Reliability Engineering and System Safety 222 (2022) 108365

Contents lists available at ScienceDirect

Reliability Engineering and System Safety

journal homepage: www.elsevier.com/locate/ress

Reduction of Petri net maintenance modeling complexity via Approximate Bayesian Computation

Manuel Chiachío [a,b,*], Ali Saleh [a,b], Susannah Naybour [c], Juan Chiachío [a,b], John Andrews [c]

[a] Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain
[b] Dept. Structural Mechanics & Hydraulics Engineering, University of Granada, 18071 Granada, Spain
[c] Resilience Engineering Group, University of Nottingham, University Park, Nottingham, UK

ARTICLE INFO

Keywords:
Petri nets
Model similarity
Bayesian inference
Approximate Bayesian Computation
Maintenance models

ABSTRACT

The accurate modeling of engineering systems and processes using Petri nets often results in complex graph representations that are computationally intensive, limiting the potential of this modeling tool in real life applications. This paper presents a methodology to properly define the optimal structure and properties of a reduced Petri net that mimic the output of a reference Petri net model. The methodology is based on Approximate Bayesian Computation to infer the plausible values of the model parameters of the reduced model in a rigorous probabilistic way. Also, the method provides a numerical measure of the level of approximation of the reduced model structure, thus allowing the selection of the optimal reduced structure among a set of potential candidates. The suitability of the proposed methodology is illustrated using a simple illustrative example and a system reliability engineering case study, showing satisfactory results. The results also show that the method allows flexible reduction of the structure of the complex Petri net model taken as reference, and provides numerical justification for the choice of the reduced model structure.

1. Introduction

In most countries of the developed world, the lifespan of a large part of critical infrastructures (such as transportation and energy assets) is approaching their end. Methods and models for optimal asset management are thus a necessity to avoid an uncontrolled increase of system failures and unexpected downtimes, whilst keeping maintenance and inspection costs under reasonably lower levels [1].

In the literature, a number of mathematical and computational techniques for optimal asset management have matured over the last decades aiming at optimizing the maintenance and operational costs [2]. Some of these methods include soft-computing techniques such as genetic algorithms [3], neural networks [4], and support vector machines [5]; along with event-based models like Markov chains [6], Bayesian networks [7], and Petri nets [8–11], to name but the most relevant ones.

Among the aforementioned approaches, Petri nets (PNs) [12] are typically regarded as powerful modeling tools for maintenance and inspection modeling due to their ability to account for resource availability, concurrency, and synchronization, which are common aspects that underline the majority of the asset management models [13,14]. The basic concepts relative to the theory of PNs are summarized in [15], whereas practical reliability engineering applications of PNs can be found in [16,17]. Moreover, PNs can be combined with other computational techniques such as object-oriented programming [9], fuzzy sets [18], expert system modeling [13,19], etc., which greatly increase their suitability for maintenance and inspection modeling in engineering problems.

Irrespectively, one of the main challenges in adopting PNs to model maintenance of engineering systems is the handling of complex processes, including their inter-dependencies and workflows, such as those appearing in real-world reliability engineering and maintenance domain. The latter requires large PN representations which are difficult (if not impossible) to interpret graphically, and also heavy computational resources for their numerical evaluation [31,32]. To efficiently approach this practical challenge, an increasing number of researchers have reported progress towards the improvement of the efficiency of PN simulation of large maintenance models [33]. These methods require specialist computational hardware and still can be expensive in terms of cost and computational time. Other existing alternatives consist in reducing the PN complexity to partially alleviate their computational cost [21]. These techniques share the common principle of using a set of rules to shorten specific graphical sub-structures commonly found in PNs while preserving the overall PN behavior unviolated. Table 1 lists the main recent contributions to the literature on this topic and

[*] Corresponding author at: Andalusian Research Institute in Data Science and Computational Intelligence, University of Granada, 18071 Granada, Spain.
E-mail address: mchiachio@ugr.es (M. Chiachío).

https://doi.org/10.1016/j.ress.2022.108365
Received 2 September 2021; Received in revised form 27 January 2022; Accepted 30 January 2022
Available online 19 February 2022

Table 1
Summary of recent relevant works about PN reduction methodologies.

Method	Description	Benefits	Limitations	Refs.
Reduction of PN structure following defined rules.	PN reduction in a rule-based way, considering model sub-structures.	Reduction rules are well understood preventing illogical reductions by the user.	Reductions limited to the pre-defined rules. Reduction error not always quantified and limited transitions types considered.	[20–24]
Symmetry reduction for time Petri net state classes	Explore the symmetries in the system to enumerate equivalence states instead of all the reachable states	It can be used with state class abstraction. Can be applied to construct State Class Graph of TPNs	Beneficial only for the cases where symmetries exists	[25]
Counting PN markings from reduction equations.	Reduction by removing redundant places and transitions and representing the effects of reductions by linear equations	Can be combined with other techniques like symmetry reductions. Keep track between the markings of initial and reduced PN.	The method can only reduce PNs with specific properties (redundant transitions, redundant places, or place agglomerations).	[26,27]
Discovering PNs from event logs	The firing sequence of transitions in PN structures is used to recreate event logs.	PN structures and firing times can be discovered. Not limited to PNs with exponential probability models.	Methods can yield different equivalent PN structures. Limited probability models governing the PN delay times are considered.	[28]
Identification of PN time parameters through comparison to event sequences.	PN time parameters are discovered for two normal transitions, in conflict, to match a pre-known event sequence.	The algorithm shows good convergence for simple event sequences.	Normal distribution for the transitions of the PN is assumed. Large event sequences lead to a low computational efficiency.	[29]
Reconstructing PN model parameters through comparison to data.	PN model parameters are discovered by matching the model output to an expected output.	Allows constant probability values within the PN model to be discovered.	The approach is not used to discover probability distributions governing PN transition firing times.	[30]

briefly indicates their key benefits and limitations. From a generic perspective, the existing methods are capable of effectively reducing the model complexity, however this is at the cost of adopting a number of assumptions about the types of nodes that can be used and their parameters, which limit their applicability to real-world cases, precisely where the benefits of these methods can be fully exploited. Hence, there is a need for more versatile and efficient PN reduction techniques to deal with large and complex system reliability and maintenance models.

In this context, this paper proposes a novel methodology for PN complexity reduction based on model similarity with respect to a reference PN model through Bayesian learning. More specifically, the proposed approach finds the most plausible values of a set of model parameters governing the reduced PN by comparing both model outputs, namely the reduced and the reference one, within an Approximate Bayesian Computation framework [34,35]. The resulting inference equations are analytically intractable, hence the ABC-SubSim algorithm [36] is adopted whereby the probability density functions (PDFs) of the model parameters are approximated by samples which represent random realizations of the parameter values from the reduced network. The ABC-SubSim method, which is used in this work due to its ability to perform the parameter inference within multi-dimensional parameter spaces and under very low approximation error [36], produces parameter samples through Markov Chain Monte Carlo (MCMC) sampling over a nested sequence of subsets until a predefined approximation error is reached. To overcome the well-known manual scaling required in MCMC sampling [37], this paper also provides an improved version of the ABC-SubSim algorithm with respect to the original one in [36] by adaptive scaling of the MCMC sampling across the subsets. Finally, and as by product, the proposed inference method also produces a quantitative measure of the reduced model suitability through Bayesian evidence computation [38], which aids the task of model selection in the frequent case of multiple reduced models available.

The resulting methodology is versatile enough to be applied to any generic PN model with firing times governed by any probability distribution, or even to high-level PN variants like the referred high-level Petri Nets (HLPN) [39], which allow consideration of nodes with complex behavior like reset and probability transitions, inhibitor arcs, numerical and predicate places, etc. Here, the proposed approach is tested on a real-world case study about a two-pump lube oil hydraulic system whose inspection, maintenance, and renewal process is modeled through a reference, yet complex, HLPN. Three candidate reduced PN models are defined following a criteria of increasing complexity allowing their suitability to be evaluated. The results show that the proposed reduction methodology is able to infer the parameters governing the inspection, maintenance and renewal processes of the equivalent reduced PN models in liaison with the reference HLPN, demonstrating a high level of accuracy in approximation to the reference model alongside a significant reduction in computational cost. The results also show that the reduced models can provide the reliability curves of the two-pump engineering system with high accuracy with respect to the one obtained using the reference HLPN.

The remainder of the paper is organized as follows. Section 2 presents the fundamentals behind the proposed computational method from basic principles of PNs to the ABC-SubSim method. Section 3 provides description of the adaptive scaling method for the ABC-SubSim algorithm. The mathematical basis and computational aspects of the PN reduction by ABC-SubSim method is given in Section 4. Section 5 illustrates the approach over a simple PN architecture before presenting the approach in application to an engineering case study in Section 6. Section 7 discusses the main findings and provides a comparative analysis, and finally Section 8 gives concluding remarks.

2. Fundamentals

2.1. Petri net modeling

A PN is a directed bipartite graph consisting of two types of nodes, known as *places* and *transitions*. The transitions, which are symbolized using boxes, are responsible of the dynamic behavior of the PN, and enable the system to move from one state to another. The places, which are symbolized using circles, represent particular states of the system (e.g. a component in a failure state or an inspection activity which is currently in progress) [15]. The distribution of tokens over the places at a specific execution time determines the state of the PN, and is expressed as a vector of integer values referred to as *marking*.

Mathematically, a PN can be defined as a tuple $\mathfrak{N} = \left\langle \mathbf{P}, \mathbf{T}, \mathbf{F}, \mathbf{W}, \mathbf{M}_0 \right\rangle$ [15], where $\mathbf{P}$ and $\mathbf{T}$ denote the set of n_p places and n_t transitions of the PN, whose connections are expressed through the set of edges $\mathbf{F} \subseteq (\mathbf{P} \times \mathbf{T}) \cup (\mathbf{T} \times \mathbf{P})$. $\mathbf{W} : \mathbf{F} \to \mathbb{N}_{>0}$ is a weight function which assigns a natural number (1 by default) to each arc within $\mathbf{F}$, and finally $\mathbf{M}_0 \in \mathbb{N}^{n_p}$ is the initial marking. Additionally, in this work the term ${}^{\bullet}\mathbf{P}_t$ will be used to denote the *pre-set* of t, namely the set of input places of transition t, and $\mathbf{P}_t^{\bullet}$ for the set of output places of transition t, also referred to as the *post-set* of t. The dynamics of a PN can be mathematically described through a state equation defined as follows:

$$\mathbf{M}_{k+1} = \mathbf{M}_k + \mathbf{A}^T \mathbf{u}_k \tag{1}$$

where $\mathbf{M}_k$ is the marking at time step k, being $k \in \mathbb{N}$ a time index. $\mathbf{A}$ is the *incidence matrix*, which can be obtained as the result of subtracting the *backward incidence matrix* $\mathbf{A}^- = \left[a_{ij}^-\right]$ from the *forward incidence matrix* $\mathbf{A}^+ = \left[a_{ij}^+\right]$. The terms a_{ij}^- and a_{ij}^+ coincide with the weights w_{ij}^- and w_{ij}^+ of the arcs of transition t_i from its pre-set and post-set places p_j, respectively, where $i = 1, \dots, n_t$, $j = 1, \dots, n_p$. Finally, the term $\mathbf{u}_k = (u_{1,k}, u_{2,k}, \dots, u_{n_t,k})^T$ in Eq. (1) is the *firing vector* whose elements are binary values which adopt the unity if the transition is fired and 0 otherwise, according to the *firing rule* [15].

In practical applications dealing with the modeling of dynamical systems, transitions are typically assigned time delays, referred to as τ. The resulting PNs are called Timed Petri Nets (TPN) if the delays are deterministic, and Stochastic Petri Nets (SPN) if the delays are specified by a probability model [15].

In this work, PN variants referred to as high-level Petri nets (HLPN) [39] are used to consider higher complexities in the systems. One of the variants is the *probability transitions* whose production of tokens to their output arcs is carried out under a specified probability. In mathematical terms, if t_i is a so-called probability transition, then a probability value π_{iz} is assigned to every z^{th} output arc from its post-set $\mathbf{P}_{t_i}^{\bullet}$. Thus, the associated elements to t_i from the forward matrix can be expressed as:

$$a_{iz}^+ = w_{iz}^+ \mathbb{I}_z \tag{2}$$

where w_{iz}^+ is the output arc weight from transition t_i to place $p_z \in \mathbf{P}_{t_i}^{\bullet}$, and $\mathbb{I}_z$ is a Boolean indicator given as follows:

$$\mathbb{I}_z = \begin{cases} 1, & \text{if} \quad z = z' \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

In the last equation, z' is the index of the output arc chosen according to the associated probability of each output arc. Finally note that HLPN models used in this work make also use of *inhibitor arcs* [15] (represented with a small circle as ending, like those depicted in Fig. 4) which produce the opposite effect of the firing rule, i.e. they prevent a transition from being enabled once its pre-set places are marked.

Fig. 1 is provided to illustrate a sample HLPN of five places and four transitions to model the maintenance of an engineering component under abnormal condition. Places p_1, p_2, p_3 and p_4 describe the normal, degraded, critical, and failed states of a component, respectively, whereas place p_5 accounts for the number of times the component changes to any malfunction state. To represent the probability of moving from the normal to any abnormal state, the probability transition t_1 is used, whose delay time τ_1 accounts for the time required before the component malfunction. Transitions t_2, t_3 and t_4 describe maintenance actions to return the component to its normal condition, and their delay times τ_2, τ_3 and τ_4, describe the times of the maintenance operations.

2.2. Approximate Bayesian computation method

The approximate Bayesian computation (ABC) method [34] is under the category of Bayesian statistics and related to the Bayesian model updating approach [38] in those cases where the *likelihood function* is difficult, or analytically intractable. In Bayesian model updating, the interest is about updating the prior knowledge about a set parameters $\theta \in \boldsymbol{\Theta} \subset \mathbb{R}^d$ of a parameterized model based on the information gained from a set of data, $y \in \mathcal{D} \subset \mathbb{R}^{\ell}$. The term $\mathcal{D}$ is the *observation space* which contains all possible observational outcomes according to a specific model class $\mathcal{M}$. If multiple candidate models classes are available, this methodology also provides a framework to assess the plausibility of each model class.

The ABC technique produces posterior samples of the pairs $(\theta, x) \in \mathcal{Z} \subseteq \boldsymbol{\Theta} \times \mathcal{D}$ which makes the model response $x = x(\theta) \sim p(x|\theta)p(\theta)$ lay within a defined region around the data y. This region can be formally defined as:

$$\mathcal{N}_\epsilon(y) = \left\{ x \in \mathcal{D} : \rho\big(\eta(x), \eta(y)\big) \leqslant \epsilon \right\} \tag{4}$$

where $\rho(\cdot) : \mathbb{R}^{d \times \ell} \to \mathbb{R}$ is a metric function to evaluate the closeness of the simulated model output x to the data y, ϵ is a tolerance parameter, and $\eta(\cdot)$ is a summary statistic [40] which, if required, allows the comparison between x and y in a weak manner. The referred posterior samples produce an approximation of the posterior PDF $p_\epsilon(\theta, x|y) = p\big(\theta, x | x \in \mathcal{N}_\epsilon(y)\big)$ which assigns higher probability density values to those pairs $(\theta, x) \in \mathcal{Z}$ that satisfy the condition $\rho\big(\eta(x), \eta(y)\big) \leqslant \epsilon$.

2.3. ABC-SubSim method

As previously mentioned, the success of an ABC method is dependent on the choice of certain hyper-parameters like the summary statistic $\eta(\cdot)$, the metric ρ, and a tolerance value ϵ. Among them, the tolerance hyper-parameter is of special importance since it entails a critical trade-off between accuracy of the posterior approximation and computational cost. In the literature, a number of techniques can be found to address this trade-off by combining the ABC principles with efficient sampling algorithms. The reader is referred to [35] for a comprehensive overview of the ABC methods. Irrespective of the efficiency demonstrated by some of the referred methods, using $\epsilon \to 0$ still translates into heavy computation except for those which sequentially adapt ϵ until reaching a desirable small value [41]. Among those, the so called approximate Bayesian computation by subset simulation, also referred to as ABC-SubSim algorithm [36], has been proved to be one of the most efficient ABC algorithms in the literature having been included in several well-known ABC user-platforms such as ABCpy [42] and Pi4U [43] (https://github.com/cselab/pi4u), and thus is the algorithm adopted for the PN reduction methodology presented in this paper.

The ABC-SubSim depends on the SubSet Simulation, originally presented in [44] as a method to avoid costly or inaccurate rare event simulation that arises due to the existence of one or several small failure regions. By Subset Simulation, the rare event simulation is avoided by the introduction of several intermediate regions $\mathcal{S}_j \subset \mathcal{Z}$ with larger probabilities by generating conditional samples $(\theta, x) \in \mathcal{S}_j, j = 1, \dots, m$, where:

$$\mathcal{S}_j = \left\{ (\theta, x) \in \mathcal{Z} : \rho(\eta(x), \eta(y)) \leq \epsilon_j \right\} \tag{5}$$

In the last equation, the intermediate tolerances ϵ_j follow a decreasing sequence $\epsilon_1 \geqslant \epsilon_2 \ldots \geqslant \epsilon_m \equiv \epsilon$.

A practical way to implement the ABC-SubSim is by fixing the size of the samples generated in $\mathcal{S}_{j-1}$ that lie in $\mathcal{S}_j$ as NP_0, where N is the amount of samples that populate the j^{th} simulation level, and P_0 is a conditional probability acting as algorithm hyper-parameter defined by the modeler [45]. The outcome of this algorithm is a set of $(\theta, x) \in \mathcal{S}$ samples that populate the posterior PDF $p_\epsilon(\theta, x|y)$. Also, and as by-product, the algorithm straightforwardly produces a measure of the *evidence* (also referred to as *marginal likelihood*) of a candidate reduced PN model via the conditional probabilities involved in the simulation [36], as follows:

$$P_\epsilon(y|\mathcal{M}_j) = P(\mathcal{S}_1) \prod_{j=2}^{m} P(\mathcal{S}_j | \mathcal{S}_{j-1}) \approx P_0^m \tag{6}$$

where $\mathcal{M}_j$ indicates the j^{th} model class within a set of available models, which in our case, corresponds to any available reduced PN model. In Section 4, a pseudo-code implementation of the ABC-SubSim algorithm is given within the context of the PN reduction methodology proposed in this work.

3. ABC-SubSim with adaptive sampling

As stated in the last section, in ABC-SubSim algorithm NP_0 samples are selected at every j^{th} simulation level acting as *seeds* of the $\mathcal{S}_j$ subset, which generate new samples via MCMC sampling. In this work, the Modified Metropolis Hastings algorithm [44] (MMA) is used as MCMC method to generate successive chain values from each seed.

Irrespectively, this method requires a tricky selection of a *proposal PDF* used to evolve values in each Markov chain due its strong impact on the quality of the ABC posterior approximation along with on the speed of convergence of the algorithm [46]. In the literature, the optimal scaling of the standard deviation of the proposal PDF has been considered within the context of Subset Simulation by manual calibration and preliminary trials, thus increasing the computational cost [36,37,45].

To avoid the aforementioned drawback, Vakilzadeh et al. [47] proposed an adaptive method to tune the proposal standard deviation per simulation level for the ABC-SubSim algorithm with soundly results. However the performance of the method in [47] requires the introduction of three new sensible hyper-parameter to the algorithm (see inputs N_a, ξ_1 and λ_j in [47, Algorithm 2]), which again needs modeler expertise to tune them. More recently, [48] proposed an adaptive method to tune the conditional probability hyper-parameter P_0 which also helps to scale the proposal standard deviation, as by product. As in [47], the resulting sampling scheme in [48] is automatic, however it is not efficient in avoiding high sampling rejections in subsets of small size.

In this section, a method based on samples' values distributions is presented to obtain appropriate standard deviation automatically, thus avoiding the modeler's intervention. The method is based on the idea that each dimension of the seeds has its own standard deviation. Also the method assumes that the best region to search the value of the proposals is located around their seed, but should not include another seeds. To obtain the seed k' closest to the k^{th} seed, the distances from each seed to all the remaining seeds are calculated. Note that the closest seed to another is the seed with the minimum-strictly positive distance to it, because otherwise a zero distance would imply repetition of the seed, which cannot be properly considered as a seed. In this work, the $\mathbb{L}_2$-distance is used for calculating distance $\jmath(\cdot)$ between two seeds as follows:

$$\jmath_{k-k'} = \left[\sum_{q=1}^{d}\left(\frac{\theta^{(k)}_{j(q)} - \theta^{(k')}_{j(q)}}{\widetilde{\theta}_{j(q)}}\right)^2\right]^{1/2} \tag{7}$$

where $\theta^{(k)}_{j(q)}$ and $\theta^{(k')}_{j(q)}$ are the q^{th}-components of the θ-values of seeds k and k', respectively, where $q = 1,\dots,d$, and $k = 1,\dots,k',\dots,NP_0$. The term $\widetilde{\theta}_q$ is the range of the seed parameter values in the q^{th}-dimension, which is calculated as follows:

$$\widetilde{\theta}_{(q)} = \| \max\left\{\theta^{(k)}_{(q)}\right\}_{k=1}^{NP_0} - \min\left\{\theta^{(k)}_{(q)}\right\}_{k=1}^{NP_0} \| \tag{8}$$

where the subscript j has been avoided for simpler notation. Then, for each of the seeds of the j^{th} simulation level, the standard deviation in each dimension is obtained as the component of the distance from the seed to its closest seed in that dimension, as follows:

$$\sigma^{(k)}_{(q)} = \theta^{(k)}_{(q)} - \theta^{(k')}_{(q)} \tag{9}$$

Remark 1. Since the θ-values are not equally distributed within the parameter space $\boldsymbol{\Theta}$, hence the normalization in Eqs. (7) through $\widetilde{\theta}_q$ enables the parameter components $q = 1,\dots,d$ to equally contribute to the distance $\jmath$. Otherwise, if $\widetilde{\theta}_q$ were discarded, then the effect of each dimension q on $\jmath$ would be larger as the θ-values in that dimension increase.

Remark 2. A limitation of Eq. (9) is that in case that the closest seed k' has the same value as seed k within any dimension q, then $\sigma^{(k)}_q$ will result in zero leading to a lack of exploration for this seed in this dimension. To avoid this critical situation, and in order for the proposals to be fairly distributed in all dimensions around any k^{th} seed, then the proposal standard deviations for each seed in all dimension are normalized, as follows:

$$\tilde{\sigma}^{(k)}_q = \frac{\sigma^{(k)}_q}{\widetilde{\theta}_q} \tag{10}$$

Finally, the standard deviations are recalculated based on the maximum normalized standard deviation among all dimensions. Mathematically:

$$\sigma^{(k)}_q = \widetilde{\theta}_q \cdot \max\left\{\tilde{\sigma}^{(k)}_q\right\}_{q=1}^{d} \tag{11}$$

The application of this automatic scaling method for the proposal standard deviation results in an algorithm that gives a step-wise improvement to the intermediate posterior regions which are broadly sampled at increasingly higher resolutions. This differs from existing versions of the ABC-SubSim algorithm and MCMC algorithms for Subset Simulation [37], where the whole possible region is sampled uniformly resulting in repeated sampling of unnecessary areas, thus increasing the amount of dependent samples and decreasing the acceptance rate. The advantages of using the proposed method for calculating standard deviation are shown in the illustrative example given in Section 5.

4. Petri Net model similarity by ABC-SubSim

This section introduces the method that allows a reduced PN mimics a bigger reference one with quantified uncertainty. The proposed PN reduction method implements the ABC-Subsim inference method explained above to allow the reduced PN, referred to as $\mathcal{P}^{(r)}$, to closely approximate the behavior of a reference one, denoted here as $\mathcal{P}^{(s)}$, so that its signal outputs from a selection of places can be closely replicated over time by those from $\mathcal{P}^{(r)}$.

The method assumes that a reduced PN $\mathcal{P}^{(r)}$, whose behavior wants to be mimicked with respect to $\mathcal{P}^{(s)}$, is available. At this standpoint is important to note that $\mathcal{P}^{(r)}$ must be chosen in a manner that retains the capability to sufficiently reproduce the reliability estimators along with the maintenance activities to be modeled whilst reducing the complexity of the reference one $\mathcal{P}^{(s)}$. Indeed, the reduced PN must have the capacity to reproduce the key outputs of $\mathcal{P}^{(s)}$ and incorporate any behaviors for optimization of the system under consideration, thus it is recommended that the net structure of $\mathcal{P}^{(r)}$ is decided based on the knowledge of the modeler. Irrespective to the previous comments, this work is not focused on the methods for net reduction of PNs and the reader is referred to [49] for rules for optimal simplification of PN models.

Thus, let us assume that $\mathcal{P}^{(r)}$ is available and that its behavior is intended to reproduce the reference one based on a selection of comparison places $\mathbf{P}^{(r)} = \{p^{(r)}_u\}_{u=1}^{n_r}$ for the reduced PN, and $\mathbf{P}^{(s)} = \{p^{(s)}_v\}_{v=1}^{n_s}$ for the reference one, where $\mathbf{P}^{(r)}$ and $\mathbf{P}^{(s)}$ are not necessarily of same size. The basis of comparison for both PN models is their marking at a specific prediction time n restricted to the referred comparison places, namely $\mathbf{M}^{(r)}_n = \left(M^{(r)}_n(1),\dots,M^{(r)}_n(u),\dots,M^{(r)}_n(n_r)\right)$ and $\mathbf{M}^{(s)}_n = \left(M^{(s)}_n(1),\dots,M^{(s)}_n(v),\dots,M^{(s)}_n(n_s)\right)$ for the reduced and reference PNs, respectively.

Next, let us assume that a subset of the $n^{(r)}_t$ transitions of $\mathcal{P}^{(r)}$ are selected such that their delay times are parameterized through a set of uncertain parameter vector $\theta = \{\theta_1,\theta_2,\dots,\theta_d\} \in \boldsymbol{\Theta} \subset \mathbb{R}^d$ whose prior information is known through their component-wise prior PDF $p(\theta_1),\dots,p(\theta_d)$, so that $p(\theta) = p(\theta_1)\cdot p(\theta_2)\cdot\dots\cdot p(\theta_d)$. Therefore, the marking of the reduced PN depends upon the uncertain parameters θ, i.e., $\mathbf{M}^{(r)}_n = \mathbf{M}^{(r)}_n(\theta)$ henceforth the challenge relies on obtaining the set of θ-values which makes $\mathbf{M}^{(r)}_n(\theta) \approx \mathbf{M}^{(s)}_n$, $\forall n$.

Algorithm 1 Petri net similarity algorithm by ABC-SubSim method

```
**Inputs:**
$p_0 \in [0,1]$ $\{P_0 = 0.2$ is recommended$\}$. $N$, {number of samples per intermediate level}; $\epsilon$,
{tolerance}; $n_r$, {number of output places to compare}; $d$, {dimension of the $\theta$-parameter space}.
**Algorithm:**
1:  Sample $\left[\left(\theta_0^{(1)}, x_0^{(1)}\right), \dots, \left(\theta_0^{(n)}, x_0^{(n)}\right), \dots, \left(\theta_0^{(N)}, x_0^{(N)}\right)\right]$, where $(\theta, x) \sim p(\theta)p(x|\theta)$
2:  Set $j = 0$, $\epsilon_j = \infty$
3:  while $\epsilon_j > \epsilon$ do
4:      $j \leftarrow j + 1$
5:      for $n : 1, \dots, N$ do
6:          for $u : 1, \dots, n_r$ do
7:              Evaluate the centroid linkage distance $d(u)$ according to Eq. (12)
8:          end for
9:          Obtain $max\{d(u)\}$ and $min\{d(u)\}$ and normalize the distances according to Eq. (13) to obtain $\rho_j(\theta^{(n)})$
10:     end for
11:     Renumber $\left[\left(\theta_{j-1}^{(n)}, x_{j-1}^{(n)}\right), n : 1, \dots, N\right]$ so that $\rho_j^{(1)} \leqslant \rho_j^{(2)} \leqslant \dots \rho_j^{(N)}$
12:     Fix $\epsilon_j = \frac{1}{2}\left(\rho_j^{(NP_0)} + \rho_j^{(NP_0+1)}\right)$
13:     if $\epsilon_j \leqslant \epsilon$ then
14:         End algorithm
15:     end if
16:     for $k = 1, \dots, NP_0$ do
17:         Select as a seed $\left(\theta_j^{(k),1}, x_j^{(k),1}\right) = \left(\theta_{j-1}^{(k)}, x_{j-1}^{(k)}\right) \sim p\left(\theta, x|(\theta, x) \in \mathcal{S}_j\right)$
18:         (Requires $k > 1$) Calculate the distance $J_{k-k'}$, according to Eq. (7), where $k' = 1, \dots, NP_0, k' \neq k$
19:         Set $\theta_j^{k'}$ as the closest seed by $\min(J_{k-k'})$, excluding repeated seeds
20:         for q= 1,..., $d$ do
21:             Obtain $\sigma_{(q)}^{(k)} = \theta_{(q)}^{(k)} - \theta_{(q)}^{(k')}$
22:             Obtain $\widetilde{\theta}_q$ according to Eq. (8) and normalize $\tilde{\sigma}_q^{(k)} = \sigma_q^{(k)}/\widetilde{\theta}_q$
23:         end for
24:         Set component-wise proposal standard deviation $\sigma_{(q)}^{(k)} = \widetilde{\theta}_q \cdot \max\left\{\tilde{\sigma}_q^{(k)}\right\}, q = 1, \dots, d$
25:         Run MMA [44] with proposal standard deviation $\sigma_{(q)}^{(k)}$ to generate $1/P_0$ states of a Markov chain lying in $\mathcal{S}_j$: $\left[\left(\theta_j^{(k),1}, x_j^{(k),1}\right), \dots, \left(\theta_j^{(k),1/P_0}, x_j^{(k),1/P_0}\right)\right]$
26:     end for
27:     Renumber $\left[(\theta_j^{(k),i}, x_j^{(k),i}) : k = 1, \dots, NP_0; i = 1, \dots, 1/P_0\right]$ as
28:     $\left[(\theta_j^{(1)}, x_j^{(1)}), \dots, (\theta_j^{(N)}, x_j^{(N)})\right]$
29: end while
```

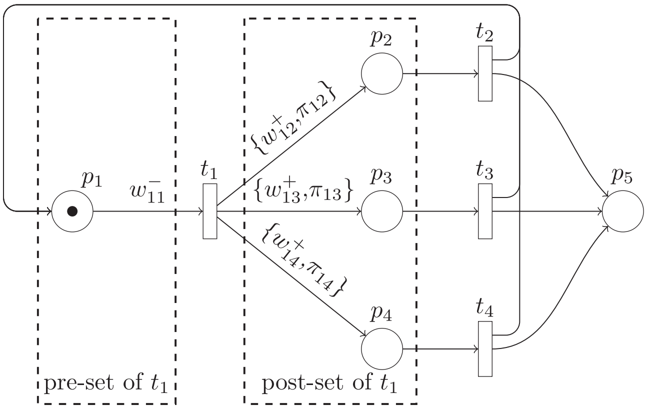


Fig. 1. High Level Petri net used as illustrative example. The figure provides indication of the arc weights and production probabilities of probability transition t_1.

Under the Approximate Bayesian computation perspective, the latter implies obtaining the posterior PDF of the model parameters and their corresponding marking response $(\theta, x) \in \mathcal{S}$ as $p_\epsilon(\theta, x|y)$, where $x = x(\theta) \doteq \mathbf{M}_{0-n}^{(r)}$, the simulated marking of the reduced PN $\mathcal{P}^{(r)}$ from initial time to n, and y refers to the data which, in this case, correspond to the marking of the reference one $\mathcal{P}^{(s)}$ from 0 to n, i.e, $y \doteq \mathbf{M}_{0-n}^{(s)}$. As with any ABC method, the marking similarity $x(\theta) \approx y$ is established under a tolerance value ϵ, along with through a metric $\rho(\cdot)$ and a summary statistic $\eta(\cdot)$.

Thus, the next step to apply the ABC inference method is to define $\rho(\cdot)$ and $\eta(\cdot)$ to properly compare the response markings from $\mathcal{P}^{(r)}$ and $\mathcal{P}^{(s)}$. Several distance measures are available in the literature, and among them, the *centroid linkage* distance [50] was revealed as the most suitable one since it involves quantifying the distance between the mean values of two comparable clusters. To apply this specific metric, the amount of comparison places from $\mathbf{P}^{(r)}$ and $\mathbf{P}^{(s)}$ should be equal, thus the comparison is carried out through the marking of the respective places in a one-by-one basis. Since $n_r = n_s$ in this particular case, the indexes u and v can be fused into a single one (e.g., u), thus $x(u, \theta)$ and $y(u)$ denote the markings up to time n for the reduced and reference PN model, respectively. Let us now split the markings $x(u, \theta)$ and $y(u)$ into a number of overlapping sub-vectors $\widetilde{x}_h(u, \theta)$ and $\widetilde{y}_h(u)$, $h = 1, \dots, H$, each of same length and with an overlapping percentage of 40%. Then, the centroid linkage distance applied to places $p_u^{(r)}$ and $p_u^{(s)}$, referred to here as $d(u) \equiv d(u, u) = d(v, v)$, $u = 1, \dots, n_r$, compares the mean values of the sliding markings under a $\mathbb{L}_2$-norm, mathematically:

$$d(u) = \left(\sum_{h=1}^{H}\left(\bar{\widetilde{x}}_h(u, \theta) - \bar{\widetilde{y}}_h(u)\right)^2\right)^{1/2} \tag{12}$$

where $\bar{\widetilde{x}}(\cdot)$ and $\bar{\widetilde{y}}(\cdot)$ refer to the mean of the h^{th} sliding marking of the reduced and reference PN models, respectively. Note that calculating the metric $d(u)$ from each comparison places $p_u^{(r)}$ and $p_u^{(s)}$, where $u = 1, \dots, n_r$, will result in a vector of metrics whereby an overall metric $\rho = \rho(\theta)$ can be obtained as follows:

$$\rho(\theta) = \sum_{u=1}^{n_r} \frac{d(u) - \min\{d(u)\}_{u=1}^{n_r}}{\parallel \max\{d(u)\}_{u=1}^{n_r} - \min\{d(u)\}_{u=1}^{n_r} \parallel} \tag{13}$$

where ρ only depends upon the model parameters θ through $d(u)$, as can be seen from Eq. (12). Note also that in Eq. (13), the individual distances $d(u)$ are normalized such that their influences for the overall metric ρ vary within $[0, 1]$.

At this standpoint, the ABC-SubSim method can be applied whereby posterior samples of the model parameters θ and their corresponding marking response $x = x(\theta)$, are obtained as output. A pseudo-code implementation of the PN model similarity method by ABC-SubSim has been provided as Algorithm 1. To produce the initial subset (see Step 1 of the Algorithm 1), samples of the model parameters are drawn from the prior PDF $p(\theta)$ by which $\mathcal{P}^{(r)}$ can be simulated via Monte Carlo simulation to obtain $x = x(\theta)$ over a time duration n. The process continues with the production of subsets until the metric falls below a specified threshold ϵ, meaning that ABC-SubSim has reached the posterior region of the model parameters. Once a posterior region in the parameter space has been found, the error associated with the reduced PN can be straightforwardly identified from the ABC-SubSim tolerance value ϵ.

In summary, the proposed method updates a number of uncertain parameters from a proposed reduced PN such that the signal outputs over time of a reference PN can be closely replicated. In this method, care must be taken on which parameters to update within the reduced PN, since it is possible to fit multiple parameters involving high model complexity and over-fitting to the reference PN. On the contrary, using few of them can imply very low accuracy of the marking from the reduced model with respect to the reference one, thus obtaining meaningless results from a reliability engineering point of view. Through the evidence calculation presented in Eq. (6), a trade-off between complexity and accuracy can be established, not only at the level of possible parameterizations of a reduced PN model, but also at the level of different potential reduced PN schemes with respect to a reference one. This will be explored in the example presented in Section 6.

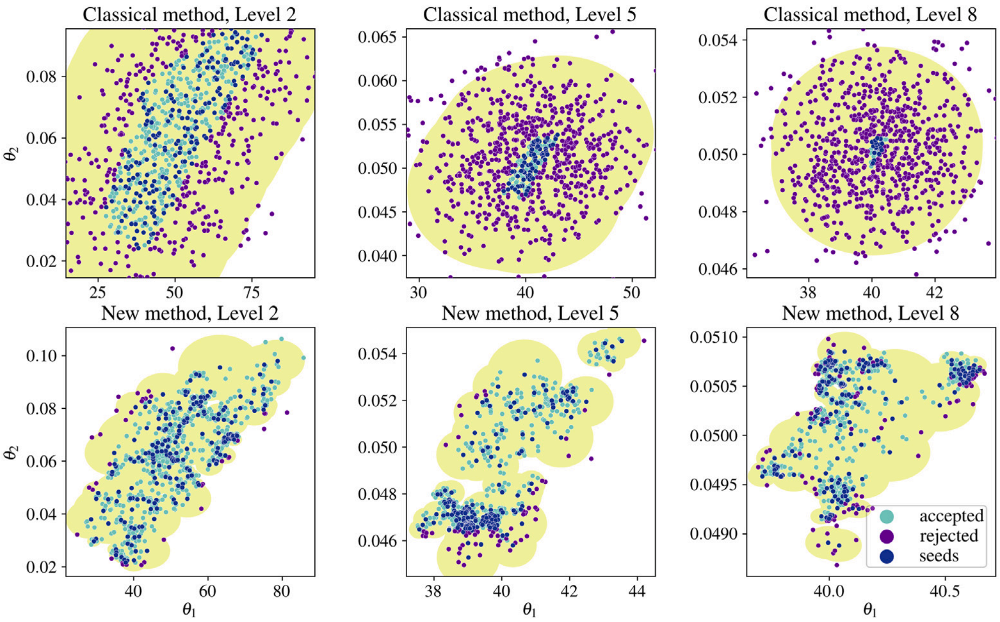


Fig. 2. Parameter updating of the θ_1 and θ_2 parameters from the illustrative Petri net example of Section 5. Top panels show the results when using the classical proposal standard deviation method, whereas the bottom ones present the results using the proposed adaptive method for scaling the standard deviation. The shaded areas drawn using yellow color indicate the proposal's areas.

5. Illustrative example

This section gives an example to illustrate the use of the proposed method on a simple PN architecture. The example also illustrates the advantages of using the proposed adaptive sampling method for the ABC-SubSim algorithm.

To this end, the PN structure shown in Fig. 1 is considered again. The reader is referred to Section 2.1 for a description of the PN structure. Here, all arcs' weights are set to 1, and the production probabilities from t_1 to $\{p_2, p_3, p_4\}$ are chosen so that $\{\pi_{12} = 0.9999, \pi_{13} = 0.00009, \pi_{14} = 0.00001\}$, respectively. The delay times of transitions t_1, t_2, t_3 and t_4 are chosen as $\{\tau_1 = 40, \tau_2 = 0.05, \tau_3 = 0.1, \tau_4 = 10\}$, respectively, which are used to obtain synthetic reference data. Then, $\{\tau_1, \tau_2\}$ are assumed to be unknown and updatable parameters using the proposed method. Their prior distributions are assumed as uniforms, as follows: $\theta_1 \sim \mathcal{U}[10, 100]$, $\theta_2 \sim \mathcal{U}[0.01, 1]$. In this example, the delay time of t_1 is chosen to be much greater than that of t_2 in the reference case, to show the ability of the model in solving critical cases where the algorithm has to infer parameters whose components belong to very different ranges of values.

In this example, the places p_1, p_2 and p_5 were considered as comparison places thus their markings $\mathbf{M}_n^{(s)}$ and $\mathbf{M}_n^{(r)}(\theta)$ for the reference and reduced PN model, respectively, were used to compute the ABC-SubSim metric. The centroid linkage metric with one sliding region for each signal was used for places p_1 and p_2. However, since that p_5 is a counter place, only its last marking was considered when calculating its metric. The original ABC-SubSim algorithm and the one with adaptive sampling presented in Section 2.3, were used to comparatively examine their performance. Each ABC-SubSim run employed $N = 1000$ samples per simulation level and a maximum of eight simulation levels.

Fig. 2 shows plots of posterior samples of the θ_1, θ_2 parameters produced at the second, fifth, and eighth simulation level of the ABC-SubSim algorithm. The plots also depict the seeds with their accepted and rejected proposals, along with their proposals' areas. The proposal area for each seed is represented by a yellow color shade ellipse whose dimension-wise semi-axis is twice the standard deviation in that dimension, in order for the ellipse to contain 95th percentile of the proposals. The results show that the posterior samples lie around $\theta_1 = 40$ and $\theta_2 = 0.05$ for both, the classical proposal standard deviation method and the proposed method to adaptive select the aforementioned algorithm hyper-parameter. Top panels (obtained using the classical standard deviation method) reveal that the amount of rejected samples highly increases as the algorithm progresses towards deeper simulation levels, which is a well-known drawback of the original ABC-SubSim algorithm. These rejected samples, which are drawn as repeating samples, conform the majority of samples within the final posterior PDF, thus significantly decreasing the acceptance rate and the statistical information of the posterior. However, posterior samples obtained using the proposed adaptive method for the ABC-SubSim are drawn around the seeds' area leading to an increased acceptance rate of new samples even when the range of values is very small, as shown in the 6th simulation level from Fig. 2 (bottom-right panel). As overall consequence, higher resolution for the ABC-SubSim inference is achieved by adopting lower tolerance values ϵ, since the statistical noise of the sampling method of the algorithm is reduced at a minimum.

6. Case study

The previous section illustrated a case where a SPN model with two uncertain parameters was compared by itself using synthetic data obtained by taking the parameters as known. In this section, the proposed Bayesian methodology is applied to infer the model parameters of a set of reduced PN models based on the response of a larger reference PN model of an engineering system.

6.1. Description of the engineering system and its maintenance model

The engineering system to be modeled corresponds to a two-pumps system to support the pressure of a lube oil supply facility which operates in a continuous regimen, as depicted in Fig. 3. The system contains a main pump, that is usually in operation except when a failure occurs whereupon a drop in pressure would be observed since the supply flow would be less than the demand flow. In this case, an auxiliary pump will then come into play to recover the pressure of the system. Further details on this system along with their physical data can be found in [51, Chapter 7].

In this case study, an insufficient system pressure is classified as a failed system state, which can occur only if both pumps fail. The

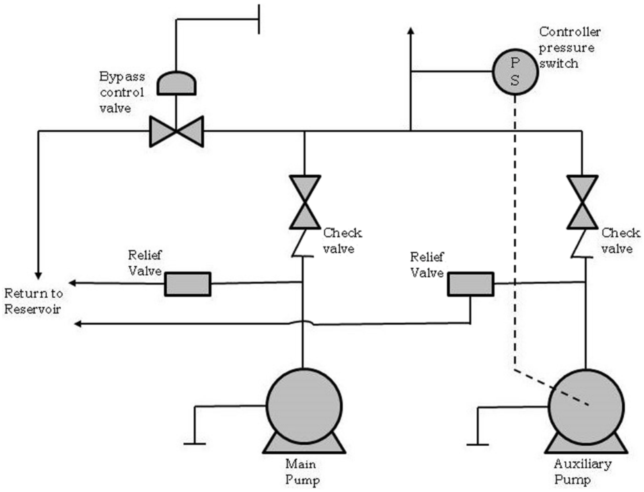


Fig. 3. Engineering diagram of the two pumps system considered as case study. The diagram shows a duty/standby arrangement of the pumps with indication of the auxiliary components of the system, like valves, starters, pressure sensor controllers, and connections.

failure of a pump occurs when there is insufficient outflow from the pump to maintain the oil pressure, and it is revealed through a pressure sensor over the main pump. In this example, the back-up pump is assumed to have unrevealed failures, since it is usually not in operation. Both pumps are assumed to be inspected on a periodic basis whose results can reveal either a failure or a degraded condition in any or both pumps. Here, degradation can happen at a different rate in each pump and is defined as a state where the pump has sufficient capacity to support the system, but maintenance is required. Mathematically, the pump degradation is assumed to be modeled as a two parameter Weibull PDF representing the time to failure, namely $\mathcal{W}(12,1)$ for the main pump, and $\mathcal{W}(30,2)$ for the auxiliary one.

After inspection, a maintenance action is activated if both pumps are in the failed state or, one of them is in the failed state whilst the other is in a degraded state. Another option is that any of the pumps was in a degraded state, then an opportunistic maintenance action is activated under a probability, which in this case study equals 0.2. After maintenance, each pump is returned to the *working* state, thus recovering the overall system to normal operation. The following assumptions are also adopted for modeling the maintenance of the system under study:

(a) The rate of degradation of the back-up pump does not change when the main pump fails;
(b) The pressure sensor associated with the main pump will always immediately reveal a failed state of the main pump;
(c) The components in the remaining parts of the system are always in the working state;
(d) Inspection is always successful in discovering a degraded or failed state;
(e) If a degraded state is identified but not rectified in time, then a failure can occur;
(f) If there is a failed state in either of the pumps, then maintenance occurs as soon as a maintenance team is available.

6.2. Petri net maintenance model acting as reference

In this section, a HLPN is proposed to model the system behavior along with its inspection and maintenance actions, as previously described in Section 6.1. The HLPN model is depicted in Fig. 4, and it will be used as the reference net for the reduction by ABC-SubSim numerical experiments that will be shown further below.

In this HLPN, places p_1 and p_2 model the overall working and failed states of the system, respectively. Since the pumps are assumed to operate in parallel, thus both must be in the failed state for a system failure to occur. In such a case, transition t_1 immediately marks place p_2. Then, if any of the pumps returns to the working state, the marking of place p_2 is removed and place p_1 is marked again by either transition t_2 or t_3.

For the back-up pump, places p_3, p_4, and p_5 represent the *working*, *degraded*, and *failed* states, respectively. When place p_9 is marked then an inspection action is enabled which lasts until place p_8 is marked. If degradation is discovered, then place p_7 is marked and maintenance is requested. Analogously, when place p_6 is marked then there is a discovered failure of the pump and, as well, maintenance is required.

The main pump states are represented on the right hand side of the graph. In this part of the model, places p_{10}, p_{11} and p_{12} corresponds to the *good*, *degraded*, and *fail* states, respectively. Similarly to the architecture described for the back-up pump, a token in place p_{15} indicates that inspection is enabled for the main pump, which is underway until place p_{14} is marked. If a degradation is discovered, then place p_{13} is marked whereupon maintenance is requested.

The remaining nodes of the model govern the maintenance scheduling for the pumps. Specifically, place p_{18} and transition t_{25} model the availability of maintenance resources which produce a token in p_{19} indicating that maintenance is possible. When place p_{17} is marked then maintenance is requested for a pump in the system due to an identified failure in one of the pumps. A token in p_{22} indicates that maintenance is available to any of the pumps to improve a degraded state, whereas, in an analogous way, a token in p_{23} then indicates maintenance to repair a failure state of any pump. If neither of the pumps are in the failed state, then the available maintenance resources can be assigned to repair a degraded state, which is modeled by the marking of place p_{21} after enabling of transitions t_{26} and t_{27}. In this model, inspections are assumed to be carried out every 6 weeks with a maintenance interval of 3 weeks, taken both terms in average. The rest of parameters are described in Table 5.

Initially, the system starts in the working state represented by one token in places p_1, p_3, p_{10}, p_9, p_{15} and p_{18}. The Monte Carlo simulation has been used to simulate the response of the reference PN model described above using $N = 4000$ samples for every stochastic transitions of the model. The results, which have been produced in week units, are shown using dark line in Fig. 6 for places p_1 and p_2, and also in Fig. 9 for places p_5 and p_{12}.

6.3. Reduced Petri net maintenance models

This section presents three different reduced SPN models, namely $\mathcal{P}_1^{(r)}, \mathcal{P}_2^{(r)}$, and $\mathcal{P}_3^{(r)}$, to approximate the reference one, $\mathcal{P}^{(s)}$, defined in the previous section. The reduced models are parameterized with a set of uncertain parameters that will be described in the following subsections, which together with their prior information, conform the model classes $\mathcal{M}_1, \mathcal{M}_2$, and $\mathcal{M}_3$. The uncertain parameters are updated using the proposed Bayesian methodology using the marking of the places p_1, p_2, p_5, and p_{12}, from $\mathcal{P}^{(s)}$, as data. Algorithm 1 is applied using $P_0 = 0.2$, $N = 4000$ samples per simulation level, and a tolerance $\epsilon = 0.035$. Gaussian densities are used as proposal PDFs for the MCMC sampling of Algorithm 1 using adaptive scaling of their standard deviations, as proposed in Section 3.

6.3.1. Reduced model 1

Fig. 1 shows the first reduced model $\mathcal{P}_1^{(r)}$ which is based on a simple net structure of two transitions, t_1 and t_2, and two places, p_1 and p_2. This reduced SPN extracts the overall behavior of both pumps in the system along with their maintenance and inspection strategies to solely provide an estimate of the system states, namely working or failed state, represented by places p_1 and p_2, respectively. The transition t_1 governs the time to total system failure whereas t_2 stands for the

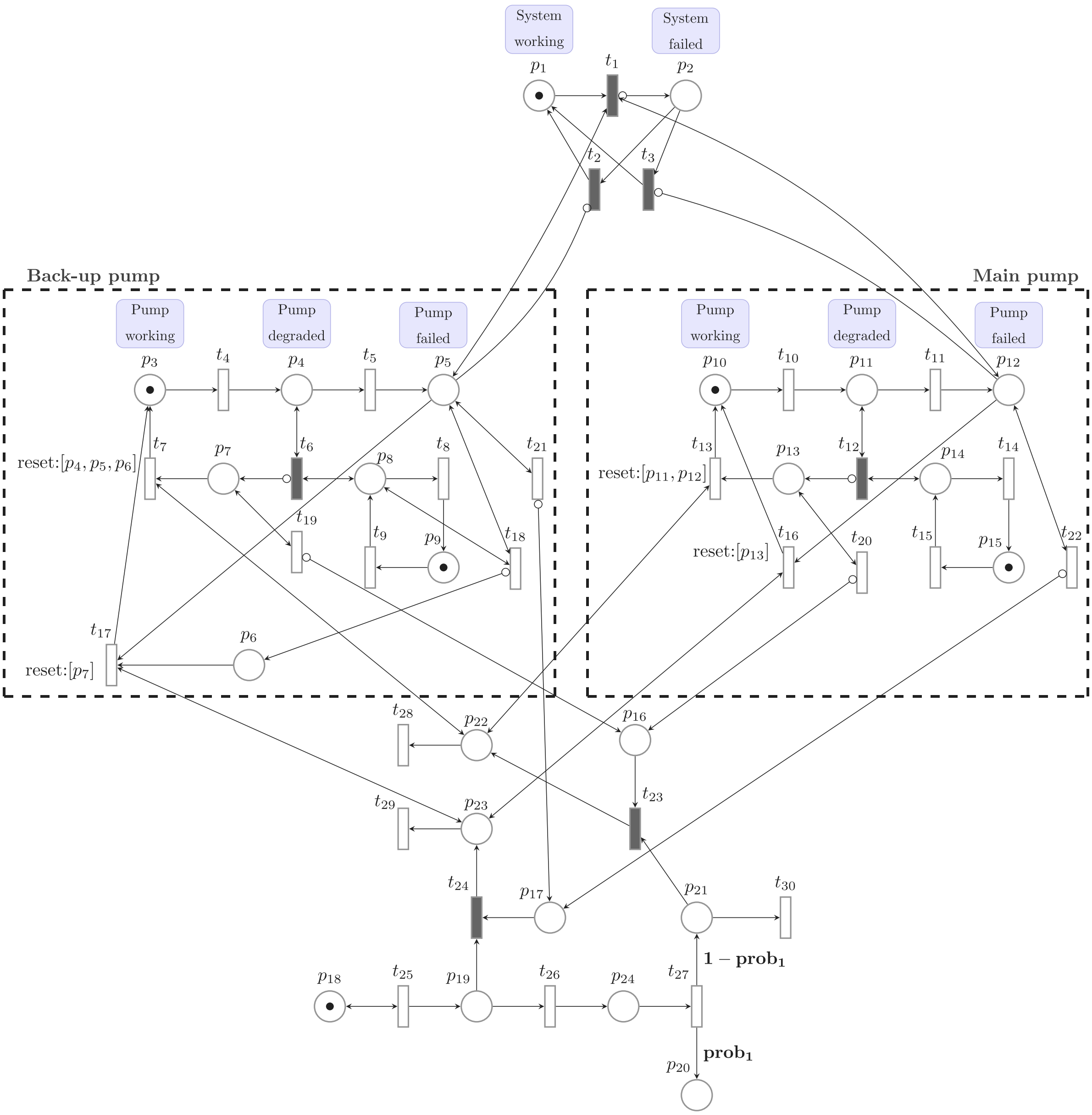


Fig. 4. Reference HLPN model of the case study presented in Section 6. The dashed rectangles indicate nodes representing the state of the main and back-up pumps. The remaining places and transitions model maintenance scheduling on the system level. The dark small rectangles indicate instant transitions whereas the blue text labels provide some explanatory information about key places.

Table 2
Description of the transitions from the PN model shown in Fig. 1.

Transition	Type	Distribution	Parameters	Description of action
t_1	Stochastic	Normal	$\theta_1, 20$	Initiates system failure
t_2	Stochastic	Normal	$\theta_2, 1$	Initiates system repair

time to repair of the system. Both transition times, namely τ_1 and τ_2 (expressed in week units), are modeled as stochastic transitions assumed as Gaussians, i.e., $\tau_1 \sim \mathcal{N}(\theta_1, 20)$ and $\tau_2 \sim \mathcal{N}(\theta_2, 20)$, where θ_1, θ_2 are considered as the uncertain parameters of this reduced model (Table 2 provides an overview about the transitions). The prior PDFs of the model parameters are modeled as uniform PDFs within the $[0, 200]$ and $[0, 18]$ intervals, namely $p(\theta_1|\mathcal{M}_1) = \mathcal{U}(0, 200)$ and $p(\theta_2|\mathcal{M}_1) = \mathcal{U}(0, 18)$, respectively. The metric values are obtained by comparing the marking of places p_1 and p_2 of $\mathcal{P}^{(s)}$ with respect to places p_1 and p_2 in the reduced one.

Fig. 5 shows the resulting posterior PDF of the uncertain parameters θ_1, θ_2 after the application of algorithm 1. The values within the posterior region that represent the *Maximum a posteriori* (MAP) were selected as point-valued estimates of θ that make the reduced model better match the behavior of the reference one. These values are $\theta_{1,MAP} = 36.9$, and $\theta_{2,MAP} = 1.0$ in weeks units, for transitions t_1 and t_2, respectively. The MAP values were then used to comparatively reproduce the marking sequence of the output places of $\mathcal{P}_1^{(r)}$ during a 500 weeks time. Additionally, the 10th and 90th percentiles of the results were calculated after simulating the reduced model using all the θ-samples from the posterior region given in Fig. 5. The results are shown in Fig. 6 for reference places p_1 (left panel) and p_2 (right one). These results demonstrate that the inferred reduced model $\mathcal{P}_1^{(r)}$ evaluated using θ_{MAP} is able to capture the throughput of tokens passing through the comparison places p_1 and p_2 with good accuracy. Note also from Fig. 6 that the overall response of $\mathcal{P}_1^{(r)}$ when using all possible model parameters values taken from the inferred posterior PDF, under-predicts the reference model response since there are some

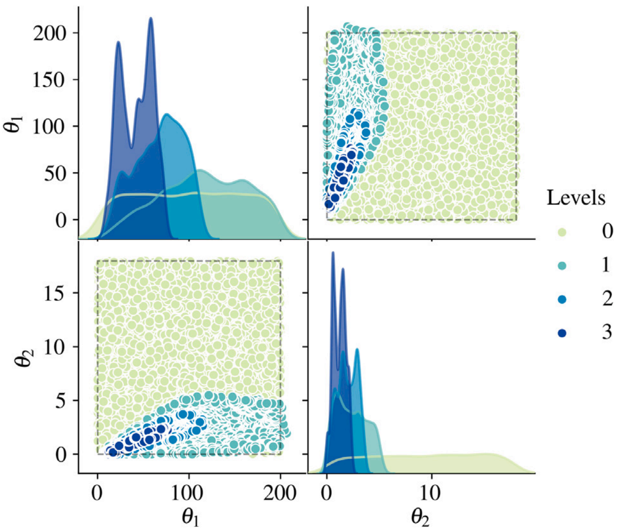


Fig. 5. Scatter plot representation of posterior samples in the θ-space as ABC-SubSim output for the first reduced SPN model. On the diagonal, kernel density estimates are shown for the marginal posterior PDF of θ_1 and θ_2 at different simulation levels, differentiated using a color scale indicated at the right side of the plot.

Table 3
Description of the transitions from the SPN model shown in Fig. 7.

Transition	Type	Distribution	Parameters	Description of action
t_1	Symbolic	None	–	Fires instantaneously
t_2	Symbolic	None	–	Fires instantaneously
t_3	Symbolic	None	–	Fires instantaneously
t_4	Stochastic	Gaussian	$\theta_1, 2$	Initiates failure (back-up pump)
t_5	Stochastic	Gaussian	$\theta_2, 1$	Initiates repair (back-up pump)
t_6	Stochastic	Gaussian	$\theta_3, 1$	Initiates failure (main pump)
t_7	Stochastic	Gaussian	$\theta_4, 1$	Initiates repair (main pump)

outliers which are not well covered within the 10th and 90th probability band. The latter can be viewed as a consequence of the extremely reduced net structure of $\mathcal{P}_1^{(r)}$ with respect to the reference one, since this model summarizes the behavior of the entire system into a single cycle (failure–repair) of transitions for both pumps.

6.3.2. Reduced model 2

The second reduced model slightly increases the complexity of the previous one by considering simple sub-nets for each of the pumps in the system, specifically $\{p_3, p_4, t_4, t_5\}$ for the back-up pump, and $\{p_5, p_6, t_6, t_7\}$ for the main one, as depicted in Fig. 7. These sub-nets are complemented with three transitions $\{t_1, t_2, t_3\}$ that govern the overall system states, namely *working* and *failed* states, represented by places p_1 and p_2, respectively. Similarly to the reference model, the pumps are operating in parallel such that both must be in the failed state for a failure to occur.

In this reduced structure, each of the pump's sub-net only consists of the working and failed states, corresponding to p_3 and p_4 for the back-up pump, and to p_5 and p_6 for the main one, respectively. Transitions t_4, t_5 and t_6, t_7 govern the times to failure and times to repair of the respective pumps, which are assumed to be modeled as Gaussian PDFs as follows: $\tau_4 \sim \mathcal{N}(\theta_1, 2); \tau_5 \sim \mathcal{N}(\theta_2, 1); \tau_6 \sim \mathcal{N}(\theta_3, 1); \tau_7 \sim \mathcal{N}(\theta_4, 1)$, where $\theta = \{\theta_1, \dots, \theta_4\}$ are considered as the uncertain model parameters of this model. The component-wise prior PDFs of these uncertain parameters are known and considered as uniforms, given by: $\theta_1 \sim \mathcal{U}[0, 100]$, $\theta_2 \sim \mathcal{U}[0, 20]$, $\theta_3 \sim \mathcal{U}[0, 80]$ and $\theta_4 \sim \mathcal{U}[0, 20]$, respectively, where these values have been considered sufficiently large to allow the method to obtain the posterior values within them. Table 3 gives further information about the referred transitions. In this model the marking sequences of places $\mathbf{P}^{(s)} = \{p_1, p_2, p_5, p_{12}\}$ in the reference model were compared with the sequences of places $\mathbf{P}_2^{(r)} = \{p_1, p_2, p_4, p_6\}$ in the reduced model respectively, since they hold the same interpretation. The data for the Bayesian inference are gathered here as the vector $y = \left[M(1)_{0-500}^{(s)}, M(2)_{0-500}^{(s)}, M(5)_{0-500}^{(s)}, M(12)_{0-500}^{(s)}\right]$, where $M(\cdot)_{0-500}^{(s)}$ refers to the marking of a specific place of the reference PN for a 500 weeks-period.

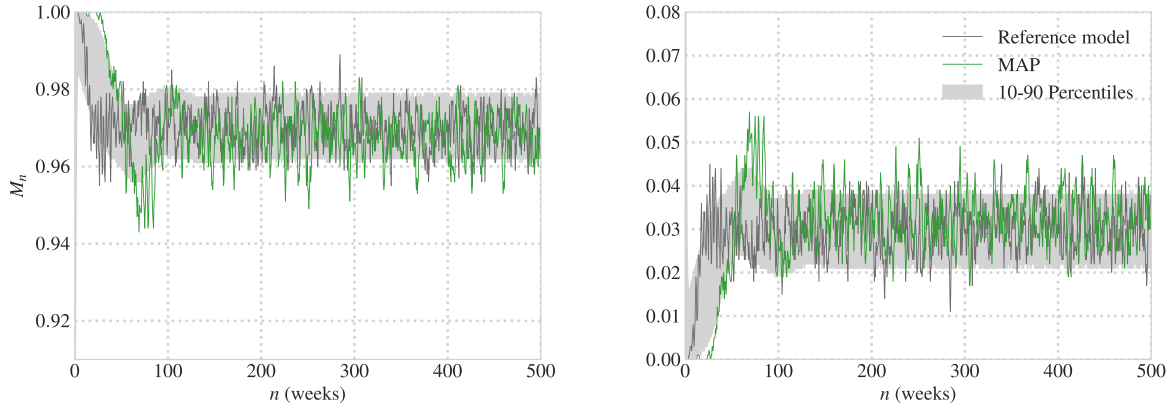
Algorithm 1 was used to update the uncertain parameters θ so as to make $\mathcal{P}_2^{(r)}(\theta)$ working similar to $\mathcal{P}^{(s)}$. Fig. 8 shows the output of the Bayesian inference of θ with respect to the data y taken from $\mathcal{P}^{(s)}$. These results disclose MAP values for θ given by $\theta_{1,MAP} = 11.96, \theta_{2,MAP} = 6.07, \theta_{3,MAP} = 4.38$ and $\theta_{4,MAP} = 0.002$ weeks, respectively. As with the previous section, the reduced model was simulated using the MAP values to comparatively obtain the marking sequences of the output places from $\mathcal{P}_2^{(r)}$ with respect to those from $\mathcal{P}^{(s)}$, and the results are shown in Fig. 9. This figure also represent the 90th and 10th percentile for the average markings sequences obtained through the use of all posterior samples from Fig. 8. The results show that the simulations from the reduced model can capture the response from the reference one when performing a posterior statistical averaging from the posterior samples. However, when observing the model response from the θ_{MAP} (in green), there are periodic peaks in the marking sequence of the reduced model that cannot be seen in that of the reference one (in gray). This reflects that this reduced model has limited capacity to reliably reproduce the behavior of the reference model, perhaps due to lack of extra nodes to better capture the system behavior including not only failure and recovering, but also maintenance actions. The latter will be tested in the following section.

6.3.3. Reduced model 3

The third reduced model $\mathcal{P}_3^{(r)}$ considered in this study is based on an extension of the reduced model $\mathcal{P}_2^{(r)}$ presented before by adding three extra nodes to the sub-nets on the lower left hand side and the lower right hand side of the graph, as depicted in Fig. 10. The new nodes correspond to $\{t_5, p_5, t_7\}$ for the sub-net of the back-up pump and to $\{t_9, p_8, t_{11}\}$ for the one of main pump, and allow each pump operation being modeled through three states: working, degraded, and the failed state, respectively. As with the previous reduced models presented above, the pumps operate in parallel such that both must be in the failed state for a system failure to occur. The system failure is manifested through a token in p_2, otherwise the system is in the working state, represented by one token in p_1.

In this reduced model, the times to degradation and failure are governed by the transitions t_4 and t_5 for the back-up pump, and by transitions t_8 and t_9 for the main one. Transitions t_6, t_7 and t_{10}, t_{11}, model the respective maintenance after a degraded and failed state are revealed in the back-up and main pump, respectively. The time delays associated to the aforementioned maintenance transitions are assumed to be modeled as Gaussian PDFs as follows: $\tau_6 \sim \mathcal{N}(\theta_1, 1); \tau_7 \sim \mathcal{N}(\theta_2, 1); \tau_{10} \sim \mathcal{N}(\theta_3, 1); \tau_{11} \sim \mathcal{N}(\theta_4, 1)$, being $\theta = \{\theta_1, \dots, \theta_4\}$ the uncertain model parameters. Similarly to the previous reduced model, the component-wise prior PDFs of these uncertain parameters are considered as uniforms, which in this case are given by: $\theta_1 \sim \mathcal{U}[0, 20]$, $\theta_2 \sim \mathcal{U}[0, 40]$, $\theta_3 \sim \mathcal{U}[0, 40]$ and $\theta_4 \sim \mathcal{U}[0, 4]$, respectively. Note that the prior interval for θ_4 is significantly smaller than the rest, and the reason is due to shorter repair times expected for the main pump under failure.

Under these settings, Algorithm 1 has been applied to obtain the posterior PDF of θ that makes the response of model $\mathcal{P}_3^{(r)}$ similar to the that from the reference one, $\mathcal{P}^{(s)}$. The model response is evaluated through the marking sequences of places $\mathbf{P}_3^{(r)} = \{p_1, p_2, p_5, p_8\}$ in comparison to the marking from places $\mathbf{P}^{(s)} = \{p_1, p_2, p_5, p_{12}\}$, in the reference model. Likewise the previous cases, the data for the Bayesian inference are given by the marking of the comparison places $\mathbf{P}^{(s)}$ over 500 weeks-period simulation, i.e, $y = \left[M(1)_{0-500}^{(s)}, M(2)_{0-500}^{(s)}, M(5)_{0-500}^{(s)}, M(12)_{0-500}^{(s)}\right]$. The ABC-SubSim algorithm led to the posterior inference



(a) Average marking from p_1 (system reliability) (b) Average marking from p_2 (probability of system failure)

Fig. 6. Outputs of the evaluation of reference Petri net shown in Fig. 4 (in dark color) and the first reduced Petri net model $\mathcal{P}_1^{(r)}$, represented using green line. The results are shown for places p_1 and p_2 from both PN models.

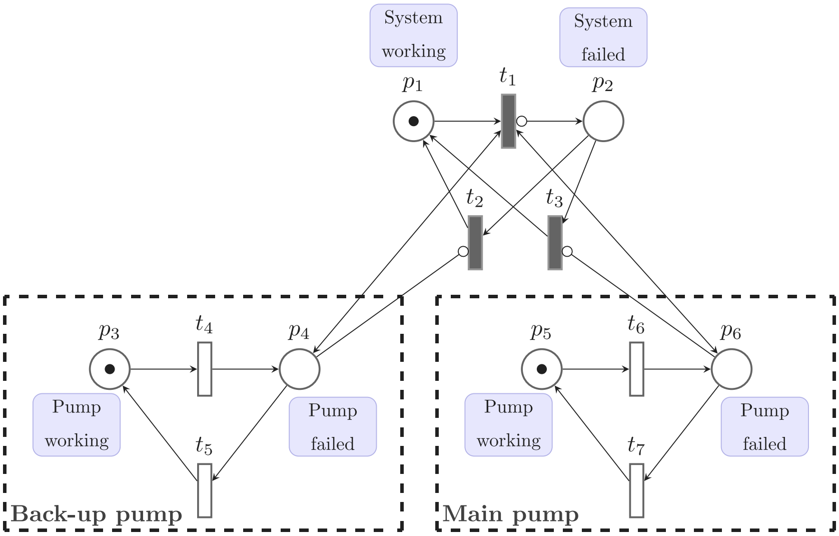


Fig. 7. Second reduced SPN model of the case study presented in Section 6. The dashed rectangles indicate nodes representing the state of the main and back-up pumps. The dark small rectangles indicate instant transitions whereas the blue text labels provide some explanatory information about key places.

results shown in Fig. 11. Note from these results that the posterior PDF of parameters depicts a curved region around θ_1, θ_2, and θ_3, θ_4. The latter means that these parameters (two-by-two) are highly correlated under an inversely decreasing correspondence, which is a consequence of the maintenance structure applied to each of the pumps, such that the higher the maintenance time applied during the degraded state of a pump (before the failure occurs), the lower the maintenance is required for recovering failures. Note also that there is no clear correlations between any of the parameters θ_1, θ_2, with respect to any from the duo θ_3, θ_4, which reflects the assumption that failure or degradation in any of the pumps can appear and progress independently in each of them.

Finally, the MAP estimates from the posterior PDF, namely $\theta_{MAP} = \{8.99, 14.15, 9.57, 1.66\}$ expressed in weeks units, were used to comparatively obtain the response of the reduced model with respect to the reference one, as shown in Fig. 12, with a shade between the 10th and 90th percentile of the results in the background. The results show that the reduced model follows similar average values and has a similar level of noise to the output from the reference model. Indeed, the average markings reveal closer matching than their corresponding outputs from the two previous reduced models (see Fig. 6, Fig. 9).

7. Discussion

This section provides discussion on the results obtained above along with on the limitations of the methodology.

7.1. Discussion on the results

The results obtained in the previous section have shown that the proposed method, via ABC-SubSim algorithm, allows us obtaining reduced PN models whose behavior are approximately equal to the one of a bigger reference model. Further insights about the suitability of the three reduced PN models $\mathcal{P}_1^{(r)}, \mathcal{P}_2^{(r)}$, and $\mathcal{P}_3^{(r)}$, are given in Table 4 under a comparative perspective. This table also summarizes some technical information about the reduced models and their updating procedure.

One of the main performance indicators of the reduced models is given through the fitting error with respect to the reference one. This can be obtained as the lowest possible distance value $\rho = \rho(\theta)$ (recall Eq. (13)) obtained using the posterior PDF of θ. A lower fitting error indicates a better approximation to the reference model, and vice-versa. In this case study, the reduced models $\mathcal{P}_1^{(r)}$ to $\mathcal{P}_3^{(r)}$ have been inferred using the ABC approach under the same tolerance value, namely $\epsilon = 0.035$, hence their fitting errors with respect to the marking y from the reference model, will be approximately the same.

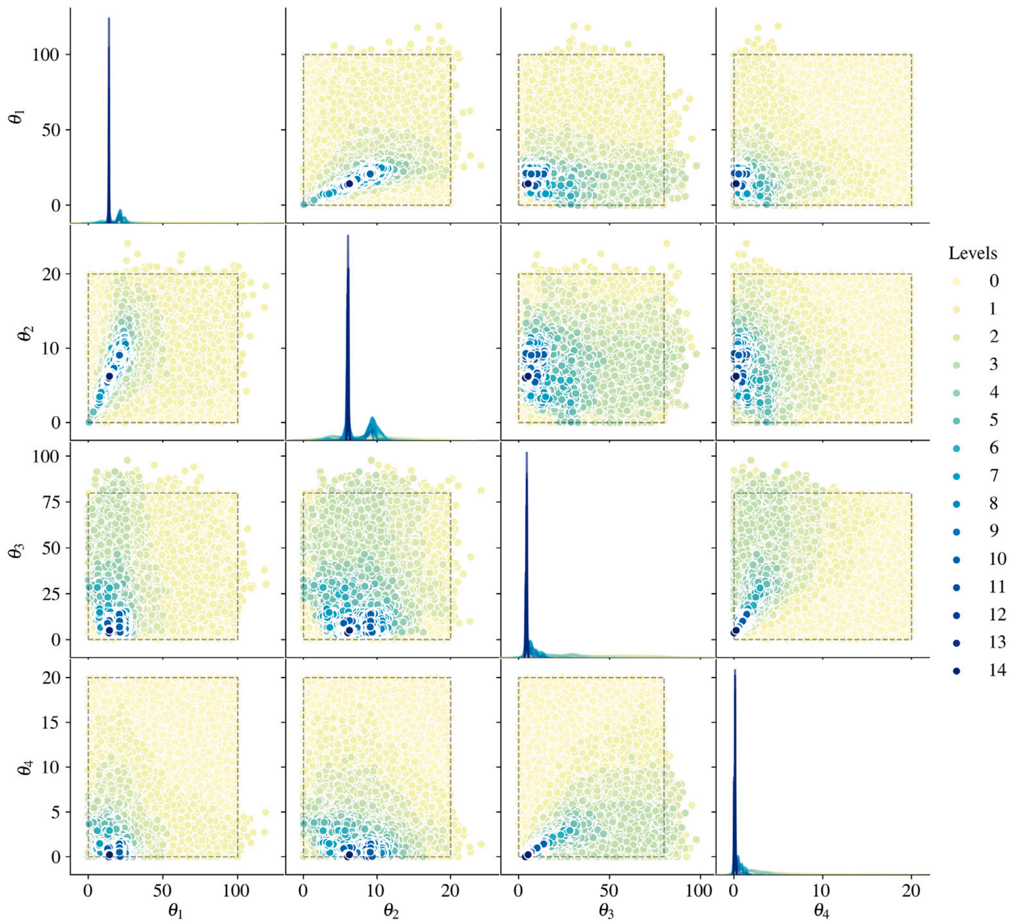


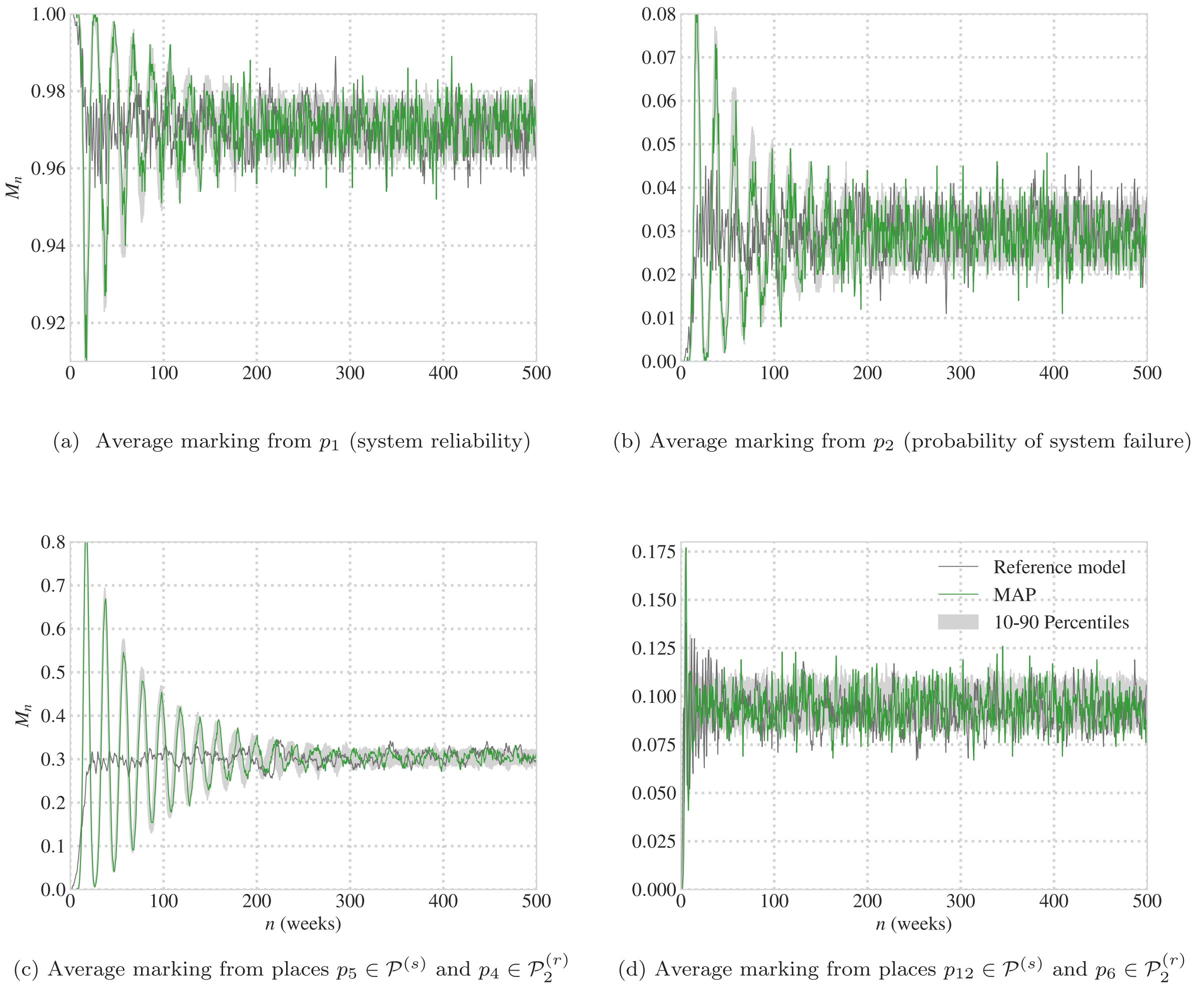
Fig. 8. Scatter plot representation of posterior samples in the θ-space as ABC-SubSim output for the second reduced Petri net model. On the diagonal, kernel density estimates are shown for the marginal posterior PDF of $\theta_1, \ldots, \theta_4$ at different simulation levels, differentiated using a color scale indicated at the right side of the plot.

Table 4
Summary of the comparative results obtained in Section 7.1 for the three reduced models considered in the case study presented in Section 6.3. The computational costs values presented in the fifth, sixth, fourteenth, and fifteenth rows are expressed in seconds.

	$\mathcal{P}_1^{(r)}$	$\mathcal{P}_2^{(r)}$	$\mathcal{P}_3^{(r)}$	$\mathcal{P}^{(s)}$
# of transitions/(timed transitions)	2/(2)	7/(4)	11/(8)	30/(22)
# of uncertain parameters	2	4	4	–
# of comparison places	2	4	4	–
Class evidence	0.2^3	0.2^{14}	0.2^4	–
Computational cost for inference	18,494	221,519	278,418	–
Computational cost for simulation	1.19	25.90	16.85	270.30
$P_{f,SYS}^{(ROB)}$	0.0290	0.0290	0.0279	0.0289
$P_{f,SYS}^{(MAP)}$	0.030	0.0283	0.0274	0.0289
$P_{f,SYS}^{*}$	0.0286	0.0286	0.0281	0.0289
# of uncertain parameters	2	4	4	–
# of comparison places	2	2	2	–
Class evidence	0.2^5	0.2^5	0.2^2	–
Computational cost for inference	22,804	160,616	121,343	–
Computational cost for simulation	1.28	15.40	14.72	270.30
$P_{f,SYS}^{(ROB)}$	0.0288	0.0288	0.0288	0.0289
$P_{f,SYS}^{(MAP)}$	0.0297	0.0279	0.0301	0.0289
$P_{f,SYS}^{*}$	0.0283	0.0281	0.0290	0.0289

Thus, a more insightful manner to compare the performance of the three candidates is through Bayesian model class selection [38], and particularly through their evidence calculation $P_\epsilon(y|\mathcal{M}_j)$ (recall Eq. (6)), assuming that the model candidates have the same prior probability. In this case, the class candidates $\mathcal{M}_j, j = 1,2,3$ correspond to $\mathcal{P}_1^{(r)}, \mathcal{P}_2^{(r)}$, and $\mathcal{P}_3^{(r)}$, respectively. The results shown in Table 4, indicate that $\mathcal{P}_1^{(r)}$ has higher evidence than the rest of competing candidates. From a practical point of view, the latter implies that if a sample response evaluation is requested from each of the reduced models, then $\mathcal{P}_1^{(r)}$ would produce, with higher probability, a marking sequence approximately equal to data y, taken as reference. However, the latter result should be discussed in line with the complexity of data taken as reference for each of the candidate model classes. Note that $\mathcal{P}_1^{(r)}$ is inferred taking into consideration the marking of only two places (p_1 and p_2), in contrast to the four places taken as reference for $\mathcal{P}_2^{(r)}$ and $\mathcal{P}_3^{(r)}$. Since the evidence computation is benefited by simpler models [38], then a new evidence calculation is presented in Table 4 under consideration of only two reference places (p_1 and p_2) for all the model candidates. The results, which are presented below the horizontal dashed line of the table and under a tolerance value of $\epsilon = 0.015$, indicate that $\mathcal{P}_3^{(r)}$ is the preferred model among the candidate ones. This is because it showed the highest evidence among the three models when considering the same number of places. Also, $\mathcal{P}_3^{(r)}$ is more explanatory since it is more descriptive to the real case example while having a low simulation-computational cost.

The computational cost to obtain the inference of the reduced models is also considered here for discussion. The results show that in general, the cost of inferring a reduced PN model increases in a non-linear manner with increasing model size. These results, which have been obtained using an Intel® $i7$ 3.2 GHz CPU, suggest that inferring a reduced model with respect to a reference one can be computationally expensive if the reduced model has some complexity in terms of amount of PN nodes. Notwithstanding, it is worth mentioning



(a) Average marking from p_1 (system reliability)

(b) Average marking from p_2 (probability of system failure)

(c) Average marking from places $p_5 \in \mathcal{P}^{(s)}$ and $p_4 \in \mathcal{P}_2^{(r)}$

(d) Average marking from places $p_{12} \in \mathcal{P}^{(s)}$ and $p_6 \in \mathcal{P}_2^{(r)}$

Fig. 9. Outputs of the evaluation of reference Petri net $\mathcal{P}^{(s)}$ shown in Fig. 4 (in dark color), and the second reduced Petri net model $\mathcal{P}_2^{(r)}$, represented using green line. The results are shown for places p_1, p_2, p_5, p_{12} from the reference model, and p_1, p_2, p_4, p_6, from the reduced one.

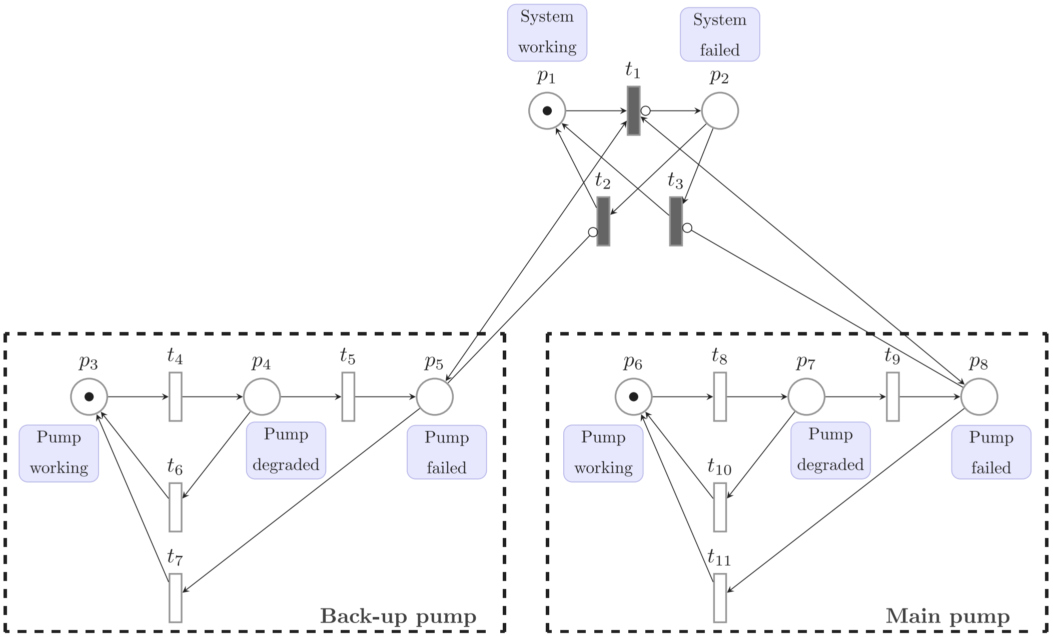


Fig. 10. Third reduced SPN model of the case study presented in Section 6. The dashed rectangles indicate nodes representing the state of the main and back-up pumps. The dark small rectangles indicate instant transitions whereas the blue text labels provide some explanatory information about key places.

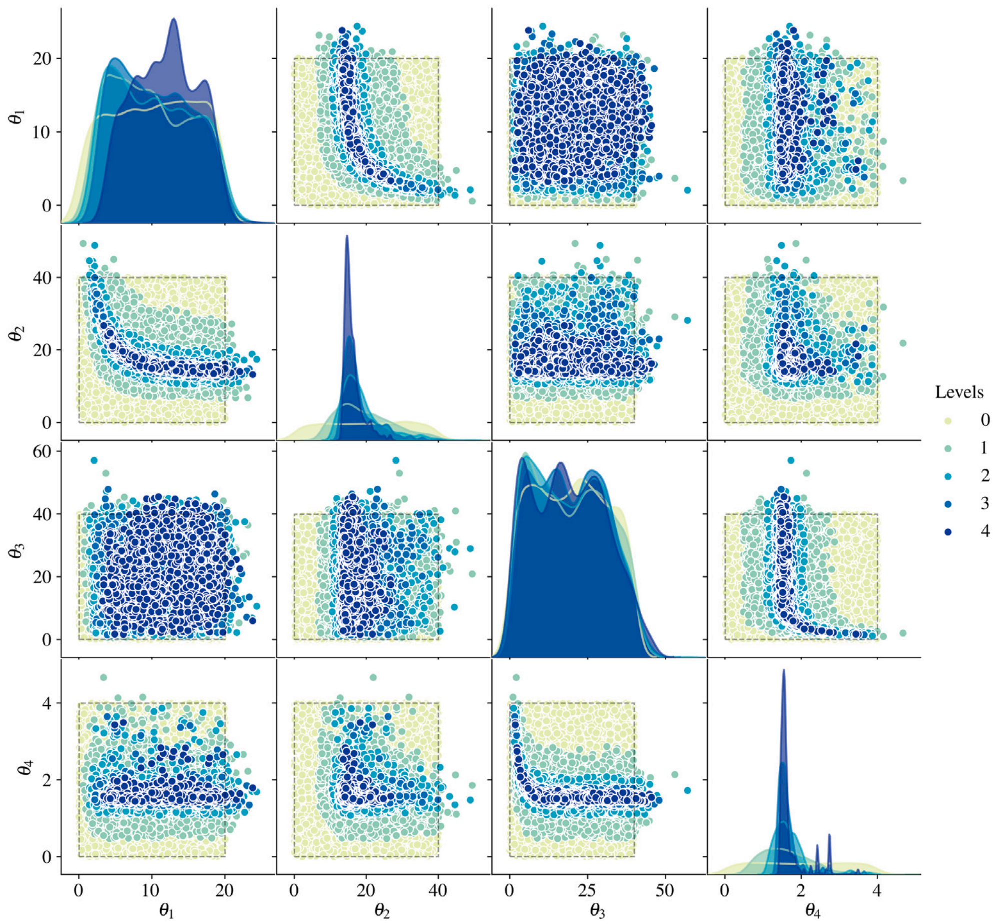
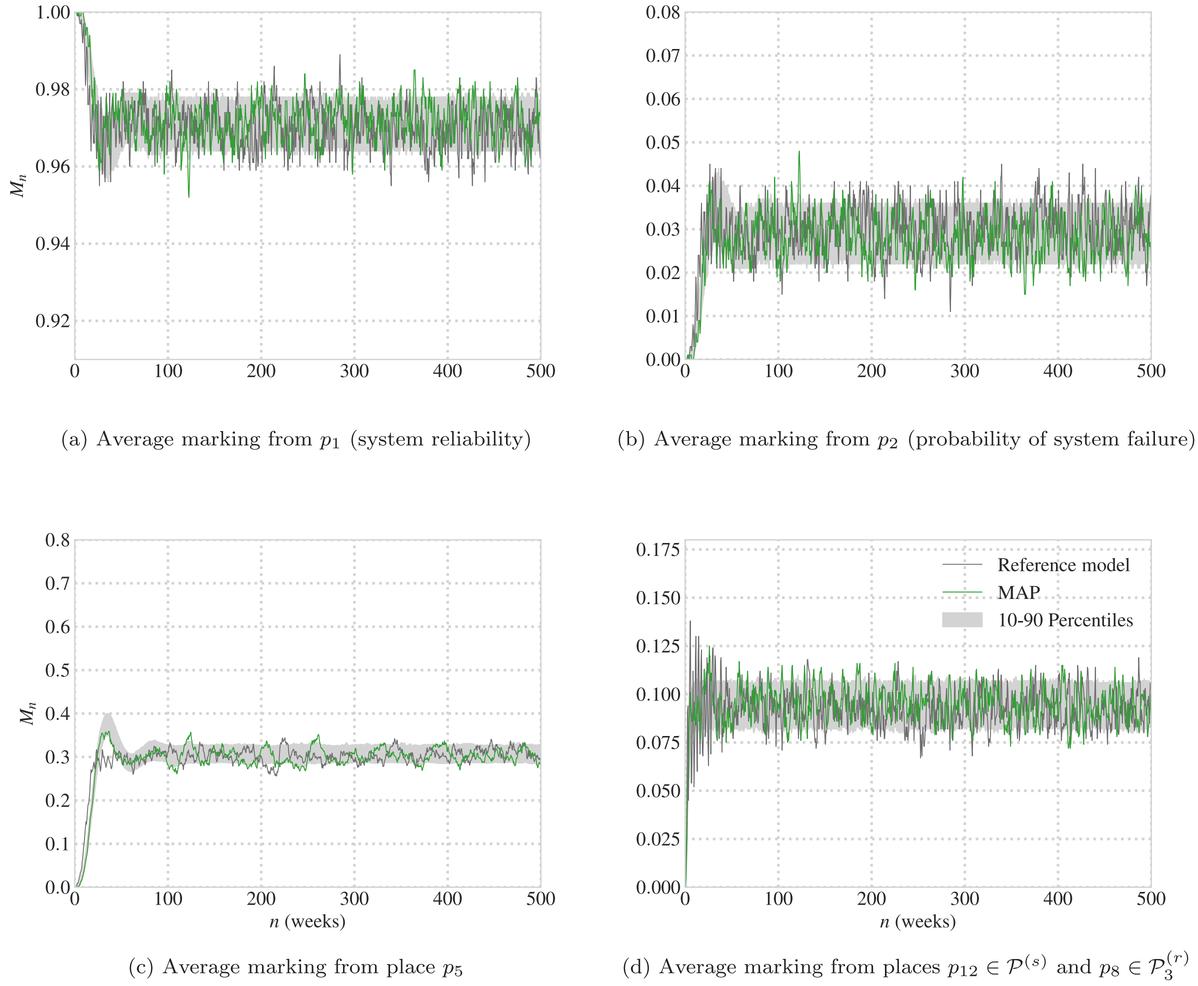


Fig. 11. Scatter plot representation of posterior samples in the θ-space as ABC-SubSim output for the third reduced Petri net model. On the diagonal, kernel density estimates are shown for the marginal posterior PDF of $\theta_1, \ldots, \theta_4$ at different simulation levels, differentiated using a color scale indicated at the right side of the plot.

Table 5
Description of the transitions from the HLPN shown in Fig. 4.

Transition	Type	Distribution	Parameters	Description of action
t_1	Symbolic	None	–	Fires instantaneously
t_2	Symbolic	None	–	Fires instantaneously
t_3	Symbolic	None	–	Fires instantaneously
t_4	Stochastic	2-Parameter Weibull	20, 2	Initiates degradation (back-up pump)
t_5	Stochastic	2-Parameter Weibull	3, 1	Initiates failure (back-up pump)
t_6	Symbolic	None	–	Fires instantaneously
t_7	Timed (reset)	Constant	0.1	Initiates maintenance on degradation (back-up pump)
t_8	Timed	Constant	0.01	Initiates inspection (back-up pump)
t_9	Stochastic	Gaussian	6, 0.5	Inspection delay (back-up pump)
t_{10}	Stochastic	2-Parameter Weibull	12, 1	Initiates degradation (main pump)
t_{11}	Stochastic	2-Parameter Weibull	2, 1	Initiates failure (main pump)
t_{12}	Symbolic	None	–	Fires instantaneously
t_{13}	Timed (reset)	Constant	0.1	Initiates maintenance on degradation (back-up pump)
t_{14}	Timed	Constant	0.01	Initiates inspection (main pump)
t_{15}	Stochastic	Gaussian	6, 0.5	Inspection delay (main pump)
t_{16}	Timed (reset)	Constant	0.1	Initiates maintenance on failure (main pump)
t_{17}	Timed	Constant	0.1	Initiates maintenance on failure (back-up pump)
t_{18}	Timed	Constant	0.01	Request maintenance on failure (back-up pump)
t_{19}	Timed	Constant	0.01	Request maintenance on degradation (back-up pump)
t_{20}	Timed	Constant	0.01	Request maintenance on degradation (main pump)
t_{21}	Timed	Constant	0.01	Request maintenance on failure (back-up pump)
t_{22}	Timed	Constant	0.01	Request maintenance on failure (main pump)
t_{23}	Symbolic	None	–	Fires instantaneously
t_{24}	Symbolic	None	–	Fires instantaneously
t_{25}	Stochastic	Gaussian	3, 0.25	Initiates a periodic maintenance
t_{26}	Timed	Constant	0.01	Assigns maintenance resources for degraded pump state
t_{27}	Probability	None	$prob_1 = 0.2$	Assigns maintenance resources
t_{28}	Timed	Constant	0.2	Removes maintenance resource
t_{29}	Timed	Constant	0.2	Removes maintenance resource
t_{30}	Timed	Constant	0.01	Removes maintenance resource



(a) Average marking from p_1 (system reliability)

(b) Average marking from p_2 (probability of system failure)

(c) Average marking from place p_5

(d) Average marking from places $p_{12} \in \mathcal{P}^{(s)}$ and $p_8 \in \mathcal{P}_3^{(r)}$

Fig. 12. Outputs of the evaluation of reference Petri net $\mathcal{P}^{(s)}$ shown in Fig. 4 (in dark color), and the third reduced Petri net model $\mathcal{P}_3^{(r)}$, represented using green line. The results are shown for places p_1, p_2, p_5, p_{12} from the reference model, and p_1, p_2, p_5, p_8, from the reduced one.

Table 6
Description of the transitions from the SPN model shown in Fig. 10.

Transition	Type	Distribution	Parameters	Description of action
t_1	Symbolic	None	–	Fires instantaneously
t_2	Symbolic	None	–	Fires instantaneously
t_3	Symbolic	None	–	Fires instantaneously
t_4	Stochastic	2-Parameter Weibull	30, 2	Initiates degradation (back-up pump)
t_5	Stochastic	2-Parameter Weibull	5, 1	Initiates failure (back-up pump)
t_6	Stochastic	Gaussian	$\theta_1, 1$	Initiates repair on degradation (back-up pump)
t_7	Stochastic	Gaussian	$\theta_2, 1$	Initiates repair on failure (back-up pump)
t_8	Stochastic	2-Parameter Weibull	12, 1	Initiates failure (main pump)
t_9	Stochastic	2-Parameter Weibull	3, 1	Initiates failure (main pump)
t_{10}	Stochastic	Gaussian	$\theta_3, 1$	Initiates repair on degradation (main pump)
t_{11}	Stochastic	Gaussian	$\theta_4, 1$	Initiates repair on failure (main pump)

that this operation is only carried out once during the model inference for the reduction of the PN reference model.

The computational cost is also evaluated through the time required to produce a number of independent forward simulations of each reduced model. Table 4 shows simulation time of 1000 repetitive runs of each corresponding model for a 500-weeks period evaluation. Note that the computational cost of model simulation also increases in a non-linear manner with the number of transitions, and is specially affected by the delay time of each transition to fire. The latter explains that $\mathcal{P}_3^{(r)}$, whose delay times are higher than those from $\mathcal{P}_2^{(r)}$ as can be seen by the posterior θ values in Fig. 11 and also in Table 6, takes less simulation time than the second one although it has more transitions. In any case, the computational cost of simulation of any of the reduced PN models is significantly lower than that of the reference model.

Next, the average probability of system failure for each model is provided for comparison. The probability of system failure, denoted here as $P_{f,SYS}$, has been obtained as a Monte Carlo estimator through the amount of visiting tokens in place p_2 with respect to all amount of model evaluations. $P_{f,SYS}^{(\text{ROB})}$ in Table 4 shows the robust probability of system failure, obtained by Bayesian averaging of the model response evaluated using the posterior samples of θ obtained through the inference. Whereas, $P_{f,SYS}^{(\text{MAP})}$ and $P_{f,SYS}^{*}$ are the average probabilities of failure of each reduced model by adopting the maximum a posteriori sample (θ_{MAP}) and the best sample (closest to $\mathcal{P}^{(s)}$), respectively. These values have been averaged after 1000 independent runs of every PN model response. The results show that the third reduced model shows the best approximation when using the MAP sample.

Finally, observe also for the third model that the parameters governing the maintenance transitions of the main pump, θ_3 and θ_4, largely show the reduction of the delay parameter governing transition t_{11}. The latter corresponds to the time to repair on failure of the main pump, and this result is in line with expectations. Indeed, since the failure is revealed, maintenance is expected shortly after the failure.

7.2. Discussion on the methodology

The previous section provided a summary of the analysis completed on the different reduced PN models presented in this case study, and discussed how this procedure can be used to rank the reduced models to aid in decision making. The proposed method overcomes bucket of limitations found in the literature summarized by limitations to predefined rules [20–23], difficulties in dealing with complexities [52,53], and limitations to the type of transition used [20–23,52,53]. On the other hand, this section highlights the challenges faced by this methodology along with its main limitations.

The proposed reduced PN methodology has revealed that the lower the size of comparison places in the reference model, the greater the level of approximation by the reduced one. By the contrary, it is possible to fit every parameter in the reduced PN to gain the closest possible approximation to the reference one, however, the computational effort counteracts the gain made by reducing the model size. Furthermore, inferring multiple parameters can result in an scenario whereby the reference output can be closely approximated but the parameters might not be inline with reality. Hence, a counterbalance between computational efficiency and efficacy should be considered. In this regards, a special consideration must be undertaken to the choice of a proper summary statistic and metric function for the model comparison, overall in those situations where the solution space is not smooth; for example, when a change in a parameter value causes a spike in the metric value. A proper summary statistic could include both the system state and the maintenance history, although this aspect is left as further research steps on this work.

Finally, with the reduced model, there may be dependencies that are contained in the reference model that are not considered during the reduction process. This could be problematic if the reduced PN alone is used for optimization of maintenance and inspection planning. To address this, the use of an ad-hoc metric for the parameter updating process can be explored, like for example, the failure probability of the system at each time. Another option is to combine the maintenance actions at each time to give a summary statistic for each proposed set of parameter values. These potential solutions to the referred particular situations can be considered as desirable extensions of this research work.

8. Concluding remarks

A novel methodology was presented to allow the reduction of the complexity of PN models while retaining the behavior of the models for key outputs. Given a proposed reduced PN model, the reduction was carried out through Approximate Bayesian Computation whereby a number of uncertain parameters from the reduced model were inferred based on the response of a reference PN. The inference was performed through an adaptive version of the ABC-SubSim algorithm which included a new method for sampling MCMC chain values to avoid manual calibration and preliminary trials beside preserving the statistical quality of the samples. An illustrative example has been provided to help the reader to easily conceptualize the procedure, and a case study has been used to demonstrate some of the challenges faced in a real-world application. The following are some concluding remarks:

(i) The methodology can be applied to complex PN without restrictions (like HLPN) on the type of transitions that can be reduced;
(ii) A set of candidate reduced models proposed by the modeler can be assessed to give a comparative measure of their suitability;
(iii) Further work can consider combining the proposed approach with alternative methods for quantifying the model similarity;
(iv) More research is needed to explore optimal summary statistics to extend the application of the proposed method to extremely complex PN models.

CRediT authorship contribution statement

Manuel Chiachío: Writing – original draft, Supervision, Project administration, Methodology, Funding acquisition, Conceptualization. **Ali Saleh:** Writing – original draft, Software, Investigation, Formal analysis. **Susannah Naybour:** Software, Formal analysis, Data curation. **Juan Chiachío:** Writing – review & editing, Validation, Investigation. **John Andrews:** Writing – review & editing, Resources.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

This paper is part of the ENHAnCE ITN project (https://www.h2020-enhanceitn.eu/) funded by the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No. 859957. The authors would like to thank the Lloyd's Register Foundation (LRF), a charitable foundation in the U.K. helping to protect life and property by supporting engineering-related education, public engagement, and the application of research. The authors gratefully acknowledge the support of these organizations which have enabled the research reported in this paper.

References

[1] Zio E. Reliability engineering: Old problems and new challenges. Reliab Eng Syst Saf 2009;94(2):125–41.
[2] Carvalho TP, Soares FA, Vita R, Francisco RdP, Basto JP, Alcalá SG. A systematic literature review of machine learning methods applied to predictive maintenance. Comput Ind Eng 2019;137:106024.
[3] Compare M, Martini F, Zio E. Genetic algorithms for condition-based maintenance optimization under uncertainty. European J Oper Res 2015;244(2):611–23.
[4] Izquierdo J, Márquez AC, Uribetxebarria J. Dynamic artificial neural network-based reliability considering operational context of assets.. Reliab Eng Syst Saf 2019;188:483–93.
[5] Bian C, Yang S, Huang T, Xu Q, Liu J, Zio E. Degradation state mining and identification for railway point machines. Reliab Eng Syst Saf 2019;188:432–43.
[6] Chen Y, Liu Y, Jiang T. Optimal maintenance strategy for multi-state systems with single maintenance capacity and arbitrarily distributed maintenance time. Reliab Eng Syst Saf 2021;211:107576.
[7] Zhang H, Marsh DWR. Managing infrastructure asset: Bayesian networks for inspection and maintenance decisions reasoning and planning. Reliab Eng Syst Saf 2021;207:107328.
[8] Qu C, Zhang L, Zhang B, Gao M. Maintenance resource scheduling modeling by petri net. In: 2009 9th international conference on electronic measurement & instruments. IEEE; 2009, p. 4–861.
[9] Sheng J, Prescott D. Using a novel hierarchical coloured Petri net to model and optimise fleet spare inventory, cannibalisation and preventive maintenance. Reliab Eng Syst Saf 2019;191:106579.
[10] Chahrour N, Nasr M, Tacnet J-M, Bérenguer C. Deterioration modeling and maintenance assessment using physics-informed stochastic Petri nets: Application to torrent protection structures. Reliab Eng Syst Saf 2021;210:107524.
[11] Lee J, Mitici M. An integrated assessment of safety and efficiency of aircraft maintenance strategies using agent-based modelling and stochastic Petri nets. Reliab Eng Syst Saf 2020;202:107052.
[12] Petri CA. Kommunikation mit automaten (Ph.D. thesis), Institut fur Instrumentelle Mathematik an der Universitat Bonn; 1962.

[13] Taleb-Berrouane M, Khan F, Amyotte P. BayesIan stochastic Petri nets (BSPN)-a new modelling tool for dynamic safety and reliability analysis. Reliab Eng Syst Saf 2020;193:106587.

[14] Andrews J, Fecarotti C. System design and maintenance modelling for safety in extended life operation. Reliab Eng Syst Saf 2017;163:95–108.

[15] Murata T. Petri Nets: Properties, analysis and applications. Proc IEEE 1989;77:541–80.

[16] Murad CA, de Melani AHA, de Michalski MAC, Netto AC, de Souza GFM. Estimation of operational and maintenance tasks influence on equipment availability through Petri net modeling. In: 2020 annual reliability and maintainability symposium (rams). 2020, p. 1–6. http://dx.doi.org/10.1109/RAMS48030.2020.9153582.

[17] Latsou C, Dunnett SJ, Jackson LM, et al. A new methodology for automated Petri net generation: Method application. Reliab Eng Syst Saf 2019;185:113–23.

[18] Knezevic J, Odoom E. Reliability modelling of repairable systems using Petri nets and fuzzy lambda–tau methodology. Reliab Eng Syst Saf 2001;73(1):1–17.

[19] Chiachío M, Chiachío J, Prescott D, Andrews J. A new paradigm for uncertain knowledge representation by plausible Petri nets. Inform Sci 2018;453:323–45.

[20] Senderovich A, Rogge-Solti A, Gal A, Mendling J, Mandelbaum A, Kadish S, Bunnell CA. Data-driven performance analysis of scheduled processes. In: International conference on business process management. Springer; 2016, p. 35–52.

[21] Senderovich A, Shleyfman A, M. Weidlich M, A. Gal A, Mandelbaum A. To aggregate or to eliminate? Optimal model simplification for improved process performance prediction. Inf Syst 2018;78:96–111.

[22] Berthomieu B, Le Botlan D, Dal Zilio S. Counting Petri net markings from reduction equations. Int J Softw Tools Technol Transf 2020;22:163–81.

[23] Bønneland FM, Dyhr J, Jensen PG, Johannsen M, Srba J. Stubborn versus structural reductions for Petri nets. J Log Algebr Methods Program 2019;102:46–63.

[24] Padma V. A study on flexible manufacturing systems using petr net reduction. IOP Conf Ser: Mater Sci Eng 2021;1130.

[25] Bourdil P-A, Berthomieu B, Dal Zilio S, Vernadat F. Symmetry reduction for time Petri net state classes. Sci Comput Program 2016;132:209–25.

[26] Berthomieu B, Le Botlan D, Dal Zilio S. Counting Petri net markings from reduction equations. Int J Softw Tools Technol Transf 2020;22(2):163–81.

[27] Berthomieu B, Le Botlan D, Dal Zilio S. Petri Net reductions for counting markings. In: International symposium on model checking software. Springer; 2018, p. 65–84.

[28] Rogge-Solti A, Mans RS, van der Aalst WM, Weske M. Repairing event logs using timed process models. In: OTM confederated international conferences“ on the move to meaningful internet systems”. Springer; 2013, p. 705–8.

[29] Leclercq E, Lefebvre D, Medhi S. Identification of timed stochastic Petri net models with normal distributions of firing periods. 13, 2009, p. 948–53,

[30] Buchholz R, Krull C, Horton G. Reconstructing model parameters in partially-observable discrete stochastic systems. 2011, p. 159–74.

[31] Li X-Y, Liu Y, Lin Y-H, Xiao L-H, Zio E, Kang R. A generalized Petri net-based modeling framework for service reliability evaluation and management of cloud data centers. Reliab Eng Syst Saf 2021;207:107381.

[32] Yan R, Dunnett S. Improving the strategy of maintaining offshore wind turbines through Petri net modelling. Appl Sci 2021;11(2):574.

[33] Yianni PC, Neves L, Andrews J. Accelerating Petri-net simulations using NVIDIA graphics processing units. European J Oper Res 2018;265:361–71.

[34] Marjoram P, Molitor J, Plagnol V, Tavare S. Markov chain Monte Carlo without likelihoods. Proc Natl Acad Sci 2003;100(26):15324–8.

[35] Sisson SA, Fan Y, Beaumont M. Handbook of Approximate Bayesian Computation. CRC Press; 2018.

[36] Chiachio M, Beck J, Chiachio J, Rus G. Approximate Bayesian computation by subset simulation. SIAM J Sci Comput 2014;36:A1339–58.

[37] Papaioannou I, Betz W, Zwirglmaier K, Straub D. MCMC algorithms for subset simulation. Probab Eng Mech 2015;41:89–103.

[38] Beck JL. BayesIan system identification based on probability logic. Struct Control Health Monit 2010;17(7):825–47.

[39] Jensen K, Rozenberg G. High-level petri nets: theory and application. Springer Science & Business Media; 2012.

[40] Fearnhead P, Prangle D. Constructing summary statistics for approximate Bayesian computation: Semi-automatic approximate Bayesian computation. J R Stat Soc Ser B Stat Methodol 2012;74(3):419–74.

[41] Beaumont M, Cornuet J-M, Marin J-M, Robert C. Adaptive approximate Bayesian computation. Biometrika 2009;96:983–90.

[42] Dutta R, Schoengens M, Pacchiardi L, Ummadisingu A, Widmer N, Onnela J-P, Mira A. ABCpy: A high-performance computing perspective to approximate Bayesian computation. 2020, arXiv:1711.04694.

[43] Hadjidoukas P. High performance framework for Bayesian uncertainty quantification and optimization. 2021, URL: https://github.com/cselab/pi4u/tree/master/inference (Accessed Feb 25, 2021).

[44] Au S-K, Beck J. Estimation of small failure probabilities in high dimensions by subset simulation. Probab Eng Mech 2001;16:263–77.

[45] Zuev KM, Beck JL, Au S-K, Katafygiotis LS. BayesIan post-processor and other enhancements of subset simulation for estimating failure probabilities in high dimensions. Comput Struct 2012;92:283–96.

[46] Gelman A, Roberts G, Gilks W. Efficient metropolis jumping rules. Bayesian statistics, Vol. 5, Oxford University Press Oxford; 1996.

[47] Vakilzadeh MK, Huang Y, Beck JL, Abrahamsson T. Approximate Bayesian computation by subset simulation using hierarchical state-space models. Mech Syst Signal Process 2017;84:2–20.

[48] Barros J, Chiachio M, Chiachio J, Cabanillas F. Adaptive approximate Bayesian computation by subset simulation for structural model calibration. Comput Aided Civ Infrastruct Eng 2021. http://dx.doi.org/10.1111/mice.12762, (in press).

[49] Pham D, Karaboga D. Intelligent optimisation techniques. 2nd ed.. Springer-Verlag Berlin Heidelberg; 2000.

[50] Desa MM, Desa E. Encyclopedia of distances. Springer-Verlag Berlin Heidelberg; 2009.

[51] Forsthoffer MS. More best practices for rotating equipment. Butterworth-Heinemann; 2017.

[52] Ciardo G, Trivedi KS. A decomposition approach for stochastic Petri net models. In: Proceedings of the fourth international workshop on petri nets and performance models. IEEE Computer Society; 1991, p. 74–5.

[53] Woodside CM, Li Y. Performance Petri net analysis of communications protocol software by delay-equivalent aggregation. In: Proceedings of the fourth international workshop on petri nets and performance models. IEEE Computer Society; 1991, p. 64–5.