Importance Function Derivation for RESTART Simulations of Petri Nets

Armin Zimmermann
System and Software Engineering
Faculty of Computer Science and Automation
Technische Universität Ilmenau, Germany
Email: armin.zimmermann@tu-ilmenau.de

Paulo Maciel
Modeling of Distributed and Concurrent Systems
Informatics Center
Federal University of Pernambuco, Recife, Brazil
Email: prmm@cin.ufpe.br

***Abstract*—The RESTART method is an efficient algorithm to estimate low probabilities by simulation. It forces a rare event to be visited more frequently by splitting promising paths at certain thresholds. The achievable speedup depends heavily on the quality of the importance function, which controls the splitting behavior. It is well-known from the literature that the specification of a good importance function is a subtle task for non-trivial systems. In this paper we present ideas that have the potential to support finding an importance function for models expressed as a stochastic Petri net. It is shown how the frequency of a rare event of interest can be estimated a-priori to set the number of thresholds close to its optimal value. A measure for the distance between a current marking and the rare state is given as well. Both contributions rely on an algebraic formulation of the Petri net structure and use results for their further analysis as a linear programming problem.**

I. Introduction

A well-known open issue in rare-event simulation with the RESTART method [1] is the specification of a suitable importance function [2], [3]. For general models, this still requires expert knowledge both of the individual model and the subtleties of RESTART, although considerable results have been reported for special cases such as for one-dimensional chains. Treatments of the theoretical background of RESTART can be found in [4]–[6].

The goal of our work is directed towards variations of RESTART allowing an implementation in a software tool for users without background knowledge about the algorithm. As the underlying model, we choose stochastic Petri nets [7]–[9] in this paper, which represent a well-known model for complex systems with synchronization, concurrency, and resource constraints. Ideally, any Petri net model containing a performance measure influenced by a rare event should be analyzable by a tool without additional information such as importance function or number of thresholds.

The first implementation step in our software tool TimeNET [10] for rare-event simulation was done by Kelling [11], [12]. Later on, the algorithms were extended to more general performance measures [13], and furthermore to colored stochastic Petri nets [14]. Among others, the implementation has been applied to safety-critical train control systems [15].

Other implementations of the RESTART technique include SPNP [16] for stochastic Petri nets and ASTRO [1]. UltraSAN supports stochastic activity networks, which are similar to stochastic Petri nets, and offer importance sampling [17] as well as stratified sampling [18].

There are several performance measure types that are hard to estimate because of rare events. Standard questions found in the literature derive the probability of a rare state, or the probability of reaching a state before returning to the initial state (set). In this paper we take a look at the expected number of (rare) events instead, which means to derive a (very small) transition throughput in a Petri net. This measure can be used to assess the expected number of catastrophic events in the design of a highly reliable system.

In the literature, it is generally accepted that specifying a good importance function is a hard task. The goal of this paper is to show that structural information contained in a (simple, i.e., uncolored) Petri net model can be exploited to partially parameterize a RESTART simulation. Linear algebraic techniques that are well-known in the Petri net community allow to describe structure and behaviour with matrix equations [19].

Basic terminology of Petri nets and their rare-event simulation is recalled in the subsequent section. A method to estimate the rare event frequency in a stochastis Petri net model is presented in Section III. It uses structural model information and is based on the bounds technique proposed by Campos et. al. [20]. Prior knowledge of rare event probability magnitude is crucial when defining thresholds to improve RESTART efficiency [2].

Section IV shows how a heuristic importance function based on the estimated number of transition firings from the current state (marking) to the rare event can be automatically derived. A lower bound for the length of the remaining firing sequence to a state of interest can be obtained from a linear algebraic formulation of the Petri net. Related work for the touched two topics is covered in Sections III and IV. Finally, conlusions and acknowledgements are given.

II. Rare-Event Simulation of Stochastic Petri Nets

A *Stochastic Petri net* can be formally defined as a tuple

$$\mathsf{SPN} = (P, T, \mathbf{Pre}, \mathbf{Post}, \Lambda, W, \mathbf{m}_0)$$

with the elements described in the following (additional details like immediate transitions, inhibitor arcs, transition priorities etc. are intentionally ignored here for simplicity).

P is the set of *places*, which may contain tokens and thus constitute the state variables of the Petri net. The *marking* $\mathbf{m}$ of the Petri net associates a (non-negative integer) number of tokens to each place $\mathbf{m} : P \to \mathbb{N}$, and denotes the *state* of the underlying stochastic process. The marking can also be viewed as a vector of natural numbers with the size of the number of places.

$$\mathbf{m} \in \mathbb{N}^{|P|} = \big(\mathbf{m}(p_1), \mathbf{m}(p_2), \ldots, \mathbf{m}(p_{|P|})\big)$$

We denote by $M = \{\mathbf{m} \mid \forall p \in P : \mathbf{m}(p) \in \mathbb{N}\}$ the set of all theoretically possible markings of a Petri net.

T represents the set of *transitions*. Obviously, a node of a Petri net may either be a place or transition ($T \cap P = \emptyset$), and a net should not be empty $T \cup P \neq \emptyset$.

$\mathbf{Pre} : P \times T \to \mathbb{N}$ describes the multiplicities (or cardinalities) of the *input arcs* that connect places to transitions. If there is no input arc connecting place p_i to transition t_j, then $\mathbf{Pre}(p_i, t_j) = 0$. Similarly, $\mathbf{Post} : P \times T \to \mathbb{N}$ denotes the multiplicities of *output arcs* connecting transitions to places.

The *delay* Λ of a transition specifies the time that a transition needs to be enabled before it fires. The delay is defined by a probability distribution function that describes the possibly random delay time. It can be zero for *immediate transitions*. W maps each immediate transition to a real number, which is interpreted as the *firing weight* and used to derive relative probabilities in conflict solving.

$\mathbf{m}_0$ denotes the *initial marking* of the model. Because $\mathbf{m}_0$ is a marking, it is of the form $\mathbf{m}_0 : P \to \mathbb{N}$.

The set of *input* and output places of a transition t is defined by the subsets of places $p \in P$ for which $\mathbf{Pre}(p, t) > 0$ or $\mathbf{Post}(p, t) > 0$, respectively.

The *pre-* and *post-incidence* (or backward and forward incidence) *functions* of a Petri net, $\mathbf{Pre}$ and $\mathbf{Post}$, map pairs of transitions and places to the respective arc cardinality. Both $\mathbf{Pre}$ and $\mathbf{Post}$ can be interpreted as matrices with dimension $|P| \times |T|$ containing natural numbers. The *token flow* or (or *incidence*) *matrix* $\mathbf{C}$ captures the flows of tokens between transitions and places, and can be obtained from the incidence matrices as $\mathbf{C} = \mathbf{Post} - \mathbf{Pre}$. Thus, $\mathbf{C}(p, t)$ stores the change of tokens in p when t is fired.

The dynamics of a Petri net are defined as follows. A transition t is said to be *enabled* in a marking $\mathbf{m}$ if there are enough tokens available in each of its input places: $\forall p \in P : \mathbf{m}(p) \geq \mathbf{Pre}(p, t)$. Whenever a transition becomes newly enabled, a *remaining firing time* (RFT) is randomly drawn from its associated firing time distribution. The RFTs of all enabled transitions decrease with identical speed until one of them reaches zero (race enabling semantics).

The fastest transition (in case of multiple ones, a probabilistic choice decides) will *fire*, and change the current marking $\mathbf{m}$ to a new one $\mathbf{m}'$ by removing the necessary number of tokens from the input places and adding tokens to output places. We denote this by $\mathbf{m} \xrightarrow{t} \mathbf{m}'$ with $\forall p \in P : \mathbf{m}'(p) = \mathbf{m}(p) + \mathbf{C}(p, t)$.

If $\exists t \in T : \mathbf{m} \xrightarrow{t} \mathbf{m}'$, we say that $\mathbf{m}'$ is *directly reachable* from $\mathbf{m}$. A finite *firing sequence* of length k starting in a marking $\mathbf{m}$ is an ordered sequence of transitions $t_1 \ldots t_k$ such that $\mathbf{m} \xrightarrow{t_1} \mathbf{m}_1 \xrightarrow{t_2} \ldots \xrightarrow{t_k} \mathbf{m}_k$. The set $RS(\mathbf{m}_0)$ of all reachable markings of a Petri net from a given initial marking $\mathbf{m}_0$ can be derived iteratively: Denote $RS^{[0]}(\mathbf{m}_0) = \{\mathbf{m}_0\}$, i.e., with a zero-length firing sequence only the marking itself is "reachable". The set of markings reachable by a firing sequence with length not greater than $i + 1$ can then be iteratively defined:

$$RS^{[i+1]}(\mathbf{m}_0) = RS^{[i]}(\mathbf{m}_0) \cup \{\mathbf{m}' \mid \exists \mathbf{m} \in RS^{[i]}(\mathbf{m}_0), t \in T : \mathbf{m} \xrightarrow{t} \mathbf{m}'\}$$

Finally, $RS(\mathbf{m}_0) = \lim_{i=0\ldots\infty} RS^{[i]}(\mathbf{m}_0)$, although we restrict ourselves to finite state spaces in which $RS(\mathbf{m}_0)$ is the fixed point of the step $RS^{[i]}(\mathbf{m}_0) \to RS^{[i+1]}(\mathbf{m}_0)$.

The RESTART Algorithm

A simulation algorithm estimates the value of a *performance measure*. Typical elements of such measures can be the specified using the concept of *rewards* [21]. *Rate rewards* are associated with states of the stochastic process, and gained over time with a certain rate. An example is the cost of a machine when it is active. On the other hand, *impulse rewards* are earned or paid when a certain event is executed. This can be used, e.g., to model the revenue for a finished part when the associated transition fires.

It is clear that the estimation of all types of rewards may suffer from a rare event situation. In the literature, most of the approaches describe rare-event simulation techniques for either the question of how probable a rare state is in steady state (a rate reward), or what the probability is that the stochastic process visits a rare state before returning to the set of "normal" (initial) states. Less attention (at least explicitly) has been paid to impulse rewards, although they may be sped up similarly [13]. In this paper we only consider impulse rewards in the Petri net setting, i.e., the question of how often a transition will fire on average in steady-state.

Thus we assume that the goal of a simulation is to estimate the throughput (or firing frequency) γ of a transition t_{rare} that only fires rarely. Let the set of all reachable states of a model be denoted by $RS_0 = RS \subseteq M$, and the initial state of the system by $\mathbf{m}_0$ as usual.

A standard simulation would require a very long run time until t_{rare} has been fired sufficiently often to estimate γ. The idea of importance splitting techniques in general is to let t_{rare} happen more frequently by concentrating on promising paths in the reachability set. A standard simulation would usually deviate from these and take paths leading "away" from regions of the reachability set that would enable t_{rare}. If we can find a measure of "how far away" from enabling t_{rare} a marking is, it becomes possible to decide which paths are more likely to succeed and should be followed more frequently. Such a

measure can often be deduced from the actual application: rare failures may be the result of a continuous wear-and-tear, and unavailability or losses because of blocking happen after a buildup in buffers. Corresponding regions of the state set with increasing probability of reaching t_{rare} can then be defined. The samples resulting from such a guided simulation have to be corrected by the ratio w.r.t. the probability that the path would have been taken in a standard simulation to avoid a bias.

Formally, define L regions $RS_1 \dots RS_L$ of the overall state space RS_0 such that all markings enabling t_{rare} are included in RS_L and

$$RS_L \subset RS_{L-1} \subset \dots \subset RS_1 \subset RS_0$$

The conditional probabilities $\mathrm{P}\{RS_{i+1} \mid RS_i\}$ are much easier to estimate than γ because each one of them is not rare if the RS_i are chosen properly. The measure of interest can then be obtained with the product of the conditionals, when $\gamma(RS_i)$ denotes the transition throughput if only markings of RS_i are considered[1].

$$\gamma = \gamma(RS_L) \prod_{i=0}^{L-1} \mathrm{P}\{RS_{i+1} \mid RS_i\}$$

An importance splitting simulation measures the conditional probability of reaching a state out of set RS_{i+1} after starting in RS_i by a Bernoulli trial. If RS_{i+1} is hit, the entering state is stored and the simulation trial is split into R_{i+1} trials (the *splitting factors*). The simulation follows each of the trials to see whether RS_{i+2} is hit and so on. A trial starting at RS_i is canceled after leaving RS_i if it did not hit RS_{i+1}. Simulation of paths inside RS_0 and RS_L is not changed.

An estimator of γ using R_0 independent replications is then [16], [22]

$$\widehat{\gamma} = \frac{\gamma(RS_L)}{R_0 R_1 \dots R_{L-1}} \sum_{i_0=1}^{R_0} \dots \sum_{i_{L-1}=1}^{R_{L-1}} \mathbf{1}_{i_0} \mathbf{1}_{i_0 i_1} \dots \mathbf{1}_{i_0 i_1 \dots i_{L-1}}$$

if we denote by $\mathbf{1}_{i_0 i_1 \dots i_j}$ the result of the Bernoulli trial at stage j, which is either 1 or 0 depending on its success.

Several variants of RESTART have been considered in the literature [23]. We follow the approach taken in [11], [16], which can be characterized as *fixed splitting* and *global step* according to [23]. The first aspect corresponds to the number of trials into which a path is split when it reaches a region closer to the event of interest. The second issue governs the sequence in which the different trials are executed. Global step has the advantage to store fewer intermediate simulation states.

An estimator for steady-state measures in a RESTART implementation needs correction factors that take into account the splitting. We adopt the method of [16], where *weights* are maintained during the simulation run, which capture the relative importance of the current path elegantly. Whenever our transition of interest t_{rare} fires during the simulation run, its impulse reward value is multiplied by the current weight of the run and accumulated in an intermediate reward variable.

[1]Obviously $\mathrm{P}\{RS_0\} = 1$.

Importance Function

The RESTART simulation algorithm must be able to detect for each entered state, to which region RS_i it belongs. Technically, this is implemented using an *importance function* f_I [2], [3] which returns a real value for each state. The value $f_I(\mathbf{m}) \in \mathbb{R}$ should estimate how close to the event (or state) of interest the current marking is. Guidelines on how to choose such a function are given in [5].

A set of *thresholds* (denoted by $Thr_i \in \mathbb{R}, i = 1 \dots L$) divides the range of importance values such that the state set RS_i can be obtained for a state[2].

$$\begin{aligned} \forall i \in \{0 \dots L\} \quad &: \quad Thr_{i+1} > Thr_i \\ \mathbf{m} \in RS_i \quad &\iff \quad f_I(\mathbf{m}) \geq Thr_i \end{aligned}$$

Setting of threshold values is discussed below. We say that the simulation is in a *level* i if the current state $\mathbf{m}$ belongs to $RS_i \setminus RS_{i+1}$.

$$Level(\mathbf{m}) = i \iff Thr_i \leq f_I(\mathbf{m}) < Thr_{i+1} \qquad (1)$$

The reduction in computation time results from estimating the conditional probabilities $\mathrm{P}\{RS_{i+1} \mid RS_i\}$, which are not rare if the sets RS_i are selected properly. Even more computational effort is saved by discarding paths that leave a set RS_i, and which are therefore deemed unsuccessful. The speedup is optimal when the importance is equal for all states on the boundaries of regions [2]. Under simplifying assumptions, this is achieved if the sets are chosen such that [24]

$$\begin{aligned} L &= -\frac{1}{2} \ln(\mathrm{P}\{A\}) \\ \mathrm{P}\{RS_{i+1} \mid RS_i\} &= e^{-2} \\ R_i &\approx \frac{1}{\mathrm{P}\{RS_{i+1} \mid RS_i\}} = e^2 \end{aligned}$$

Later results of the same authors recommend an alternative setting such that $\mathrm{P}\{RS_{i+1} \mid RS_i\} = 1/2$, if it is possible to set the thresholds dense enough [6]. It should be noted that the optimal conditional probabilities as well as number of retrials do not depend on the model.

The numbers of retrials R_i can only be set approximately, because they have to be natural numbers. Apart from that, the optimal setting of the RESTART parameters for a given model and performance measure is not trivial. The first problem is that, following the equations above, the final result needs to be known before the simulation has been started. The second problem is how the importance function and especially the thresholds should be set. The problem of finding good thresholds is e.g. discussed in [5], [16]. Finally, even if the optimal settings are known, the model structure might require to set them differently.

However, even if the optimal efficiency might not be reached easily, experiences show that the technique works robustly for a wide range of applications [4], [6]. If it is at least possible to specify the optimal number of thresholds, they should be

[2]We assume $Thr_0 = -\infty$ and $Thr_{L+1} = \infty$ here to simplify notation.

set such that every $\mathrm{P}\{RS_{i+1} \mid RS_i\}$ is as close as possible to e^{-2} (or $1/2$ following [4]).

The advantage of RESTART compared to standard simulation actually increases for lower probabilities of visiting RS_L (and thus smaller γ), and speedups of several orders of magnitude have been reported. Formulas for the asymptotical speedup to be achieved with a RESTART simulation using optimal parameters are derived in [4], [6].

Application of the RESTART method is restricted by two problems: how many levels should be chosen, and what is a useful importance function? Preferably, both questions should be answered automatically by a future software tool implementation. The number of levels depends on the unkown value of $\mathrm{P}\{RS_L\}$ or γ. The following two sections present ideas to tackle these issues in a Petri net setting.

III. A-Priori Estimation of Rare Event Frequencies

The computation of approximated quantitative values of stochastic Petri nets can be done through several techniques, e.g. *response time approximation* [25], [26], where relatively accurate results are achieved with a computational effort that is still high. The goal of most approximation techniques is not a faster computation, but to cope with models that are too complex to be analyzed exactly. A first estimate of a probability in the rare event simulation context is obtained with a backward simulation for the simple case of finite-state Markov chains [2].

For certain classes of Petri nets, efficient algorithms based on linear programming problems (LPP) exist for the computation of upper and lower bounds of performance measures [27]. In the following, we denote with $\chi_+[t_i]$ and $\chi_-[t_i]$ the upper and lower bound of the real throughput $\chi[t_i]$ of transition t_i ($\gamma = \chi[t_{rare}]$) in steady state.

The basis of an algebraic treatment of Petri nets is their incidence matrix $\mathbf{C}$ and state equation. If we denote by $\sigma \in \mathbb{N}^{|T|}$ a vector that counts for each transition t of the Petri net the number of firings that occurred in some firing sequence starting at the initial marking $\mathbf{m}_0$, the finally reached marking $\mathbf{m}$ is given by the *state equation* $\mathbf{m} = \mathbf{m}_0 + \mathbf{C} \cdot \sigma$ [19].

There are two important types of linear invariants that can be obtained from the state equation. A vector $\mathbf{y}$ is called a *P-flow*[3] iff

$$\mathbf{y}\cdot\mathbf{C} = 0 \quad \longrightarrow \quad \mathbf{y}\cdot\mathbf{m} = \mathbf{y}\cdot\mathbf{m}_0 + \mathbf{y}\cdot\mathbf{C}\cdot\sigma = \mathbf{y}\cdot\mathbf{m}_0 = \text{constant}$$

because the weighted sum of tokens specified by $\mathbf{y}$ for every reachable marking $\mathbf{m}$ is constant (see e.g. [19]).

Non-trivial and non-negative P-flows (i.e. if $\mathbf{y} \geq \mathbf{0}$) are called *P-semiflows*. They correspond to *conservative* parts of a model, where tokens are neither lost nor created.

It is possible to derive transition invariants similar to place invariants: A vector $\mathbf{x} \in \mathbb{N}^{|T|}$ is called a *T-flow* if the following equation holds.

$$\mathbf{C}\cdot\mathbf{x} = 0 \quad \longrightarrow \quad \mathbf{m}' = \mathbf{m} + \mathbf{C}\cdot\mathbf{x} = \mathbf{m}$$

[3]Sometimes informally called *Place invariant* in the literature.

A non-negative, non-trivial T-flow is called *T-semiflow*, and describes multi sets of transitions that, when fired in a sequence, always lead back to the first marking. This does however not guarantee that such a sequence is in fact always executable.

P- and T-semiflows are said to be *minimal* if there is no smaller P-semiflow. An alternative condition is that there is no other semiflow of the same type that has a strictly smaller support. The set of minimal semiflows is unique for a model; it represents the basis of a vector space containing all P- or T-semiflows. Efficient algorithm for their computations exist, see e.g. [19], [28], [29].

Performance bounds can be derived with the following method for any kind of Petri net, but they are more exact if the models are restricted to the class of *FRT-nets* (short for freely related T-semiflows). Informally, this requires mainly that if there are transitions in conflict, the probabilities of firing each of them can be computed from the net structure. Hence conflicts are only allowed between transitions that are in *equal conflict relation*, that is, their pre-incidence function is the same: $\mathbf{Pre}[\cdot, t_i] = \mathbf{Pre}[\cdot, t_k]$. Additionally, there must not be different T-semiflows for which the relative throughput cannot be computed from the net structure.

Conflicts are restricted to immediate transitions for the implemented algorithm, which can be achieved for most net structures. Moreover, the net has to be *structurally bounded* and *structurally life*, which should be the case for most correctly modeled systems; and timed transitions are considered as having *infinite server* firing semantic (single server behavior can be emulated by adding a few elements).

First, routing rates at conflict of the Petri net system are derived for all pairs of transitions t_i, t_j which are in conflict (denoted by $\mathrm{EC}(t_i, t_j)$).

$$\forall t_i, t_j \in T : \mathrm{EC}(t_i, t_j) \quad \text{if} \quad \mathbf{Pre}(t_i) = \mathbf{Pre}(t_j)$$

The relation is an equivalence relation, which divides the set of transitions T into a set of nonempty sets $T_1 \ldots T_k$ of transitions which have no common elements and together form the set T. Every transition t belongs to exactly one of the sets T_i.

The relative *visit ratios* $\mathbf{v}^{(1)}[t_i]$ of transitions are computed in the following, denoting the number of times that transition t_i fires in steady-state in relation to transition t_1:

$$\forall T_i, \forall t_1 \ldots t_k \in T_i : \begin{array}{rcl} r_2\mathbf{v}[t_1] - r_1\mathbf{v}[t_2] &=& 0 \\ r_3\mathbf{v}[t_2] - r_2\mathbf{v}[t_3] &=& 0 \\ \cdots & & \\ r_k\mathbf{v}[t_{k-1}] - r_{k-1}\mathbf{v}[t_k] &=& 0 \end{array} \tag{2}$$

where r_k denotes the relative routing rate at conflict of transition t_k. These rates are given in the model definition for the immediate transitions by their *weights* (relative firing probabilities). They are computed iteratively following

$$r_1 = 1;\ r_{i+1} = r_i \frac{W(t_{i+1})}{W(t_i)} \tag{3}$$

Because of the definition of the FRT net class all transitions are a member of a T-semiflow, and all T-semiflows are related

by relative visiting rates set by the conflict probabilities. The *vector of visit ratios* $\mathbf{v}^{(1)}$ is therefore uniquely defined.

The homogeneous system of linear equations can be expressed in matrix form as $\mathbf{R}[T_i] \cdot \mathbf{v}^{(1)} = 0$ where $\mathbf{R}[T_i]$ is a $|T_i| \times |T|$ matrix and combined to the *routing matrix* $\mathbf{R}$:

$$\mathbf{R} = \begin{pmatrix} \mathbf{R}[T_1] \\ \vdots \\ \mathbf{R}[T_n] \end{pmatrix} \tag{4}$$

The information that transitions belonging to the same T-semiflow need to have corresponding visit ratios is added by extending the routing matrix $\mathbf{R}$ by the token flow matrix $\mathbf{C}$. This results in the following system of linear equations normalized for transition t_1:

$$\begin{pmatrix} \mathbf{C} \\ \mathbf{R} \end{pmatrix} \cdot \mathbf{v}^{(1)} = 0, \quad \mathbf{v}^{(1)}[t_1] = 1 \tag{5}$$

Solving this system of equations results in the visit ratios of all transitions of the net with respect to transition t_1.

Little's Law can be applied to each subnet of a model that consists of a timed transition t_1 and (one of its) input place(s) p. We denote in the following by $\overline{\mathbf{m}}[p]$ the mean number of tokens in place p, the average token waiting (or residence) time in p by $\overline{r}[p]$, and by $\chi[t_1]$ the throughput of transition t_1. Applying Little's Law to such a subnet leads to

$$\overline{\mathbf{m}}[p] = (\mathbf{Pre}[p, \cdot] \cdot \chi)\, \overline{r}[p] \tag{6}$$

$$= \mathbf{Pre}[p, t_1]\, \chi[t_1]\, \overline{r}[p] \tag{7}$$

The equation also holds for immediate transitions, because both firing time (and thus token waiting time) as well as marking is zero then.

$$\overline{\mathbf{m}}[p] = \overline{r}[p] = 0 \tag{8}$$

It is obvious that the mean token waiting time $\overline{r}[p]$ in place p is at least as long as the mean service time $\overline{s}$ of the output transition t_1 (which is given by its firing delay specification Λ).

$$\overline{r}[p] \geq \overline{s}[t_1] \tag{9}$$

leading to

$$\overline{\mathbf{m}}[p] \geq \mathbf{Pre}[p, t_1]\, \chi[t_1]\, \overline{s}[t_1] \tag{10}$$

$\mathbf{Pre}[p, t_j] = 0$ for $j \neq 1$ holds because there are no conflicting timed transitions. Thus

$$\overline{\mathbf{m}}[p] \geq \sum_{j=1}^{m} \mathbf{Pre}[p, t_j]\, \chi[t_j]\, \overline{s}[t_j] \tag{11}$$

Multiplication with the average interfiring time (the inverse of the throughput) $\Gamma[t_1] = 1/\chi[t_1]$ results in

$$\Gamma[t_1]\, \overline{\mathbf{m}}[p] \geq \sum_{j=1}^{m} \mathbf{Pre}[p, t_j]\, \Gamma[t_1]\, \chi[t_j]\, \overline{s}[t_j] \tag{12}$$

and since $\Gamma[t_1]\, \chi[t_j] = \dfrac{\chi[t_j]}{\chi[t_1]} = \mathbf{v}^{(1)}[t_j]$,

$$\Gamma[t_1]\, \overline{\mathbf{m}}[p] \geq \sum_{j=1}^{m} \mathbf{Pre}[p, t_j]\, \mathbf{v}^{(1)}[t_j]\, \overline{s}[t_j] \tag{13}$$

We define *average service demands* $\overline{\mathbf{D}}^{(1)}$ relative to transition t_1 as

$$\overline{\mathbf{D}}^{(1)}[t_i] = \mathbf{v}^{(1)}[t_i]\, \overline{s}[t_i] \tag{14}$$

By multiplying the visit ratio with the service demand of a transition, a measure of the relative workload of the transition corresponds to the service demand, which can be expressed in matrix form (Eq. (13))

$$\Gamma[t_1]\, \overline{\mathbf{m}} \geq \mathbf{Pre} \cdot \overline{\mathbf{D}}^{(1)} \tag{15}$$

Following the definition and properties of P-semiflows [28] we know that $\mathbf{y} \cdot \mathbf{m} = \mathbf{y} \cdot \mathbf{m}_0$ for all markings $\mathbf{m}$ of the Petri net reachable from the initial marking $\mathbf{m}_0$. It can be concluded that for the average marking $\mathbf{y} \cdot \overline{\mathbf{m}} = \mathbf{y} \cdot \mathbf{m}_0$ holds. Using this together with Eq. (15) results in a lower bound for the average interfiring time of transition t_1:

$$\Gamma[t_1] \geq \max_{\mathbf{y} \in \{\text{P-semiflows}\}} \frac{\mathbf{y} \cdot \mathbf{Pre} \cdot \overline{\mathbf{D}}^{(1)}}{\mathbf{y} \cdot \mathbf{m}_0} \tag{16}$$

The search for a lower bound in Eq. (16) can be formulated as a fractional programming problem:

$$\begin{array}{lll} \Gamma[t_1] = & \text{maximum} & \dfrac{\mathbf{y} \cdot \mathbf{Pre} \cdot \overline{\mathbf{D}}^{(1)}}{\mathbf{y} \cdot \mathbf{m}_0} \\ & \text{subject to} & \mathbf{y} \cdot \mathbf{C} = 0 \\ & & \mathbf{1} \cdot \mathbf{y} > 0 \\ & & \mathbf{y} \geq 0 \end{array} \tag{17}$$

which can be rewritten because for life systems $\mathbf{y} \cdot \mathbf{m}_0 > 0$ holds:

$$\begin{array}{lll} \Gamma[t_1] = & \text{maximum} & \mathbf{y} \cdot \mathbf{Pre} \cdot \overline{\mathbf{D}}^{(1)} \\ & \text{subject to} & \mathbf{y} \cdot \mathbf{C} = 0 \\ & & \mathbf{y} \cdot \mathbf{m}_0 = 1 \\ & & \mathbf{y} \geq 0 \end{array} \tag{18}$$

where $\mathbf{y}$ is a P-semiflow and $\mathbf{C}$ denotes the token flow (incidence) matrix. Equation (18) constitutes a standard linear programming problem (LPP) which can be solved efficiently in practice. An interpretation of the LPP is to search for the "slowest subsystem" among the ones defined by P-semiflows in isolation, similar to a bottleneck analysis.

Throughput upper bounds χ_+ can now be computed from the mean interfiring time $\Gamma[t_1]$ of t_1

$$\chi_+[t_1] = \frac{1}{\Gamma[t_1]} \tag{19}$$

The bounds for all other transitions are directly calculated from the result and the relative corresponding visit ratios.

$$\chi_+[t_i] = \chi_+[t_1]\, \mathbf{v}^{(1)}[t_i] \tag{20}$$

A pessimistic upper bound for the average interfiring rate of transition t_1 is computed by assuming that the worst case

for firing this transition again is after having fired all other transitions the number of times that their visit ratio specifies:

$$\Gamma[t_1] \leq \sum_{t \in T} \mathbf{v}^{(1)}[t]\, \overline{s}[t] = \sum_{t \in T} \overline{\mathbf{D}}^{(1)}[t] \tag{21}$$

which leads to obvious lower bounds χ_- for the transition throughputs:

$$\chi_-[t_1] = \frac{1}{\sum_{t \in T} \overline{\mathbf{D}}^{(1)}[t]} \tag{22}$$

$$\chi_-[t_j] = \chi_-[t_1]\, \mathbf{v}^{(1)}[t_j] \tag{23}$$

Upper and lower bounds of transition throughputs can be used to estimate throughput values of transitions with a weighted sum.

$$\chi[t_i] \approx \alpha \chi_-[t_i] + (1-\alpha)\chi_+[t_i]$$

Based on such an estimation, a roughly correct splitting factor and number of levels is chosen below.

IV. A Heuristic Importance Function for Stochastic Petri Nets

While a good importance function for a splitting simulation can be found for special cases with some background knowledge, it would be hard to give model-level rules for the general case. Although the RESTART method can be used with a "black box" simulation model, the importance function can only be defined with some information about the model (and understanding of its behavior). If we restrict ourselves to stochastic Petri nets as the base model, we may still not know what the model describes, but we can apply Petri net analysis results to derive an importance function.

Recall that the regions $RS_1 \ldots RS_L$ try to capture states from which it is equally likely to arrive at the rare state of interest. An obvious heuristic to organize states of any stochastic discrete-event system into these regions depends on how many state transitions are necessary to reach the state (or event) of interest. In Petri net terms, we are interested in the number of transition firings that are necessary to reach any $\mathbf{m}_{rare}$ that will enable the transition of interest t_{rare} from a given current state $\mathbf{m}$. This can of course only be an approximate heuristic as it does not take into account transition firing delays $\Lambda(t_\cdot)$, which may influence the actual probabilities to reach the rare marking significantly. In this paper, however, we only consider the structural information.

Formally, we would like to derive the length k of the shortest firing sequence of transitions $t_1 \ldots t_k$ that will lead from our current marking $\mathbf{m}$ to any marking out of $M_{rare} = \{\mathbf{m} \in M \mid \forall p \in P : \mathbf{m}(p) \geq \mathbf{Pre}(p, t_{rare})\}$, which enable t_{rare}.

We define the *state distance* δ [30] between two markings (called *shortest path distance* in [31])

$$\delta(\mathbf{m}, \mathbf{m}') = n \iff \mathbf{m}' \in \left(RS^{[n]}(\mathbf{m}_0) \setminus RS^{[n-1]}(\mathbf{m}_0) \right)$$

For completeness, the distance is set to infinity if the state is not reachable: $\delta(\mathbf{m}, \mathbf{m}') = \infty \iff \mathbf{m}' \notin RS(\mathbf{m})$. Following the notation in [31], this distance metric is extended to conditions ψ, i.e., Boolean expressions on markings $\psi : M \to \{\text{True}, \text{False}\}$:

$$\delta(\mathbf{m}, \psi) = \min \{\, \delta(\mathbf{m}, \mathbf{m}') \mid \mathbf{m} \in M, \psi(\mathbf{m}') = \text{True} \,\}$$

Obvioulsy, the first marking-based distance specification is then just a special case.

Setting ψ as $\forall p \in P : \mathbf{m}(p) \geq \mathbf{Pre}(p, t_{rare})$ will then result in $\delta(\mathbf{m}, \psi)$ specifying the shortest distance to enable t_{rare}.[4]

Computing exact state distances is similar to the general issue of reachability, and search algorithms based on the reachability graph could be used just like it is, for instance, done in model checking. As long as the reachability graph is small enough to be completely derived and stored, the distances δ could be analyzed and stored a-priori.

Such a solution is a side result of the algorithm presented in [30]. The authors describe an efficient algorithm for the reachability graph generation using a novel edge-valued multiple-valued decision diagram. A *token distance matrix* describing the shortest firing sequence lengths $\delta(\mathbf{m}_0, \mathbf{m})$ to enable any node $\mathbf{m}$ when starting from the initial state $\mathbf{m}_0$ is derived on the fly. This distance is interesting to discuss exploration strategies for the reachability graph. A very efficient fixed-point algorithm generates and stores the exact distances from the initial marking to all reachable states, on the fly together with the actual RG generation. This algorithm could be extended to compute the distances from every state.

However, (exact) solutions based on a reachability graph search are not useful if the reachability graph is too large to be handled, which is one of the reasons to apply simulation instead of a numerical analysis. Moreover, its computational cost both in memory and time depends on the size of the complete underlying stochastic process. A hybrid combination in cases where simulation is used only because of non-memoryless delay distributions, while the state space is manageable and the distances could be completely computed a-priori and used during the RESTART simulation, could be an interesting future task. In this paper, however, we focus on heuristics that can be derived from the Petri net structure and do not require a state space exploration.

Let $h(\mathbf{m}, \psi)$ denote a heuristic estimate of $\delta(\mathbf{m}, \psi)$. To be useful for a heuristic search, it should comply with some basic properties. Following the notation in [31], h is *admissible* (or *optimistic*) if $\forall \mathbf{m} \in M : h(\mathbf{m}, \psi) \leq \delta(\mathbf{m}, \psi)$, i.e., it never overestimates the actual distance. Furthermore it is *monotone* if for each marking $\mathbf{m}'$ directly reachable from $\mathbf{m}$ holds

$$\mathbf{m} \xrightarrow{t} \mathbf{m}' \longrightarrow h(\mathbf{m}', \psi) - h(\mathbf{m}, \psi) \leq 1$$

Monotone heuristics with

$$\forall \mathbf{m} \in M : \psi(\mathbf{m}) = \text{True} \longrightarrow h(\mathbf{m}, \psi) = 0$$

are admissible [31].

[4] Reachable sets $RS^{[i]}$ and distance δ could have been defined based on a finally enabled rare transition t_{rare} alternatively, but the chosen way of description can be used for both (sub-)markings and transitions, and maps better to the later linear algebraic approach.

The authors of [31] present some heuristics for shortest paths in Petri nets. The first is based on a simple Hamming distance, while another one uses subnet refinements but requires a coverability graph analysis for the subnet(s). A deadlock search heuristic is also presented.

For restricted classes of Petri nets the state distance can be derived much easier. Shortest paths are derived from the net structure in polynomial time in [32] for 1-bounded nets, i.e., in which there must not be more than one token in a place in each marking. The method ignores input arcs of transitions and checks how many transitions must fire to mark a place. This idea is similar to planning approaches in artificial intelligence applications.

For the restricted subclass of *homogeneous synchronized choice Petri nets*, reachability is directly decidable in polynomial time based on a structural representation, and the shortest firing sequence can be derived exactly [33].

For life and safe *extended free-choice nets* (and similar other Petri net subclasses), lower and upper bounds for shortest paths between markings are proved in [34]. However, as the bounds depend on the overall number of transitions and are not marking-specific, they cannot be used as a distance heuristic.

Building on an algebraic description of Petri nets, a new distance heuristic is proposed in Equation 24. Just like for the bounds in Section III, a lower bound for the distance can be derived with the solution of a linear programming problem containing model properties as restrictions.

$$\begin{array}{lrcl} \text{minimize} & h(\mathbf{m},\psi) & = & \mathbf{x}\,\mathbf{1} \\ \text{subject to} & \mathbf{m}' & = & \mathbf{m} + \mathbf{C}\,\mathbf{x} \\ & \mathbf{x} & \geq & 0 \\ & \mathbf{m}' & \geq & 0 \\ & \psi(\mathbf{m}') & = & \text{True} \end{array} \tag{24}$$

The minimized result in the first equation counts the number of transition firings by multiplying a firing vector $\mathbf{x}$ by a vector of ones with corresponding size. The second equation contains the state (or marking) equation, and the third and fourth ensure that only nonnegative number of firings or tokens are allowed. Finally, the individual condition ψ must hold in the reached marking(s). For the case of the distance to enable transition t_{rare} it would be $\mathbf{m}(\mathbf{m}') \geq \mathbf{Pre}(\mathbf{m}', t_{rare})$.[5]

This heuristic will not be exact, but result in a lower bound of the distance: There is no guarantee that the transition multi set described by the firing count vector $\mathbf{x}$ in the LPP will be fireable in a sequence. This usually happens because of priorities or inhibitor arcs, as they are not part of the structural description and reduce the reachability graph.

Admissibility:

Obviously, $h(\mathbf{m}_{rare}, \psi) = 0$ holds if $\psi(\mathbf{m}_{rare}) = \text{True}$ with $\mathbf{m} = \mathbf{m}_{rare}$ and $\mathbf{x} = (0, \ldots, 0)^{\mathrm{T}}$.

The heuristic is monotone as well: Select any two markings that are directly reachable $\mathbf{m} \xrightarrow{t} \mathbf{m}'$, and assume that $h(\mathbf{m}', \psi) = n$. Then there must be a firing count vector $\mathbf{x}'$ with $n = \mathbf{x}'\,\mathbf{1}$ and $\mathbf{m}_{rare} = \mathbf{m}' + \mathbf{C}\,\mathbf{x}'$. Because $\mathbf{m}'$ is directly reachable from $\mathbf{m}$ by firing t we know that $\mathbf{m}' = \mathbf{m} + \mathbf{C}(., t)$. Thus a firing count vector $\mathbf{x}$ that is incremented for the fired transition t can be found, such that

$$\forall t_i \in T : \mathbf{x}(t_i) = \begin{cases} \mathbf{x}'(t_i) + 1 & \text{if } t_i = t \\ \mathbf{x}'(t_i) & \text{else} \end{cases}$$

which will solve the state equation $\mathbf{m}_{rare} = \mathbf{m} + \mathbf{C}\,\mathbf{x}$ and for which holds $\mathbf{x}\,\mathbf{1} + 1 = \mathbf{x}'\,\mathbf{1}$. Therefore, the heuristic $h(\mathbf{m}, \psi)$ will be $n + 1$ at most. It could also be n in cases where a shorter firing sequence exists that does not visit $\mathbf{m}'$. It thus follows that the heuristic is admissible. □

Applying the method to the initial marking $\mathbf{m}_0$, we estimate the maximum possible number of levels by $h_{\max} = h(\mathbf{m}_0, \psi)$ with the condition ψ set to $\mathbf{m}(\mathbf{m}') \geq \mathbf{Pre}(\mathbf{m}', t_{rare})$ as stated earlier.

Based on the distance heuristic and frequency estimation, we can now define an importance function f_I for stochastic Petri nets:

$$f_I(\mathbf{m}) = h_{\max} - h(\mathbf{m}, \psi)$$

This function will start with value zero at the initial marking, and gradually increases until $h_{\max}$ for states in which the condition holds. Please note that there are of course cases in which other markings in the Petri net exist for which the distance of $\mathbf{m}_0$ and thus $h_{\max}$ is exceeded; however, these markings will then all belong to the same splitting region as $\mathbf{m}_0$ and no further splitting will be done "below". This means that also the initial marking should be chosen with care to allow a maximum of regions.

Splitting probabilities $p_k = \mathrm{P}\{RS_k \mid RS_{k-1}\}$ should be equal and chosen between $1/2$ and e^{-2} (c.f. Section II), leading to an optimal number of tresholds $L_{\text{opt}} = \frac{\ln \gamma}{\ln p_k}$. For an example case of $\gamma = 10^{-20}$, this would result in $23 < L < 66$ levels and thresholds, depending on the selected optimality assumption. The number of desired levels may thus be easily greater than an available maximum, thus we set $L = \min(L_{\text{opt}}, h_{\max})$ to have unique thresholds.

It remains to set the levels, which is done equidistantly as we have no better heuristic information.

$$Thr_i = \left\lfloor i\,\frac{h_{\max}}{L} \right\rfloor$$

V. Conclusion

The paper presented two ideas to simplify the search for a good importance function when a stochastic Petri net is simulated using the RESTART method. Using results based on an algebraic formulation of the Petri net structure, an a-priori estimation of rare event frequencies as well as a distance measure from a rare state have been presented. The first estimate helps to set a near-optimal number of splitting thresholds, while the second result can be used as a rough guideline for the importance function.

In the future, we plan to implemented the ideas in our tool TimeNET, which will then allow to assess the achievable speedup for application examples.

[5] In cases like these, the fourth equation is not necessary, as it is included in the last one.

An open question to be investigated is how well a distance measure that uses only structural information will guide a simulation path whose "rareness" depends on transition firing times as well. In addition to that it is necessary to check the actual computational effort of the heuristics to select a proper tradeoff between importance function computation and the actual rare-event simulation.

ACKNOWLEDGMENT

The authors would like to thank Javier Campos and Enrique Teruel for their valuable comments on an algebraic formulation of the distance estimation.

REFERENCES

[1] M. Villén-Altamirano and J. Villén-Altamirano, "RESTART: A straightforward method for fast simulation of rare events," in *Proc. Winter Simulation Conference*, 1994, pp. 282–289.

[2] M. J. Garvels, J.-K. C. Van Ommeren, and D. P. Kroese, "On the importance function in splitting simulation," *European Transactions on Telecommunications*, vol. 13, no. 4, pp. 363–371, 2002.

[3] J. Villén-Altamirano, "Importance functions for RESTART simulation of highly-dependable systems," *Simulation*, vol. 83, no. 12, pp. 821–828, dec 2007.

[4] M. Villén-Altamirano and J. Villén-Altamirano, "Analysis of RESTART simulation: Theoretical basis and sensitivity study," *European Transactions on Telecommunications*, vol. 13, no. 4, pp. 373–385, 2002.

[5] ——, "On the efficiency of RESTART for multidimensional systems," *ACM Transactions on Modeling and Computer Simulation*, vol. 16, no. 3, pp. 251–279, Jul. 2006.

[6] ——, "Optimality and robustness of RESTART simulation," in *Proc. 4th Workshop on Rare Event Simulation and Related Combinatorial Optimisation Problems*, Madrid, Spain, Apr. 2002.

[7] P. J. Haas, *Stochastic Petri Nets: Modelling, Stability, Simulation*, ser. Springer Series in Operations Research. Springer Verlag, 2002.

[8] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, and G. Franceschinis, *Modelling with Generalized Stochastic Petri Nets*, ser. Series in parallel computing. John Wiley and Sons, 1995.

[9] A. Zimmermann, *Stochastic Discrete Event Systems*. Springer, Berlin Heidelberg New York, 2007.

[10] ——, "Dependability evaluation of complex systems with TimeNET," in *Proc. Int. Workshop on Dynamic Aspects in Dependability Models for Fault-Tolerant Systems (DYADEM-FTS 2010)*, Valencia, Spain, Apr. 2010.

[11] C. Kelling, "A framework for rare event simulation of stochastic Petri nets using RESTART," in *Proc. of the Winter Simulation Conference*, 1996, pp. 317–324.

[12] C. Kelling and G. Hommel, "Rare event simulation with an adaptive "RESTART" method in a Petri net modeling environment," in *Proc. 4th Int. Workshop on Parallel and Distributed Real-Time Systems*, Los Alamitos, CA, USA, Apr. 1996, pp. 229–234.

[13] A. Zimmermann, "Applied restart estimation of general reward measures," in *Proc. 6th Int. Workshop on Rare Event Simulation (RESIM 2006)*, Bamberg, Germany, Oct. 2006, pp. 196–204.

[14] ——, "RESTART simulation of colored stochastic Petri nets," in *Proc. 7th Int. Workshop on Rare Event Simulation (RESIM 2008)*, Rennes, France, Sep. 2008, pp. 143–152.

[15] A. Zimmermann and G. Hommel, "Towards modeling and evaluation of ETCS real-time communication and operation," *Journal of Systems and Software*, vol. 77, pp. 47–54, 2005.

[16] B. Tuffin and K. S. Trivedi, "Implementation of importance splitting techniques in stochastic Petri net package," in *Computer Performance Evaluation, Modelling Techniques and Tools — 11th Int. Conf., TOOLS 2000*, ser. Lecture Notes in Computer Science, C. U. S. Boudewijn R. Haverkort, Henrik C. Bohnenkamp, Ed., vol. 1786. Schaumburg, IL, USA: Springer Verlag, 2000, pp. 216–pp.

[17] W. Obal and W. Sanders, "Importance sampling simulation in UltraSAN," *Simulation*, vol. 62, no. 2, pp. 98–111, 1994.

[18] A. Jalaly and M. Abdollahi Azgomi, "Rare event simulation of stochastic activity networks using partition of the region technique," in *Proc. 24th UK Performance Engineering Workshop (UKPEW'08)*, Imperial College London, UK, jul 2008, pp. 107–122.

[19] M. Silva, E. Teruel, and J. M. Colom, "Linear algebraic and linear programming techniques for the analysis of place/transition net systems," in *Lectures on Petri Nets I: Basic Models*, ser. Lecture Notes in Computer Science, G. Reisig, W.; Rozenberg, Ed. Springer Verlag, 1998, vol. 1491, pp. 309–373.

[20] M. Silva and J. Campos, "Structural performance analysis of stochastic Petri nets," in *Proc. IEEE Int. Performance and Dependability Symposium*, Erlangen, Apr. 1995, pp. 61–70.

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

[22] P. Glasserman, P. Heidelberger, P. Shahabuddin, and T. Zajic, "Splitting for rare event simulation: Analysis of simple cases," in *Proc. Winter Simulation Conference*, 1996, pp. 302–308.

[23] M. J. Garvels and D. P. Kroese, "A comparison of RESTART implementations," in *Proc. 1998 Winter Simulation Conference*, 1998.

[24] M. Villén-Altamirano, J. Villén-Altamirano, J. Gamo, and F. Fernández-Cuesta, "Enhancement of the accelerated simulation method RESTART by considering multiple thresholds." in *Proc. 14th Int. Teletraffic Congress*. Elsevier Science Publishers B. V., 1994, pp. 797–810.

[25] H. Jungnitz, B. Sanchez, and M. Silva, "Approximate throughput computation of stochastic marked graphs," *J. Parallel and Distributed Computing*, vol. 2, pp. 282–295, 1992.

[26] C. J. Pérez-Jiménez and J. Campos, "A response time approximation technique for stochastic general P/T systems," in *Proc. Symp. Industrial and Manufacturing Systems, 2nd IMACS Int. Multiconf. on Computational Engineering in Systems Applications (CESA '98)*, Nabeul-Hammamet, Tunisia, Apr. 1998.

[27] J. Campos and M. Silva, "Structural techniques and performance bounds of stochastic Petri net models," in *Advances in Petri Nets 1992*, ser. Lecture Notes in Computer Science, G. Rozenberg, Ed. Springer Verlag, 1992, vol. 609, pp. 352–391.

[28] J. M. Colom and M. Silva, "Convex geometry and semiflows in P/T nets. A comparative study of algorithms for computation of minimal P-semiflows," in *Advances in Petri Nets 1990*, ser. Lecture Notes in Computer Science, G. Rozenberg, Ed. Springer Verlag, 1991, vol. 483, pp. 79–112.

[29] N. Treves, "A comparative study of different techniques for semi-flows computation in place/transition nets," in *Advances in Petri Nets 1989*, ser. Lecture Notes in Computer Science, G. Rozenberg, Ed. Springer Verlag, 1990, vol. 424, pp. 434–452.

[30] G. Ciardo and R. Siminiceanu, "Using edge-valued decision diagrams for symbolic generation of shortest paths," in *Proc. 4th Int. Conf. Formal Methods in Computer-Aided Design (FMCAD 2002)*, ser. LNCS 2517. Spring, nov 2002.

[31] S. Edelkamp and S. Jabbar, "Action planning for directed model checking of Petri nets," *Electronic Notes in Theoretical Computer Science.*, vol. 149, no. 2, pp. 3–18, 2006.

[32] B. Bonet, P. Haslum, S. L. Hickmott, and S. Thiébaux, "Directed unfolding of Petri nets," *Trans. Petri Nets and Other Models of Concurrency*, vol. 1, no. 1, pp. 172–198, 2008, lNCS 5100, Springer.

[33] D. Y. Chao, "Reachability and firing sequences of homogeneous synchronized choice Petri nets," *Journal of Information Science and Engineering*, vol. 21, pp. 129–152, 2005.

[34] P. K. Mishra, "Lower and upper bounds of shortest paths in reachability graphs," *Int. Journal of Mathematics and Mathematical Sciences*, vol. 2004, no. 57, pp. 3023–3036, 2004.