic and stochastic behavior, and the possible presence of multiple concurrently enabled transitions with GEN distributions.

A. Partially stochastic Time Petri Nets

A *partially stochastic Time Petri Net* (psTPN) is a tuple:

$$psTPN = < P; T^c; T^{nc}; A^-; A^+; A^\bullet; m_0; EFT; LFT; \mathcal{C}; \mathcal{F} > \qquad (1)$$

- P is a set of places.
- T^c and T^{nc} are disjoint sets of transitions associated with controllable and non-controllable times-to-fire, respectively, and their union is denoted by T.
- $A^- \subseteq P \times T$, $A^+ \subseteq T \times P$, and $A^\bullet \subseteq P \times T$ are sets of precondition, postcondition and inhibitor arcs, respectively. Place p is said to be an input place, an output place or an inhibitor place for transition t if $\langle p, t\rangle \in A^-$, $\langle t, p\rangle \in A^+$, or $\langle p, t\rangle \in A^\bullet$, respectively.
- $m_0 : P \rightarrow \mathbb{N}$ is the (initial) marking associating each place with a non-negative number of tokens.
- EFT and LFT associate each transition $t \in T$ with a static firing interval made of an Earliest and a (possibly infinite) Latest Firing Time:
$$EFT : T \rightarrow \mathbb{Q}_0^+ \qquad LFT : T \rightarrow \mathbb{Q}_0^+ \cup \{+\infty\} \qquad (2)$$
where $\mathbb{Q}_0^+$ denotes the set of non-negative rational numbers, $EFT(t) \leq LFT(t)\ \forall\ t \in T^{nc}$, and $EFT(t) < LFT(t)\ \forall\ t \in T^c$.
- $\mathcal{C}$ associates each transition with a weight $\mathcal{C} : T \rightarrow \mathbb{R}^+$.
- $\mathcal{F}$ associates each transition $t \in T^{nc}$ with a static probability distribution $F_t()$ defined over its static firing interval $[EFT(t), LFT(t)]$. For simplicity and without loss of generality, we assume that the Probability Distribution Function of every non-controllable transition supported over a non-pointlike domain is an absolutely continuous function, which can thus be expressed as the integral of a Probability Density Function:
$$F_t(x) = \int_0^x f_t(y)dy. \qquad (3)$$

The state of a psTPN is a pair $s = \langle m, \tau\rangle$, where $m : P \rightarrow \mathbb{N}$ is a marking and $\tau : T \rightarrow \mathbb{R}_0^+$ associates each transition with a (dynamic) time-to-fire. The state evolves according to two clauses of *firability* and *firing*.

- *Firability:* A transition t_0 is enabled if each of its input places contains at least one token and none of its inhibiting places contains any token; it is firable if it is enabled and its time-to-fire $\tau(t_0)$ is not higher than that of any other enabled transition: $\tau(t_0) \leq \tau(t_i) \quad \forall\ t_i \in T^e(m)$, where $T^e(m)$ denotes the set of transitions enabled by marking m. When multiple transitions are firable, the choice is resolved through a random switch determined by $\mathcal{C}$:

$$Prob\{t_0 \text{ is selected}\} = \frac{\mathcal{C}(t_0)}{\sum\limits_{t_i \text{ firable}} \mathcal{C}(t_i)} \tag{4}$$

- *Firing:* When a transition t_0 fires, the state $s = \langle m, \tau \rangle$ is replaced by a new state $s' = \langle m', \tau' \rangle$, which we write as $s \overset{t_0}{\to} s'$. Marking m' is derived from m by removing a token from each input place of t_0, and by adding a token to each output place of t_0, as usual in Petri Nets:

$$\begin{array}{ll} m_{tmp}(p) = m(p) - 1 & \forall\ p.\ \langle p, t_0 \rangle \in A^- \\ m'(p) = m_{tmp}(p) + 1 & \forall\ p.\ \langle t_0, p \rangle \in A^+ \end{array} \tag{5}$$

Transitions that are enabled both by the intermediate marking m_{tmp} and by the final marking m' are said *persistent*, while those that are enabled by m' but not by m or by m_{tmp} are said *newly-enabled*. As in [5][38], if t_0 is still enabled after its own firing, it is always regarded as newly-enabled.
The time-to-fire τ' of any transition enabled by the new marking m' is computed in a different manner for transitions that are persistent and newly-enabled after the firing of t_0. For any persistent transition t_i, the time-to-fire is reduced by the time elapsed in the previous state:

$$\tau'(t_i) = \tau(t_i) - \tau(t_0) \tag{6}$$

For any newly-enabled controllable transition $t_c \in T^c$, the time-to-fire takes a non-deterministic value in the static firing interval:

$$\tau(t_c) \in [EFT(t_c), LFT(t_c)] \tag{7}$$

Besides, for any newly-enabled non-controllable transition $t_{nc} \in T^{nc}$, the time-to-fire is sampled according to the static probability distribution $F_{t_{nc}}()$:

$$\begin{array}{c} \tau(t_{nc}) \in [EFT(t_{nc}), LFT(t_{nc})] \\ \text{with } Prob\{\tau(t_{nc}) \leq x\} = F_{t_{nc}}(x) \end{array} \tag{8}$$

B. Related works

Combination of nondeterministic and stochastic behavior is addressed in various continuous-time formalisms with different aims. In Continuous Time Markov Decision Processes (CTMDP), discrete controllable choices are embededd within a Continuous Time Markov Chain. Generalized Semi-Markov Decision Processes (GSMDP) [44] extend the model by allowing multiple non-exponentially distributed timers. For both CTMDP and GSMDP, the analysis combines controllable actions and exogenous events in order to formulate a decision problem: while controllable actions can be executed according to any selected policy, exogenous events are associated with a generally distributed time to fire and execute according to a race policy when no controllable actions are chosen. In [44], generally distributed timers are replaced by continuous Phase Types so as to reduce the problem to the solution of a CTMDP. As a major difference with respect to psTPNs, in both CTMDP and GSMDP, controllable actions determine immediate choices rather than values of continuous timers.

In [25], Continuous Probabilistic Timed Automata (CPTA) are introduced as a stochastic variant of Timed Automata combining non-deterministic and probabilistic choices, with generally distributed clocks. Approximation of the underlying stochastic process through a discrete sampling of the region graph provides a discrete abstraction for model checking verification, but incurs into an explosion of process states which limits practical application.

The case of a controller that can determine the value of temporal parameters ranging within continuous domains is addressed in [43] with reference to the model of Stochastic Timed Input-Output Automata (STIOA). In this model, input actions represent controllable events that can be fired at any selected time, while output actions are non-controllable events associated with a duration uniformly distributed within a finite interval. By leveraging on the assumption of uniform distribution, generation of timers that maximize the probability of conclusive execution is reduced to the maximization of the volume of zones including times of output actions that let the IUT run along the selected run. As opposed to STIOA, psTPNs also permit to constrain the time interval within which controllable actions can be forced to fire; more importantly, non-controllable actions can also assume any GEN distribution supported over any connected domain. The two extensions better fit the needs of real-time testing: in this context, controllable actions are usually subject to some kind of timeliness restriction; moreover, the distribution of non-controllable actions generally comes from Execution Time analysis [40], which usually yields profiling measures that can be conveniently fitted through GEN distributions. As a counterpart, the higher expressivity changes the complexity of the analysis, requiring that the volume of feasible timings be equipped with a measure of probability.

III. Characterization of the probability of conclusive execution

We recall the essential concepts of symbolic analysis of the non-deterministic TPN that underlies a psTPN model (Sect.III-A), and we develop them so as to identify the set of timings that let the model run along a selected run (Sect.III-B) and to characterize their probability distribution as a function of values given to the firing-times of controllable transitions (Sect.III-C). A small example illustrates the derivation (Sect.III-D).

A. State classes in symbolic analysis of psTPN models

The set of feasible behaviors of a TPN can be conveniently covered through *state classes* [38], each including a set of states with the same marking and with valuations of the vector of times to fire within a continuous set D. Formally:

$$State\ class = \langle m, D \rangle, \tag{9}$$

where m is a marking and D is a continuous set of times-to-fire of transitions enabled by m:

$$D \subseteq (\mathbb{R}_{\geq 0})^{||T^e(m)||}, \tag{10}$$

$||T^e(m)||$ denoting the cardinality of the set $T^e(m)$ of transitions enabled by m.

As usual in symbolic state space analysis of models with nondeterministic timing, we assume that in the initial class times to fire of the N enabled transitions range within a Difference Bounds Matrix (DBM) [15][38] zone:

$$D = \begin{cases} \tau_i - \tau_j \leq b_{ij} \\ \tau_* = 0 \\ \forall\ i \neq j \in [0, N-1] \cup \{*\} \end{cases} \tag{11}$$

where $\tau_* = 0$ represents the ground time at which the class was entered and $b_{ij} \in \mathbb{Q} \cup \{\pm\infty\}$ are rational values determining the boundaries of the firing domain. Under this assumption, it is easily shown that all reachable classes are still in DBM form, which enables efficient and compact encoding of the firing domain. In particular, a non-empty DBM zone has a unique *normal form* of representation where coefficients $-b_{ji}$ and b_{ij} belong to $\mathbb{Q} \bigcup \{\pm\infty\}$ and coincide with the minimum and maximum values that can be attained by the difference $\tau_i - \tau_j$, respectively. In particular, $-b_{*i}$ and b_{i*} turn out to represent the minimum and the maximum values of τ_i which yield solutions for system D, respectively. The normal form is univocally identified by the condition:

$$b_{ij} \leq b_{ih} + b_{hj} \qquad \forall i, j, h \in [0, N-1] \cup \{*\} \text{ with } i \neq j \neq h \neq i \tag{12}$$

and it can be derived as the solution of an all-shortest-path problem in time $O(N^3)$ with respect to the number N of enabled transitions, or even in time $O(N^2)$ in a repeated derivation exploiting warm restart [38].

The relation of reachability among states is covered through a symbolic reachability relation between state classes [32]:

Definition 1: Given two state classes $S = \langle m, D \rangle$ and $S' = \langle m', D' \rangle$, we say that S' is a successor of S through t_0, and we write $S \overset{t_0}{\Longrightarrow} S'$, if and only if S' contains all and only the states that are reachable from some state collected in S through some feasible firing of t_0.

Enumeration of the relation $S \overset{t}{\Longrightarrow} S'$ produces a State Class Graph (SCG) [5], [38] that provides a compact representation for the set of feasible firing sequences: by transitive closure of *Definition* (1), if the TPN is in a marking m and its times-to-fire are distributed over a domain D, then a firing sequence ρ can be executed if and only if ρ is a path originating from class $S^0 = \langle m, D \rangle$ in the SCG. In this case, the state at the end of the sequence has the marking of the class $S^N = \langle m^N, D^N \rangle$ reached by the path, and its times-to-fire take a value within D^N. Thus, any path in the SCG identifies a qualitative sequence of transition firings that can be executed with a continuous set of timings, here called *symbolic run*.

Symbolic runs comprise a robust yet effective abstraction for the selection of test-cases [12], enabling significant selection criteria that cover the variety of reachable logical states and feasible event sequences [7]. Moreover, symbolic runs represent a suitable test-case abstraction even in the presence of densely-distributed non-controllable timers, where the specification of a run with determined pointlike timings would result in a null probability of obtaining a conclusive test.

B. Domain of feasible timings along a symbolic run

We consider a symbolic run ρ that originates when class S^0 is entered, visits classes S^1 through S^{N-1}, and terminates when class S^N is reached (see Fig.1 for an example). To give identity to activations of transitions that are enabled in multiple classes along ρ, a transition t_i newly-enabled in S^n will be denoted with t_i^n in S^n and in all the subsequent classes where it is persistent. For instance, in Fig.1, t_1^0 and t_1^1 denote the activations of transition t_1 in classes S^0 and S^1, respectively; besides, t_4^1 denotes the activation of transition t_4 that is newly-enabled in S^1 and persistent until the firing that enters class S^4.

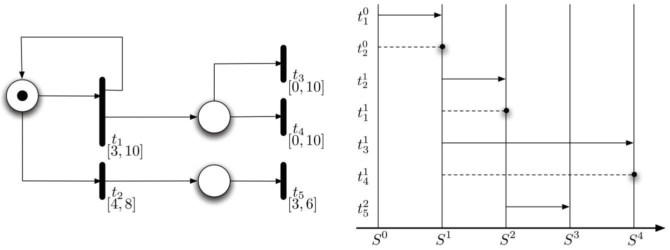


Figure 1. A simple net and the schema of one of its symbolic runs: $\rho = S^0 \xrightarrow{t_1} S^1 \xrightarrow{t_2} S^2 \xrightarrow{t_5} S^3 \xrightarrow{t_3} S^4$. Dotted lines denote transitions that do not come to fire; a dot marks the point where the transition is disabled.

Each transition activation t_i^n is associated with an *absolute virtual firing-time* (firing-time, for short) τ_i^n, which is the sum of the time-to-fire taken by t_i at its newly-enabling plus the time elapsed from the start of the sequence ρ to the firing that enters S^n. We say that τ_i^n is *absolute* as it is referred to the start time of the run, and that it is *virtual* as t_i^n might not come to fire, either because it is disabled or because the sequence ρ terminates. In Fig.1, this is for instance the case of t_1^1, which is disabled by t_2^1.

To represent the relations among transition activations, we consider four natural-valued functions $\iota(n)$, $\nu(n)$, $\gamma(j,n)$ and $\delta(j,n)$, such that: $\iota(n)$ is the index of the transition that enters class S^n, while $\nu(n)$ is the index of its enabling class (i.e. $t_{\iota(n)}^{\nu(n)}$ is the transition activation that enters the class S^n); for any transition activation t_j^n enabled but not fired along ρ, $\gamma(j,n)$ is the index of the transition that disables t_j^n or terminates the sequence ρ, and $\delta(j,n)$ is the index of its enabling class (i.e. $t_{\gamma(j,n)}^{\delta(j,n)}$ is the transition activation that disables t_j^n or terminates the sequence ρ). For instance, in Fig.1: class S^1 is entered through the firing of t_1^0 and thus $t_{\iota(1)}^{\nu(1)} = t_1^0$; t_1^1 is disabled by t_2^1 and thus $t_{\gamma(1,1)}^{\delta(1,1)} = t_2^1$; for simplicity, the schema omits to show the fictitious activation $t_{\iota(0)}^{\nu(0)}$ which enters the initial class S^0 at time $\tau_{\iota(0)}^{\nu(0)} = 0$. Note that, for each activation, all the four indexing functions can be derived in time $O(N)$ with respect to the length N of the symbolic run ρ.

The set of firing-times for transitions that are consistent with firing intervals of transitions and let the model execute along the symbolic run ρ turns out to be a DBM zone:

Theorem 1: Let S^0 be a class where all ebabled transitions are newly-enabled, and let ρ be the symbolic run that originates in S^0 and reaches S^N after visiting classes S^1 through S^{N-1}. A vector $\boldsymbol{\tau}$ of valuations for the firing-times of transitions enabled along ρ is consistent with the semantics of the model and with the constraints that let it execute along ρ if and only if it belongs to the following DBM zone $D_{\boldsymbol{\tau}}$:

$$D_{\boldsymbol{\tau}} = \begin{cases} EFT(t_i) & & \\ \leq \tau_i^n - \tau_{\iota(n)}^{\nu(n)} & \forall\ t_i^n \text{ enabled along } \rho & (a) \\ \leq LFT(t_i) & & \\ \tau_x^n \geq \tau_{\gamma(x,n)}^{\delta(x,n)} & \forall\ t_x^n \text{ enabled, not fired along } \rho & (b) \\ \tau_{\iota(n+1)}^{\nu(n+1)} \geq \tau_{\iota(n)}^{\nu(n)} & \forall\ n \in [0, N-1] & (c) \end{cases} \tag{13}$$

Proof: - if

According to the semantics of psTPNs, a vector $\boldsymbol{\tau}$ capturing a feasible timing for ρ satisfies the following constraints:

- For each activation t_i^n, the time-to-fire taken at the newly enabling falls in the interval $[EFT(t_i), LFT(t_i)]$. Besides, this time-to-fire can be expressed as the difference $\tau_i^n - \tau_{\iota(n)}^{\nu(n)}$ between the firing-time of t_i^n itself and the firing-time $\tau_{\iota(n)}^{\nu(n)}$ of the transition that enters the enabling class of t_i^n:

$$EFT(t_i) \leq \tau_i^n - \tau_{\iota(n)}^{\nu(n)} \leq LFT(t_i) \quad \forall\ t_i^n \text{ enabled along } \rho$$

- The firing-time of any transition t_x^n that does not come to fire is not lower than the firing-time of the transition that disables t_x^n itself or terminates the sequence ρ:

$$\tau_x^n \geq \tau_{\gamma(x,n)}^{\delta(x,n)} \quad \forall\ t_x^n \text{ enabled but not fired along } \rho$$

- According to the sequencing of ρ, class S^n is entered not later than S^{n+1}:

$$\tau_{\iota(n+1)}^{\nu(n+1)} \geq \tau_{\iota(n)}^{\nu(n)} \quad \forall\ n \in [0, N-1]$$

■

Proof: - only if:

Ab absurdo, assume that $\boldsymbol{\tau}$ is not a feasible timing for the trace ρ. Since all transitions are newly-enabled within the execution of ρ, Eq.(13-*(a)*) guarantees that the time-to-fire taken by any transition t_i^n falls within the firing interval $[EFT(t_i), LFT(t_i)]$ and can thus be accepted by the psTPN model. Thus, there must be some class S^z, encountered along ρ, where the transition that comes to fire is not the transition $t_a^{n_a}$ expected according to ρ, but some other transition $t_b^{n_b}$. This implies that $\tau_b^{n_b} < \tau_a^{n_a}$.

If $t_b^{n_b}$ is expected to come to fire along ρ in some class encountered after S^z, then by transitive closure of Eq.(13-*(c)*) it must be $\tau_b^{n_b} \geq \tau_a^{n_a}$, which contradicts with $\tau_b^{n_b} < \tau_a^{n_a}$. Otherwise, if $t_b^{n_b}$ is not expected to come to fire along ρ, then by transitive closure of Eq.(13-*(c)*) a transition $t_{\gamma(b,n_b)}^{\delta(b,n_b)}$ must exist whose firing disables $t_b^{n_b}$. This implies that $\tau_{\gamma(b,n_b)}^{\delta(b,n_b)} \geq \tau_a^{n_a}$ and $\tau_b^{n_b} \geq \tau_{\gamma(b,n_b)}^{\delta(b,n_b)}$ (for Eq.(13-*(b)*)). For transitivity between the two equations, we obtain the absurd: $\tau_b^{n_b} \geq \tau_a^{n_a}$. ■

The result of Theorem 1 was originally proven in [38], with a different argument and under a partially different formulation. With respect to that formulation, the global set of Eq.(13) includes in the set of unknown values also firing-times of transition activations that are enabled but not fired along ρ, and concentrates in a single inequality the constraint introduced by the EFT and the LFT of each enabled transition. This provides the basis that permits to associate the set of timings with the measure of probability induced by the distributions of non-controllable transitions.

The global set $D_{\boldsymbol{\tau}}$ has null measure iff there is a deterministic transition t_i (i.e., a transition such that $EFT(t_i) = LFT(t_i)$), that is activated along ρ in some class S^n. By applying the variable substitution $\tau_i^n = \tau_{\iota(n)}^{\nu(n)} + EFT(t_i)$, the deterministic firing-time τ_i^n can be removed from the set of variables in $D_{\boldsymbol{\tau}}$. If also $\tau_{\iota(n)}^{\nu(n)}$ is deterministic, the substitution process is iterated until encountering an enabling activation that either has a non-pointlike firing-time or is the fictitious activation $t_{\iota(0)}^{\nu(0)}$. According to this, in the subsequent treatment, we assume that non-controllable activations enabled along ρ have firing-times distributed over a non-pointlike support, and that the global set $D_{\boldsymbol{\tau}}$ has non-null measure.

C. Probability of conclusive execution

We evaluate here the probability $P_{OK}(\boldsymbol{r})$ that the model runs along ρ when the vector of firing-times of controllable activations is set equal to $\boldsymbol{r}$. To this end, we assume the following notational conventions:

- $\boldsymbol{\theta}^{nc}$ is the random vector of times-to-fire of non-controllable activations, and $\boldsymbol{y}$ is a valuation for $\boldsymbol{\theta}^{nc}$;
- $\boldsymbol{\tau}^{nc}$ is the random vector of (absolute virtual) firing-times of non-controllable activations, and $\boldsymbol{s}$ is a valuation for $\boldsymbol{\tau}^{nc}$;
- $\boldsymbol{r}$ is a valuation for the vector of (absolute virtual) firing-times of controllable activations;
- $l(i,n)$ is the position of t_i^n in a serialized representation of the activations along ρ.

According to Theorem 1, $P_{OK}(\boldsymbol{r})$ is the probability that the vector of firing-times of controllable and non-controllable activations takes a value in the set $D_{\boldsymbol{\tau}}$ of solutions of the global set of Eq.(13):

$$P_{OK}(\boldsymbol{r}) = Prob\{\langle \boldsymbol{r}, \boldsymbol{\tau}^{nc}(\boldsymbol{r}, \boldsymbol{\theta}^{nc}) \rangle \in D_{\boldsymbol{\tau}}\} \tag{14}$$

where: $\boldsymbol{\tau}^{nc}(\boldsymbol{r}, \boldsymbol{\theta}^{nc})$ is the random vector of firing-times of non-controllable transitions when firing-times of controllable transitions are set equal to $\boldsymbol{r}$ and times-to-fire of non-controllable transitions are the random vector $\boldsymbol{\theta}^{nc}$.

If $f_{\boldsymbol{\theta}^{nc}}(\boldsymbol{y})$ and $\boldsymbol{I}^{nc}$ are the Probability Density Function and the support of $\boldsymbol{\theta}^{nc}$, respectively, then, according to the Law of Total Probability, $P_{OK}(\boldsymbol{r})$ can be expressed as:

$$P_{OK}(\boldsymbol{r}) = \int_{\boldsymbol{I}^{nc}} In_{D_{\boldsymbol{\tau}}}(\boldsymbol{r}, \boldsymbol{\tau}^{nc}(\boldsymbol{r}, \boldsymbol{y})) f_{\boldsymbol{\theta}^{nc}}(\boldsymbol{y}) d\boldsymbol{y} \tag{15}$$

where $In_{D_{\boldsymbol{\tau}}}$ is the indicator function of the set $D_{\boldsymbol{\tau}}$:

$$In_{D_{\boldsymbol{\tau}}}(\boldsymbol{r},\boldsymbol{s}) = \begin{cases} 1 & \text{if } \langle \boldsymbol{r},\boldsymbol{s}\rangle \in D_{\boldsymbol{\tau}} \\ 0 & else \end{cases} \tag{16}$$

Times-to-fire of non-controllable transitions activated along ρ are Independent Random Variables, each distributed according to its respective static density function and supported within its static firing interval. According to this, $\boldsymbol{I}^{nc}$ is the Cartesian product of static firing intervals of non-controllable transitions activated along ρ, and $f_{\boldsymbol{\theta}^{nc}}(\boldsymbol{y})$ can be expressed in product-form as:

$$f_{\boldsymbol{\theta}^{nc}}(\boldsymbol{y}) = \prod_{\forall\ t_i^n\ .\ t_i \in T^{nc}} f_{t_i}(y_{l(i,n)}) \tag{17}$$

Given a vector of firing-times $\boldsymbol{r}$ for controllable activations, the relation between firing-times $\boldsymbol{\tau}^{nc}$ and times-to-fire $\boldsymbol{\theta}^{nc}$ of non-controllable activations can be made explicit as:

$$\begin{array}{ll} \tau_i^n - \tau_{\iota(n)}^{\nu(n)} = \theta_i^n & \forall\ t_i^n \text{ such that } t_i^n \in T^{nc} \wedge t_{\iota(n)}^{\nu(n)} \in T^{nc} \\ \tau_i^n - r_{l(\iota(n),\nu(n))} = \theta_i^n & \forall\ t_i^n \text{ such that } t_i^n \in T^{nc} \wedge t_{\iota(n)}^{\nu(n)} \in T^{c} \end{array} \tag{18}$$

The function $\boldsymbol{\tau}^{nc}(\boldsymbol{r},\boldsymbol{\theta}^{nc})$ is thus a linear transformation that can be expressed as:

$$\begin{array}{l} \boldsymbol{\theta}^{nc} = \mathbf{G}\cdot\boldsymbol{\tau}^{nc} - \boldsymbol{H}(\boldsymbol{r}) \\ \boldsymbol{\tau}^{nc} = \mathbf{G}^{-1}\cdot(\boldsymbol{\theta}^{nc} + \boldsymbol{H}(\boldsymbol{r})) \end{array} \tag{19}$$

where: $\mathbf{H}(\boldsymbol{r})$ is a column vector of size equal to the number N^{nc} of non-controllable activations, whose elements are either null or elements of vector $\boldsymbol{r}$; and, $\mathbf{G}$ is a square matrix of size $N^{nc} \times N^{nc}$ that satisfies the following properties: all the elements on the diagonal are equal to 1 (i.e., $G_{ii} = 1$); on each row, there is at most one extra-diagonal non-null element, and in the case this is equal to -1 (i.e., $\forall\ i,\ \text{if}\ \exists\ j \neq i\ .\ G_{ij} \neq 0$ then $G_{ij} = -1\ \wedge\ \forall\ h \neq i, j\ G_{ih} = 0$). Moreover, if firing-times are encoded so as to follow the order of activations (i.e., $l(i,n) \geq l(j,m)$ iff $\tau_{\iota(n)}^{\nu(n)} \geq \tau_{\iota(j)}^{\nu(m)}$), then all non-null elements are in the left-diagonal part of $\mathbf{G}$. According to these properties, it is easily proven that:

$$det(\mathbf{G}) = 1 \tag{20}$$

By applying the variable substitution $\boldsymbol{y} = \mathbf{G}\cdot\boldsymbol{s} - \boldsymbol{H}(\boldsymbol{r})$, the integral of Eq.(15) can thus be rewritten as:

$$\begin{array}{ll} P_{OK}(\boldsymbol{r}) = & \displaystyle\int_{\mathbf{G}^{-1}\cdot(\boldsymbol{I}^{nc}+\boldsymbol{H}(r))} In_{D_{\boldsymbol{\tau}}}(\boldsymbol{r},\boldsymbol{s}) \\ & \cdot f_{\boldsymbol{\theta}^{nc}}(\mathbf{G}\cdot\boldsymbol{s} - \boldsymbol{H}(\boldsymbol{r}))\cdot det(\mathbf{G})d\boldsymbol{s} \end{array} \tag{21}$$

where $\mathbf{G}^{-1}\cdot(\boldsymbol{I}^{nc} + \boldsymbol{H}(r))$ is the image of the set $\boldsymbol{I}^{nc}$ through the transformation $\boldsymbol{\theta}^{nc} \rightarrow \mathbf{G}^{-1}\cdot(\boldsymbol{\theta}^{nc} + \boldsymbol{H}(\boldsymbol{r}))$. This image is the set of firing-times of non-controllable activations that are covered when controllable firing-times are set equal to $\boldsymbol{r}$ and non-controllable times-to-fire range within their respective firing intervals.

Besides, the indicator function $In_{D_{\boldsymbol{\tau}}}(\boldsymbol{r},\boldsymbol{s})$ is equal to 1 iff $\boldsymbol{s} \in D_{\boldsymbol{\tau}}^{\boldsymbol{r}}$ and it equals 0 elsewhere, $D_{\boldsymbol{\tau}}^{\boldsymbol{r}}$ being the projection of $D_{\boldsymbol{\tau}}$ on the space of firing times of non-controllable activations:

$$D_{\boldsymbol{\tau}}^{\boldsymbol{r}} = \{\boldsymbol{s} | \langle \boldsymbol{r},\boldsymbol{s}\rangle \in D_{\tau}\} \tag{22}$$

According to this, $D_{\boldsymbol{\tau}}^{\boldsymbol{r}} \subseteq \mathbf{G}^{-1}\cdot(\boldsymbol{I}^{nc} + \boldsymbol{H}(\boldsymbol{r}))$, and we can rewrite the integral of Eq.(21) as:

$$P_{OK}(\boldsymbol{r}) = \int_{D_{\boldsymbol{\tau}}^{\boldsymbol{r}}} f_{\boldsymbol{\theta}^{nc}}(\mathbf{G}\cdot\boldsymbol{s} - \boldsymbol{H}(\boldsymbol{r}))d\boldsymbol{s} \tag{23}$$

where, according to Eq.(17):

$$\begin{array}{l} f_{\boldsymbol{\theta}^{nc}}(\mathbf{G}\cdot\boldsymbol{s} - \boldsymbol{H}(\boldsymbol{r})) \\ = \displaystyle\prod_{t_i^n \in T^{nc}.\ t_{\iota(n)}^{\nu(n)} \in T^{nc}} f_{t_i}(s_{l(i,n)} - s_{l(\iota(n),\nu(n))}) \\ \cdot \displaystyle\prod_{t_i^n \in T^{nc}.\ t_{\iota(n)}^{\nu(n)} \in T^{c}} f_{t_i}(s_{l(i,n)} - r_{l(\iota(n),\nu(n))}) \end{array} \tag{24}$$

and the domain of $P_{OK}(\boldsymbol{r})$ is the projection $D_{\boldsymbol{\tau}}^{\boldsymbol{s}}$ of the set $D_{\boldsymbol{\tau}}$ on the space of firing-times of controllable activations:

$$D_{\boldsymbol{\tau}}^{\boldsymbol{s}} = \{\boldsymbol{r} | \langle \boldsymbol{r},\boldsymbol{s}\rangle \in D_{\boldsymbol{\tau}}\}. \tag{25}$$

The following result guarantees that P_{OK} is continuous, which may take relevance in subsequent optimization steps:

Theorem 2: Function P_{OK} is continuous over $D_{\boldsymbol{\tau}}^{\boldsymbol{s}}$, provided that static density functions in the psTPN model are continuous in their respective supports.

Proof: Let $\boldsymbol{r} + \boldsymbol{\epsilon}$ denote the vector obtained by adding a quantity ϵ to every component of $\boldsymbol{r}$. To prove that P_{OK} is continuous, we show that $\lim_{\boldsymbol{\epsilon}\rightarrow\mathbf{0}} P_{OK}(\boldsymbol{r}+\boldsymbol{\epsilon}) - P_{OK}(\boldsymbol{r}) = 0$. According to Eq.(23), the limit can be written as:

$$\begin{array}{l} \displaystyle\lim_{\boldsymbol{\epsilon}\rightarrow\mathbf{0}} P_{OK}(\boldsymbol{r}+\boldsymbol{\epsilon}) - P_{OK}(\boldsymbol{r}) \\ = \displaystyle\lim_{\boldsymbol{\epsilon}\rightarrow\mathbf{0}} \int_{D_{\boldsymbol{\tau}}^{\boldsymbol{r}+\boldsymbol{\epsilon}}} f_{\boldsymbol{\theta}^{nc}}(\mathbf{G}\cdot\boldsymbol{s} - \boldsymbol{H}(\boldsymbol{r}+\boldsymbol{\epsilon}))d\boldsymbol{s} \\ - \displaystyle\int_{D_{\boldsymbol{\tau}}^{\boldsymbol{r}}} f_{\boldsymbol{\theta}^{nc}}(\mathbf{G}\cdot\boldsymbol{s} - \boldsymbol{H}(\boldsymbol{r}))d\boldsymbol{s}. \end{array} \tag{26}$$

Evaluation of the integral of Eq.(26) can be performed through subsequent marginalizations over non-controllable variables. Without loss of generality, we assume that $\boldsymbol{s} = \langle s_0\rangle$ is made of a single non-controllable variable. According to this:

$$\begin{array}{l} \displaystyle\lim_{\boldsymbol{\epsilon}\rightarrow\mathbf{0}} P_{OK}(\boldsymbol{r}+\boldsymbol{\epsilon}) - P_{OK}(\boldsymbol{r}) \\ = \displaystyle\lim_{\boldsymbol{\epsilon}\rightarrow\mathbf{0}} \int_{D_{\boldsymbol{\tau}}^{\boldsymbol{r}+\boldsymbol{\epsilon}}} f_{\boldsymbol{\theta}^{nc}}(\mathbf{G}\cdot s_0 - \boldsymbol{H}(\boldsymbol{r}+\boldsymbol{\epsilon}))ds_0 \\ - \displaystyle\int_{D_{\boldsymbol{\tau}}^{\boldsymbol{r}}} f_{\boldsymbol{\theta}^{nc}}(\mathbf{G}\cdot s_0 - \boldsymbol{H}(\boldsymbol{r}))ds_0 \\ = \displaystyle\lim_{\boldsymbol{\epsilon}\rightarrow\mathbf{0}} \int_{E_s(\boldsymbol{r}+\boldsymbol{\epsilon})}^{L_s(\boldsymbol{r}+\boldsymbol{\epsilon})} f_{\boldsymbol{\theta}^{nc}}(\mathbf{G}\cdot s_0 - \boldsymbol{H}(\boldsymbol{r}+\boldsymbol{\epsilon}))ds_0 \\ - \displaystyle\int_{E_s(\boldsymbol{r})}^{L_s(\boldsymbol{r})} f_{\boldsymbol{\theta}^{nc}}(\mathbf{G}\cdot s_0 - \boldsymbol{H}(\boldsymbol{r}))ds_0 \\ = \displaystyle\lim_{\boldsymbol{\epsilon}\rightarrow\mathbf{0}} F_{\boldsymbol{\theta}^{nc}}(\mathbf{G}\cdot L_s(\boldsymbol{r}+\boldsymbol{\epsilon}) \\ -\boldsymbol{H}(\boldsymbol{r}+\boldsymbol{\epsilon})) - F_{\boldsymbol{\theta}^{nc}}(\mathbf{G}\cdot E_s(\boldsymbol{r}+\boldsymbol{\epsilon}) - \boldsymbol{H}(\boldsymbol{r}+\boldsymbol{\epsilon})) \\ -F_{\boldsymbol{\theta}^{nc}}(\mathbf{G}\cdot L_s(\boldsymbol{r}) - \boldsymbol{H}(\boldsymbol{r})) - F_{\boldsymbol{\theta}^{nc}}(\mathbf{G}\cdot E_s(\boldsymbol{r}) - \boldsymbol{H}(\boldsymbol{r})) \end{array} \tag{27}$$

where $E_s(\boldsymbol{r})$ and $L_s(\boldsymbol{r})$ denote the minimum and the maximum values of s_0 such that $\langle \boldsymbol{r}, s_0\rangle \in D_{\boldsymbol{\tau}}$, and $F_{\boldsymbol{\theta}^{nc}}$ denotes the integral function of $f_{\boldsymbol{\theta}^{nc}}$. Function $f_{\boldsymbol{\theta}^{nc}}$ is continuous since it is defined as the product of static density functions associated with transitions in the model, which are continuous by hypothesis. Therefore, $F_{\boldsymbol{\theta}^{nc}}$ is continuous. Since integration bounds $E_s(\boldsymbol{r})$ and $L_s(\boldsymbol{r})$ are also continuous functions [8], the limit of Eq.(27) turns out to be equal to 0. ■

D. An illustrative example

We illustrate the steps of the derivation on the small example shown in Fig.2. Transitions t_1 and t_2 model two concurrent activities α_1 and α_2, whose Execution Times have expolynomial density functions $f_{t_1}(t) = 0.25\, t$ over $[1, 3]$ and $f_{t_2}(t) = \frac{e^4}{e^2-1} \cdot e^{-t}$ over $[2, 4]$, respectively. We assume that the test driver is able to delay the beginning of activity α_1 of a controllable offset in the range $[1, 3]$, which is accounted by transition t_0. The objective of the analysis is the identification of firing-times for t_0 that maximize the probability of execution of the symbolic run $\rho = t_0 \rightarrow t_2 \rightarrow t_1$.

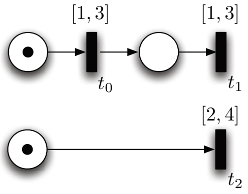


Figure 2. The psTPN model of two concurrent activities.

The space $D_{\boldsymbol{\tau}}$ of solutions of the global set of Eq.(13) takes the following form:

$$D_{\boldsymbol{\tau}} = \begin{cases} 1 \leq r_{l(0,0)} \leq 3 \\ 1 \leq s_{l(1,1)} - r_{l(0,0)} \leq 3 \\ 2 \leq s_{l(2,0)} \leq 4 \\ r_{l(0,0)} - s_{l(2,0)} \leq 0 \\ s_{l(2,0)} - s_{l(1,1)} \leq 0, \end{cases} \tag{28}$$

where $r_{l(0,0)}$, $s_{l(1,1)}$, and $s_{l(2,0)}$ are valuations of firing-times of transitions t_0, t_1, and t_2, respectively, and $s_{l(\iota(1),\nu(1))}$ has been made explicit and expressed as $r_{l(0,0)}$. The normal form of $D_{\boldsymbol{\tau}}$ turns out to be:

$$D_{\boldsymbol{\tau}} = \begin{cases} 1 \leq r_{l(0,0)} \leq 3 \\ 2 \leq s_{l(1,1)} \leq 6 \\ 2 \leq s_{l(2,0)} \leq 4 \\ 1 \leq s_{l(1,1)} - r_{l(0,0)} \leq 3 \\ -3 \leq r_{l(0,0)} - s_{l(2,0)} \leq 0 \\ -3 \leq s_{l(2,0)} - s_{l(1,1)} \leq 0. \end{cases} \tag{29}$$

The projection $D_{\boldsymbol{\tau}}^{\boldsymbol{r}}$ of $D_{\boldsymbol{\tau}}$ on the space of firing-times of non-controllable activations t_1^1 and t_2^0 is obtained from the normal form of $D_{\boldsymbol{\tau}}$ by disregarding constraints that involve valuations of controllable activation t_0^0:

$$D_{\boldsymbol{\tau}}^{\boldsymbol{r}} = \begin{cases} 2 \leq s_{l(1,1)} \leq 6 \\ 2 \leq s_{l(2,0)} \leq 4 \\ 0 \leq s_{l(1,1)} - s_{l(2,0)} \leq 3. \end{cases} \tag{30}$$

According to Eq.(17), function $f_{\boldsymbol{\theta}^{nc}}(\boldsymbol{y})$ is expressed as:

$$f_{\boldsymbol{\theta}^{nc}}(\boldsymbol{y}) = f_{t_1}(y_{l(1,1)}) \cdot f_{t_2}(y_{l(2,0)}) = 0.25 \cdot y_{l(1,1)} \cdot \frac{e^4}{e^2-1} \cdot e^{-y_{l(2,0)}} \tag{31}$$

The relation between firings-times $\boldsymbol{\tau}^{nc} = \langle \tau_1^1, \tau_2^0 \rangle$ and times-to-fire $\boldsymbol{\theta}^{nc} = \langle \theta_1^1, \theta_2^0 \rangle$ of non-controllable activations t_1^1 and t_2^0 is:

$$\begin{aligned} \theta_1^1 = &\ \tau_1^1 - r_0^0 \\ \theta_2^0 = &\ \tau_2^0, \end{aligned} \tag{32}$$

which is expressed in vectorial form as:

$$\langle \theta_1^1, \theta_2^0 \rangle = \mathbf{G} \cdot \langle \tau_1^1, \tau_2^0 \rangle - \mathbf{H}(\boldsymbol{r}) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \langle \tau_1^1, \tau_2^0 \rangle - \begin{bmatrix} r_{l(0,0)} \\ 0 \end{bmatrix}. \tag{33}$$

According to Eqs.(23, 24), function $f_{\boldsymbol{\theta}^{nc}}(\mathbf{G} \cdot \boldsymbol{s} - \mathbf{H}(\boldsymbol{r}))$ is expressed as:

$$\begin{aligned} & f_{\boldsymbol{\theta}^{nc}}(\mathbf{G} \cdot \boldsymbol{s} - \mathbf{H}(\boldsymbol{r})) = f_{\boldsymbol{\theta}^{nc}}(s_{l(1,1)} - r_{l(0,0)}, s_{l(2,0)}) \\ & = f_{t_1}(s_{l(1,1)} - r_{l(0,0)}) \cdot f_{t_2}(s_{l(2,0)}) \\ & = 0.25 \cdot (s_{l(1,1)} - r_{l(0,0)}) \cdot \frac{e^4}{e^2-1} \cdot e^{-s_{l(2,0)}} \end{aligned} \tag{34}$$

and the probability $P_{OK}(\boldsymbol{r})$ of conclusive execution is finally written as:

$$\begin{aligned} & P_{OK}(\boldsymbol{r}) = P_{OK}(r_{l(0,0)}) \\ & = \begin{cases} 1.632121 \cdot 10^{-33}\ r_{l(0,0)}\ e^{-r_{l(0,0)}} + 6.973703\ e^{-r_{l(0,0)}} \\ +0.019565\ (r_{l(0,0)})^2 - 0.195647\ r_{l(0,0)} + 0.332600 \\ \qquad\qquad\qquad\qquad \text{if } r_{l(0,0)} \in [1,2] \\ \\ -3.678794 \cdot 10^{-34}\ r_{l(0,0)}\ e^{-r_{l(0,0)}} - 1.571870\ e^{-r_{l(0,0)}} \\ -0.195647\ r_{l(0,0)} + 1.489118 + 0.019565\ (r_{l(0,0)})^2 \\ \qquad\qquad\qquad\qquad \text{if } r_{l(0,0)} \in [2,3] \end{cases} \end{aligned} \tag{35}$$

As shown in Fig. 3, $P_{OK}(r_{l(0,0)})$ attains its maximum value 0.963353 with $r_{l(0,0)} = 2$.

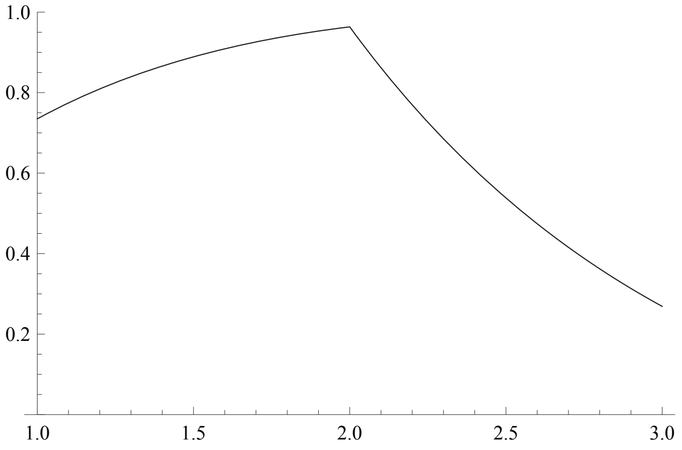


Figure 3. The plot of function $P_{OK}(r_{l(0,0)})$ representing the probability of execution of the sequence $\rho = t_0 \rightarrow t_2 \rightarrow t_1$ in the example of Fig. 2 as a function of the firing-time $r_{l(0,0)}$ of controllable transition t_0.

IV. Probabilistic input generation

Function $P_{OK}(\boldsymbol{r})$ can be applied in the generation of inputs that reduce the number of test repetitions needed to guarantee a conclusive test with a given probability. We report here experimental results obtained through an implementation of the theory, with the twofold aim of exemplifying the applicability of the proposed technique in the engineering of Real-Time SW, and of highlighting the practical impact of dominating factors of complexity.

A. Symbolic and numerical calculus implementation

The theory of Sect.III-C has been implemented as a Java component of the Sirio Framework [10], [11], [6], which supports enumeration of the SCG of a psTPN model, manipulation of DBM domains [15], [38], and symbolic representation of expolynomial functions. The new component allows construction of domain $D_{\boldsymbol{\tau}}$ and evaluation of function $P_{OK}(\boldsymbol{r})$ for any given symbolic run ρ; moreover, it exploits JLink library [41] to compose a chain tool with Wolfram Mathematica [42].

The implementation allows two different approaches to evaluation and maximization of $P_{OK}(\boldsymbol{r})$.

Symbolic evaluation of $P_{OK}(\boldsymbol{r})$: In principle, function $P_{OK}(\boldsymbol{r})$ can be derived by evaluating the integral of Eq.(23) in symbolic form through the Sirio Framework, as the integrand function $f_{\boldsymbol{\theta}^{nc}}()$ is an expolynomial supported over a DBM domain [8]. The integration is performed by repeatedly marginalizing $f_{\boldsymbol{\theta}^{nc}}()$ over non-controllable variables through Fourier-Motzkin elimination [14]. Function $P_{OK}(\boldsymbol{r})$ can then be maximized through a numerical optimization algorithm in order to obtain the values of firing-times of controllable activations that maximize the probability of conclusive test execution:

$$P_{max} = \max_{\boldsymbol{r} \in D^{\boldsymbol{s}}_{\boldsymbol{\tau}}} P_{OK}(\boldsymbol{r}). \tag{36}$$

Numerical evaluation of $P_{OK}(\boldsymbol{r})$: The integral of Eq.(23) can be numerically evaluated for a finite set of valuations of the vector $\boldsymbol{r}$ of firing-times of controllable activations, assuming the maximum obtained value as the maximum probability of conclusive test execution. In our implementation, domain $D^{\boldsymbol{s}}_{\boldsymbol{\tau}}$ is covered through a regular grid that takes an equal number of samples for each controllable variable, applying a light uniform perturbation to each sample to reduce border effects. The numerical integral is evaluated on a compact factored form of function $f_{\boldsymbol{\theta}^{nc}}()$ through Wolfram Mathematica [42] using Genz-Malik algorithm [4]. This belongs to the class of Globally Adaptive Algorithms, which estimate an integral through the sum of integral estimates evaluated over a partition of the integration domain in disjoint subregions, according to the following scheme: *1)* choose a subregion and partition it in subregions; *2)* apply an integration rule to the subregions obtained at step 1; *3)* update the global integral and error estimates; *4)* check some convergence criterion and go back to step 1 if it is not satisfied.

The symbolic and numerical approaches have different capabilities and limits. In the functional perspective, symbolic evaluation provides the exact analytic form of function $P_{OK}(\boldsymbol{r})$, and thus enables further symbolic derivations (e.g., in the calculus of derivatives that might be useful in the optimization step). Whereas, the numerical evaluation approach computes $P_{OK}(\boldsymbol{r})$ for selected values $\boldsymbol{r}$ of firing-times of controllable activations (in this case, derivatives could still be evaluated but only through numerical differentiation). The same optimization technique could then be chained to both approaches and run on the same samples of $P_{OK}(\boldsymbol{r})$ to determine the values $\boldsymbol{r}$ that maximize $P_{OK}(\boldsymbol{r})$.

In the performance perspective, symbolic evaluation is much more fragile with respect to the curse of dimensionality, which impacts in particular on the evaluation of repeated projections of $f_{\boldsymbol{\theta}^{nc}}()$ that eliminate non-controllable variables, producing an incremental partitioning of the DBM integration domain in the piecewise representation of the integrand function. Specifically, domain partitioning and polynomial degree grow with the length of the trace and with the concurrency degree of the model [8]. In the present practice, symbolic evaluation is also hurdled by the present implementation of the Sirio Framework [10], [11], [6] which is not efficient in the representation of product forms.

Numerical evaluation avoids the curse of dimensionality of symbolic integration by computing samples of $P_{OK}(\boldsymbol{r})$ through numerical integration. The approach is still sensitive to the number of computed samples, but this can be kept under control, either by trading accuracy for efficiency or by selecting the samples to be evaluated through some higher level algorithm. Computational complexity of the numerical evaluation approach is also affected by the number of expolynomial terms of the integrand function $f_{\boldsymbol{\theta}^{nc}}()$, which is in the order of $O((m \cdot k)^{N^{nc}})$, where m and k are the maximum number of expolynomial terms and the maximum polynomial degree, respectively, of the static density function of any transition enabled along ρ, and N^{nc} is the number of non-controllable activations enabled along ρ. To cope with the problem, our implementation avoids the expansion of expolynomial terms of $f_{\boldsymbol{\theta}^{nc}}()$ before computing the numerical integral through Wolfram Mathematica. The approach also slightly increases the effort of Genz-Malik algorithm to converge to the exact integral value with an acceptable accuracy, since the algorithm is natively defined for hyperrectangular integration domains, but is here run on DBM integration domains. In the experiments reported in the following Section, N^{nc} reaches values in the order of 20, while k and m equals 5 and 3, respectively, for some function terms. A not adequately factorized representation may reach a number of terms in the order of 10^{18}. This visibly impacts not only on the space occupation of the representation, but also on the number of mathematical operations, and consequently on time complexity and accuracy of its evaluation.

B. Probabilistic strategies for input generation

Function $P_{OK}(\boldsymbol{r})$ and its points of maximum can be used to generate inputs in a sequence of N independent experiments so as to obtain a controlled probability of a conclusive execution of the test. Different test repetition schemes can be defined, attaining different trade-offs between the computational effort in the off-line test planning and the success probability $P(N)$ that at least one of the N repetitions yields a conclusive execution of the test. In particular, we focused the experimentation on three strategies, which we refer to as *optimal sampling*, *proportional sampling* and *uniform sampling*:

- in the *optimal sampling* scheme all experiments are performed with the same optimal set of parameters; since experiments are independent, this yields a geometric distribution for the success probability:

$$P_{opt}(N) = 1 - (1 - P_{max})^N \tag{37}$$

- in the *proportional sampling* scheme, controllable timers are sampled according to a randomization probability density function $h : D^{\boldsymbol{s}}_{\boldsymbol{\tau}} \to \mathbb{R} \cap [0, 1]$ with $\int_{D^{\boldsymbol{s}}_{\boldsymbol{\tau}}} h(\boldsymbol{r})d\boldsymbol{r} = 1$. In this case, the success probability takes the form:

$$\begin{aligned} &P_{prop}(N) = 1 - \left(\int (1 - P_{OK}(\boldsymbol{r}))\, h(\boldsymbol{r})d\boldsymbol{r}\right)^N \\ &= 1 - \left(1 - \int P_{OK}(\boldsymbol{r})h(\boldsymbol{r})d\boldsymbol{r}\right)^N \end{aligned} \tag{38}$$

  In general, $h(\boldsymbol{r})$ can be conveniently defined so as to give higher probability to the values of controllable timers that yield a higher probability of conclusive execution of the test. In particular, this can be achieved by assuming:

$$h^k(\boldsymbol{r}) = \frac{P_{OK}(\boldsymbol{r})^k}{\int P_{OK}(\boldsymbol{r})^k d\boldsymbol{r}}$$

  with $k \geq 1$. Note that, with $k = 1$, sampling will be proportional to the probability of conclusive execution; when $k \to \infty$, sampling tends to the optimal one. In general:

$$P_{prop_k}(N) = 1 - \left(1 - \frac{\int P_{OK}(\boldsymbol{r})^{k+1} d\boldsymbol{r}}{\int P_{OK}(\boldsymbol{r})^k d\boldsymbol{r}}\right)^N \tag{39}$$

- in the *uniform sampling* scheme, stochastic information is neglected and a uniform selection method is used, assuming $h(\boldsymbol{r})$ to be uniform over $D^{\boldsymbol{s}}_{\boldsymbol{\tau}}$. Following this approach, the distribution P_{OK} is not needed during the execution, but it will anyway condition the resulting success probability:

$$P_{unif}(N) = 1 - \left(\frac{1}{V(D^{\boldsymbol{s}}_{\boldsymbol{\tau}})} \cdot \int 1 - P_{OK}(\boldsymbol{r})d\boldsymbol{r}\right)^N \tag{40}$$

  where $V(D^{\boldsymbol{s}}_{\boldsymbol{\tau}})$ is the volume of $D^{\boldsymbol{s}}_{\boldsymbol{\tau}}$.

C. Experimentation on Real-Time task-sets

Implementation of symbolic and numerical approaches to input generation enables their concrete application to the development process of Real-Time SW. This becomes relevant in the context of Model Driven Development, where a formal specification of system requirements and design is translated into an IUT through a structured process [35], [18], [26]. In so doing, the specification turns out to be a valid abstraction of the IUT, allowing reliable generation of test inputs based on the model structure.

A class of real-time workloads: We consider the case of N concurrent *tasks* T_n with $n \in [1, N]$ which must be completed within a deadline P. Each task is a sequence of atomic *chunks*, and chunks belonging to different tasks may be subject to mutual exclusion constraints. We assume that temporal parameters can be regarded as independent variables: this is not always completely true, but comprises an assumption that is necessarily made in most quantitative approaches to Real-Time schedulability analysis. In our setting, this may result in a deviation between expected and actual statistics, which in turn may partially degradate the achieved performance.

Task	Jitter	Chunk	Exclusions	BCET	WCET	α	β
T_1	$[0, 10]$	C_{11}	-	5	10	5	3
		C_{12}	C_{22}	10	15	3	2
T_2	$[0, 10]$	C_{21}	-	5	20	6	3
		C_{22}	C_{12}	5	10	2	1

Table I
PARAMETERS OF A WORKLOAD MADE OF 2 TASKS.

The release time of each task is subject to a jitter delay depending on the availability of inputs external to the task-set, which we assume to be controllable during the testing stage.

The Execution Time of each chunk is guaranteed to range between a Best Case Execution Time (BCET, also denoted by B for short) and a Worst Case Execution Time (WCET, W for short), but its actual value is not controllable. We assume that its statistics is acquired through some measurement-based profiling technique [26], [40], and represented through some expolynomial fitting distribution f_{ET}. Without loss of generality, we consider here approximation through Gamma distributions, truncated over the Execution Time range and reduced by subtracting a linear function so as to obtain $f_{ET}(B) = 0$ and $f_{ET}(W) = 0$:

$$f_{ET}(t) = \begin{cases} c \cdot \left[\Gamma_{\alpha,\beta}(t) - \left(\frac{W-t}{W-B} \cdot \Gamma_{\alpha,\beta}(B) + \frac{B-t}{B-W} \cdot \Gamma_{\alpha,\beta}(W)\right)\right] \\ \hfill \text{if } t \in [B, W] \\ 0 \hfill \text{otherwise} \end{cases}$$

where

$$\Gamma_{\alpha,\beta}(t) = \frac{\beta^{-\alpha} \cdot e^{-\frac{t}{\beta}} \cdot t^{\alpha-1}}{(a-1)!}$$

and

$$\begin{aligned} c \;\;=\;\; & \left[\int_B^W \left(\Gamma_{\alpha,\beta}(t) - \left(\frac{W-t}{W-B} \cdot \Gamma_{\alpha,\beta}(B)\right.\right.\right. \\ + \;\; & \left.\left.\left.\frac{B-t}{B-W} \cdot \Gamma_{\alpha,\beta}(W)\right) dt\right)\right]^{-1} \end{aligned}$$

Following this scheme, the Execution Time of the m-th chunk of the n-th task, denoted by $C_{n,m}$, can be completely characterized through the four-tuple $\langle B_{n,m}, W_{n,m}, \alpha_{n,m}, \beta_{n,m}\rangle$.

Comparing implementation approaches: Table I specifies a case with 2 tasks, each made of two chunks. For each task, the Table reports the characterization of Jitter and, for each chunk, the chunks in the relation of mutual exclusion and the characterization of the Execution Time. Fig. 4 reports the plots of the four corresponding probability density functions, while the overall task-set is represented as a psTPN in Fig.5.

Without loss of generality, we are here interested in testing the symbolic run $\rho_2 = t_0 \to t_5 \to t_1 \to t_3 \to t_6 \to t_4 \to t_7 \to t_8$, which was identified through nondeterministic analysis [38] as one of the sequences that can attain the Worst Case Completion Time.

Both numerical and symbolic approaches to input generation turn out to be viable with the complexity of the 2-tasks model of the selected run ρ_2. In the symbolic approach, a DBM domain is initially computed for the 8 firing-times of the transitions enabled along ρ_2. The overall computation, including the enumeration of the SCG, takes less than 3 minutes on a 2GHz Dual Core processor. The function $P_{OK}(x_0^0, x_1^0)$

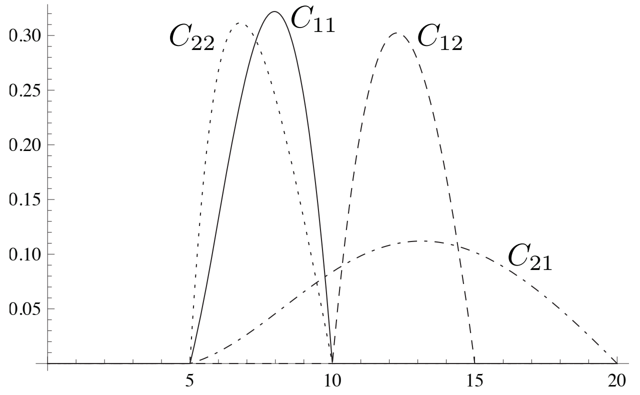


Figure 4. Plot of the Execution Time distributions for the 4 chunks in Table I.

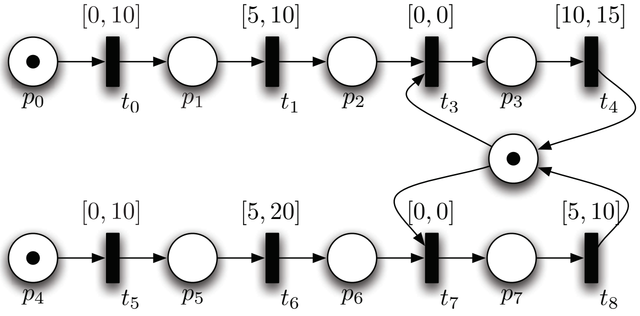


Figure 5. psTPN model of the task-set made of 2 tasks specified in Table I.

encoding the probability of conclusive test execution has a piecewise form over a partition in 2 DBM zones with 1319 and 1037 exp-monomial terms, which are not reported here for the sake of readibility but which are easily encoded and managed in the Sirio Framework [10], [11], [6]. Fig.6 displays the plot of the function and the expression of the two zones that partition its support.

The maximum of $P_{OK}(x_0^0, x_5^0)$ is evaluated through Wolfram Mathematica [42] using the Nelder-Mead [31] technique in about 0.5 seconds, yielding a maximum success probability $P_{max} = 0.147092$ attained at $\boldsymbol{r}_{max} = \langle 0, 1.37 \rangle$. Approximate evaluation through the numerical approach would yield slightly different values for $\boldsymbol{r}_{max}$ with no substantial changes for P_{max}: with 5 samples for each controllable variable, $P_{max}^5 = 0.14448$, attained at $\boldsymbol{r}_{max}^5 = \langle 0, 2.50 \rangle$; with 10 samples for each controllable variable, $P_{max}^{10} = 0.146943$, attained at $\boldsymbol{r}_{max}^{10} = \langle 0, 1.11 \rangle$.

Fig.7-(a) plots success probability $P(N)$ as a function of the number N of repetitions of the test, with the three test repetition schemes of Sect. IV-B. The Figure also plots success probability of a basic random testing approach (thick dotted line), where each transition of the model is sampled according to its own distribution.

As expected, the optimal sampling strategy performs far better than uniform sampling. Proportional sampling shows a good performance, which can be further improved by incrementing the parameter k (Fig.7-(a) reports results for $k = 2$, 5, 10 with thin dashed lines). Table II highlights the gain of the proposed approach, reporting for each strategy the minimum number of test repetitions needed to attain a given probability of conclusive test.

Fig. 7-(b) compares optimal and proportional sampling with the approximate approach of [43]. To this end, we consider the case where non-controllable transitions in the IUT have the GEN distribution specified in the model of Fig.5, but test inputs are generated assuming that they are uniformly distributed over their respective supports. This scenario can concretely occur when distributions of Execution Times are not known, and the assumption of maximum entropy is advocated to assume uniform distributions [3], or it can also be an intentional simplification aimed at partially reducing the complexity of derivation of $P_{OK}()$. In this case, while $P_{OK}()$ is still distributed as in the exact analysis (and in the plot of Fig.6), input values are derived as the optimum of a deviated function $\tilde{P}_{OK}()$. Note that the performance that is obtained through an optimal sampling of $\tilde{P}_{OK}()$ is lower than that obtained through a proportional sampling. The difference is actually not impressive in the quantitative perspective, but significant in the qualitative one: this comprises a neat example of how the proportional sampling strategy tends to mitigate degradation of the success probability deriving from inaccurate estimation of Execution Time distributions.

Stressing complexity: Tables III and IV describe two further examples with 4 and 6 tasks, respectively, and Fig.8 reports their psTPN representation. In these cases, experimentation was focused on the generation of inputs that sensitize the two symbolic runs $\rho_4 = t_{15} \rightarrow t_{16} \rightarrow t_5 \rightarrow t_9 \rightarrow t_{17} \rightarrow t_0 \rightarrow t_1 \rightarrow t_3 \rightarrow t_4 \rightarrow t_6 \rightarrow t_7 \rightarrow t_{10} \rightarrow t_8$ (13 events, 4 of which are controllable) for the 4-tasks model and $\rho_6 = t_{15} \rightarrow t_{16} \rightarrow t_{18} \rightarrow t_{19} \rightarrow t_5 \rightarrow t_{24} \rightarrow t_{25} \rightarrow t_9 \rightarrow t_{17} \rightarrow t_0 \rightarrow t_{20} \rightarrow t_{26} \rightarrow t_1 \rightarrow t_3 \rightarrow t_{21} \rightarrow t_{22} \rightarrow t_4 \rightarrow t_6 \rightarrow t_7 \rightarrow t_{10} \rightarrow t_{23} \rightarrow t_{27} \rightarrow t_{13}$ (23 events, 6 of which are controllable) for the 6-tasks model.

Both the task-sets and the target test cases are expressly designed so as to stress the factors of complexity of the proposed approach, beyond the limits of a reasonable test case planning. In particular, the two selected runs are identified so as to exercise all the chunks of the entire task-set, which in the practice would be more conveniently covered through a suite of separate shorter tests; moreover, this is attained by combining control over 4 and 6 parameters, while a subset of them would be sufficient.

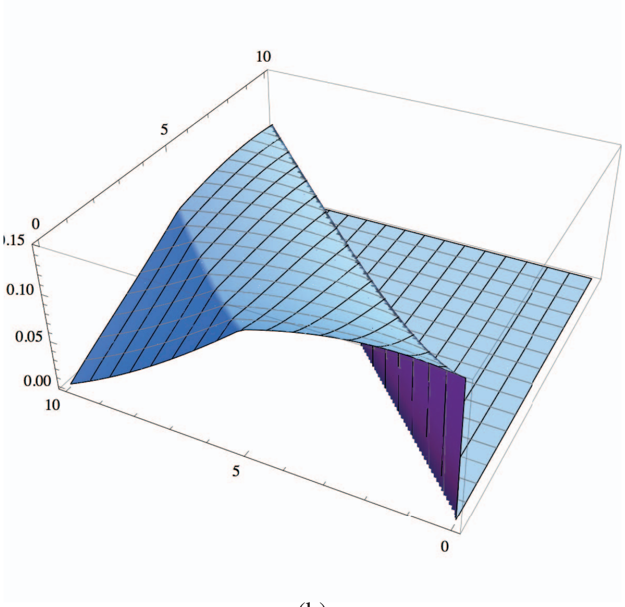
On both the 4-tasks and 6-tasks cases, the symbolic approach is not viable for our present implementation. Whereas, the numerical procedure with 5 samples per variable finds the best input values in 20 seconds for the 4-tasks example ($P_{max} = 2.063845 \cdot 10^{-5}$ attained at $\boldsymbol{r}_{max}^5 = \langle 8.68, 7.31, 7.74, 1.27 \rangle$) and in 6 minutes for the 6-tasks case ($P_{max} = 6.971576 \cdot 10^{-8}$ attained at $\boldsymbol{r}_{max}^5 = \langle 10, 6.57, 9.98, 0.12, 3.29, 6.70 \rangle$).

In the 4-tasks model, the assumption of optimized inputs actually makes the difference in making the success feasible: if we assume that durations are in microseconds (which could

$$D_1 \quad = \quad \left\{ \begin{array}{rcl} 0 \leq & x_0^0 & \leq 5 \\ 5 \leq & x_5^0 & \leq 10 \\ -10 \leq & x_0^0 - x_5^0 & \leq -5 \end{array} \right.$$

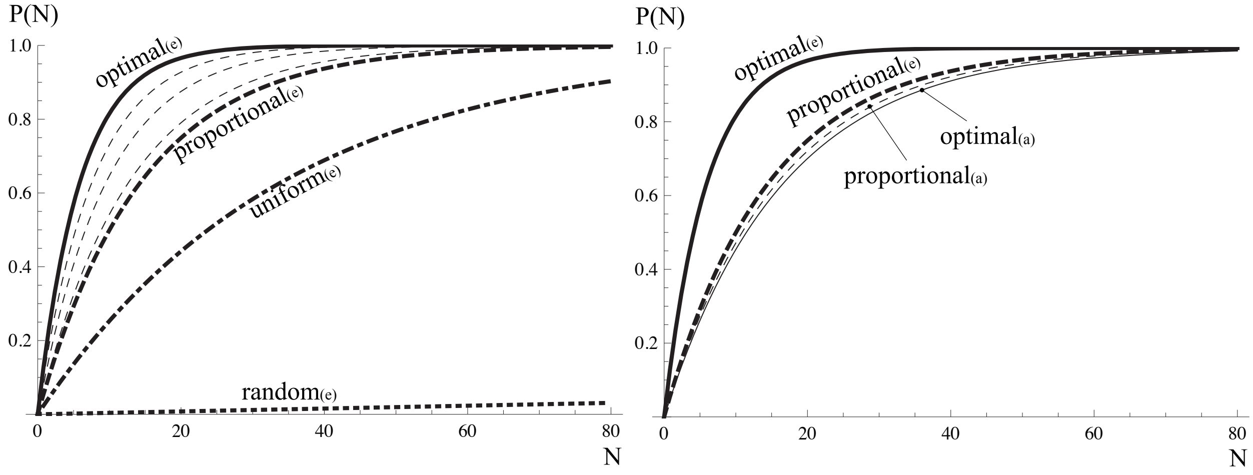
$$D_2 \quad = \quad \left\{ \begin{array}{rcl} 0 \leq & x_0^0 & \leq 10 \\ 0 \leq & x_5^0 & \leq 10 \\ -5 \leq & x_0^0 - x_5^0 & \leq 0 \end{array} \right.$$

(a)



(b)

Figure 6. (a) The DBM zones that constitute the partitioned domain of function $P_{OK}(x_0^0, x_1^0)$, representing the probability of execution of the run $\rho_2 = t_0 \rightarrow t_5 \rightarrow t_1 \rightarrow t_3 \rightarrow t_6 \rightarrow t_4 \rightarrow t_7 \rightarrow t_8$ as a function of controllable transitions t_0 and t_5, in the example of Fig.5; (b) plot of function $P_{OK}(x_0^0, x_1^0)$.



(a) (b)

Figure 7. (a) Probability of conclusive test execution as a function of the number of test repetitions, with optimal sampling (continuous thick line), proportional sampling with order $k = 1, 2, 5, 10$ (dashed lines, bottom-up for increasing values of k), uniform sampling (dot-dashed thick line), random testing (dotted thick line). (b) Comparison of optimal sampling (continuous thick line) and proportional sampling (dashed thick line) based on exact estimates of function $P_{OK}()$ (marked by subscript "(e)") with respect to the approach of [43] (continuous thin line) and its combination with proportional sampling (dashed thin line), which are based on approximate estimates of $P_{OK}()$ (marked by subscript "(a)").

	Optimal sampling	Proportional sampling	Uniform sampling	Random testing
0.8	11	25	57	4087
0.9	15	35	80	5846
0.95	19	45	104	7606
0.99	29	68	159	11691

Table II
NUMBER OF TEST REPETITIONS NEEDED TO ATTAIN A GIVEN SUCCESS PROBABILITY THROUGH DIFFERENT REPETITION SCHEMES.

Task	Jitter	Chunk	Exclusions	BCET	WCET	α	β
T_1	$[0, 10]$	C_{11}	-	5	10	5	3
		C_{12}	C_{22}, C_{33}	10	15	3	2
T_2	$[0, 10]$	C_{21}	-	5	20	6	3
		C_{22}	C_{12}, C_{32}, C_{41}	5	10	2	1
T_3	$[0, 10]$	C_{31}	-	10	20	5	4
		C_{32}	C_{22}, C_{41}	5	10	4	1
		C_{33}	C_{12}	10	20	1	4
T_4	$[0, 10]$	C_{41}	C_{22}, C_{32}	5	10	1	1

Table III
PARAMETERS OF A WORKLOAD MADE OF 4 TASKS.

Task	Jitter	Chunk	Exclusions	BCET	WCET	α	β
T_1	$[0, 10]$	C_{11}	-	5	10	5	3
		C_{12}	C_{22}, C_{33}, C_{51}, C_{61}	10	15	3	2
T_2	$[0, 10]$	C_{21}	-	5	20	6	3
		C_{22}	C_{12}, C_{32}, C_{41}, C_{61}	5	10	2	1
T_3	$[0, 10]$	C_{31}	-	10	20	5	4
		C_{32}	C_{22}, C_{41}	5	10	4	1
		C_{33}	C_{12}, C_{51}, C_{53}	10	20	1	4
T_4	$[0, 10]$	C_{41}	C_{22}, C_{32}	5	10	1	1
T_5	$[0, 10]$	C_{51}	C_{12}, C_{33}	10	15	3	2
		C_{52}	-	5	10	1	1
		C_{53}	C_{33}	5	10	1	1
T_6	$[0, 10]$	C_{61}	C_{12}, C_{22}	5	15	4	1
		C_{62}	-	10	20	1	4

Table IV
PARAMETERS OF A WORKLOAD MADE OF 6 TASKS.

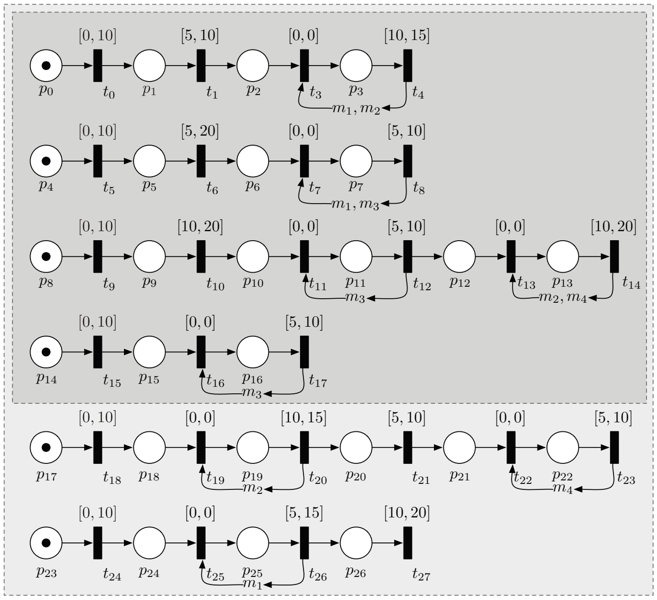


Figure 8. psTPN models of a task set made of 4 tasks (inner box) and 6 tasks (outer box), synchronized on 4 mutual exclusive resources. Places accounting for mutual exclusion among chunks are omitted for readability and substituted by their names $m_1, \ldots, m_4$.

be a realistic case in the Real-Time practice), the target run ρ_4 has a maximum duration of 160 μs; using optimal values, the test is covered with probability 0.9 within 111567 repetitions, i.e., in about 18 seconds; with random testing the behavior would be virtually impossible to observe. In the 6-tasks case, the target run ρ_6 is somehow beyond the limit of a reasonable test planning: with optimized inputs, the case is covered with probability 0.9 within 3.3×10^7 repetitions, which would take 137 minutes as the maximum duration of ρ_6 is 250 μs. This definitely suggests that a suite of shorter runs may permit to reach an acceptable coverage with a reduced number of test repetitions.

V. CONCLUSIONS

In the execution of real-time tests, non-controllable temporal parameters may prevent effective application of input values and produce inconclusive behaviors. In this case, a quantitative approach is needed to select input values for controllable parameters that maximize the probability of a conclusive test.

We characterized the problem so as to account for the combined effect of the stochastic characterization of non-controllable timers, the structure of concurrency of the model, and the sequencing constraints of the selected test. In particular, this all yields an explicit representation of the probability of conclusive execution as a function of input values given to controllable timers.

The proposed technique can be applied in a constructive perspective to generate the inputs through an optimization approach or through simpler yet effective sub-optimal techniques, which may be implemented through symbolic or numerical calculus. Both strategies were implemented through a toolchain that composes the Sirio Framework [10], [11], [6] with Wolfram Mathematica [42]. The overall framework can also be applied to set a reference of optimal performance for the evaluation of various input generation techniques, such as the basic straight approach of random testing or the approximate approach of [43].

In the theoretical perspective, achieved results suggest the opportunity of addressing input generation for test cases specified through partial order restrictions rather than through a complete sequence. In the applicative perspective, the approach opens the way to experimentation in formal Model-Driven Development frameworks.

References

[1] R. Alur and D. L. Dill. A Theory of Timed Automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.

[2] B. Beizer. Black-Box Testing: Techniques for Functional Testing of Software and Systems. *Wiley*, 1995.

[3] S. Bernardi, J. Campos, and J. Merseguer. Timing-failure risk assessment of UML design using Time Petri Net bound techniques. *IEEE Transactions on Industrial Informatics*, 7(1):90–104, February 2011.

[4] J. Berntsen, T. O. Espelid, and A. Genz. An Adaptive Algorithm for the Approximate Calculation of Multiple Integrals. *ACM Trans. Math. Softw.*, 17, December 1991.

[5] B. Berthomieu and M. Diaz. Modeling and Verification of Time Dependent Systems Using Time Petri Nets. *IEEE Trans. on SW Eng.*, 17(3):259–273, March 1991.

[6] G. Bucci, L. Carnevali, L. Ridi, and E. Vicario. Oris: a Tool for Modeling, Verification and Evaluation of Real-Time Systems. *Int. Journal of Software Tools for Technology Transfer*, 12(5):391 – 403, 2010.

[7] L. Carnevali, L. Grassi, and E. Vicario. A Tailored V-Model Exploiting the Theory of Preemptive Time Petri Nets. In *Proc. of the Ada-Europe Int. Conf. on Reliable SW Tech.*, pages 87–100. Springer-Verlag, 2008.

[8] L. Carnevali, L. Grassi, and E. Vicario. State-Density Functions over DBM Domains in the Analysis of Non-Markovian Models. *IEEE Trans. on SW Eng.*, 35(2):178–194, 2009.

[9] L. Carnevali, L. Ridi, and E. Vicario. Partial stochastic characterization of timed runs over DBM domains. In *Proc. of the* 9^{th} *International Workshop on Performability Modeling of Computer and Communication Systems*, Sept. 2009.

[10] L. Carnevali, L. Ridi, and E. Vicario. A Framework for Simulation and Symbolic State Space Analysis of Non-Markovian Models. In *SAFECOMP*, pages 409–422, 2011.

[11] L. Carnevali, L. Ridi, and E. Vicario. Sirio: A Framework for Simulation and Symbolic State Space Analysis of non-Markovian Models. In 8^{st} *Int. Conf. on Quantitative Evaluation of Systems (QEST '11)*, pages 153–154, 2011.

[12] L. Carnevali, L. Sassoli, and E. Vicario. Sensitization of symbolic runs in real-time testing using the ORIS tool. In *Proc. of the IEEE Conf. on Emerging Technologies and Factory Automation (ETFA)*, Sept. 2007.

[13] T. S. Chow. Testing software design modeled by finite-state machines. *IEEE Trans. Softw. Eng.*, 4(3):178–187, May 1978.

[14] G. B. Dantzig and B. C. Eaves. Fourier-Motzkin elimination and its dual. *Journal of Combinatorial Theory*, 14(3):288–297, 1973.

[15] D. Dill. Timing Assumptions and Verification of Finite-State Concurrent Systems. *Proc. Workshop on Computer Aided Verification Methods for Finite State Systems*, 1989.

[16] A. En-Nouaary, R. Dssouli, and F. Khendek. Timed Wp-Method: Testing Real-Time Systems. *IEEE Trans. on SW Eng.*, 28(11), 2002.

[17] S. Fujiwara, G. Bochmann, F. Khendek, M. Amalou, and A. Ghedamsi. Test Selection Based on Finite-State Models. *IEEE Trans. on SW Eng.*, 17(2):591–603, 1991.

[18] G. Karsai and J. Sztipanovits and A. Ledeczi and T. Bapty. Model-Integrated Development of Embedded Software. *Proc. of the IEEE*, 91:145–164, Jan. 2003.

[19] A. Hessel, K. Larsen, M. Mikucionis, B. Nielsen, P. Pettersson, and A. Skou. Testing real-time systems using UPPAAL. In *Formal Methods and Testing, LNCS 4949*, pages 77–117. Springer, 2008.

[20] A. Hessel and P. Pettersson. A Global Algorithm for Model-Based Test Suite Generation. *Electr. Notes Theor. Comput. Sci.*, 190(2):47–59, 2007.

[21] R. M. Hierons, K. Bogdanov, J. P. Bowen, R. Cleaveland, J. Derrick, J. Dick, M. Gheorghe, M. Harman, K. Kapoor, P. Krause, G. Lüttgen, A. J. H. Simons, S. Vilkomir, M. R. Woodward, and H. Zedan. Using formal specifications to support testing. *ACM Comput. Surv.*, 41(2):9:1–9:76, Feb. 2009.

[22] C. Jard and T. Jéron. TGV: theory, principles and algorithms, A tool for the automatic synthesis of conformance test cases for non-deterministic reactive systems. *Software Tools for Technology Transfer (STTT)*, 6, October 2004.

[23] M. Krichen and S. Tripakis. Black-Box Conformance Testing for Real–Time Systems. *Int. SPIN Workshop on Model Checking of SW*, 2004.

[24] M. Krichen and S. Tripakis. An expressive and implementable formal framework for testing real-time systems. In F. Khendek and R. Dssouli, editors, *Testing of Communicating Systems*, volume 3502 of *Lecture Notes in Computer Science*, pages 375–375. Springer Berlin / Heidelberg, 2005.

[25] M. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Verifying Quantitative Properties of Continuous Probabilistic Timed Automata. In LNCS, editor, *Proc. CONCUR'00*, volume 1877, pages 123–137, 2000.

[26] L. Carnevali and L. Ridi and E. Vicario. Putting preemptive Time Petri Nets to work in a V-Model SW life cycle. *IEEE Trans. on SW Engineering*, 37(6), Nov./Dec. 2011.

[27] K. G. Larsen, M. Mikucionis, and B. Nielsen. Online Testing of Real-Time Systems Using UPPAAL: Status and Future Work. In Ed Brinksma and Wolfgang Grieskamp and Jan Tretmans, editor, *Perspectives of Model-Based Testing*, number 04371 in Dagstuhl Seminar Proceedings. Internationales Begegnungs- und Forschungszentrum fuer Informatik (IBFI), Schloss Dagstuhl, Germany, 2005.

[28] M. G. Merayo, M. Núñez, and I. Rodríguez. Formal testing from timed finite state machines. *Computer Networks*, 52(2):432–460, 2008.

[29] H. Muccini, A. Bertolino, and P. Inverardi. Using Software Architecture for Code Testing. *IEEE Trans. on SW Eng.*, 30(3):160–171, 2004.

[30] G. J. Myers. *The Art of Software Testing, Second edition. Revised and updated by Tom Badgett and Todd M. Thomas, with Corey Sandler.* John Wiley & Sons, Inc., New York, 2004.

[31] J. A. Nelder and R. Mead. A simplex method for function minimization. *The Computer Journal*, 7(4):308–313, January 1965.

[32] W. Penczek and A. Polrola. Specification and Model Checking of Temporal Properties in Time Petri Nets and Timed Automata. In *Proc. of the* 25^{th} *Int. Conf. on Application and Theory of Petri Nets (ICATPN)*, Bologna, Italy, June 2004.

[33] Radio Technical Commission for Aeronautics. *DO-178B, Software Considerations in Airborne Systems and Equipment Certification*, 1992.

[34] J. Schmaltz and J. Tretmans. On conformance testing for timed systems. In *6th Int. Conf. on Formal Modeling and Analysis of Timed Systems, FORMATS'08, LNCS 5215*, pages 250–264. Springer, 2008.

[35] D. C. Schmidt. Model–Driven Engineering. *IEEE Computer*, pages 1–2, February 2006.

[36] J. Springintveld, F. Vaandrager, and P. D'Argenio. Testing timed automata. *Theoretical Computer Science*, 254(1-2):225–257, 2001. Previously appeared as Technical Report CTIT-97-17, University of Twente, 1997.

[37] J. Tretmans. Model based testing with labelled transition systems. In *Formal Methods and Testing, LNCS 4949*, pages 1–38. Springer, 2008.

[38] E. Vicario. Static Analysis and Dynamic Steering of Time Dependent Systems Using Time Petri Nets. *IEEE Trans. on SW Eng.*, 27(1):728–748, August 2001.

[39] E. Vicario, L. Sassoli, and L. Carnevali. Using Stochastic State Classes in Quantitative Evaluation of Dense-Time Reactive Systems. *IEEE Trans. on SW Eng.*, 35(5):703–719, 2009.

[40] R. Wilhelm, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, D. Whalley, G. Bernat, C. Ferdinand, R. Heckmann, T. Mitra, F. Mueller, I. Puaut, P. Puschner, J. Statshulat, and P.Stenstroem. Priority Inheritance Protocols: The Worst Case Execution-Time problem: Overview of methods and survey of tools. *ACM Trans. Emb. Comp. Sys.*, 7(3):1–53, 2008.

[41] Wolfram Connection Technologies. *Java Toolkit: J/Link*. http://www.wolfram.com/solutions/mathlink/jlink/.

[42] Wolfram Research, www.wolfram.com. *Mathemathica 5.2*.

[43] N. Wolovick, P. D'Argenio, and H. Qu. Optimizing probabilities of real-time test case execution. In *Int. Conf. on Software Testing Verification and Validation (ICST)*, pages 1–10, April 2009.

[44] H. L. S. Younes and R. G. Simmons. Solving generalized semi-Markov decision processes using continuous phase-type distributions. In *AAAI'04: Proceedings of the 19th national conference on Artifical intelligence*, pages 742–747. AAAI Press / The MIT Press, 2004.

Laura Carnevali Laura Carnevali received the Bachelor Degree in Informatics Engineering, the Doctoral Degree in Informatics Engineering, and the Ph.D. in Informatics, Multimedia, and Telecommunications Engineering from the University of Florence in 2004, 2006, and 2010, respectively. She is a post-doc fellow at the Department of Systems and Informatics of the University of Florence. Her research is focused on correctness verification and performance evaluation of real-time systems, with specific interest on integration of formal methods in the development life cycle of real-time software and stochastic characterization of timed models.

Lorenzo Ridi Lorenzo Ridi received the Bachelor Degree in Informatics Engineering in 2005, the Master Degree in Informatics Engineering in 2007 and the Ph.D in Informatics, Multimedia and Telecommunications Engineering in 2011. He is now a research fellow at the Software Technologies Laboratory (STLab) of the University of Florence. His research activity mainly focuses on formal techniques for the specification, the qualitative verification and the quantitative validation of stochastic time-dependent systems, and their integration in the development lifecycle of real-time software.

Enrico Vicario Enrico Vicario is a Full Professor of Computer Science at the School of Engineering of the University of Florence, where he received the Doctoral Degree in Electronics Engineering and the Ph.D. in Informatics and Telecommunications Engineering, in 1990 and 1994, respectively. His research is presently focused on formal methods for model driven development, correctness verification of real-time systems, and quantitative evaluation of concurrent non-Markovian models.