Implicit Reward Structures for Implicit Reliability Models

Giulio Masetti*, Leonardo Robol*†, Silvano Chiaradonna*, Felicita Di Giandomenico*
* ISTI-CNR, Pisa, Italy, † University of Pisa,
{giulio.masetti, leonardo.robol, silvano.chiaradonna, felicita.digiandomenico}@isti.cnr.it

***Abstract*—A new methodology for effective definition and efficient evaluation of dependability-related properties is proposed. The analysis targets systems composed of a large number of components, each one modeled implicitly through high-level formalisms such as SPNs. Since the component models are implicit, the reward structure that characterizes the dependability properties has to be implicit as well. Therefore, we present a new formalism to specify those reward structures. The focus here is on component models that can be mapped to Stochastic Automata with one or several absorbing states, so that the system model can be mapped to a Stochastic Automata Network with one or several absorbing states. Correspondingly, the new reward structure defined on each component's model is mapped to a reward vector, so that the dependability-related properties of the system are expressed through a newly introduced measure defined starting from those reward vectors. A simple, yet representative, case study is adopted, to show the feasibility of the method.**

I. Introduction

Stochastic state-space model-based approaches are widely used to perform dependability, performance or performability analysis of complex systems. Generally, such systems are composed of many components interacting with each other in an intricate manner. This makes the model-based analysis very challenging, both with respect to the definition of the overall model, which must represent all the individual components and their specific interdependencies, and the explosion of the state-space [1]. Thus, highly expressive formalisms for the definition of the model and high computational efficiency for its solution are being increasingly addressed, and our contribution goes in these directions.

Resorting to the popular modularity and compositionality paradigm, system complexity and largeness are typically managed through the composition of individual submodels, which represent system components at the desired level of abstraction [2]–[7]. Individual submodels can be defined concisely by adopting a high-level formalism, and the resulting overall model can then be obtained through composition operators, such as the approaches based on named replication of stochastic template models recently proposed in [8]–[11].

To tackle the state-space explosion, a successful research line in the literature consists in representing and manipulating the state-space implicitly, i.e., not listing the states, but describing them through more complex data structures [12], [13]. However, despite the great research effort, underway for decades, there are still areas where improvements are very welcome, to keep pace with the growing complexity of systems on the one hand, and the need to maintain high accuracy of the analysis on the other hand. This paper provides a contribution in this research area, adopting Superposed SPN (SSPN) as high level formalism, inspired to the Superposed GSPN (SGSPN) proposed in [13], and focusing on strategies based on Kronecker algebra [14]. SSPN is an extension of Stochastic Petri Net (SPN) [15], where the interdependencies occurring among submodels are captured through synchronization of transitions and the underlying stochastic process is a Continuous Time Markov Chain (CTMC) [16], [17]. It is a sufficiently general and powerful formalism, suitable to the purpose of our study. Through Kronecker algebra [14], interpreting the system model as the transition-based synchronization of submodels allows to map each high-level submodel to a Stochastic Automaton, and then to manipulate implicitly the underlying low-level model as a Stochastic Automata Network (SAN) [18].

So far, the community has focused almost exclusively on implicit models, without considering the reward structure defined on top of the system model, which is at the basis of the evaluation of dependability, performance or performability measures in classical model-based analysis [19], [20]. Reward models have been widely used both at high-level, e.g., with Stochastic Activity Networks [19], Generalized Stochastic Petri Net (GSPN) [21] or Stochastic Reward Nets [22], and at the level of CTMC, which is extended to a Markov reward model [20], where a reward rate is attached to each state of the CTMC and is represented by a reward vector.

Following the approach adopted in [19], a reward structure can be defined on the SSPN model. The solution of the SSPN model involves automatic generation and analysis of the corresponding Markov reward model. Although the underlying CTMC model is automatically generated starting from a concise SSPN model, one major drawback can be the largeness of its state-space [22].

Until recently, only descriptor matrices were available to the modelers in the context of CTMC and there was no solid numerical foundation onto which to develop the very idea of the implicit reward vector. An implicit reward vector, defined on top of an SAN and based on Kronecker algebra, and a new numerical solution for the resulting Markov reward model generated from an SSPN structured in submodels, are proposed in [10]. Such solution addresses efficient evaluation of the Mean-Time-To-Absorption in SSPN, but it is restricted to one absorbing state only.

Targeting systems representable through a composition of submodels, each one characterized by a small number of states

and of interactions with other submodels, this paper presents three original developments. Their feasibility and effectiveness are shown through a representative case study, which includes component-based systems with multiple failure modes and rather complex patterns of error propagation.

The first contribution is the definition of a high-level implicit reward structure based on submodels (inspired by the explicit one in [19]), given concisely in terms of the reward structure on top of each SSPN submodel. Differently from the explicit one, this new reward structure can be automatically transformed into the implicit reward vector defined on top of the underlying SAN, in an easy and direct way.

The second contribution is a strategy based on Kronecker algebra to numerically solve an SAN reward model with multiple absorbing states. Mathematical formulas are used to express and numerically evaluate the k-moments (e.g., mean or variance) of instant-of-time or interval-of-time reward variables [19] to absorbing state (i.e., for time going to infinity). Examples of these measures are the Mean Time To Failure (MTTF) and the variance of the time to failure, or the conditional MTTF [23], [24], where one among multiple causes of failure can be addressed (e.g., safe or unsafe shutdown, imperfect coverage or exhaustion of employed redundancy [23]).

The third contribution is the characterization of the relation between the reward variables, defined through the implicit reward structure proposed for the high-level SSPN model, and the formulas proposed for the reward variables at level of SAN. The outcome is that the Markov reward model and the formulas used for the solution can be automatically generated from the corresponding reward model and measures, concisely described at level of SSPN.

The rest of the paper is structured as follows. Section II introduces the reference modeling context and the measures of interest considered in this paper at level of both stochastic Petri net and Markov chain. The formulas for the explicit evaluation of the measures at the Markov chain level are also included. The derivation of the explicit formulas for the moments of the accumulated reward to absorption is given in Appendix C. In Section III, a new implicit reward structure based on submodels at the stochastic Petri net level is proposed. Examples of reward structures are then shown in Appendix F. Section IV presents the expressions, based on the Kronecker Algebra, which directly map the measures defined at the level of stochastic Petri net on the level of Markov chain. The solution methodology proposed to evaluate the measures is also included. To illustrate the effectiveness of the proposed approach, Section V presents the case study and its relevant measures, while the results of the performance evaluation obtained by applying the proposed solution method are discussed in Section VI. Section VII reviews related works, while Section VIII draws conclusions and outlines some future extensions. Finally, a few Appendices are included, to detail mathematical developments as well as the table of acronyms and symbols.

II. Modeling context and measures of interest

In this section, the reference modeling context and the measures of interest considered in this paper, at level of both stochastic Petri net and Markov chain, are introduced. The formulas for the explicit evaluation of the measures at Markov chain level are also included.

A. Stochastic Petri net level

This paper considers a system model M with a finite number of states, composed of n submodels $M_1, \ldots, M_n$. The submodels are defined using the SPN high-level formalism. They are synchronized through transitions, forming an SSPN network with a small number of interconnections.

A marking μ of an SPN represents the state of the model at a particular instant of time. It is a function $\mu : P \mapsto \mathbb{N}$ describing the number of tokens in each place, where P is the set of places and $\mathbb{N}$ is the set of nonnegative integers. The notation $\#p$ is adopted instead of $\#(p, \mu)$ to indicate the number of tokens in place p in marking μ [22], whenever the marking μ is clear from the context. A marking $\mu \in \mathbb{N}^P$ is usually represented as a formal sum $\sum_p \#p \cdot p$ or as a $|P|$-dimensional vector, where $|\ldots|$ denotes the size of a set. For example, given $P = \{A, B, C\}$, $\mu = 2B + C = (0, 2, 1)$ is the marking with $(\#A = 0, \#B = 2, \#C = 1)$.

Let $\mathcal{S} \subseteq \mathbb{N}^P$ and $\mathcal{S}^{(i)} \subseteq \mathbb{N}^{P_i}$ be the reachability sets (i.e., the sets of all the markings which are reachable from the initial marking) of M and M_i, respectively. Each submodel has a small number of markings (i.e., $|\mathcal{S}^{(i)}|$ is small) with at least one absorbing marking. By definition, in an absorbing marking of an SPN no transition of the SPN is enabled and the evolution of the SPN has reached a final marking.

Each marking of an SPN corresponds to a state of the underlying CTMC. Once the correspondence between markings and states is defined, each marking is of the same type as the corresponding state. This paper addresses only SPN models for which the underlying CTMC satisfies the requirements of the proposed solution method, as stated in Section II-B, where the underlying CTMC class is characterized explicitly.

Figure 1 shows the general scheme of an SPN model, exemplified for the case where 4 places are explicitly represented for each submodel M_i: E_i, W_i, B_i and C_i. In particular, B_i and C_i are used to represent the two absorbing markings considered for each M_i given by: $(\#B_i = 1, \#C_i = 0)$ and $(\#B_i = 0, \#C_i = 1)$. The absorbing markings of the overall SPN model correspond to those markings for which all the submodels reached the absorbing marking, i.e., when $\sum_i (\#B_i + \#C_i) = n$. A detailed example with one or two absorbing markings for each M_i is given by the SPN models shown in Figures 3 and 4 for the case study described in Section V-B.

Let $\mathcal{A}$ be the set of all absorbing markings (states) of the model M, and $\mathcal{B}$ be a subset of absorbing markings with $\mathcal{B} \subseteq \mathcal{A}$. Let Y_∞ denote the random variable representing the reward accumulated by the model M in transient states until absorption into $\mathcal{A}$. Let $Y_{\infty|\mathcal{B}}$ be the conditional random variable representing the reward accumulated by the model M

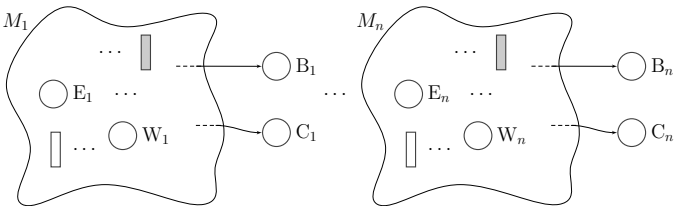


Fig. 1. General scheme of the SPN model composed of n submodels $M_1, \ldots, M_n$, synchronized through (shaded) transitions.

in transient states until absorption into $\mathcal{A}$, given that the model eventually absorbs into $\mathcal{B}$.

The measures of interest addressed in this paper are the following performability (i.e., performance and dependability) metrics defined on the SPN model M with absorbing states:

- The moments of Y_∞, namely
$$\mathcal{M}_k(Y_\infty) = \mathbb{E}\big[Y_\infty^k\big], \tag{1}$$
in particular, the Mean Reward To Absorption (MRTA) with $MRTA = \mathbb{E}[Y_\infty]$, the variance $Var(Y_\infty) = \mathbb{E}[Y_\infty^2] - \big(\mathbb{E}[Y_\infty]\big)^2$ and possibly an approximation of order k of the distribution of Y_∞.
- The conditional mean $\mathrm{MRTA}_{|\mathcal{B}} = \mathbb{E}[Y_{\infty|\mathcal{B}}]$, i.e., the conditional mean reward to absorption, given that the model eventually absorbs into $\mathcal{B}$.
- $\pi_{\mathcal{B}}(\infty)$, i.e., the probability that the model eventually absorbs into $\mathcal{B}$.

This kind of measures is often taken as metrics (or as intermediate values useful to define metrics) of relevant reliability-related properties [24]–[26].

When the absorbing markings represent an unrecoverable failure and the accumulated reward is the time, then Y_∞ and $Y_{\infty|\mathcal{B}}$ reduce to the random variables $T_{\mathcal{A}}$ and $T_{\mathcal{A}|\mathcal{B}}$, representing the time to absorption into $\mathcal{A}$ and the time to absorption into $\mathcal{A}$ given that the model eventually absorbs into $\mathcal{B}$, respectively. Thus, important cases of the above metrics are the following:

- The mean time to failure $MTTF = \mathbb{E}[T_{\mathcal{A}}]$, and the variance $Var(T_{\mathcal{A}})$.
- $\mathrm{MTTF}_{|\mathcal{B}}$, i.e., the conditional mean time to failure given that the model eventually absorbs into $\mathcal{B}$.

$\mathrm{MRTA}_{|\mathcal{B}}$ and $\mathrm{MTTF}_{|\mathcal{B}}$ are particularly important when there is more than one absorbing state. This is the case, for example, when it is required to discriminate among multiple causes of failure, such as to distinguish between safe and unsafe shutdown (in a safety critical system), or to distinguish a failure due to imperfect coverage from a failure due to the exhaustion of redundancy (in a fault-tolerant system) [23].

Following the approach proposed in [22], and focusing on marking oriented reward (rate reward) structure, the reward is defined at the SPN level by the function
$$\mathcal{R} : \mathbb{N}^P \mapsto \mathbb{R}, \tag{2}$$
where $\mathbb{R}$ is the set of real numbers. For every $\mu \in \mathcal{S}$, $\mathcal{R}(\mu)$ is the (positive, negative or zero) reward obtained while the model M is in the marking μ. Let $\bar{\mathcal{S}} \subseteq \mathcal{S}$ be the set of reachable markings of the SPN that includes only the elements having a non-zero reward assignment.

Given an SPN model with a reward structure of this type, the measures of interest, here specialized to the case of infinite time, are specified in terms of the following reward variables, inspired by those proposed in [19]: i) the interval-of-time reward variable Y_∞, ii) the conditional interval-of-time reward variable $Y_{\infty|\mathcal{B}}$ and iii) the instant-of-time reward variable V_∞. Their formulation is as follows:
$$Y_\infty(\mathcal{R}) = \sum_{\mu\in\mathcal{S}} \mathcal{R}(\mu)\cdot J_\infty^\mu, \tag{3}$$
$$Y_{\infty|\mathcal{B}}(\mathcal{R}) = Y_\infty(\mathcal{R}) \mid \bigvee_{\mu\in\mathcal{B}} [I_\infty^\mu = 1], \tag{4}$$
$$V_\infty(\mathcal{R}) = \sum_{\mu\in\mathcal{S}} \mathcal{R}(\mu)\cdot I_\infty^\mu, \tag{5}$$
where the vertical bar $\mid$ in Equation (4) is the symbol for conditional event, the symbol $\bigvee$ is the logical operator "or", J_t^μ is a random variable counting the total time the SPN spends in the marking μ during the interval of time $[0, t]$, and I_t^μ is an indicator random variable representing the event $[I_t^\mu = 1]$ that at time t the SPN is in the marking μ. Notice that $J_\infty^\mu = \int_0^\infty I_t^\mu dt$. Moreover, J_t^μ and I_t^μ are random variables, depending on the random event "the SPN is in the marking μ at time t".

In the following, to simplify the notation, $\mathcal{M}_{k,\mathcal{R}}$ will be used to denote $\mathcal{M}_k(Y_\infty(\mathcal{R}))$.

Since there are absorbing markings in the model, the measures of interest specified in terms of Equations (3) and (4) can only be defined under the condition that the reward associated to absorbing markings in Equation (3) is zero, i.e., $\mathcal{R}(\mu) = 0\ \forall \mu \in \mathcal{A}$.

For a given $\mathcal{B}$, defining
$$\mathcal{R}(\mu) = \begin{cases} 1 & \text{if } \mu \in \mathcal{B}, \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$
results in obtaining $\pi_{\mathcal{B}}(\infty)$ from Equation (5) as follows:
$$\pi_{\mathcal{B}}(\infty) = \sum_{\mu\in\mathcal{B}} P(I_\infty^\mu = 1). \tag{7}$$

For a given $\mathcal{B}$ and $\mathcal{R}$, applying the definition of conditional mean of a random variable, $\mathrm{MRTA}_{|\mathcal{B}}$ is obtained from Equations (4) and (7) as follows:
$$\begin{aligned} \mathrm{MRTA}_{|\mathcal{B}} &= \mathbb{E}[Y_{\infty|\mathcal{B}}(\mathcal{R})] && \text{(by definition)} \\ &= \mathbb{E}[Y_\infty(\mathcal{R}) \mid \bigvee_{\mu\in\mathcal{B}} [I_\infty^\mu = 1]] && \text{(from Equation (4))} \\ &= \frac{\mathbb{E}[Y_\infty(\mathcal{R}) \text{ and } \bigvee_{\mu\in\mathcal{B}} [I_\infty^\mu = 1]]}{\pi_{\mathcal{B}}(\infty)}, && (8) \end{aligned}$$
being "the conditional expectation $E[Y \mid \mathcal{B}]$ of a random variable Y given an event $\mathcal{B}$" equal to $E[Y \text{ and } \mathcal{B}]/P(\mathcal{B})$, where $\mathcal{B} = \bigvee_{\mu\in\mathcal{B}} [I_\infty^\mu = 1]$, and $P(\mathcal{B}) = \pi_{\mathcal{B}}(\infty)$ from Equation (7). The conditional expectation $\mathbb{E}[Y_\infty(\mathcal{R}) \mid \bigvee_{\mu\in\mathcal{B}} [I_\infty^\mu = 1]]$

represents the mean reward accumated by the SPN until the occurrence of the event $\bigvee_{\mu\in\mathcal{B}}[I^{\mu}_{\infty}=1]$, given that the event has occurred, where the event is the absorption of the SPN into any of the markings of the set $\mathcal{B}$.

Notice that the formulas used in this section to define the measures of interest are generic, since they do not take into account the structure of the system in submodels. Moreover, depending on the specific SPN and reward structure, the measures of interest may be defined only when the corresponding formulas converge to a finite value, as time approaches infinity.

B. Markov chain level

The numerical solution requires generating the stochastic process $\{X(t)\in\mathcal{S},\, t\geq 0\}$ underlying the SPN model. It is a time-homogeneous continuous-time Markov chain (CTMC) with discrete (finite) state space $\mathcal{S}=\{1,2,\ldots,n_{\text{reach}}\}$, where n_{reach} denotes the number of reachable states. To every marking $\mu_i\in\mathcal{S}$ of the SPN corresponds a state $i\in\mathcal{S}$ of the CTMC. Call $\boldsymbol{Q}=[q_{ij}]$ the Infinitesimal Generator matrix consisting of the direct transition rates from state i to j, for $i\neq j$, and $q_{ii}=-\sum_{j,j\neq i}q_{ij}$, and $\boldsymbol{\pi}(t)\in\mathbb{R}^{n_{\text{reach}}}$ the state probability vector at time t such that $\sum_i \pi_i(t)=1$ for every $t\geq 0$.

The states in $\mathcal{S}$ are partitioned into transient states $\mathcal{T}$ and absorbing states $\mathcal{A}$, i.e., $\mathcal{S}=\mathcal{T}\cup\mathcal{A}$, with $\mathcal{T}\neq\emptyset$ and $\mathcal{A}\neq\emptyset$. It is assumed that the modeled system is in a working state at $t=0$, i.e., $\pi_i(0)=0$ for $i\in\mathcal{A}$. For $\pi_i(0)$, with $i\notin\mathcal{A}$, any probability value can be used. Hence, there can be several transient states with initial positive probability less than 1, or a single initial transient state with probability 1. The proposed solution method requires that the state space contains no irreducible subset consisting of more than one state. But it is not required that an absorbing state be reachable from every transient state.

After relabeling the absorbing states as the last states, the CTMC is represented through the infinitesimal generator matrix as follows:

$$\boldsymbol{Q}=\left[\begin{array}{ccc|c|c|c|c} & \boldsymbol{Q}_{\mathcal{T}} & & \boldsymbol{v}_1 & \boldsymbol{v}_2 & \ldots & \boldsymbol{v}_{n_{\text{abs}}} \\ \hline 0 & \ldots & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & \ldots & 0 & 0 & 0 & 0 & 0 \\ \hline & \vdots & & \vdots & \vdots & \vdots & \vdots \\ \hline 0 & \ldots & 0 & 0 & 0 & 0 & 0 \end{array}\right], \tag{9}$$

where $\boldsymbol{Q}_{\mathcal{T}}$ is the submatrix that contains the rates of transitions among the transient states, $n_{\text{abs}}=|\mathcal{A}|$, and $\boldsymbol{v}_i$ are the column vectors that contain the rates of transitions from transient states to $i\in\mathcal{A}$.

Differently from the standard notation adopted in the performance and dependability community, in this paper column vectors (instead of row vectors) are employed to ease the notation, so the state probability equation is expressed as:

$$\frac{d\boldsymbol{\pi}^T(t)}{dt}=\boldsymbol{\pi}^T(t)\cdot\boldsymbol{Q}. \tag{10}$$

In the following, $\boldsymbol{e}_s$ denotes the s-th element of the standard basis of $\mathbb{R}^m$ (where the value of m depends on the context), i.e., the column vector $\boldsymbol{e}_s=(0,\ldots,1,\ldots,0)^T$ with 1 at position s, while $\boldsymbol{e}$ denotes the column vector with all the entries equal to 1.

At Markov chain level, the reward column vector $\boldsymbol{r}=(r_1,\ldots,r_{n_{\text{reach}}})^T$ is obtained, for each state $i\in\mathcal{S}$, from the SPN level reward function $\mathcal{R}$ as follows:

$$r_i=\mathcal{R}(\mu_i)\text{ with }\mu_i\in\mathcal{S}. \tag{11}$$

Notice that, for the measure defined in Equation (3) (and consequently for Equation (4) that is defined in terms of Equation (3)), the proposed solution method requires that $r_i=0\;\forall i\in\mathcal{A}$.

The vectors $\boldsymbol{r}_{\mathcal{T}}$ and $\boldsymbol{\pi}_{\mathcal{T}}(t)$ are the vector $\boldsymbol{r}$ and $\boldsymbol{\pi}(t)$, respectively, restricted to the transient states $\mathcal{T}$.

Given the reward vector $\boldsymbol{r}$, the reward variable Y_∞ can be defined at Markov chain level as follows:

$$\begin{aligned} Y_\infty(\boldsymbol{r}) &= \int_0^\infty r_{X(t)}dt=\int_0^\infty \boldsymbol{r}^T\cdot\boldsymbol{e}_{X(t)}dt \\ &= \int_0^\infty \sum_{s\in\mathcal{S}} r_s I^s_t dt=\boldsymbol{J}\cdot\boldsymbol{r}, \end{aligned} \tag{12}$$

where $\boldsymbol{J}=(J^1_\infty,\ldots,J^{n_{\text{reach}}}_\infty)$. In the following, the measures of interest are expressed in terms of a matrix function of $\boldsymbol{Q}$, the vector $\boldsymbol{\pi}(0)$ and the reward vector $\boldsymbol{r}$. This opens up the possibility to derive expressions based on generic reward vectors on the one hand, and to fully exploit implicit representations of all the matrices and the vectors that are involved in the computation (as shown in Section IV) on the other hand. The formula for the moments $m_{k,r}$ of $Y_\infty(\boldsymbol{r})$ can be written as:

$$m_{k,r}=k!(-1)^k\boldsymbol{\pi}^T_{\mathcal{T}}(0)(\boldsymbol{Q}^{-1}_{\mathcal{T}}\text{diag}(\boldsymbol{r}_{\mathcal{T}}))^{k-1}\boldsymbol{Q}^{-1}_{\mathcal{T}}\boldsymbol{r}_{\mathcal{T}}. \tag{13}$$

The derivation of this formula is given in Appendix C. Actually, Equation (13) can be evaluated explicitly solving k linear systems to derive the solutions $\boldsymbol{x}^{(1)},\ldots,\boldsymbol{x}^{(k)}$:

$$\begin{cases} \boldsymbol{Q}_{\mathcal{T}}\boldsymbol{x}^{(1)}=\boldsymbol{r}, \\ \boldsymbol{Q}_{\mathcal{T}}\boldsymbol{x}^{(i)}=\text{diag}(\boldsymbol{r}_{\mathcal{T}})\boldsymbol{x}^{(i-1)}, \quad \text{for } i=2,\ldots,k, \end{cases} \tag{14}$$

and performing a dot product:

$$m_{k,r}=k!(-1)^k\boldsymbol{\pi}^T_{\mathcal{T}}(0)\cdot\boldsymbol{x}^{(k)}. \tag{15}$$

This way, well-known issues related to numerical precision when evaluating the moments of a random variable are mitigated (although not entirely avoided because the linear systems can be stiff).

A particular case of Equation (13), i.e., when the reward accumulated in transient states is the time, is the formula used to evaluate explicitly the moments of $T_{\mathcal{A}}$. It can be easily derived exploiting the Laplace transform, as done in Equations (10) to (62) of [26]:

$$m_k(T_{\mathcal{A}})=(-1)^k\boldsymbol{\pi}^T_{\mathcal{T}}(0)\boldsymbol{Q}^{-k}_{\mathcal{T}}\Big(\boldsymbol{e}-\boldsymbol{e}_{\mathcal{A}}\Big), \tag{16}$$

where $\boldsymbol{e}_{\mathcal{A}}$ denotes the column vector $\boldsymbol{e}_{\mathcal{A}}=\sum_{s\in\mathcal{A}}\boldsymbol{e}_s$.

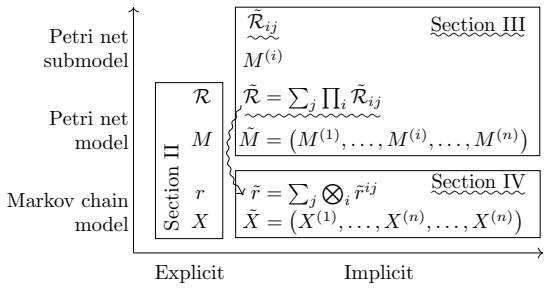


Fig. 2. Classification of the models and reward structures on top of them according to: explicit versus implicit representation and level of modeling abstraction. Newly introduced definitions for reward structures are underlined with a wavy line. The wavy arrow highlights one of the contributions of Section IV-A: how to project reward structures from high-level to low-level models.

Given the absorbing state $a \in \mathcal{A}$ (generalizable to $\mathcal{B} \subseteq \mathcal{A}$), the measure $\mathrm{MRTA}_{|a}$ is obtained like in Equation (8) as follows:

$$\mathrm{MRTA}_{|a} = \frac{\mathbb{E}[Y_\infty(\mathcal{R}) \text{ and } X(\infty) = a]}{\pi_a(\infty)}. \tag{17}$$

Similarly (see Appendix D for details), the formulas to evaluate explicitly $\mathbb{E}[Y_\infty(\mathcal{R}) \text{ and } X(\infty) = a]$ and $\pi_a(\infty)$ are:

$$\mathbb{E}[Y_\infty(\mathcal{R}) \text{ and } X(\infty) = a] = \boldsymbol{\pi}_{\mathcal{T}}^T(0)\boldsymbol{Q}_{\mathcal{T}}^{-1}\mathrm{diag}(\boldsymbol{r}_{\mathcal{T}})\boldsymbol{Q}_{\mathcal{T}}^{-1}\boldsymbol{v}_a, \tag{18}$$

$$\pi_a(\infty) = -\boldsymbol{\pi}_{\mathcal{T}}^T(0)\boldsymbol{Q}_{\mathcal{T}}^{-1}\boldsymbol{v}_a. \tag{19}$$

The formulas for $\mathrm{MRTA}_{|\mathcal{B}}$ and $\pi_{\mathcal{B}}(\infty)$ can be easily obtained by replacing $\boldsymbol{v}_a$ with $\boldsymbol{v}_{\mathcal{B}}$ in Equations (18) and (19), with $\boldsymbol{v}_{\mathcal{B}} = \sum_{a \in \mathcal{B}} \boldsymbol{v}_a$.

As for the formulas of Section II-A, the ones above do not take into account that the system model is structured in submodels. Moreover, they are oriented to an explicit representation of all the model states. Depending on the size of the problem at hand, the explicit evaluation of the measures of interest through the above equations could be impossible, due to the state-space explosion problem. However, these equations are the basis for deriving new formulas that exploit the implicit representation of matrices and vectors, (i.e., not listing the states, but describing them through more complex data structures), then used by the new solution method to overcome the state-space explosion. Moreover, in order to support the automatic and direct generation of the implicit expressions used at level of Markov chain to evaluate measures of interest defined at level of stochastic Petri net, a new implicit reward structure also needs to be introduced.

In the following, the terms explicit and implicit refer to reward structures, formulas and solution methods that are oriented to a direct representation of all the states of the model or to a representation that does not list all the states, respectively. The introduction of implicit reward structures and measures takes advantage of the existence of already well-established (explicit) reward alternatives, since correctness of the new implicit-based solutions is performed through a mapping from the former to the latter (see Appendices A and B for details).

To guide the reader, Figure 2 illustrates the classification and notation adopted through the paper for the modeling formalisms and the reward structures according to two categories: explicit versus implicit representation and level of modeling abstraction. The symbols (whose meaning will be described the first time they are used) are grouped according to the sections where they are defined. To distinguish between already published concepts, such as those summarized in this section, from new concepts and mapping introduced in this paper, underlined text using the wavy line is adopted, as well as the wavy arrow for the mapping. Other relevant information are not shown in Figure 2, such as indicator random variables and measures definition, because they follow the same symbolic code of the reward structures.

Although the focus of this paper is on the implicit side of the figure, relation with explicit formalisms is relevant, both because they inspire/constrain the new developments (see Section III-B) and because they are essential to verify —for small models— the correctness of the new definitions and solution method.

III. Implicit reward model at level of Stochastic Petri Net

This section presents a new implicit reward structure based on submodels (top right corner of Figure 2). The definitions adhere to: same representation power of the reward structure in Section II, reasonably simple translation to the corresponding definitions of Section II to preserve semantics and enhance models verification, direct correspondence with reward vectors in Section IV-A and clear identification of parameters impacting on the solution method performance (details in Section IV-B).

A. Implicit reward structure

The new reward $\tilde{\mathcal{R}}$ is defined in terms of the reward functions $\tilde{\mathcal{R}}_{ij}$ associated to the submodel M_i. Informally, the non-zero reward rate associated by $\tilde{\mathcal{R}}$ to a marking is given as the sum of products of the rewards defined on the marking restricted to the submodels.

Given a marking μ intended as a formal sum, the marking μ restricted to M_i, denoted $\mu^{(i)}$, can be derived as follows:

$$\mu^{(i)} = \sum_{p \in P_i} \#(p, \mu) \cdot p, \tag{20}$$

The marking $\mu^{(i)}$ is also denoted as the projection of μ on the submodel M_i.

Viceversa, given the markings $\mu^{(1)}, \ldots, \mu^{(n)}$, the marking μ can be obtained as the formal sum:

$$\mu = \sum_i \mu^{(i)}. \tag{21}$$

Likewise, each set of absorbing markings $\mathcal{B}$ can be defined from the sets of absorbing markings $\mathcal{B}^{(i)}$ of M_i, and viceversa, denoting $\mathcal{B}^{(i)}$ the projection of $\mathcal{B}$ on M_i:

$$\mathcal{B} = \{\mu^{(1)} + \ldots + \mu^{(n)} \mid \mu^{(i)} \in \mathcal{B}^{(i)}, i = 1, \ldots, n\}, \tag{22}$$

$$\mathcal{B}^{(i)} = \{\sum_{p \in P_i} \#(p, \mu) \cdot p \mid \mu \in \mathcal{B}\}. \tag{23}$$

Let $\tilde{\mathcal{R}}_j^{\Pi}$ be the reward rate function associated to each marking of the SSPN, with $j = 1, \ldots, n_\Pi$ for a given n_Π:

$$\tilde{\mathcal{R}}_j^{\Pi} : \mathbb{N}^P \mapsto \mathbb{R}.$$

The non-zero rewards defined by each $\tilde{\mathcal{R}}_j^{\Pi}$ for some marking are summed by $\tilde{\mathcal{R}}$ to obtain the final reward associated to each marking.

Let $\tilde{\mathcal{R}}_{ij}$ be a function of reward associated to the markings of M_i, with $j = 1, \ldots, n_\Pi$:

$$\tilde{\mathcal{R}}_{ij} : \mathbb{N}^{P_i} \mapsto \mathbb{R}.$$

For $\mu^{(i)} \in \mathbb{N}^{P_i}$, $\tilde{\mathcal{R}}_{ij}(\mu^{(i)})$ is the reward attached to $\mu^{(i)}$.

For a given j, each non-zero reward defined by $\tilde{\mathcal{R}}_j^{\Pi}$ for some marking is obtained as:

$$\tilde{\mathcal{R}}_j^{\Pi}(\mu) = \prod_i \tilde{\mathcal{R}}_{ij}(\mu^{(i)}), \tag{24}$$

where $\mu^{(i)}$ is the marking μ restricted to M_i, as defined in Equation (20).

Let $\bar{\mathbb{S}}_{ij}$ be a set of markings of M_i, i.e., $\bar{\mathbb{S}}_{ij} \subseteq \mathbb{S}^{(i)}$, including only the elements that deserve a non-zero reward explicitly assigned via $\tilde{\mathcal{R}}_{ij}$, being 0 the default value.

When $\bar{\mathbb{S}}_{ij} \neq \emptyset$, the function $\tilde{\mathcal{R}}_{ij}$ is explicitly defined for each marking $\mu^{(i)}$:

$$\tilde{\mathcal{R}}_{ij}(\mu^{(i)}) = \begin{cases} r, & \text{if } \mu^{(i)} \in \bar{\mathbb{S}}_{ij}, \\ 0 & \text{otherwise,} \end{cases} \tag{25}$$

where $r \in \mathbb{R}$ and $r \neq 0$.

When, for a given j, $\bar{\mathbb{S}}_{ij} = \emptyset$, but $\exists k$ such that $\bar{\mathbb{S}}_{kj} \neq \emptyset$, then it is assumed that the function $\tilde{\mathcal{R}}_{ij}$ is defined by default as follows:

$$\tilde{\mathcal{R}}_{ij}(\mu^{(i)}) = \begin{cases} 1 & \text{if } \mu^{(i)} \in \mathbb{S}^{(i)}, \\ 0 & \text{otherwise.} \end{cases} \tag{26}$$

This ensures that the projection $\mu^{(i)}$ of μ on M_i does not impact on $\tilde{\mathcal{R}}_j^{\Pi}(\mu)$ in Equation (24), as shown in Appendix F.

Otherwise, if, for a given j, $\bar{\mathbb{S}}_{ij} = \emptyset$ for $i = 1, \ldots, n$, then it is assumed by default that $\tilde{\mathcal{R}}_{ij}(\mu^{(i)}) = 0$ for each $\mu^{(i)}$, hence $\tilde{\mathcal{R}}_j^{\Pi}(\mu) = 0$ for each μ.

Informally, the contribution of $\tilde{\mathcal{R}}_j^{\Pi}$ to the reward of the overall SSPN for a given marking is obtained as the product of the rewards that are either assigned explicitly or by default through $\tilde{\mathcal{R}}_{ij}$ to the marking restricted to each submodel.

The new reward $\tilde{\mathcal{R}}$ is defined in terms of $\tilde{\mathcal{R}}_{ij}$ through n_Π sums of functions $\tilde{\mathcal{R}}_j^{\Pi}$:

$$\tilde{\mathcal{R}}(\mu) = \sum_{j=1}^{n_\Pi} \tilde{\mathcal{R}}_j^{\Pi}(\mu). \tag{27}$$

Thus, the parameter n_Π is the number of $\tilde{\mathcal{R}}_j^{\Pi}$ functions used in the definition of $\tilde{\mathcal{R}}(\mu)$, depending on the reward structure and the SPN. Intuitively, n_Π represents the maximum number of non-zero reward values that contribute to $\tilde{\mathcal{R}}(\mu)$ for a specific marking, i.e., the maximum amount of information that is required to represent the reward attached to a specific marking.

In practice, to define a reward structure $\tilde{\mathcal{R}}$, it is enough to define n_Π (i.e., one or more) reward contributions $\tilde{\mathcal{R}}_j^{\Pi}$. This means to identify, for each $\tilde{\mathcal{R}}_j^{\Pi}$, some set $\bar{\mathbb{S}}_{ij}$ of markings restricted to a submodel, and to define explicitly some $\tilde{\mathcal{R}}_{ij}$ only for the elements of $\bar{\mathbb{S}}_{ij}$. The convention is used that, when $\bar{\mathbb{S}}_{ij} \neq \emptyset$, then $\tilde{\mathcal{R}}_{ij}$ is 0 for all the elements not explicitly assigned, as shown in Equation (25), whereas when $\bar{\mathbb{S}}_{ij} = \emptyset$, then $\tilde{\mathcal{R}}_{ij}$ is defined as shown in Equation (26). Given an SSPN with an implicit reward structure, the implicit performance variables are defined as follows:

$$\tilde{Y}_\infty(\tilde{\mathcal{R}}) = \sum_{\mu \in \mathbb{S}} \int_0^\infty \sum_j \prod_i \tilde{\mathcal{R}}_{ij}(\mu^{(i)}) \tilde{I}_t^{\mu^{(i)}} dt, \tag{28}$$

$$\tilde{Y}_{\infty|\mathcal{B}}(\tilde{\mathcal{R}}) = \tilde{Y}_\infty(\tilde{\mathcal{R}}) \mid \bigvee_{\mu \in \mathcal{B}} [\tilde{I}_\infty^\mu = 1], \tag{29}$$

$$\tilde{V}_\infty(\tilde{\mathcal{R}}) = \sum_{\mu \in \mathcal{B}} \sum_j \prod_i \tilde{\mathcal{R}}_{ij}(\mu^{(i)}) \tilde{I}_\infty^{\mu^{(i)}}, \tag{30}$$

where $\tilde{I}_t^{\mu^{(i)}}$ is the implicit indicator random variable representing the event that M_i is in the marking $\mu^{(i)}$ at time t, and $\tilde{I}_\infty^\mu$ is the implicit indicator random variable defined as follows:

$$\tilde{I}_\infty^\mu = \tilde{I}_\infty^{\mu^{(1)}} \cdot \ldots \cdot \tilde{I}_\infty^{\mu^{(n)}}.$$

Notice that the domain of $\tilde{I}_t^{\mu^{(i)}}$ is the state-space $\mathcal{S}^{(i)}$ of M_i. Also, in general the integral of the product $\int \prod \tilde{I}_t^{\mu^{(i)}}$ is different from the product of the integral $\prod \int \tilde{I}_t^{\mu^{(i)}} = \prod \tilde{J}_t^{\mu^{(i)}}$, where $\tilde{J}_t^{\mu^{(i)}}$ is the random variable representing the total time that M_i is in the marking $\mu^{(i)}$ during the interval of time $[0, t]$. Therefore, the definition of $\tilde{Y}_\infty(\tilde{\mathcal{R}})$ in Equation (28) is not based on J as in Equation (3), but requires the use of the implicit indicator random variables $\tilde{I}_t^{\mu^{(i)}}$, as detailed in Appendix A. In addition, implicit rewards and measures are defined directly on top of each submodel M_i, skipping the definition at level of the overall SSPN model. This is an advantage in terms of modeling expressive power, when the submodels can be defined by a single template model.

The measures of interest are implicitly defined as follows:

$$\tilde{\mathcal{M}}_{k,\tilde{\mathcal{R}}_{ij}} = \mathbb{E}\big[\tilde{Y}_\infty(\tilde{\mathcal{R}})^k\big], \tag{31}$$

$$\tilde{\mathrm{MRTA}}_{|\mathcal{B}} = \mathbb{E}[\tilde{Y}_{\infty|\mathcal{B}}(\tilde{\mathcal{R}})], \tag{32}$$

$$\tilde{\pi}_{\mathcal{B}}(\infty) = \mathbb{E}[\tilde{V}_\infty(\tilde{\mathcal{R}})]. \tag{33}$$

Notice that the reward structure, the performance variables and the measures implicitly defined in Equations (27) to (33) are different representations of the same reward structure, performance variables and measures defined explicitly in Equations (1) to (5), (7) and (8). In particular, the implicit definition of Equation (6), used in Equation (33), for a given $\mathcal{B}$ is

$$\tilde{\mathcal{R}}_{i1}(\mu^{(i)}) = \begin{cases} 1 & \text{if } \mu^{(i)} \in \mathcal{B}^{(i)}, \\ 0 & \text{otherwise,} \end{cases} \tag{34}$$

for each $i = 1, \ldots, n$, where $\mathcal{B}^{(i)}$ is the projection of $\mathcal{B}$, as defined in Equation (23).

For a given SSPN model, at increasing values of n and of the markings of each submodel, the efficiency of the methodology proposed in Section IV-B strongly depends on the total number m_Π, with $m_\Pi \geq n_\Pi$, of non-zero reward values defined (as product of values) for some marking by all the functions $\tilde{\mathcal{R}}_j^\Pi$, for $j = 1, \ldots, n_\Pi$. Formally, the parameter m_Π is defined as

$$m_\Pi = \sum_{\mu} \sum_{j=1}^{n_\Pi} \gamma_j(\mu), \tag{35}$$

with

$$\gamma_j(\mu) = \begin{cases} 1 & \text{if } \tilde{\mathcal{R}}_j^\Pi(\mu) \neq 0, \\ 0 & \text{otherwise.} \end{cases}$$

Intuitively, m_Π represents the total number of non-zero reward values that contribute to $\tilde{\mathcal{R}}$, i.e., the maximum amount of information that is required to represent the specific reward structure for the SPN at hand. Thus, the actual definition of specific reward structures $\tilde{\mathcal{R}}$ should be based on the minimum number of functions $\tilde{\mathcal{R}}_j^\Pi$ and on the minimum number of occurrences of non-zero values explicitly obtained as product of rewards defined by $\tilde{\mathcal{R}}_{ij}$. In practice, most of the measures of interest can be defined by reward structures based only on a few functions $\tilde{\mathcal{R}}_j^\Pi$, as shown in the examples of reward structures in Appendix F. The measures defined using reward rates for which the value of m_Π is not low enough to apply efficiently the methodology of Section IV-B are not of interest for this paper.

B. Modeling power

Now, it can be shown that the new implicit reward structure based on the function $\tilde{\mathcal{R}}$ has the same modeling power as the explicit one based on $\mathcal{R}$. Let $n_r = |\bar{\mathcal{S}}|$ denote the number of non-zero reward values that are output of $\mathcal{R}$. Given a function $\mathcal{R}$, it is always possible to define the functions $\tilde{\mathcal{R}}_{1j}, \ldots, \tilde{\mathcal{R}}_{nj}$, with $j = 1, \ldots, n_\Pi$ and $n_\Pi = n_r$, such that $\tilde{\mathcal{R}}(\mu) = \mathcal{R}(\mu)$, for each $\mu \in \mathcal{S}$.

$\mathcal{R}$ can be rewritten as follows:

$$\mathcal{R}(\mu) = \begin{cases} r_j & \text{if } \mu = \mu_j \in \bar{\mathcal{S}}, \\ 0 & \text{otherwise,} \end{cases} \tag{36}$$

where $r_j \in \mathbb{R} \setminus \{0\}$ is a reward rate that does not depend on the marking and μ_j is the j-th element of $\bar{\mathcal{S}}$, $j = 1, \ldots, n_r$. For example, given $P_1 = \{\ldots, W_1, W_2, \ldots\}$ (the other places are omitted for brevity), if $r_j = sin(\#W_1 + \#W_2)$ when $\#W_1 \leq 3$ and $\#W_2 \leq 3$ then r_j can be replaced by ρ non-zero real numbers, one for each different combination of tokens in W_1 and W_2 (intended as a formal sum), obtaining:

$$\begin{aligned} &r_{j_1} = sin(0+1), \text{for } \mu_{j_1} = W_2, \\ &r_{j_2} = sin(1+0), \text{for } \mu_{j_2} = W_1, \\ &\ldots \\ &r_{j_\rho} = sin(3+3), \text{for } \mu_{j_\rho} = 3W_1 + 3W_2, \end{aligned}$$

where $sin(a+b)$ is replaced by the result obtained applying the sin function to $a+b$.

One can then define $\tilde{\mathcal{R}}_{ij}$, for $i = 1, \ldots, n$ and $j = 1, \ldots, n_\Pi$, as follows:

$$\begin{aligned} &\tilde{\mathcal{R}}_{1j}(\mu^{(1)}) = r_j, \qquad \text{if } \mu^{(1)} = \mu_j^{(1)}, \\ &\tilde{\mathcal{R}}_{ij}(\mu^{(i)}) = 1, \qquad \text{if } \mu^{(i)} = \mu_j^{(i)}, i = 2, \ldots, n, \end{aligned} \tag{37}$$

where explicit reward assignments are considered for all the submodels M_i, i.e.,

$$\bar{\mathcal{S}}_{ij} = \{\mu_j^{(i)} \mid \mu \in \bar{\mathcal{S}}\}, \tag{38}$$

where $\mu_j^{(i)}$ is μ restricted to M_i.

Hence, applying Equation (37) to Equation (24) yields

$$\tilde{\mathcal{R}}_j^\Pi(\mu) = \begin{cases} r_j & \mu = \mu_j \in \bar{\mathcal{S}}, \\ 0 & \text{otherwise,} \end{cases} \tag{39}$$

being

$$\begin{aligned} \tilde{\mathcal{R}}_j^\Pi(\mu_j) &= \tilde{\mathcal{R}}_{1j}(\mu_j^{(1)}) \cdot \tilde{\mathcal{R}}_{2j}(\mu_j^{(2)}) \cdot \ldots \cdot \tilde{\mathcal{R}}_{nj}(\mu_j^{(n)}) \\ &= r_j \cdot 1 \cdot \ldots \cdot 1 = r_j. \end{aligned}$$

From Equations (27) and (39) it is immediate to derive the definition of $\tilde{\mathcal{R}}$, which is equal to the definition given in Equation (36) for $\mathcal{R}$. This proves that $\mathcal{R}$ and $\tilde{\mathcal{R}}$ have the same modeling power. Therefore, for a given model and some given reward variables, the same measures of interest can be modeled using both reward functions. In particular, the mapping from Equation (28) to Equation (3) is trivial, and similarly is the mapping from Equation (31) to Equation (1).

IV. Implicit reward model and solution at Markov chain level

As shown in the bottom right corner of Figure 2, this section presents the implicit expressions based on the Kronecker Algebra that directly map the rewards and measures defined at the stochastic Petri net level to the corresponding ones at the Markov chain level. In more detail, the developments concern: 1) the definition of new shift matrices $\boldsymbol{S}^{\mathcal{A}}$ and $\tilde{\boldsymbol{S}}^{\mathcal{A}}$ with respect to [10] for dealing directly with the infinitesimal generator matrix $\boldsymbol{Q}$ and the corresponding implicit matrix $\tilde{\boldsymbol{Q}}$ instead of working with the transitions $\boldsymbol{Q}_{\mathcal{T}}$ among transient states; and 2) the formal definition of implicit and generic reward vectors $\tilde{\boldsymbol{r}}$, $\tilde{\boldsymbol{\pi}}(0)$ and $\tilde{\boldsymbol{e}}_{\mathcal{A}}$. In particular, a new formulation is given for $\tilde{\boldsymbol{r}}$, which has already been defined in [10], accordingly to the new addressed measures. Then, in Section IV-B the Tensor-Trains format to store matrices and vectors is first introduced, and then it is shown how it is exploited in the solution methodology proposed to evaluate the measures.

A. Implicit modeling through Kronecker Algebra

As shown in Figure 1, the SSPN model M under consideration is described in terms of a certain number of interacting submodels M_i. The number n_{reach} of reachable states in the underlying Markov chain often depends exponentially on the number of such submodels. This affects the computational complexity of numerically dealing with such stochastic processes, and is a phenomenon known as *curse of dimensionality*. Several strategies have been proposed to overcome this difficulty, and

mostly rely on tensor representation of the data; we refer the reader to the survey [27] for further details on this matter.

These approaches find representations of the infinitesimal generator with a reduced set of parameters. Among these strategies, referred to as *implicit*, the Kronecker based ones [14] fit particularly well with the logical structure of the systems under analysis and the properties of matrices and vectors described in Section II-B. In the following, the SAN formalism [13], [18] will be adopted.

Instead of studying directly the Markov chain underlying the system model M, as done in Section II-B, exploring the state-space $\mathcal{S}$ and addressing the Markov chain $\{X(t)\}$, here it is taken into account that the model M is structured in n interacting submodels. Thus, the underlying Markov chain is represented by $\{\tilde{X}(t) \mid t \geq 0\}$, where $\tilde{X}(t) = (X^{(1)}(t), \ldots, X^{(n)}(t))$, $\{X^{(i)}(t) \in \mathcal{S}^{(i)}\}$ is the Markov chain underlying M_i and $\mathcal{S}^{(i)}$ is the state-space of M_i.

Let $\tilde{\mathcal{S}} = \mathcal{S}^{(1)} \times \cdots \times \mathcal{S}^{(n)}$ be the potential state-space[1]. Notice that $\tilde{\mathcal{S}}$ includes the actual state-space $\mathcal{S}$, i.e., there exists an injection Φ such that $\Phi\big(\mathcal{S}\big) \subseteq \tilde{\mathcal{S}}$. In order to define $\tilde{\boldsymbol{Q}}$, the infinitesimal generator matrix of $\{\tilde{X}(t)\}$, trough sums and products of Kronecker, $\{\tilde{X}(t)\}$ is represented by the SAN formalism [18] as a set of n stochastic automata $\tilde{M} = \{\tilde{M}_1, \ldots, \tilde{M}_n\}$ interacting through *synchronization* transitions. Each $\tilde{M}_i$ represents $\{X^{(i)}(t)\}$ as a (small) reachable set of states $\mathcal{S}^{(i)}$ and two types of transitions, which map directly those of the corresponding high level model M_i: *local* transitions, which are local to $\tilde{M}_i$ and impact only on $\mathcal{S}^{(i)}$, and *synchronization* transitions, which can appear in, and impact on, more than one $\tilde{M}_i$.

Also, following the SAN formalism, and under reasonable conditions, it is possible to guarantee that the stochastic process $\tilde{X}(t)$, which spans $\mathcal{S}$, is indistinguishable from $X(t)$. $\mathcal{A}^{(i)} = \{a_1^{(i)}, a_2^{(i)}, \ldots, a_{|\mathcal{A}^{(i)}|}^{(i)}\} \subseteq \mathcal{S}^{(i)}$ denotes the set of absorbing states for the chain $\{X^{(i)}(t)\}$, $\tilde{\mathcal{A}} = \mathcal{A}^{(1)} \times \ldots \times \mathcal{A}^{(n)}$ denotes the set of potential absorbing states, and notice that $\Phi\big(\mathcal{A}\big) = \tilde{\mathcal{A}}$. Exploiting the *Kronecker product* on two matrices $\boldsymbol{A} = [a_{ij}] \in \mathbb{R}^{m\times n}$ and $\boldsymbol{B} \in \mathbb{R}^{h\times k}$ defined as:

$$\boldsymbol{A} \otimes \boldsymbol{B} = \begin{bmatrix} a_{11}\boldsymbol{B} & a_{12}\boldsymbol{B} & \cdots & a_{1n}\boldsymbol{B} \\ \vdots & \vdots & \ldots & \vdots \\ a_{m1}\boldsymbol{B} & a_{m2}\boldsymbol{B} & \cdots & a_{mn}\boldsymbol{B} \end{bmatrix} \in \mathbb{R}^{mh\times nk},$$

it is possible to express formally the implicit state probability $\tilde{\boldsymbol{\pi}}(0)$ as:

$$\tilde{\boldsymbol{\pi}}(0) = \boldsymbol{\pi}^{(1)}(0) \otimes \boldsymbol{\pi}^{(2)}(0) \otimes \ldots \otimes \boldsymbol{\pi}^{(n)}(0),$$

imposing that $\pi_j^{(i)}(0) = 0$ if $j \in \mathcal{A}^{(i)}$. Instead, $\tilde{\boldsymbol{Q}}$, the implicit infinitesimal generator matrix of $\{\tilde{X}(t)\}$, is expressed in terms of a compressed representation called descriptor matrix, as:

$$\tilde{\boldsymbol{Q}} = \boldsymbol{R} + \boldsymbol{W} + \boldsymbol{\Delta},$$

[1]Although we mention the space of potential states, adopted notation and solution method do not distinguish reachable states from unreachable ones, being based on both implicit vectors and implicit matrices.

where

$$\boldsymbol{R} = \bigoplus_{i=1}^{n} \boldsymbol{R}^{(i)}, \qquad \boldsymbol{W} = \sum_{\xi\in\mathcal{ST}} \bigotimes_{i=1}^{n} \boldsymbol{W}^{(\xi,i)}, \tag{40}$$

$\mathcal{ST}$ is the set of all synchronization transitions, and $\boldsymbol{R}$ and $\boldsymbol{W}$ are $|\mathcal{S}^{(i)}| \times |\mathcal{S}^{(i)}|$ matrices that represent, respectively, the local contribution (i.e., $\tilde{M}_i$ limited to the local transitions only) and the synchronization contribution (i.e., the impact of each synchronization transition on each $\tilde{M}_i$). $\boldsymbol{\Delta}$ is the diagonal matrix defined as $\boldsymbol{\Delta} = -\mathrm{diag}((\boldsymbol{R} + \boldsymbol{W})\boldsymbol{e}))$. The operator $\oplus$ is the *Kronecker sum*. A complete characterization can be found in [14]. Also, the implicit reward column vector $\tilde{\boldsymbol{r}} = (r_1, \ldots, r_{\underline{n}_{\text{reach}}})^T$ is obtained from the explicit SPN level reward function $\tilde{\mathcal{R}}_{ij}$, switching from sum of products of Equation (27) to sums of Kronecker products as follows for each state $\tilde{\boldsymbol{s}} \in \tilde{\mathcal{S}}$, with $\tilde{\boldsymbol{s}} = (s_1, \ldots, s_n)$:

$$\tilde{r}_{\tilde{\boldsymbol{s}}} = \tilde{r}_{s_1,\ldots,s_n} = \sum_{j-1}^{n_\Pi} \bigotimes_{i=1}^{n} \tilde{r}_{s_i}^{i,j} \tag{41}$$

where, fixed i and j,

$$\tilde{r}_{s_i}^{i,j} = \begin{cases} \tilde{\mathcal{R}}_{ij}(\mu^{(i)}) & \text{if } \mu^{(i)} = \mu_{s_i}^{(i)} \in \mathcal{S}^{(i)}, \\ 0 & \text{otherwise,} \end{cases} \tag{42}$$

where $\mu_{s_i}^{(i)}$ is the s_i-th element of $\mathcal{S}^{(i)}$.

A clear correspondence is then established between $\tilde{V}_t(\tilde{\mathcal{R}})$ and $\tilde{V}_t(\tilde{\boldsymbol{r}})$, and $\tilde{Y}_t(\tilde{\mathcal{R}})$ and $\tilde{Y}_t(\tilde{\boldsymbol{r}})$, as detailed in Appendices A and B. As in [10], if each $X^{(i)}(t)$ has a single absorbing state, then $\tilde{X}(t)$ has a single absorbing state. In addition, if the absorbing state of $X^{(i)}(t)$ is the last state in $\mathcal{S}^{(i)}$ for each $i = 1, \ldots, n$, then (because of the lexicographical ordering dictated by the Kronecker product) the absorbing state of $\tilde{X}(t)$ is the last state in $\tilde{\mathcal{S}}$. This can be extended to the case of multiple absorbing states, i.e., to a generic set $\mathcal{A}$, with $|\mathcal{A}| > 0$.

Switching from explicit to implicit representation, the implicit submatrix $\tilde{\boldsymbol{Q}}_{\mathcal{T}}$, corresponding to the submatrix $\boldsymbol{Q}_{\mathcal{T}}$ used in Section II-B in the explicit expressions for the measures of interest, is infeasible to be defined through a permutation of the original matrix $\tilde{\boldsymbol{Q}}$ that switches the last states with the states in $\mathcal{A}$. Instead, the shift matrix $\boldsymbol{S}^{\mathcal{A}}$, defined as:

$$\boldsymbol{S}^{\mathcal{A}} = \boldsymbol{Q}(\mathrm{diag}(\boldsymbol{e}_{\mathcal{A}}) + \boldsymbol{I}), \tag{43}$$

where $\boldsymbol{I}$ is the identity matrix, is introduced so that

$$\big(\boldsymbol{Q} - \boldsymbol{S}^{\mathcal{A}}\big)^{-1} = \begin{bmatrix} \boldsymbol{Q}_{\mathcal{T}} & \\ & -\boldsymbol{I} \end{bmatrix}^{-1} = \begin{bmatrix} \boldsymbol{Q}_{\mathcal{T}}^{-1} & \\ & -\boldsymbol{I} \end{bmatrix},$$

since the inverse of a block diagonal matrix is a block diagonal matrix whose blocks are the inverse of those of the original matrix. Then

$$\begin{aligned} -\boldsymbol{\pi}^T(0)(\boldsymbol{Q}-\boldsymbol{S}^{\mathcal{A}})^{-1}\mathrm{diag}(\boldsymbol{r}) & \\ = -\boldsymbol{\pi}_{\mathcal{T}}^T(0)\boldsymbol{Q}_{\mathcal{T}}^{-1}\mathrm{diag}(\boldsymbol{r}_{\mathcal{T}}) + \boldsymbol{\pi}_{\mathcal{A}}^T(0)\cdot 0 & \\ = -\boldsymbol{\pi}_{\mathcal{T}}^T(0)\boldsymbol{Q}_{\mathcal{T}}^{-1}\mathrm{diag}(\boldsymbol{r}_{\mathcal{T}}). & \end{aligned} \tag{44}$$

Thus, applying Equation (44), Equations (13), (18) and (19) can be rewritten by replacing $\boldsymbol{\pi}^T_{\mathcal{T}}(0)$, $\boldsymbol{Q}_{\mathcal{T}}$ and $\boldsymbol{r}_{\mathcal{T}}$ with $\boldsymbol{\pi}^T(0)$, $\boldsymbol{Q}-\boldsymbol{S}^{\mathcal{A}}$ and $\boldsymbol{r}$, respectively, obtaining:

$$m_{k,r} = k!(-1)^k \boldsymbol{\pi}^T(0)((\boldsymbol{Q}-\boldsymbol{S}^{\mathcal{A}})^{-1}\mathrm{diag}(\boldsymbol{r}))^{k-1} (\boldsymbol{Q}-\boldsymbol{S}^{\mathcal{A}})^{-1}\boldsymbol{r}, \tag{45}$$

$$\mathbb{E}[Y_\infty(\mathcal{R}) \text{ and } X(\infty)=a] = \boldsymbol{\pi}^T(0)(\boldsymbol{Q}-\boldsymbol{S}^{\mathcal{A}})^{-1}\mathrm{diag}(\boldsymbol{r})(\boldsymbol{Q}-\boldsymbol{S}^{\mathcal{A}})^{-1}\boldsymbol{v}_a, \tag{46}$$

$$\pi_a(\infty) = -\boldsymbol{\pi}^T(0)(\boldsymbol{Q}-\boldsymbol{S}^{\mathcal{A}})^{-1}\boldsymbol{v}_a. \tag{47}$$

Introducing $\tilde{\boldsymbol{e}} = \boldsymbol{e}\otimes\ldots\otimes\boldsymbol{e}$, $\tilde{\boldsymbol{e}}_{\mathcal{A}} = \boldsymbol{e}_{\mathcal{A}^{(1)}}\otimes\ldots\otimes\boldsymbol{e}_{\mathcal{A}^{(n)}}$ and observing that $\tilde{\boldsymbol{S}}^{\mathcal{A}}$ corresponds exactly to $\boldsymbol{S}^{\mathcal{A}}$, i.e.,

$$\tilde{\boldsymbol{S}}^{\mathcal{A}} = \tilde{\boldsymbol{Q}}\cdot\big(\mathrm{diag}(\boldsymbol{e}_{\mathcal{A}^{(1)}})\otimes\cdots\otimes\mathrm{diag}(\boldsymbol{e}_{\mathcal{A}^{(n)}})+\boldsymbol{I}\otimes\cdots\otimes\boldsymbol{I}\big), \tag{48}$$

the implicit expressions of Equations (45) to (47) can be easily defined replacing the explicit symbols with the corresponding implicit ones and obtaining respectively:

$$\tilde{m}_{k,r} = k!(-1)^k \tilde{\boldsymbol{\pi}}^T(0)((\tilde{\boldsymbol{Q}}-\tilde{\boldsymbol{S}}^{\mathcal{A}})^{-1}\mathrm{diag}(\tilde{\boldsymbol{r}}))^{k-1} (\tilde{\boldsymbol{Q}}-\tilde{\boldsymbol{S}}^{\mathcal{A}})^{-1}\tilde{\boldsymbol{r}}, \tag{49}$$

$$\mathbb{E}[\tilde{Y}_\infty(\tilde{\mathcal{R}}) \text{ and } \tilde{X}(\infty)=a] = \tilde{\boldsymbol{\pi}}^T(0)(\tilde{\boldsymbol{Q}}-\tilde{\boldsymbol{S}}^{\mathcal{A}})^{-1}\mathrm{diag}(\tilde{\boldsymbol{r}})(\tilde{\boldsymbol{Q}}-\tilde{\boldsymbol{S}}^{\mathcal{A}})^{-1}\tilde{\boldsymbol{Q}}\tilde{\boldsymbol{e}}_a, \tag{50}$$

$$\tilde{\pi}_a(\infty) = -\tilde{\boldsymbol{\pi}}^T(0)(\tilde{\boldsymbol{Q}}-\tilde{\boldsymbol{S}}^{\mathcal{A}})^{-1}\tilde{\boldsymbol{Q}}\tilde{\boldsymbol{e}}_a, \tag{51}$$

where $\tilde{\boldsymbol{e}}_a$ is the implicit vector corresponding to $\boldsymbol{e}_a$, and $\tilde{\boldsymbol{Q}}\tilde{\boldsymbol{e}}_a = (\boldsymbol{v}_a^T, 0, \ldots, 0)^T$. Here, it can be appreciated the ability to define $\tilde{\boldsymbol{e}}_a$ implicitly in terms of Kronecker products because it directly relates to the implicit indicator random variable $\tilde{I}^a_\infty$. Particular cases of Equations (49) and (50) are:

$$\tilde{m}_k(T_{\mathcal{A}}) = (-1)^k \tilde{\boldsymbol{\pi}}^T(0)(\tilde{\boldsymbol{Q}}-\tilde{\boldsymbol{S}}^{\mathcal{A}})^{-k}(\tilde{\boldsymbol{e}}-\tilde{\boldsymbol{e}}_{\mathcal{A}}), \tag{52}$$

$$\mathbb{E}[T_{\mathcal{A}} \text{ and } \tilde{X}(\infty)=a] = \tilde{\boldsymbol{\pi}}^T(0)(\tilde{\boldsymbol{Q}}-\tilde{\boldsymbol{S}}^{\mathcal{A}})^{-2}\tilde{\boldsymbol{Q}}\tilde{\boldsymbol{e}}_a. \tag{53}$$

The implicit formulas for $\tilde{\mathrm{MRTA}}_{|\mathcal{B}}$ and $\tilde{\pi}_{\mathcal{B}}(\infty)$ can be easily obtained by replacing $\tilde{\boldsymbol{e}}_a$ (or the corresponding $\boldsymbol{v}_a$) with $\tilde{\boldsymbol{e}}_{\mathcal{B}}$ (or $\boldsymbol{v}_{\mathcal{B}}$) in Equations (50) and (51), with $\tilde{\boldsymbol{e}}_{\mathcal{B}} = \sum_{a\in\mathcal{B}}\tilde{\boldsymbol{e}}_a$.

All the manipulations of this subsection preserve the Kronecker structure, and in particular the evaluation of Equations (45), (50) and (51) takes advantage of that.

B. Solution method

The computation method presented in [23], [26], [28] can be easily generalized to the evaluation of Equations (45) to (47), following an approach similar to that adopted to evaluate Equation (13), which is based on the solution of a chain of at least one linear system and then a dot product, as shown in Equations (14) and (15). In order to adapt the explicit computation of Equations (45) to (47) to implicitly evaluate Equations (49) to (51), it is necessary to choose a compressed representation of $\tilde{\boldsymbol{Q}}$, $\tilde{\boldsymbol{S}}^{\mathcal{A}}$, $\tilde{\boldsymbol{\pi}}^T(0)$, $\tilde{\boldsymbol{r}}$, the implicit solution vectors $\tilde{\boldsymbol{x}}^{(k)}$ and in general all the data structures involved in the solution of the linear systems. In fact, the product of sums of Kronecker products can be represented as the sum of Kronecker products, but the number of addends is so high that, in the long run, it soon becomes too large to handle without compression.

For the computations in this paper the tensor trains representation has been selected and adopted also for vectors as detailed below, thus extending the tensor trains representation proposed in [10]. Recalling that the potential state-space $\tilde{\mathcal{S}}$ is the Cartesian product of the component state spaces $\mathcal{S}^{(i)}$ for $i=1,\ldots,n$, each vector $\boldsymbol{u}=(u_1,\ldots,u_{|\tilde{\mathcal{S}}|})$ involved in the solution of Equations (49) to (51) can be expressed in *tensor form*, as an array $\tilde{\boldsymbol{u}}$ with n indices, such that $u_k = \tilde{u}_{k_1,\ldots,k_n}$. An approximate low-rank representation for this tensor $\tilde{\boldsymbol{u}}$, known as a *Canonic Polyadic Decomposition* (CPD), is

$$\tilde{u}_{k_1,\ldots,k_n} \approx \sum_{j\in\mathcal{I}} \tilde{u}^j_{k_1}\otimes\ldots\otimes\tilde{u}^j_{k_n},$$

where, for some finite index set $\mathcal{I}$, the minimal cardinality of $\mathcal{I}$ is the *tensor rank* of the approximation of $\tilde{\boldsymbol{u}}$. Unfortunately, computing such approximation can be rather challenging, both from the mathematical and numerical perspective [27].

Hence, several alternative low-rank formats have been proposed over the years for tensors. In this work, the Tensor Train (TT) format is considered [29]. A fast Tensor Train Singular Value Decomposition (TT-SVD) [29] can be designed, which allows to perform recompression of the tensors under consideration after performing arithmetic operation. Thus, iterative methods for solving linear systems are enabled in this context. In particular, if the low-rank structure is preserved throughout the computations, this format allows to avoid the exponential complexity in the number of dimensions.

A tensor train representation of the tensor form of a vector $\tilde{\boldsymbol{u}}$ is a collection of order 3 tensors $\boldsymbol{\mathcal{N}}^{(i)}$ of size $\rho_i\times n_i\times\rho_{i+1}$ such that[2] $\rho_1=\rho_n=1$, and

$$\tilde{u}_{k_1,\ldots,k_n} = \sum_{h_1,\ldots,h_{n-1}} \mathcal{N}^{(1)}_{1,k_1,h_1}\mathcal{N}^{(2)}_{h_1,k_2,h_2}\cdot\ldots\cdot\mathcal{N}^{(n)}_{h_{n-1},k_n,1}. \tag{54}$$

The tensors $\boldsymbol{\mathcal{N}}^{(i)}$ are called *carriages*, hence the name *tensor train* [29]. The tuple $(\rho_2,\ldots,\rho_{n-1})$ is called the TT-rank of the tensor $\tilde{\boldsymbol{u}}$.

Similarly, matrices $|\tilde{\mathcal{S}}|\times|\tilde{\mathcal{S}}|$ can be represented as tensors by subdividing the row and column indices. More precisely, a matrix $\boldsymbol{A}\in\mathbb{C}^{n_1\cdots n_n\times n_1\cdots n_n}$ can be viewed (up to permuting the indices) as a larger vector of the vector space $\mathbb{C}^{n_1^2\times\ldots\times n_n^2}$. This vector can be stored in the TT-format as described in (54). Such an arrangement makes computing matrix-vector product and matrix-matrix product relatively simple to implement; we refer the reader to [29] for further details.

After each matrix-vector multiplication and vector-vector or matrix-matrix addition, the TT-rank tends to grow and then managing the data structure is more and more computationally intensive. Likely, it is possible to *round* all the tensor trains, applying the TT-SVD decomposition, so that the TT-rank is made closer to the actual amount of information stored within the tensors.

[2] In particular, $\boldsymbol{\mathcal{N}}^{(1)}$ and $\boldsymbol{\mathcal{N}}^{(n)}$ are matrices, instead of order 3 tensors, because they have one dimension with only 1 index.

It can be easily proved that any Kronecker sum with ℓ terms yields a tensor train vector of TT-ranks bounded by ℓ (component-wise). More specifically, the tensor rank (obtained through the CPD decomposition) is always a component-wise bound for the TT-ranks. Hence, to efficiently exploit this structure one should make sure that in Equation (27) the number n_Π is not too large. Throughout the iterations, the ranks are kept under control by repeatedly recompressing the tensors using the TT-SVD decomposition, as described above. The experiments have shown that the ranks tend to remain bounded when the topology of the network is close to a linear graph. This closely mimics the order of multiplication in the carriages of the tensor train representation. It can be conjectured that, in case of more complex topologies, it may be beneficial to consider tensor representations obtained by constructing carriages with indices multiplied with similar interconnections; examples include *tensor networks* and the *hierarchical Tucker format*. For further information on these more complex data-structures, the reader is referred to [27].

The original explicit and implicit definitions of the shift matrices $\boldsymbol{S}^{\mathcal{A}}$ and $\tilde{\boldsymbol{S}}^{\mathcal{A}}$, as shown in Equations (43) and (48), respectively, promote the adoption of a different solver with respect to that used in [10], where a different definition of shift matrix is proposed. In particular, among the two most known linear system solvers already available for the tensor trains (*DMRG* and *AMEn*), here *AMEn* [30] has been selected. AMEn is an iterative optimization-based solver. In each iteration, it selects one by one, back and forth, the n dimensions of the problem and minimizes the residue in the selected dimension. The key idea is to maintain the TT-ranks sufficiently small during the iterations to allow AMEn to cover enough search space. How small the TT-ranks should be depends on the available RAM, while how many iterations are needed to reach the desired precision depends on the model at hand. As for all optimization methods, sometimes the procedure stagnates in a local minimum and then no acceptable solution is obtained.

From the above description, model characteristics that are expected to impact the performance of the method are:

MC1: the number of submodels n. This corresponds to the number of carriages in TT format for matrices and vectors and has no direct effect on the TT ranks,

MC2: the number of synchronization transitions $|\mathcal{ST}|$. The TT-ranks can grow sensibly after an addition (even after rounding), so the number of addends in Equation (40) directly impacts on the memory footprint of the vectors,

MC3: the number m_Π of non-zero reward values that contribute to $\tilde{\mathcal{R}}(\mu)$ in Equation (27). As for $|\mathcal{ST}|$, additions can lead to intractable TT-ranks,

MC4: the number of absorbing states $|\mathcal{B}|$ involved in conditioned measures,

MC5: dimension and complexity (seen from the SVD point of view) of $\boldsymbol{R}^{(i)}$ and $\boldsymbol{W}^{(\xi,i)}$ in Equation (40). As it will be discussed in Section VII, only if the memory occupancy of $\tilde{\boldsymbol{Q}}$ is negligible with respect to that of the vectors the method has chances to be useful in practice,

MC6: stiffness. Many solution methods can have troubles if the ratio between the largest and smallest rate in the infinitesimal generator matrix is large. This may arise in dependability models, for example, when fault (error) handling (recovery) rates are several orders of magnitude (sometimes 10^6 times) larger than fault (error) occurrence rates [31]. Stiffness impacts on TT rounding too.

The kind of system models, where the presented method is expected to perform at best, is that with a large number of simple submodels, where $|\mathcal{ST}| \approx n$, the reward structures involve small values of m_Π (e.g., [10]) and only relatively few of the absorbing states are of interest.

The implementation of the solution method has been written in MATLAB [32] and is freely available[3].

V. Case study

To better understand how the method of Section IV-B behaves, a parametric case study has been designed taking into account realistic scenarios on the one hand, and the possibility of varying those parameters that are expected to have impact on the performance of the system analysis on the other hand. In order to quantitatively assess the impact of the model characteristics MC1-MC6 discussed above, the organization of the case study accounts for submodels of increasing complexity, two scenarios (each one characterized by different values of $|\mathcal{ST}|$), two reward structures (one for the time to absorption and the other for a more general performability measure) and a tunable $\mathcal{B}$.

A. Case study description

The case study consists of a computer system composed of n interconnected components $\mathcal{C}_1, \ldots, \mathcal{C}_n$, properly functioning at time 0, but that can be affected by erroneous status (caused by software or hardware faults), and equipped with error detection and recovery capabilities. Each component provides a certain service, the correctness of which may depend on the (state) information it receives from other components. Interconnections between components can be logical or physical. An interconnection is the ability to receive or send service state information. The components are constrained to communicate only with certain other components, for example for security reasons. Error propagation consists of sending and receiving messages with incorrect information, which makes the recipient's state inconsistent (erroneous) with the service it must provide.

In a properly functioning component $\mathcal{C}_i$, a fault occurs after an exponentially distributed random time with rate λ_i^E and leads to an erroneous component state. In a component $\mathcal{C}_i$ exhibiting an erroneous state, two random events can occur after an exponentially distributed random time with rate λ_i^L and λ_i^{EP}, respectively: the manifestation of the error or the propagation of the error to those components directly interconnected to it. When the error manifests itself, a component $\mathcal{C}_i$ may: 1) detect the erroneous state, with coverage η_i, and attempt to recover the correct state, or 2) not detect the erroneous state, with

[3]Code available at https://github.com/numpi/kaes. Specifically, the AMEn solver is `amen_block_solve` provided by the https://github.com/oseledets/TT-Toolbox.

probability $1-\eta_i$, and move into an unsafe state, corresponding to a permanent failure of $\mathcal{C}_i$ (service failure) denoted by $\mathcal{F}_C$. The recovery ends after an exponentially distributed random time with rate λ_i^R. It is successful with some coverage ζ_i, when $\mathcal{C}_i$ returns to be properly functioning; otherwise, with probability $1-\zeta_i$, $\mathcal{C}_i$ moves in a safe state where the service is stopped, corresponding to a permanent failure of $\mathcal{C}_i$ denoted by $\mathcal{F}_B$. To further differentiate the components, we assume that the first k components $\mathcal{C}_1, \ldots, \mathcal{C}_k$, with $0 < k \leq n$, can be affected by both $\mathcal{F}_B$ and $\mathcal{F}_C$ failures. The remaining $n-k$ components can be affected only by $\mathcal{F}_B$ failures, i.e., $\eta_i = 1$ for $i = k+1, \ldots, n$. The random times considered in each $\mathcal{C}_i$ are stochastically independent.

Let $\bar{\Delta}_i = h_1, h_2, \ldots, h_{\bar{\delta}_i}$ and $\Delta_i = \{j_1, j_2, \ldots, j_{\delta_i}\}$ be, respectively, the list of the $\bar{\delta}_i$ indices of the components towards which the error of $\mathcal{C}_i$ can propagate and the list of the δ_i indices of the components whose error can propagate to $\mathcal{C}_i$. The topology of interactions among components is given by the $n \times n$ adjacency matrix $\boldsymbol{\mathcal{T}} = [\mathcal{T}_{i,j}]$, where $\mathcal{T}_{i,j} = 1$ if $j \in \bar{\Delta}_i$, else $\mathcal{T}_{i,j} = 0$. Thus, $\boldsymbol{\mathcal{T}}$ defines an oriented graph that represents how the n components depend on each other and how they are connected to form the overall system.

Two system failure models are considered, thus originating two different scenarios from the analysis point of view.

In $Scenario1$, the system continues to work after the occurrence of $\mathcal{F}_C$ or $\mathcal{F}_B$ failures, although in a degraded manner, until all components fail.

In $Scenario2$, $\mathcal{F}_C$ is associated to the catastrophic failure of the overall system, while $\mathcal{F}_B$ is considered a benign failure of the component. The system continues to work even after the occurrence of $\mathcal{F}_B$ failures, although in a degraded manner, until the first $\mathcal{F}_C$ failure or until the $\mathcal{F}_B$ failure of all components. Moreover in $Scenario2$, each $\mathcal{C}_i$ is structured in two stochastically independent units, each of which can be affected by error, such that one or two erroneous states may occur, be detected and recovered. Each unit behaves like the individual component described above, but when an error propagates to those components directly interconnected to it, both units of these components move to an erroneous state.

This class of systems is sufficiently representative of some real systems, for example a multi-service internet platform, where components are grouped by different service providers (banking, sales, transportation, etc.) so that not all components can communicate with all the others, for reasons of security or even existing physical interconnection. Another realistic example could be an electrical system, where the components are substations grouped by different operators that interact for the control of the electrical network and therefore communications between components are constrained by privacy reasons, so not everyone can communicate with everyone else.

B. Models of the case study

The system of the case study is modeled through the SSPN formalism. The overall model is obtained composing the submodels M_i for $i = 1, \ldots, n$, exploiting the transition-synchronization approach, as dictated by the SSPN formalism. The submodels for $Scenario1$ and $Scenario2$ are depicted in Figures 3 and 4, respectively.

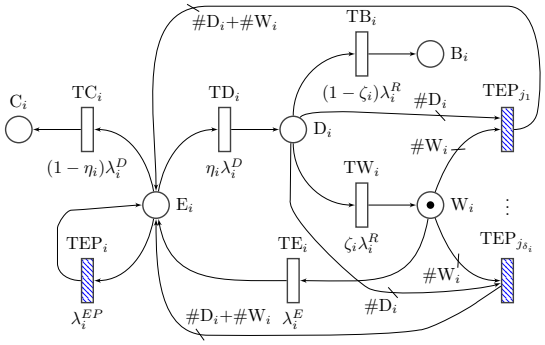


Fig. 3. $Scenario1$: SSPN model M_i. Pattern-filled transitions are synchronized based on the topology.

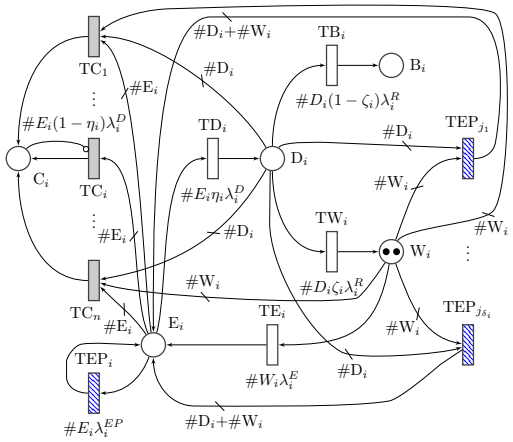


Fig. 4. $Scenario2$: SSPN model M_i. Shaded and pattern-filled transitions are synchronization transitions, with the latter synchronized based on the topology.

In Figure 3, the places W_i (initialized with one token), E_i, D_i, B_i and C_i (initialized with no token) are local to M_i and represent the states where, respectively, $\mathcal{C}_i$ works properly (one token in W_i), $\mathcal{C}_i$ is affected by an error (one token in E_i), $\mathcal{C}_i$ has detected the error (one token in D_i), an $\mathcal{F}_B$ (one token in B_i) and an $\mathcal{F}_C$ (one token in C_i) failure occurred.

The exponentially distributed transitions TE_i, TD_i, TW_i and TB_i are local to M_i and represent, respectively: the time to the occurrence of an error from the state where $\mathcal{C}_i$ works properly (i.e., $W_i = 1$), the time to detect the error (when $E_i = 1$), the time to recover from the erroneous state (when $D_i = 1$) and the time to the occurrence of $\mathcal{F}_B$ failure (when $B_i = 1$). The transitions TEP_i and TEP_{j_k} with $k = 1, \ldots, \delta_i$ are synchronization transitions used to synchronize the submodels M_i, i.e., to propagate the error that affects $\mathcal{C}_i$ to its neighbors and the error that affects each neighbor to $\mathcal{C}_i$. TEP_i represents the exponentially distributed time to the propagation of the error to $\mathcal{C}_h$, with $h \in \bar{\Delta}_i$. TEP_i is replicated in M_i and M_h, for each $h \in \bar{\Delta}_i$. In each M_h, it exists a transition TEP_{j_k} with $j_k = i$, synchronized with TEP_i, that propagates the failure occurred in M_i. The transitions TEP_{j_k} for each $k = 1, \ldots, \delta_i$

represent the time to the propagation of the error from $\mathcal{C}_{j_k}$ to $\mathcal{C}_i$. Each transition TEP_{j_k} is replicated in M_i and M_h with $j_k \in \bar{\Delta}_h$. In each M_i exists a transition TEP_h with $j_k = h$, synchronized with TEP_{j_k}, that represents the error propagation from $\mathcal{C}_h$ to $\mathcal{C}_i$.

A synchronized transition is enabled when it, and all the transitions synchronized with it, have concession. As shown in Figure 3, the transition TEP_i has concession when there is one token in the place E_i. All the transitions TEP_{j_k}, for $k = 1, \ldots, \delta_i$, always have concession, being the multiplicity of each input arc equal to $\#D_i$ or $\#W_i$, i.e., the number of tokens in the corresponding input place D_i and W_i, respectively. Thus, TEP_i is enabled as soon as there is one token in E_i. The firing of TEP_i occurs simultaneously in M_i and M_h, for each $h \in \bar{\Delta}_i$. In M_i, the firing of TEP_i removes the token from E_i and adds one token to E_i, thus the erroneous state of M_i does not change. In M_h, as shown in Figures 3 and 4 replacing i with h, the firing of TEP_{j_k}, with $j_k = i$, removes the token from W_h or D_h (if any, notice that $0 \leq \#W_h + \#D_h \leq 1$) and adds $\#W_h + \#D_h$ tokens to E_h. Thus the error of $\mathcal{C}_i$ only propagates to the not yet failed $\mathcal{C}_h$ ($\#W_h = 1$ or $\#D_h = 1$, i.e., $\#W_h + \#D_h = 1$), and without adding new tokens to E_h when the state of $\mathcal{C}_h$ is already undetected erroneous, i.e., if $\#E_h = 1$ and $\#W_h + \#D_h = 0$. The exponentially distributed transition TC_i is local to M_i in $Scenario1$ and represents the time to the occurrence of the $\mathcal{F}_C$ failure (when $E_i = 1$).

In Figure 4 the place W_i is initialized with 2 tokens, representing the state in which both units of $\mathcal{C}_i$ are working correctly. Thus, compared to the model of Figure 3, the places E_i, D_i and B_i can also contain 2 tokens. Transitions have a transition rate that depends on the tokens in the input place, as shown in Figure 4. Moreover, differently from Figure 3, TC_i is synchronized with TC_h for $h \neq i$. Thus, the firing of TC_h occurs simultaneously with the firing of TC_i with $i \neq h$, removing the tokens from D_i, E_i and W_i and adding one token to C_i. The opposite happens for the firing of TC_i. The inhibitor arc from C_i disables TC_i after the first firing.

Notice that $Scenario1$ and $Scenario2$ are characterized by a different number of synchronization transitions and a different number of absorbing states (reflecting model characteristics MC2 and MC4, respectively).

C. Measures of interest

The definition of the measures of interest has been guided by the objective of assessing: i) the ability of the approach to cope with a good variety of analysis needs; ii) the utility of the k-moments instead of just the mean and variance; and iii) the efficiency of the model solution technique. To this purpose, the measure evaluated for both case studies is:

M1: the first 4 moments of $T_{\mathcal{A}}$, exploited to compute MTTF, $Var(T_{\mathcal{A}})$, skewness and kurtosis, useful to gain insights on the "shape" of the distribution of $T_{\mathcal{A}}$ [33]. Here, the reward structure is the one detailed in Appendix F3.

Additional measures, specific for the two scenarios, have been also considered. In particular, for $Scenario1$, the MTTF conditioned to the following configurations is evaluated:

S1M2: the first 4 moments of Y_∞ where, similarly to the example shown in Appendix F1, the reward structure is

$$\tilde{\mathcal{R}}_{ij}(\mu) = 3, \text{ if } \#W_i = 1,\ i = j = 1, \ldots, n,$$
$$\tilde{\mathcal{R}}_{ij}(\mu) = 1, \text{ if } \#D_i = 1,\ i = j = 1, \ldots, n,$$
$$\tilde{\mathcal{R}}_{ij}(\mu) = 1, \text{ if } i \neq j \text{ and } i, j = 1, \ldots, n,$$

i.e., in each component, the working state produces a gain of 3, the error detection produces a gain of 1 and all the other states produce no gain.

S1M3: $\text{MRTA}_{|\mathcal{B}}$, where all components end up in $\mathcal{F}_C$. It is evaluated considering the reward structure detailed in Appendix F3 and, labeling the first k components as the ones with two absorbing states, the indicator random variables are defined by

$$\mathcal{B}^{(i)} = \{\mu^{(i)} \mid \#B_i = 0,\ \#C_i = 1\}, \qquad i = 1, \ldots, k,$$
$$\mathcal{B}^{(i)} = \{\mu^{(i)} \mid \#C_i = 1\}, \qquad i = k+1, \ldots, n.$$

S1M4: k in $\mathcal{F}_B$ and $n - k$ components in $\mathcal{F}_C$. Here

$$\mathcal{B}^{(i)} = \{\mu^{(i)} \mid \#B_i = 1,\ \#C_i = 0\}, \qquad i = 1, \ldots, k,$$
$$\mathcal{B}^{(i)} = \{\mu^{(i)} \mid \#C_i = 1\}, \qquad i = k+1, \ldots, n.$$

S1M5: $\text{MRTA}_{|\mathcal{B}}$, where 1 component moves to $\mathcal{F}_B$ and $n-1$ components move to $\mathcal{F}_C$, namely among the k components that can experience both $\mathcal{F}_C$ and $\mathcal{F}_B$ only one ends up in $\mathcal{F}_B$. Here

$$\mathcal{B}^{(i)} = \cup_{h=1}^{n} \{\mu \mid \#B_i = 1, \#C_i = 0,\ h = i \text{ or } \#B_i = 0, \#C_i = 1,\ h \neq i\}, i = 1, \ldots, k$$
$$\mathcal{B}^{(i)} = \{\mu \mid \#C_i = 1\}, i = k+1, \ldots, n$$

S1M6: 2 components move to $\mathcal{F}_B$ and $n-2$ components move to $\mathcal{F}_C$. Here, the indicator random variable is similar to the one defined in S1M5, but there are $\binom{n}{2}$ addends.

In $Scenario2$ the reward structures and indicator functions are similar to those of $Scenario1$, so they are not listed. Of course the interpretation is completely different. In particular:

S2M2: the first 4 moments of Y_∞, where, in each component, if there is at least a working subcomponent then the gain is 3, if an error is detected then the gain is 1, otherwise the gain is 0.

S2M3: there is a catastrophic failure before the components can experience benign failure.

S2M4: all components end up in $\mathcal{F}_B$, i.e., no catastrophic failure.

S2M5: before the catastrophic failure, exactly one component experiences a benign failure.

S2M6: before the catastrophic failure, exactly two components experience a benign failure.

VI. Numerical experiments

Fixed one hour as the unit of measure for time, the exponential distribution parameters of each M_i have been selected at random in the ranges: $\lambda_i^R \in [0.45, 0.55]$ (about once every two hours), $\lambda_i^D \in [0.013, 0.014]$ (about once every

72 hours), $\lambda_i^E \in [0.0015, 0.0025]$ (about once every 21 days), $\lambda_i^{EP} \in [0.013, 0.017]$ (about once every 3 days), and $\zeta_i = 0.2$.

Regarding the parameters n and η, first n has been chosen equal to 3, and all M_i have the same η that spans $[0.99, 0.999]$; then, n is increased from 3 to 20, and η_i is set to 0.995. This parameters choice allows to account for the model's characteristics CM1 and CM6 (increasing number of models and stiff models, being the ratio between largest and smallest rate in $\tilde{\boldsymbol{Q}}$ more than 10^4).

In order to enhance consistency among the results, the topology for $n = 20$ has been first generated ($\boldsymbol{\mathcal{T}}$ is depicted in Figure 5) and then the topologies for $6 \leq n \leq 19$ are extracted from that one (upper left submatrices of $\boldsymbol{\mathcal{T}}$). In order to keep the number of synchronizations low, and guarantee that the topology is connected for all n, $\boldsymbol{\mathcal{T}}$ is almost tridiagonal. Although artificial, having to adapt it to different numbers of components, the proposed topology is sufficiently varied and complex to represent real systems, as already discussed when describing the case study.

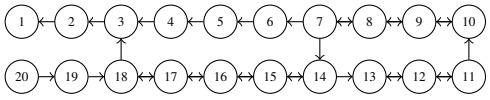


Fig. 5. Topology for $n = 20$.

An important role is expected to be played by k because it impacts on the number of absorbing states and then, in particular for $Scenario1$, on the probability of entering in one of them. If this probability is smaller than the required precision, then AMEn is expected to have troubles finding the correct solution. In order to consider models that are increasingly difficult to be solved, k has been chosen directly proportional to n. For the figures and tables shown in this section, $k = n/2$.

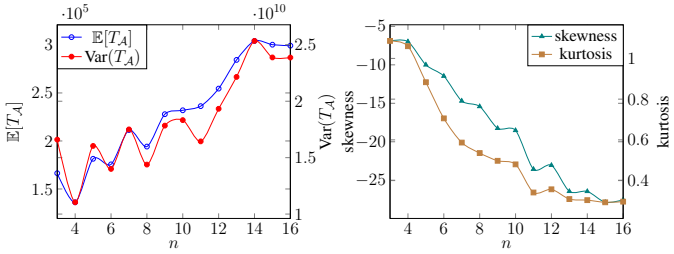
The computations have been performed on a node of a cluster with two Intel(R) Xeon(R) CPU E5-2650 v4 @ 2.20GHz processors each. The processes have been limited to 19 GB of RAM and 4 threads each, with a time limit of 48 hours.

Since the focus of the evaluation is to assess the performance of the method, among all the analyses carried out to study $Scenario1$ and $Scenario2$ only a selection is discussed here.

A. Evaluation results

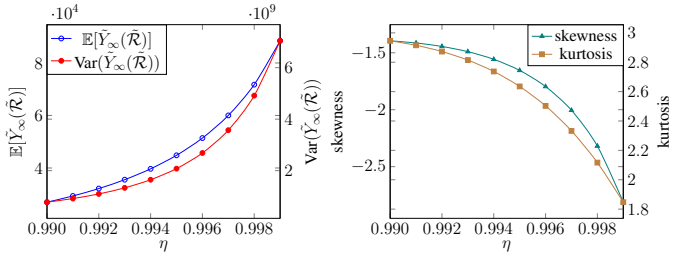
In Figure 6, M1 is shown for $Scenario1$ at increasing of n. As expected, the MTTF increases, and it is interesting that also its variability increases. Figure 6b shows that the left tail of $T_{\mathcal{A}}$ is longer and, at increasing of n, the asymmetry of the distribution increases. Kurtosis decreases, i.e., the propensity to produce outliers decreases. The 4 moments of $T_{\mathcal{A}}$ present a sort of "regularization", and in particular $\mathbb{E}[T_{\mathcal{A}}]$ increases and tends to stabilize while $\mathrm{Var}(T_{\mathcal{A}})$ is a little more shaky, as expected considering the influence of the topology. The results confirm also the role of node 14 that, by inspection of Figure 5, can be identified as a "turning point" in the topology.

Results obtained for S2M2 are further elaborated for evaluating skewness and kurtosis that are shown in Figure 7 for $Scenario2$, for n=3 and at increasing of η. In particular, it can



(a) mean and variance (b) skewness and kurtosis

Fig. 6. $Scenario1$: (a) values of $\mathbb{E}[T_{\mathcal{A}}]$ and $\mathrm{Var}(T_{\mathcal{A}})$, and (b) skewness and kurtosis, for $\eta = 0.995$ at increasing of n.



(a) mean and variance (b) skewness and kurtosis

Fig. 7. $Scenario2$: (a) values of $\mathbb{E}[\tilde{Y}_\infty(\tilde{\mathcal{R}})]$ and $\mathrm{Var}(\tilde{Y}_\infty(\tilde{\mathcal{R}}))$, and (b) skewness and kurtosis, for $n = 3$ at increasing of η.

be observed that η has a great impact on $\tilde{Y}_\infty(\tilde{\mathcal{R}})$, not only on $\mathbb{E}[\tilde{Y}_\infty(\tilde{\mathcal{R}})]$ but also on the distribution shape.

B. Performance of the method

Time related indicators are shown in Tables I and III, while Table II is related to the evaluation of the TT-ranks. Overall, they are representative of the efficiency of the proposed solution method, and the reported values confirm the expected behavior.

Notice that the evaluation of $\tilde{m}_k(T_{\mathcal{A}})$ and $m_k(Y_\infty(\boldsymbol{r}))$ requires the solution of $1, 2, 3$ or 4 linear systems of huge dimension and AMEn is either able to solve them all within the defined timeout, or it is not able to find a solution already for the first linear system. For S1M2, $\tilde{\boldsymbol{r}}$ is

$$\tilde{\boldsymbol{r}} = \hat{\boldsymbol{r}} \otimes \boldsymbol{e} \otimes \ldots \otimes \boldsymbol{e} + \boldsymbol{e} \otimes \hat{\boldsymbol{r}} \otimes \ldots \otimes \boldsymbol{e} + \ldots + \boldsymbol{e} \otimes \ldots \otimes \boldsymbol{e} \otimes \hat{\boldsymbol{r}},$$

where $\hat{\boldsymbol{r}} = (3, 0, 1, 0, 0)^T$. It can be proven that $\tilde{\boldsymbol{r}}$ has TT-rank bounded by 2 (see Appendix E). This favourably impacts on the performance of the method. Actually, in the performed computations, the max TT-rank of the solution vector is 22, a small value as expected.

In the numerical evaluations reported in Table I, for n greater than 14, AMEn was not able to converge because it remains trapped in a region characterized by a large error.

The computations for S1M3-6 deserve special comments. In particular, in S1M3 the TT-ranks are expected to remain low, in S1M4-S1M6 the TT-ranks are expected to be slightly larger because of the increasing complexity of the indicator random variable. Similarly for S2M4-S2M6 compared to S2M3. Table II confirms the expectation, so AMEn easily explores the search space even for models with a huge state-space. Table III, read by column, shows that as n increases the time

TABLE I
WALL-CLOCK TIME (IN SECONDS) REQUIRED TO EVALUATE THE FIRST 4 MOMENTS OF $Y_\infty(\boldsymbol{r})$ (M2) AT INCREASING OF n. THE VALUE 0 INDICATES A TIME BELOW 1 SECOND.

	$Scenario1$				$Scenario2$			
n	$\tilde{m}_1$	$\tilde{m}_2$	$\tilde{m}_3$	$\tilde{m}_4$	$\tilde{m}_1$	$\tilde{m}_2$	$\tilde{m}_3$	$\tilde{m}_4$
3	1	0	0	0	1	0	1	1
4	1	0	0	0	3	16	29	42
5	1	1	1	1	16	127	264	412
6	1	3	5	8	28	193	476	704
7	2	11	23	35	16	266	775	954
8	4	25	63	101	41	421	821	1707
9	5	42	114	182	284	578	1229	2073
10	12	80	196	321	340	704	1553	2505
11	20	119	361	648	409	898	2188	3924
12	75	199	523	956	464	1054	2127	4868
13	35	273	1674	12490	526	1610	2809	5075
14	550	3561	7526	13062	1313	2452	5572	10436

TABLE II
MAX TT-RANK REQUIRED TO EVALUATE S1M3-6 AND S2M3-6 AT INCREASING OF n.

	$Scenario1$				$Scenario2$			
n	S1M3	S1M4	S1M5	S1M6	S2M3	S2M4	S2M5	S2M6
3	7	7	7	7	9	12	12	12
4	13	15	14	15	25	49	30	50
5	13	17	16	17	30	68	46	62
6	15	24	18	20	40	86	68	78
7	17	20	21	22	40	89	66	76
8	18	31	23	31	42	90	65	78
9	19	31	24	33	41	104	58	85
10	18	41	26	37	44	135	67	86
11	26	38	28	34	41	132	59	76
12	23	48	33	36	34	135	56	72
13	29	52	33	42	33	128	52	65
14	37	98	48	63				
15	41		59	68				
16	39		56	73				
17	44		53	80				
18	58							
19	56							
20	59							

to convergence of AMEn increases, but in a complex way that is not easy to predict. It is also evident that $Scenario1$ and $Scenario2$ differ in terms of characteristics MC2 and MC5, being the latter more complex than the former. An important role is played by k. If a higher value of k is considered, e.g., $k = 3/4 \cdot n$, then the max TT-rank remains almost the same but AMEn is slower for all n and has troubles finding solutions starting from $n = 18$ in $Scenario1$.

Notice that, even though the computations required for S1M3 (Table III) are quite similar to those of the second moment of M2 (Table I), AMEn performs differently, confirming that an optimization-based solver can be highly sensitive to the problem at hand.

TABLE III
WALL-CLOCK TIME (IN SECONDS) REQUIRED TO EVALUATE S1M3-6 AND S2M3-6 AT INCREASING OF n. THE VALUE 0 INDICATES A TIME BELOW 1 SECOND.

	$Scenario1$				$Scenario2$			
n	S1M3	S1M4	S1M5	S1M6	S2M3	S2M4	S2M5	S2M6
3	1	1	1	0	1	1	1	1
4	1	1	1	1	3	32	12	64
5	1	1	1	1	48	271	279	523
6	1	2	2	3	197	736	449	1328
7	2	2	3	4	251	1222	709	2175
8	2	7	5	13	370	1970	841	4421
9	2	10	7	19	542	3790	898	7706
10	2	31	19	55	627	8215	1737	11048
11	7	58	31	110	798	9617	3178	14525
12	7	181	57	172	634	14645	4895	21654
13	13	329	77	411	561	17489	6278	32455
14	70	13634	568	3714				
15	261		1672	10811				
16	175		1052	12359				
17	186		1410	35767				
18	876							
19	870							
20	1212							

C. Final considerations

As a form of validation of the new solution method, comparisons with alternative techniques have been carried out, for tractable system configurations. As an example, for $n = k = 4$, $\mathcal{S}$ can be explicitly constructed and stored in memory because $|\mathcal{S}|$ is equal to $5^4 = 625$ for $Scenario1$ and $10^4 + 2^4 = 10016$ for $Scenario2$. This allowed to verify that the results obtained with the presented implicit approach and those obtained with the classical explicit approach are in agreement up to the desired numerical error[4].

As n increases, the explicit representation of $\boldsymbol{Q}$ becomes infeasible, promoting the adoption of the implicit descriptor matrix $\tilde{\boldsymbol{Q}}$, but typically the solution vector remains explicit in currently available solutions (see Section VII on related work). For the considered case study, when $n = 14$ each vector requires about 11 Gigabytes for $Scenario1$ and when $n = 10$ about 19 Gigabytes for $Scenario2$. When $n = 16$, for $Scenario1$, each vector requires about 280 Gigabytes, and when $n = 17$ it is unfeasible to store vectors explicitly, since each one requires about 1.2 Petabytes. Thus, the method presented here is —to the best of the authors' knowledge— the only one able to evaluate the defined measures of interest. It needs also to be recalled that literature studies on implicit representations mostly focus on availability analysis, differently from the reliability perspective taken in this paper.

It has also been checked that, for large models, specific entries of the reward vector $\tilde{\boldsymbol{r}}$, that are expected to be (non)zero and can be evaluated exploiting Kronecker product properties, actually match the expectations.

[4]Both the `spantree` method provided by https://github.com/numpi/kaes and the analytical solvers of Möbius [34] have been exercised.

VII. Related work

Modularity and composability are well established concepts within the modeling community. They require adequate formalisms to be applicable in practice, both from the modeling and solution perspectives. To this purpose, many approaches have been proposed, resulting in a vast and scattered literature. In this section, the focus will be only on those papers that show a direct link with the work developed in the previous sections, according to the following organization in four subtopics.

A. Implicit model as joint submodels trough Kronecker Algebra

Considering the explicit-implicit axis of Figure 2, in this paper the focus has been on the descriptor matrix (Kronecker product) approach formulated a few decades ago (e.g., in [12], [13]), and for which there was active research for several years, as documented in the survey in [14]. Two key aspects distinguish this paper from previous work: i) the relationship between the system model and the synchronization of its parts, ii) the properties of the infinitesimal generator matrix.

Concerning point i), the approach in the literature has been to first define the system model, exploiting high level formalisms as in [35]–[37], then cut it into pieces (not necessarily related to the system logical architecture), each one transformed into an SAN, and finally solved. Instead, here the focus is on system components and their modeling, possibly exploiting template models, so the approach is reversed. Since current trend in modeling large systems is to rely on template based modeling and related compositional operators (e.g., [8], [9]), the proposed approach is more suitable.

Related to point ii), the vast majority of the papers assumes an (implicit) irreducible infinitesimal generator matrix, since their Markov chains derive from availability or performance models. Instead, here the focus is on CTMC with absorbing states that derive from reliability models, so the infinitesimal generator matrix is reducible. This triggered the new definition of the shift matrix (Equation (43)) to cope with matrix invertibility.

B. Implicit matrices and vectors

Initial studies on implicit representations focused on an implicit representation of matrices and explicit representation of vectors. However, this requires clever implementations of the matrix-vector product, as in the shuffle, slice and split algorithms [38] implemented in PEPS[5], or strategies implemented in Nsolve[6] [39]. As pointed out in [14], [40], the main limitation of working with explicit vectors is that, for models with huge state-space, the memory occupancy of the vectors supersedes that of the matrices, making the solution infeasible in practice. None of the available mitigation strategies, such as storing data on hard drives and load small chunks of them only when needed [41], [42], seem to be able to balance memory with computation time for very large models. This triggered studies on strategies to deal with implicit representation of the solution vector.

[5]http://www-id.imag.fr/Logiciels/peps

[6]https://ls4-www.cs.tu-dortmund.de/download/buchholz/Programs/Nsolve

Representing $\boldsymbol{x}^{(i)}$ as the sum of Kronecker products, in analogy to Equation (41), through the steps of Equation (14) is infeasible because the addends double after a matrix-vector multiplication. To the best of the authors' knowledge, so far only two representations have been exploited to address this issue: *Tucker decomposition*, at the moment only exploited for irreducible $\boldsymbol{Q}$ produced by availability or performance models [43], and *Tensor Trains* or *matrix product states*, for irreducible $\boldsymbol{Q}$ first employed in [44] and for reducible $\boldsymbol{Q}$ in [10]. The first strategy helps to reduce the complexity in case of components with a large number of states, but does not solve the curse of dimensionality: the storage is still exponential in the number of components. The latter, instead, along with similar strategies such as the hierarchical Tucker decomposition [45], [46], reduces the complexity of most arithmetic operations to be linear in the number of components, and polynomial in the tensor-train-rank, which is often bounded in practice. Only recently the dependability community started exploring descriptor vectors. In [44] the same symbolic vector representation of this paper [29] is employed together with standard numerical solvers, such as AMEn [30].

In this paper a new systematic way to implicitly define reward vectors is introduced, moving from the simple one defined in [10]. In addition to the obtained higher computational efficiency, this extension is needed to determine higher moments of the random variable the measures of interest are based on.

C. Other forms of "implicit" representation

In the literature, several other approaches to implicit representation have been proposed. Among them, variants of the multi-valued decision diagrams as data structure and the saturation algorithm [47], [48] to explore the state-space are the most commonly investigated. Also, combinations of Multi Decision Diagrams and Kronecker Algebra have been successfully exploited, such as *matrix diagram* presented in [36], implemented (together with other strategies) in the Meddly library[7] and exploited by several tools, such as SMART[8] [40] and GreatSPN[9] [49]. However, notice that in such approaches the vectors are stored explicitly, so they experience the already discussed issues.

D. Formalisms to express reward structures

Concerning the measures of interest, this paper adapted the perspective of the work in [19], to the implicit definition of rewards, focusing on, and generalizing, a specific class of measures: the (Conditional) MTTF as worked out in [23], [24].

Other approaches have been extensively investigated in the literature. A highly impacting line of research is the definition of Continuous Stochastic Logic (CSL) started in [50] within the context of model checking, where not only single states are considered, but also paths are addressed. Popular tools, such as PRISM[10] and Storm[11], implement

[7]https://github.com/asminer/meddly.git

[8]https://github.com/asminer/smart.git

[9]https://github.com/greatspn/SOURCES.git

[10]https://www.prismmodelchecker.org

[11]https://www.stormchecker.org

extensions of this temporal logic. In particular, they allow to define both conditional probabilities and cumulative reward till absorption on submodels, hence many of the measures addressed in this paper are expressible in the CSL formalism. However, the concept of implicit solution vector is not yet available in techniques for the solution of CSL models, and the representation of the reward vector is not exploited at level of boosting efficiency of the model solution. Moreover, to the best of the authors' knowledge, there is no direct way of expressing higher moments of the random variable of interest (indirectly, this is always possible, but cumbersome). It is expected that the approach in this paper can stimulate the related community to develop similar extensions.

The template-based metamodel approach presented in [9] also allows to define rewards on submodels, but again there is no characterization of such representation with direct exploitable benefit at level of solvers' performance.

VIII. Conclusions and future work

This paper addressed implicit reward structures definition in the context of Markov Reliability modeling. The major achievements are: i) the definition of a high-level implicit reward structure based on submodels, given concisely in terms of the reward structure on top of each SSPN submodel; ii) a technique based on Kronecker algebra and Tensor Trains to numerically solve a SAN reward model with multiple absorbing states, deriving formulas to express and assess the k-moments of instant-of-time or interval-of-time reward variables to absorbing states; iii) the characterization of the relation that allows the automatic generation of the Markov reward model and the formulas used for the solution from the corresponding reward model and measures, concisely described at level of SSPN. The effectiveness of the proposed approach has been demonstrated on a rather sophisticated case study, considering a variety of scenarios and measures of interest. Both the dimension of the system under analysis and the performance of the method reached interesting levels. Of course, to better consolidate such outcomes, a more extensive evaluation campaign considering other system configurations would be desirable, planned as next step of this study.

Moving from the current developments, extensions are foreseen in several directions. Among the most immediate subject of future work there is the exploration of the SGSPN [13] formalism to describe the submodels and their interactions, adopting the approach proposed in [28] to synchronize immediate transitions. Moreover, methods to ensure that the model at level of Petri net meets the requirements of the solution method are worth investigating. Deepening the study of how to tune the parameters of the approach (mainly, the truncation parameter) is also relevant to pursue.

Recently, the evaluation of performability measures has been recasted to the evaluation of bilinear forms characterized by matrix functions [51], where matrices and vector are treated explicitly. Trying to merge the contribution of this paper with the theory of matrix function appears an interesting investigation, enlarging the perspective on implicit representation.

The focus of this paper is on Reliability models (CTMC with absorbing states), but the contribution of Section III can be adapted to define reward and measures for Availability and Performance models (CTMC with irreducible infinitesimal generator matrix).

Another extension would be to consider more general models. In this respect, it is relatively straightforward to apply the method presented here to semi-Markov models, following [52], since the DTMC structure is untouched (only the sojourn time can be generally distributed). Also Markov Regenerative Processes [26] can be considered, but the adoption of the results discussed here is more challenging.

Another promising research line is to extend the proposed approach in the context of probabilistic model checking.

Finally, in this paper the standard Kronecker algebra has been adopted, but the theory of SAN has already considered a generalized Kronecker algebra in which the rate of synchronization transition can be a function of components' state variables, following a given order among components. So, it would be interesting to investigate how to adapt the proposed solution method to address this generalization.

References

[1] A. Bondavalli, A. Fantechi, D. Latella, and L. Simoncini, "An integrated and compositional approach to design validation of embedded dependable systems," in *IEEE International Workshop On Embedded Fault-Tolerant Systems (EFTS00)*, 2000.

[2] A. Bondavalli, M. Nelli, L. Simoncini, and G. Mongardi, "Hierarchical modelling of complex control systems: Dependability analysis of a railway interlocking," *Journal of Computer Systems Science and Engineering*, vol. 16, no. 4, pp. 249–261, 2001.

[3] W. H. Sanders and J. F. Meyer, "Reduced base model construction methods for Stochastic Activity Networks," *IEEE Journal on Selected Areas in Communications*, vol. 9, no. 1, pp. 25–36, 1991.

[4] W. H. Sanders and L. M. Malhis, "Dependability evaluation using composed SAN-based reward models," *Journal of Parallel and Distributed Computing*, vol. 15, no. 3, pp. 238–254, 1992.

[5] W. H. Sanders and R. S. Freire, "Efficient simulation of hierarchical stochastic activity network models," *Discrete Event Dynamic Systems: Theory and Applications*, vol. 3, no. 2, pp. 271–300, 1993.

[6] S. Derisavi, P. Kemper, and W. H. Sanders, "Symbolic state-space exploration and numerical analysis of state-sharing composed models," *Linear Algebra Applications, Special Issue on the Conference on the Numerical Solution of Markov Chains*, vol. 386, pp. 137–166, 2004.

[7] H. Sukhwani, A. Bobbio, and K. S. Trivedi, "Largeness avoidance in availability modeling using hierarchical and fixed-point iterative techniques," *International Journal of Performability Engineering*, vol. 11, no. 4, pp. 305–319, 2015.

[8] S. Chiaradonna, F. Di Giandomenico, and G. Masetti, "A stochastic modeling approach for an efficient dependability evaluation of large systems with non-anonymous interconnected components," in *The* 28^{th} *IEEE International Symposium on Software Reliability Engineering (ISSRE2017)*, 2017, pp. 46–55.

[9] L. Montecchi, P. Lollini, and A. Bondavalli, "A template-based methodology for the specification and automated composition of performability models," *IEEE Transactions on Reliability*, pp. 1–17, 2019.

[10] G. Masetti, L. Robol, S. Chiaradonna, and F. Di Giandomenico, "Stochastic evaluation of large interdependent composed models through Kronecker algebra and exponential sums," in *Application and Theory of Petri Nets and Concurrency*, 2019, pp. 47–66.

[11] S. Chiaradonna, F. Di Giandomenico, and G. Masetti, "On identity-aware replication in stochastic modeling for simulation-based dependability analysis of large interconnected systems," *Performance Evaluation*, vol. 147, 2021.

[12] B. Plateau, J.-M. Fourneau, and K.-H. Lee, "PEPS: A package for solving complex Markov models of parallel systems," in *Modeling Techniques and Tools for Computer Performance Evaluation*, 1989, pp. 291–305.

[13] S. Donatelli, "Superposed stochastic automata: A class of stochastic Petri nets with parallel solution and distributed state space," *Performance Evaluation*, vol. 18, no. 1, pp. 21–36, 1993.

[14] P. Buchholz and P. Kemper, "Kronecker based matrix representations for large Markov models," in *Validation of Stochastic Systems*, 2004, vol. 2925, pp. 256–295.

[15] M. Ajmone Marsan, "Stochastic Petri nets: An elementary introduction," in *Advances in Petri Nets 1989*, ser. LNCS, G. Rozenberg, Ed. Springer, Berlin, Heidelberg, 1990, vol. 424, pp. 1–29.

[16] M. Ajmone Marsan, G. Balbo, and G. Conte, "A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems," *ACM Transactions on Computer Systems*, vol. 2, no. 2, pp. 93–122, 1984.

[17] B. R. Haverkort, "Markovian models for performance and dependability evaluation," in *First EEF/Euro Summer School on Trends in Computer Science, Lectures on Formal Methods and Performance Analysis*, 2001, vol. 2090, pp. 38–83.

[18] B. Plateau and W. J. Stewart, *Computational Probability*, ser. International Series in Operations Research & Management Science. Boston, MA: Springer, 2000, vol. 24, ch. Stochastic Automata Networks, pp. 113–151.

[19] W. H. Sanders and J. F. Meyer, "A unified approach for specifying measures of performance, dependability and performability," *Dependable Computing for Critical Applications, Dependable Computing and Fault-Tolerant Systems*, vol. 4, pp. 215–237, 1991.

[20] B. R. Haverkort and K. S. Trivedi, "Specification techniques for Markov reward models," *Discrete Event Dynamical Systems*, no. 3, pp. 219–247, 1993.

[21] M. Ajmone Marsan, G. Balbo, G. Chiola, G. Conte, S. Donatelli, and G. Franceschinis, "An introduction to generalized stochastic Petri nets," *Microelectronics Reliability*, vol. 31, pp. 699–725, 1991.

[22] G. Ciardo, A. Blakemore, P. F. Chimento, J. K. Muppala, and K. S. Trivedi, "Automated generation and analysis of Markov reward models using stochastic reward nets," in *Linear Algebra, Markov Chains, and Queueing Models, IMA Volumes in Mathematics and its Applications, volume 48*, 1993, pp. 145–191.

[23] H. Choi and K. S. Trivedi, "Conditional MTTF and its computation in Markov reliability models," in *Annual Reliability and Maintainability Symposium 1993 Proceedings*, 1993, pp. 56–63.

[24] H. Choi, W. Wang, and K. S. Trivedi, "Analysis of conditional MTTF of fault-tolerant systems," *Microelectronics Reliability*, vol. 38, no. 3, pp. 393–401, 1998.

[25] K. S. Trivedi and M. Malhotra, "Reliability and performability techniques and tools: A survey," in *7th ITG/GI Conference on Measurement, Modelling and Evaluation of Computer and Communication Systems*, Aachen, Germany, 1993, pp. 27–48.

[26] K. S. Trivedi and A. Bobbio, *Reliability and Availability Engineering: Modeling, Analysis, and Applications*. Cambridge University Press, 2017.

[27] L. Grasedyck, D. Kressner, and C. Tobler, "A literature survey of low-rank tensor approximation techniques," *GAMM-Mitteilungen*, vol. 36, no. 1, pp. 53–78, 2013.

[28] G. Ciardo and M. Tilgner, "On the use of Kronecker operators for the solution of generalized stochastic Petri nets," 1996, Nasa Technical Report Server 20040110963.

[29] I. V. Oseledets, "Tensor-train decomposition," *SIAM Journal on Scientific Computing*, vol. 33, no. 5, pp. 2295–2317, 2011.

[30] S. V. Dolgov and D. V. Savostyanov, "Alternating minimal energy methods for linear systems in higher dimensions," *SIAM Journal on Scientific Computing*, vol. 36, no. 5, pp. A2248–A2271, 2014.

[31] A. Bobbio and K. S. Trivedi, "An aggregation technique for the transient analysis of stiff Markov chains," *IEEE Transactions on Computers*, vol. 35, no. 9, pp. 803–814, 1986.

[32] MathWorks, *MATLAB R2018a*, The Mathworks, Inc., 2018.

[33] J. Filliben, W. F. Guthrie, N. A. Heckert, J. D. Splett, and N.-F. Zhang, "NIST/SEMATECH e-handbook of statistical methods," 2003, http://www.itl.nist.gov/div898/handbook.

[34] T. Courtney, S. Gaonkar, K. Keefe, E. W. D. Rozier, and W. H. Sanders, "Möbius 2.3: An extensible tool for dependability, security, and performance evaluation of large and complex system models," in 39^{th} *Annual IEEE/IFIP International Conference on Dependable Systems and Network (DSN 2009)*, 2009, pp. 353–358.

[35] P. Kemper, "Numerical analysis of superposed GSPNs," *IEEE Transactions on Software Engineering*, vol. 22, no. 9, pp. 615–628, 1996.

[36] G. Ciardo and A. S. Miner, "A data structure for the efficient Kronecker solution of GSPNs," in *Proceedings 8th International Workshop on Petri Nets and Performance Models*, 1999, pp. 22–31.

[37] L. Brenner, P. Fernandes, A. Sales, and T. Webber, "A framework to decompose GSPN models," in *International Conference on Application and Theory of Petri Nets*, 2005, pp. 128–147.

[38] R. M. Czekster, P. Fernandes, J.-M. Vincent, and T. Webber, "Split: A flexible and efficient algorithm to vector-descriptor product," in *VALUETOOLS*, 2007, pp. 83:1–83:8.

[39] P. Buchholz, "Hierarchical structuring of superposed GSPNs," *IEEE Transactions on Software Engineering*, vol. 25, no. 2, pp. 166–181, 1999.

[40] G. Ciardo, R. L. Jones, A. S. Miner, and R. Siminiceanu, "Logical and stochastic modeling with SMART," in *Computer Performance Evaluation. Modelling Techniques and Tools*, 2003, pp. 78–97.

[41] D. D. Deavours and W. H. Sanders, "An efficient disk-based tool for solving large Markov models," *Performance Evaluation*, vol. 33, no. 1, pp. 67–84, 1998.

[42] M. Kwiatkowska and R. Mehmood, "Out-of-core solution of large linear systems of equations arising from stochastic modelling," in *Process Algebra and Probabilistic Methods: Performance Modeling and Verification*, H. Hermanns and R. Segala, Eds., 2002, pp. 135–151.

[43] P. Buchholz, T. Dayar, J. Kriege, and M. C. Orhan, "On compact solution vectors in Kronecker-based Markovian analysis," *Performance Evaluation*, vol. 115, pp. 132–149, 2017.

[44] D. Kressner and F. Macedo, "Low-rank tensor methods for communicating Markov processes," *Quantitative Evaluation of Systems*, pp. 25–40, 2014.

[45] L. Grasedyck, "Hierarchical singular value decomposition of tensors," *SIAM Journal on Matrix Analysis and Applications*, vol. 31, no. 4, pp. 2029–2054, 2010.

[46] D. Kressner and C. Tobler, "Algorithm 941: htucker – A Matlab toolbox for tensors in hierarchical Tucker format," *ACM Transactions on Mathematical Software*, vol. 40, no. 3, pp. 1–22, 2014.

[47] G. Ciardo, "Data representation and efficient solution: A decision diagram approach," in *Formal Methods for Performance Evaluation: 7th International School on Formal Methods for the Design of Computer, Communication, and Software Systems*, 2007, pp. 371–394.

[48] G. Ciardo, G. Lüttgen, and R. Siminiceanu, "Saturation: An efficient iteration strategy for symbolic state-space generation," in *Tools and Algorithms for the Construction and Analysis of Systems*, 2001, pp. 328–342.

[49] J. Babar, M. Beccuti, S. Donatelli, and A. Miner, "GreatSPN enhanced with decision diagram data structures," in *Applications and Theory of Petri Nets*, J. Lilius and W. Penczek, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 308–317.

[50] C. Baier, B. Haverkort, H. Hermanns, and J.-P. Katoen, "Model-checking algorithms for continuous-time Markov chains," *IEEE Transactions on Software Engineering*, vol. 29, no. 6, pp. 524–541, 2003.

[51] G. Masetti and L. Robol, "Computing performability measures in Markov chains by means of matrix functions," *Journal of Computational and Applied Mathematics*, vol. 368, 2020.

[52] G. Ciardo, R. A. Marie, B. Sericola, and K. S. Trivedi, "Performability analysis using semi-Markov reward processes," *IEEE Transactions on Computers*, vol. 39, no. 10, pp. 1251–1264, 1990.

APPENDIX

A. Deriving formulas for $\tilde{V}_\infty(\tilde{\mathcal{R}})$ and $\tilde{Y}_\infty(\tilde{\mathcal{R}})$

The implicit formulation of Equation (5) is

$$\tilde{V}_t(\tilde{\mathcal{R}}) = \sum_{\mu \in \mathcal{S}} \tilde{\mathcal{R}}(\mu) \cdot \prod_{i=1}^{n} \tilde{I}_t^{\mu(i)},$$

being the logic AND represented through the product of the indicators in each submodel. Then, expanding $\tilde{\mathcal{R}}(\mu)$ as in

Equation (27), it is obtained

$$\begin{aligned}\tilde{V}_t(\tilde{\mathcal{R}}) &= \sum_{\mu\in\mathcal{S}}\Big(\sum_{j=1}^{n_\Pi}\prod_{i=1}^{n}\tilde{\mathcal{R}}_{ij}(\mu^{(i)})\Big)\cdot\prod_{i=1}^{n}\tilde{I}_t^{\mu^{(i)}}\\ &= \sum_{\mu\in\mathcal{S}}\sum_{j=1}^{n_\Pi}\Big(\prod_{i=1}^{n}\tilde{\mathcal{R}}_{ij}(\mu^{(i)})\Big)\cdot\prod_{i=1}^{n}\tilde{I}_t^{\mu^{(i)}}\\ &= \sum_{\mu\in\mathcal{S}}\sum_{j=1}^{n_\Pi}\Big(\prod_{i=1}^{n}\tilde{\mathcal{R}}_{ij}(\mu^{(i)})\cdot\tilde{I}_t^{\mu^{(i)}}\Big),\end{aligned}\tag{55}$$

that proves Equation (30). To obtain Equation (28) just notice that

$$\tilde{Y}_\infty(\tilde{\mathcal{R}}) = \sum_{\mu\in\mathcal{S}}\tilde{\mathcal{R}}(\mu)\cdot\int_0^\infty\prod_{i=1}^{n}\tilde{I}_t^{\mu^{(i)}},$$

where the integral of the product of the indicators is the sojourn time. The final step for obtaining the formula consists in exploiting the linearity of the integral.

B. Verifying correspondence between high and low level rewards

Embedding $\tilde{X}^{(i)}(t)$ into $\mathbb{R}^{|\tilde{\mathcal{S}}|}$ through the Kronecker product $\boldsymbol{e}_{s_1}\otimes\cdots\otimes\boldsymbol{e}_{s_n}$, for $\tilde{\boldsymbol{s}}=(s_1,\dots,s_n)\in\tilde{\mathcal{S}}$, it is possible to write:

$$\tilde{V}_t(\tilde{\boldsymbol{r}}) = \Big(\sum_{s_1\in\mathcal{S}^{(1)},\dots,s_n\in\mathcal{S}^{(n)}}\prod_{i=1}^{n}\tilde{I}_t^{s_i}\cdot\boldsymbol{e}_{s_1}\otimes\cdots\otimes\boldsymbol{e}_{s_n}\Big)^T\cdot\tilde{\boldsymbol{r}},$$

where $\tilde{\boldsymbol{r}}$ is defined in Equation (41) and the same reasoning on the AND of indicators of Appendix A applies. Thus, exploiting well known properties of the Kronecker product,

$$\begin{aligned}\tilde{V}_t(\tilde{\boldsymbol{r}}) &= \sum_{s_1,\dots,s_n}\sum_{j}\bigotimes_{i=1}^{n}\big(\tilde{I}_t^{s_i}\boldsymbol{e}_{s_i}^T\tilde{\boldsymbol{r}}^{i,j}\big)\\ &= \sum_{s_1,\dots,s_n}\sum_{j}\prod_{i=1}^{n}\tilde{I}_t^{s_i}\tilde{r}_{s_i}^{i,j},\end{aligned}$$

that is equal to Equation (55), taking advantage of Equation (42). For $\tilde{Y}_\infty(\tilde{\boldsymbol{r}})$ the same reasoning of Appendix A applies.

C. Moments of $Y_\infty(\boldsymbol{r})$

Applying the matrix function theory to Equation (9) yields:

$$\exp(t\boldsymbol{Q}) = \left[\begin{array}{ccc|c|c|c|c} & e^{t\boldsymbol{Q}_\mathcal{T}} & & \boldsymbol{w}_1 & \boldsymbol{w}_2 & \dots & \boldsymbol{w}_{n_{\text{abs}}}\\ \hline 0 & \dots & 0 & 1 & 0 & 0 & 0\\ \hline 0 & \dots & 0 & 0 & 1 & 0 & 0\\ \hline & \vdots & & \vdots & \vdots & \vdots & \vdots\\ \hline 0 & \dots & 0 & 0 & 0 & 0 & 1\end{array}\right],\tag{56}$$

where $\boldsymbol{w}_a = \boldsymbol{Q}_\mathcal{T}^{-1}\big(e^{t\boldsymbol{Q}_\mathcal{T}}-I\big)\boldsymbol{v}_a$ for $a\in\mathcal{A}$. From the requirement that r equals to zero on $\mathcal{A}$, it follows that $\boldsymbol{\pi}^T(0)e^{t\boldsymbol{Q}}\boldsymbol{r} = \boldsymbol{\pi}_\mathcal{T}^T(0)e^{t\boldsymbol{Q}_\mathcal{T}}\boldsymbol{r}_\mathcal{T}$. Then, recalling Equation (12) and $\mathbb{E}[Y_\infty(\boldsymbol{r})] := \int Y_\infty(\boldsymbol{r})\,d\mathbb{P}$, swapping the order of integration and knowing the formal solution of Equation (10), it is possible to write:

$$\begin{aligned}\mathbb{E}\big[Y_\infty(\boldsymbol{r})\big] &= \int\int_0^\infty \boldsymbol{e}_{X(t)}^T\cdot\boldsymbol{r}\,dt\,d\mathbb{P}\\ &= \int_0^\infty\Big(\int\boldsymbol{e}_{X(t)}^T\,d\mathbb{P}\Big)\cdot\boldsymbol{r}\,dt = \int_0^\infty\boldsymbol{\pi}^T(t)\cdot\boldsymbol{r}\,dt\\ &= \int_0^\infty\boldsymbol{\pi}^T(0)e^{t\boldsymbol{Q}}\boldsymbol{r}\,dt = -\boldsymbol{\pi}_\mathcal{T}^T(0)\boldsymbol{Q}_\mathcal{T}^{-1}\boldsymbol{r}_\mathcal{T}.\end{aligned}$$

Notice that $\boldsymbol{w}_a$ has no role in the computation of $\mathbb{E}\big[Y_\infty(\boldsymbol{r})\big]$, but it is important for $m_k\big(Y_\infty(\boldsymbol{r})\big)$ when $k>1$. The argument for the second moment of $Y_\infty(\boldsymbol{r})$ is the same as the one for $k>1$, so in the following it is assumed that $k=2$.

By a change of variable, the integral $\big(\int f(t)dt\big)^2$ can be rewritten as $\big(\int f(t)dt\big)\cdot\big(\int f(s)ds\big)$ and then as $\int f(t)f(s)\,dtds$. This enables swapping the expected value and the integral over $dtds$ as for the expected value of $Y_\infty(\boldsymbol{r})$, obtaining from Equation (12):

$$\begin{aligned}m_2(Y_\infty(\boldsymbol{r})) &= \mathbb{E}\Big[\Big(\int_0^\infty r_{X(t)}dt\Big)^2\Big]\\ &= \int_0^\infty\int_0^\infty\mathbb{E}[r_{X(t)}r_{X(s)}]\,dtds.\end{aligned}$$

In the upper right Cartesian plane, where coordinates $t,s\geq 0$, the regions $\{(t,s)|t\geq s\}$ and $\{(t,s)|s\geq t\}$ are indistinguishable from the point of view of $r_{X(t)}r_{X(s)}$ because this expression remains unchanged under the reflection with respect to the line $t=s$. Thus:

$$m_2(Y_\infty(\boldsymbol{r})) = 2\int\int_{\{(t,s)|s\geq t\}}\mathbb{E}[r_{X(t)}r_{X(s)}]\,dtds.$$

Exploiting the Markov property and the homogeneity of the CTMC, t may be considered as the new origin of time, which allows to compute the probability of being in a given state at time $s-t$ as:

$$\mathbb{P}\{X(s-t)=j,X(0)=i\} = \boldsymbol{e}_i^Te^{(s-t)\boldsymbol{Q}}\boldsymbol{e}_j,$$

and then

$$\begin{aligned}E[r_{X(t)}r_{X(s)}] &= \sum_{i,j}\pi_i(t)r_i\mathbb{P}\{X(s-t)=j,X(0)=i\}r_j\\ &= \sum_{i,j}(\boldsymbol{\pi}^T(0)e^{t\boldsymbol{Q}}\boldsymbol{e}_i)r_i\boldsymbol{e}_i^Te^{(s-t)\boldsymbol{Q}}\boldsymbol{e}_jr_j.\end{aligned}$$

Hence, $m_2(Y_\infty(\boldsymbol{r}))$ may be expressed as

$$m_2(Y_\infty(\boldsymbol{r})) = \int_0^\infty w(t)\Big(\int_t^\infty e^{(s-t)\boldsymbol{Q}}\boldsymbol{r}\,ds\Big)dt,$$

where $w(t)=\boldsymbol{\pi}^T(0)e^{t\boldsymbol{Q}}\mathrm{diag}(\boldsymbol{r})$. Solving the integrals with the help of Equation (56) yields:

$$m_2(Y_\infty(\boldsymbol{r})) = 2\boldsymbol{\pi}_\mathcal{T}^T(0)\boldsymbol{Q}_\mathcal{T}^{-1}\mathrm{diag}(\boldsymbol{r})\boldsymbol{Q}_\mathcal{T}^{-1}\boldsymbol{r}_\mathcal{T}.$$

For $k>2$ the reasoning is the same, but with the addition of two observations:

- Every time an integral of this kind is solved, a multiplicative factor -1 has to be considered. This explains the presence of the factor $(-1)^k$ in Equation (13).
- Counting the reflections that leave $r_{X(t_1)} \cdots r_{X(t_k)}$ unchanged, which are $k!$, justifies the factorial in Equation (13).

D. Evaluate $\mathrm{MRTA}_{|a}$

The closed formula for $\mathrm{MRTA}_{|a}$ can be derived following a path similar to the one adopted to obtain $m_2(Y_\infty(\boldsymbol{r}))$ starting from $\mathbb{E}[Y_\infty(\boldsymbol{r})]$. In particular:

$$\begin{aligned}&\mathbb{E}[Y_\infty(\mathcal{R}) \text{ and } X(\infty) = a] \\ &= \int_0^t \sum_k r_k \mathbb{P}\{X(u) = k,\, X(t) = a\}\, du \\ &= \boldsymbol{\pi}^T(0) \int_0^t e^{u\boldsymbol{Q}} \mathrm{diag}(\boldsymbol{r}) e^{(t-u)\boldsymbol{Q}}\, du\, \boldsymbol{e}_a,\end{aligned}$$

and, exploiting Equation (56), to

$$\boldsymbol{\pi}^T(0) \left[\begin{array}{c|c} \int_0^t e^{u\boldsymbol{Q}_\mathcal{T}} \mathrm{diag}(\boldsymbol{r}_\mathcal{T}) e^{(t-u)\boldsymbol{Q}_\mathcal{T}}\, du & \clubsuit + \spadesuit \\ 0 & 0 \end{array}\right] \boldsymbol{e}_a,$$

where

$$\begin{aligned}\clubsuit &= \int_0^t e^{u\boldsymbol{Q}_\mathcal{T}} \mathrm{diag}(\boldsymbol{r}_\mathcal{T}) \boldsymbol{Q}_\mathcal{T}^{-1} e^{(t-u)\boldsymbol{Q}_\mathcal{T}} \boldsymbol{v}_a\, du, \\ \spadesuit &= -\int_0^t e^{u\boldsymbol{Q}_\mathcal{T}} \mathrm{diag}(\boldsymbol{r}_\mathcal{T}) \boldsymbol{Q}_\mathcal{T}^{-1} \boldsymbol{v}_a\, du.\end{aligned}$$

For $t \to \infty$ it is easy to observe that $\clubsuit$ goes to zero and $\spadesuit$ goes to $\boldsymbol{Q}_\mathcal{T}^{-1} \mathrm{diag}(\boldsymbol{r}_\mathcal{T}) \boldsymbol{Q}_\mathcal{T}^{-1} \boldsymbol{v}_a$. Then, Equation (18) is proved.

A proof of Equation (19) can be found in [26], but an easier derivation (based on Equation (56)) is the following:

$$\lim_{t\to\infty} \pi_a(t) = \lim_{t\to\infty} \boldsymbol{\pi}_\mathcal{T}^T(0) \boldsymbol{Q}_\mathcal{T}^{-1} \left(e^{t\boldsymbol{Q}_\mathcal{T}} - I\right) \boldsymbol{v}_a = -\boldsymbol{\pi}_\mathcal{T}^T(0) \boldsymbol{Q}_\mathcal{T}^{-1} \boldsymbol{v}_a.$$

Notice that the reasoning presented in [26] for deriving the formula for $\mathrm{MTTF}_{|a}$ cannot be easily generalized to address $\mathrm{MRTA}_{|a}$. Therefore, the related developments presented in this paper are original.

E. TT-ranks bound for some relevant vectors

It can be proven that, for any vector $\hat{\boldsymbol{r}}$, vectors of the form

$$\tilde{\boldsymbol{r}} = \hat{\boldsymbol{r}} \otimes \boldsymbol{e} \otimes \ldots \otimes \boldsymbol{e} + \boldsymbol{e} \otimes \hat{\boldsymbol{r}} \otimes \ldots \otimes \boldsymbol{e} + \ldots + \boldsymbol{e} \otimes \ldots \otimes \boldsymbol{e} \otimes \hat{\boldsymbol{r}}$$

where $\boldsymbol{e}$ is the vector of all ones, have TT-ranks bounded by 2. Indeed, if the TT-ranks are $(\rho_1, \ldots, \rho_{n-1})$, then

$$\rho_h = \mathrm{rank}(\tilde{\boldsymbol{r}}_{i_1,\ldots,i_h;i_{h+1},\ldots,i_n}), \qquad h = 1, \ldots, n-1,$$

where the indices denote the reshaping of the vector $\tilde{\boldsymbol{r}}$ in a matrix where the first h indices are grouped as row indices, and the rest as column indices [29, Theorem 2.1]. Each addend defining $\tilde{\boldsymbol{r}}$ appears as a rank 1 term $\boldsymbol{u}\boldsymbol{v}^T$ in $\tilde{\boldsymbol{r}}_{i_1,\ldots,i_h;i_{h+1},\ldots,i_n}$, where either $\boldsymbol{u}$ or $\boldsymbol{v}$ is equal to $\boldsymbol{e}$. Hence, this matrix has rank bounded by 2.

F. Examples of reward and measures

All the examples presented in this Appendix are based on the SPN scheme shown in Figure 1.

1) Marking-dependent reward rate: An example of marking-dependent reward rate with $n_\Pi = 1$, used to evaluate the reward obtained accumulating the number of tokens in the places W_1 and E_1, is given by:

$$\tilde{\mathcal{R}}_{11}(\mu) = \#E_1 + \#W_1, \tag{57}$$

where an explicit reward assignment is only considered for the submodel M_1, i.e., $\bar{\mathcal{S}}_{11} = \{\mu \mid \#E_1 = e, \#W_1 = w,\, \forall\, e, w \in \mathbb{N}\}$ and $\bar{\mathcal{S}}_{i1} = \emptyset$ for $i = 2, \ldots, n$. In this case, from Equations (25), (26) and (57) it is derived that:

$$\begin{aligned}\tilde{\mathcal{R}}_{11}(\mu^{(1)}) &= \begin{cases} \#E_1 + \#W_1 & \text{if } \mu^{(1)} \in \bar{\mathcal{S}}_{11}, \\ 0 & \text{otherwise.}\end{cases} \\ \tilde{\mathcal{R}}_{i1}(\mu^{(i)}) &= \begin{cases} 1 & \text{if } \mu^{(i)} \in \mathcal{S}^{(i)}, \\ 0 & \text{otherwise.}\end{cases} i = 2 \ldots, n.\end{aligned} \tag{58}$$

Hence, applying Equation (58) to Equations (24) and (27) yields:

$$\tilde{\mathcal{R}}(\mu) = \tilde{\mathcal{R}}_1^\Pi(\mu) = \begin{cases} \#E_1 + \#W_1 & \text{if } \mu \in \mathcal{S}, \\ 0 & \text{otherwise,}\end{cases}$$

being

$$\begin{aligned}&\tilde{\mathcal{R}}_1^\Pi(\mu) \\ &= \tilde{\mathcal{R}}_{11}(\mu^{(1)}) \cdot \tilde{\mathcal{R}}_{21}(\mu^{(2)}) \cdot \ldots \cdot \tilde{\mathcal{R}}_{n1}(\mu^{(n)}) = \#E_1 + \#W_1.\end{aligned}$$

In this example, the value of m_Π depends on the number of tokens $\#E_1$ and $\#W_1$ that can be assigned to the places E_1 and W_1, respectively, for each marking of the SPN.

2) Minimal conditional accumulated time to absorption between two submodels: Another example of reward rate with $m_\Pi = n_\Pi = 1$, used to evaluate the minimal conditional accumulated time $\tilde{\mathrm{MRTA}}_{|\mathcal{B}}$ to absorption between the two submodels M_i and M_j conditioned by the event that one of the two submodels reached the absorbing marking, is given by the following definition:

$$\tilde{\mathcal{R}}_{i1}(\mu^{(i)}) = 1, \;\; \text{if } \#B_i = 0, \#C_i = 0,\; i = 1, \ldots n, \tag{59}$$

where explicit reward assignments are only considered for all submodels, i.e., $\bar{\mathcal{S}}_{i1} = \{\mu^{(i)} \mid \#B_i = 0, \#C_i = 0\}$. In this case, from Equations (24) to (27) and (59) it is derived that:

$$\begin{aligned}\tilde{\mathcal{R}}(\mu) &= \tilde{\mathcal{R}}_1^\Pi(\mu) \\ &= \begin{cases} 1 & \text{if } \#B_i = 0, \#C_i = 0,\; \forall i = 1, \ldots, n, \\ 0 & \text{otherwise,}\end{cases}\end{aligned}$$

being, for each μ such that $\#B_i = 0, \#C_i = 0,\; \forall i = 1, \ldots, n$:

$$\begin{aligned}&\tilde{\mathcal{R}}_1^\Pi(\mu) \\ &= \tilde{\mathcal{R}}_{11}(\mu^{(1)}) \cdot \ldots \cdot \tilde{\mathcal{R}}_{n1}(\mu^{(n)}) \\ &= 1 \cdot \ldots \cdot 1 = 1.\end{aligned}$$

To obtain $\mathcal{B}$, which represents the event that one of the two submodels M_i and M_j reached the absorbing marking,

is enough to apply Equation (22) to $\mathcal{B}^{(i)}$ and $\mathcal{B}^{(j)}$ defined as follows:

$$\mathcal{B}^{(i)} = \{\mu^{(i)} \mid \#B_i = 1, \#C_i = 0 \text{ or } \#B_i = 0, \#C_i = 1\},$$
$$\mathcal{B}^{(j)} = \{\mu^{(j)} \mid \#B_j = 1, \#C_j = 0 \text{ or } \#B_j = 0, \#C_j = 1\}.$$

3) Accumulated time to absorption of the last among all the submodels: Of particular interest is the example of reward rate with $n_\Pi = 2$, used to evaluate the moments $\tilde{\mathcal{M}}_{k,\tilde{\mathcal{R}}_{ij}}$ of the accumulated time to absorption of the last among all the submodels, that is given by:

$$\begin{aligned}
&\tilde{\mathcal{R}}_{11}(\mu^{(1)}) = 1,\\
&\tilde{\mathcal{R}}_{12}(\mu^{(1)}) = -1, \text{ if } \#B_1 = 1, \#C_1 = 0,\\
&\tilde{\mathcal{R}}_{12}(\mu^{(1)}) = -1, \text{ if } \#B_1 = 0, \#C_1 = 1,\\
&\tilde{\mathcal{R}}_{i2}(\mu^{(1)}) = 1, \text{ if } \#B_i = 1, \#C_i = 0,\ i = 2, \ldots, n,\\
&\tilde{\mathcal{R}}_{i2}(\mu^{(1)}) = 1, \text{ if } \#B_i = 0, \#C_i = 1,\ i = 2, \ldots, n,
\end{aligned} \tag{60}$$

where explicit reward assignments are considered for M_1, i.e., $\bar{\mathcal{S}}_{11} = \mathcal{S}^{(1)}$, and for all the submodels M_i, i.e., $\bar{\mathcal{S}}_{i2} = \{\mu^{(i)} \mid \#B_i = 1, \#C_i = 0 \text{ or } \#B_i = 0, \#C_i = 1\}$ for $i = 1, \ldots, n$.

In this case, from Equations (24) to (26) and (60) it is derived that:

$$\tilde{\mathcal{R}}_1^{\Pi}(\mu) = \begin{cases} 1 & \text{if } \mu \in \mathcal{S}, \\ 0 & \text{otherwise,} \end{cases} \tag{61}$$

$$\tilde{\mathcal{R}}_2^{\Pi}(\mu) = \begin{cases} -1 & \text{if } \mu \in \bar{\mathcal{S}}_2, \\ 0 & \text{otherwise,} \end{cases} \tag{62}$$

being, for each $\mu \in \mathcal{S}$:

$$\tilde{\mathcal{R}}_1^{\Pi}(\mu) = \tilde{\mathcal{R}}_{11}(\mu^{(1)}) \cdot \ldots \cdot \tilde{\mathcal{R}}_{n1}(\mu^{(n)}) = 1 \cdot \ldots \cdot 1 = 1,$$

and, for each $\mu \in \bar{\mathcal{S}}_2$:

$$\begin{aligned}
\tilde{\mathcal{R}}_2^{\Pi}(\mu) &= \tilde{\mathcal{R}}_{12}(\mu^{(1)}) \cdot \tilde{\mathcal{R}}_{22}(\mu^{(2)}) \ \cdots \ \cdot \tilde{\mathcal{R}}_{n2}(\mu^{(n)})\\
&= -1 \cdot 1 \cdot \ \cdots \ \cdot 1 = -1,
\end{aligned}$$

where $\mu^{(i)} \in \bar{\mathcal{S}}_{i2}, i = 1, \ldots, n$, and $\bar{\mathcal{S}}_2 = \cup_i \{\mu^{(i)} \mid \mu^{(i)} \in \bar{\mathcal{S}}_{i2}\}$ is the set of markings for which all submodels have reached an absorbing state.

Hence, applying Equations (61) and (62) to Equation (27) yields:

$$\tilde{\mathcal{R}}(\mu) = \begin{cases} 1 & \text{if } \mu = \mathcal{S}, \\ -1 & \text{if } \mu \in \bar{\mathcal{S}}_2, \\ 0 & \text{otherwise,} \end{cases} \tag{63}$$

where

- for each marking that satisfies the first case condition, the SPN can be in any marking,
- for each marking that satisfies the second case condition, all submodels M_i have reached an absorbing state (one token in B_i or in C_i),
- the third case condition includes the markings for which there is at least one submodel that has not reached an absorbing state.

The mean time to absorption of the last among all the submodels is evalutated by $\tilde{\mathcal{M}}_{1,\tilde{\mathcal{R}}_{ij}} = \mathbb{E}\big[\tilde{Y}_\infty(\tilde{\mathcal{R}})\big]$. In fact, considering that $Y = \tilde{Y}$, then applying Equation (63) to Equation (3) and the definition of mean of random variable yields:

$$\begin{aligned}
&\mathbb{E}[\tilde{Y}_\infty(\tilde{\mathcal{R}})] = E[Y_\infty(\tilde{\mathcal{R}})] = \mathbb{E}\big[\sum_{\mu \in \mathcal{S}} \tilde{\mathcal{R}}(\mu) \cdot J_\infty^\mu\big]\\
&= \sum_{\mu \in \mathcal{S}} \tilde{\mathcal{R}}(\mu) \cdot \mathbb{E}[J_\infty^\mu]\\
&= \sum_{\mu \in \mathcal{S}} 1 \cdot \mathbb{E}[J_\infty^\mu] + \sum_{\mu \in \bar{\mathcal{S}}_2} -1 \cdot \mathbb{E}[J_\infty^\mu] \text{ (from Equation (63))}\\
&= \sum_{\mu \in \mathcal{S} \setminus \bar{\mathcal{S}}_2} \mathbb{E}[J_\infty^\mu] + \sum_{\mu \in \bar{\mathcal{S}}_2} \mathbb{E}[J_\infty^\mu] - \sum_{\mu \in \bar{\mathcal{S}}_2} \mathbb{E}[J_\infty^\mu]\\
&= \sum_{\mu \in \mathcal{S} \setminus \bar{\mathcal{S}}_2} \mathbb{E}[J_\infty^\mu],
\end{aligned} \tag{64}$$

Equation (64) expresses the total time the SPN is in all markings such that there is at least one submodel that has not reached an absorbing state during $[0, \infty]$, obtained as the difference between:

- the total time the SPN is in all reachable markings during $[0, \infty]$,
- the total time the SPN is in all markings such that all submodels have reached an absorbing state during $[0, \infty]$.

Informally, since the reward variable is defined in Equation (3) by summing up the reward obtained for each marking, the reward obtained at each instant of time is:

- $\tilde{\mathcal{R}}(\mu) = 1$, if $\mu \in \mathcal{S} \setminus \bar{\mathcal{S}}_2$, i.e., there is at least one submodel that has not reached an absorbing state, or
- $\tilde{\mathcal{R}}(\mu) = 1 - 1 = 0$, if $\mu \in \bar{\mathcal{S}}_2 \subseteq \mathcal{S}$, i.e., all submodels have reached an absorbing state.

Therefore, the accumulated reward is the time to absorption of all submodels.

G. Acronyms and Symbols

CTMC	Continuous Time Markov Chain
GSPN	Generalized Stochastic Petri Net
MRTA	Mean Reward To Absorption
MTTF	Mean Time To Failure
SAN	Stochastic Automata Network
SGSPN	Superposed GSPN
SPN	Stochastic Petri Net
SSPN	Superposed SPN
TT	Tensor Train
TT-SVD	Tensor Train Singular Value Decomposition
a	Absorbing state
$\mathcal{A}$	Set of all absorbing markings (states) of M
$\tilde{\mathcal{A}}$	Set of all potential absorbing states of $\tilde{M}_i$
$\mathcal{A}^{(i)}$	Set of all absorbing states restricted to M_i
$\mathcal{B}$	Subset of absorbing states
$\mathrm{MRTA}_{\mid\mathcal{B}}$	Conditional mean reward to absorption given that the model eventually absorbs into $\mathcal{B}$
$\tilde{\mathrm{MRTA}}_{\mid\mathcal{B}}$	Implicit notation for $\mathrm{MRTA}_{\mid\mathcal{B}}$
$\mathrm{MTTF}_{\mid\mathcal{B}}$	Conditional mean time to failure given that the model eventually absorbs into $\mathcal{B}$

η_i	Coverage of the error detection in $\mathcal{C}_i$
ζ_i	Coverage of the error recovery in $\mathcal{C}_i$
$\mathrm{diag}(\boldsymbol{r})$	Diagonal matrix $D = [d_{ij}]$ such that $d_{ii} = r_i$
$\boldsymbol{e}, \tilde{\boldsymbol{e}}$	Column vector of all ones, and corresponding implicit vector
$\boldsymbol{e}_i, \tilde{\boldsymbol{e}}_i$	i-th element of the standard basis $\mathbb{R}^{n_{\text{reach}}}$, and corresponding implicit element
$\mathcal{F}_B$	Permanent failure of $\mathcal{C}_i$ when error recovery fails
$\mathcal{F}_C$	Permanent failure of $\mathcal{C}_i$ when an erroneous status is not detected
$\boldsymbol{I}$	Identity matrix
I_t^μ	Indicator random variable representing the event that the SPN is the marking μ at time t
J_t^μ	Random variable counting the total time the SPN spends in the marking μ during the interval of time $[0, t]$
$M, \tilde{M}$	System model at Petri net level, and underlying implicit SAN model
$M_i, \tilde{M}_i$	i-th SPN model, and underlying implicit SAN model
$\mathcal{M}_{k,\mathcal{R}}$	Explicit k-moment at Petri net level
$\tilde{\mathcal{M}}_{k,\tilde{\mathcal{R}}_{ij}}$	Implicit k-moment at Petri net level
$m_{k,r}, \tilde{m}_{k,r}$	Explicit and implicit k-moment at Markov chain level
$\mu, \mu^{(i)}$	Marking of M, and marking restricted to M_i
n	Number of system submodels
n_i	Number of reachable states of $\mathcal{S}^{(i)}$
n_Π	Number of $\tilde{\mathcal{R}}_j^\Pi$ functions used in the definition of $\tilde{\mathcal{R}}(\mu)$
m_Π	Total number of non-zero reward values defined by all the functions $\tilde{\mathcal{R}}_j^\Pi$
n_r	Cardinality of $\bar{\mathcal{S}}$
$n_{\text{reach}}, n_{\text{abs}}$	Number of reachable and absorbing states
$\mathbb{N}^P$	Set of all functions $\mu : P \mapsto \mathbb{N}$
$\#(p, \mu), \#p$	Number of tokens in place p in marking μ
p, P	Generic place of M and set of all places of M
$\boldsymbol{\pi}(t), \tilde{\boldsymbol{\pi}}(t)$	State probability vector at time t, and corresponding implicit vector
$\boldsymbol{\pi}_\mathcal{T}(t)$	Vector $\boldsymbol{\pi}(t)$ restricted to the transient states $\mathcal{T}$
$\pi_\mathcal{B}(\infty)$	Probability that M eventually absorbs into $\mathcal{B}$
$\tilde{\pi}_\mathcal{B}(\infty)$	Implicit notation for $\pi_\mathcal{B}(\infty)$
$\boldsymbol{Q}, \tilde{\boldsymbol{Q}}$	Generator and descriptor matrix
$\boldsymbol{Q}_\mathcal{T}$	Submatrix of $\boldsymbol{Q}$ restricted to transient states
$\mathcal{R}, \tilde{\mathcal{R}}$	Explicit and implicit reward at Petri net level
$\mathcal{S}$	Set of all reachable markings of M
$\bar{\mathcal{S}}$	Subset of $\mathcal{S}$ including only markings with non-zero rewards attached
$\tilde{\mathcal{R}}_{ij}$	Reward associated to the marking of M_i contributing as a factor to $\tilde{\mathcal{R}}_j^\Pi$
$\tilde{\mathcal{R}}_j^\Pi$	Addend contributing to $\tilde{\mathcal{R}}$
$\boldsymbol{r}, \tilde{\boldsymbol{r}}$	Explicit and implicit reward vector at Markov chain level
$\boldsymbol{r}_\mathcal{T}$	Vector $\boldsymbol{r}$ restricted to the transient states $\mathcal{T}$
$\boldsymbol{S}^\mathcal{A}, \tilde{S}^\mathcal{A}$	Shift matrix and corresponding descriptor
$\mathcal{S}^{(i)}$	Reachable states of M_i
$\mathcal{ST}$	Set of all synchronization transitions
$\mathcal{T}, \mathcal{S}, \tilde{\mathcal{S}}$	Transient, reachable and potential states
$T_\mathcal{A}$	Time to absorption
TT-rank	Positive integers $\rho_2, \ldots, \rho_{n-1}$ such that $\rho_i \times n_i \times \rho_{i+1}$ is the size of the order 3 tensor $\boldsymbol{\mathcal{N}}^{(i)}$, for $i = 2, ..., n-1$
$V_\infty, \tilde{V}_\infty$	Instantaneous reward variable at infinite time, and corresponding implicit notation
$\boldsymbol{v}_i$	Column vector with the rates from the transient states to the i-th absorbing state
$\{X(t), t \geq 0\}$	Markov chain underlying the SPN model
$\boldsymbol{x}^{(i)}$,$\tilde{\boldsymbol{x}}^{(i)}$	Solution of the i-th linear system $\boldsymbol{Q}_\mathcal{T} x^{(i)} = \mathrm{diag}(\boldsymbol{r}) x^{(i-1)}$ with $\boldsymbol{Q}_\mathcal{T} x^{(1)} = \boldsymbol{r}$
$\{\tilde{X}(t), t \geq 0\}$	Implicit Markov chain underlying the SPN
$Y_\infty, \tilde{Y}_\infty$	Reward accumulated in transient states until absorbtion, and corresponding implict notation
$Y_{\infty\|\mathcal{B}}, \tilde{Y}_{\infty\|\mathcal{B}}$	Conditional reward accumulated until absorbtion given that some absorbing markings $\mathcal{B}$ are reached, and corresponding implict notation