Performance Evaluation 97 (2016) 36–56

Contents lists available at ScienceDirect

Performance Evaluation

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

Survivability analysis of a sewage treatment facility using hybrid Petri nets

Hamed Ghasemieh [a,*], Anne Remke [a,b], Boudewijn R. Haverkort [a]

[a] Design and Analysis of Communication Systems, University of Twente, Netherlands
[b] University of Münster, Germany

ARTICLE INFO

Article history:
Available online 5 December 2015

Keywords:
Model checking
Dependability
Hybrid Petri nets
Case study

ABSTRACT

Waste water treatment facilities clean sewage water from households and industry in several cleaning steps. Such facilities are dimensioned to accommodate a maximum intake. However, in the case of very bad weather conditions or failures of system components, the system might not be able to accommodate all waste water. This paper models a real waste water treatment facility, situated in the city of Enschede, the Netherlands, with Hybrid Petri nets with general transitions, to analyse under which circumstances the existing infrastructure will overflow. Comparing to previous models an structural extension is proposed, and one limitation is tackled. First, we extended the hybrid Petri net formalism with *guard arcs* and *dynamic continuous transitions*, to be able to model dependencies on continuous places and the rates of continuous transitions. Secondly, we tackle the restriction of having only a single general transition, by proposing a new discretization method. We introduce to different discretization methods, and compare their efficiency in a complex case study. Using recently developed algorithms for model checking STL properties on hybrid Petri nets, the paper computes survivability measures that can be expressed using the path-based until operator. After computing measures for a wide range of parameters, we provide recommendations as to where the system can be improved to reduce the probability of overflow.

1. Introduction

Any water that has been affected in quality either by households or by industries is considered as waste water. It is usually conveyed in the sewerage system of the community to the nearest waste water treatment companies. The treatment process consists of several physical, chemical and biological cleaning steps. The goal of the process is to separate the clean water from the so-called sludge, that can later be safely disposed or used as fertilizer. The cleaned water is usually released to surface water in the area.

In the Netherlands, communities normally have contracts with waste water treatment facilities about the maximum amount of waste water that needs to be taken in by the treatment facility. Hence, these facilities are dimensioned to accommodate the treatment of a maximum amount of sewage, often *without* taking into account the possibility of unforeseen events. However, in the case of very heavy rainfall, which is hard to predict, and actually may occur more often due to climate change, it may happen that the amount of waste water in the community sewerage exceeds the available

* Corresponding author.
E-mail addresses: h.ghasemieh@utwente.nl (H. Ghasemieh), anne.remke@wwu.de (A. Remke), b.r.h.m.haverkort@utwente.nl (B.R. Haverkort).

http://dx.doi.org/10.1016/j.peva.2015.11.004
0166-5316/

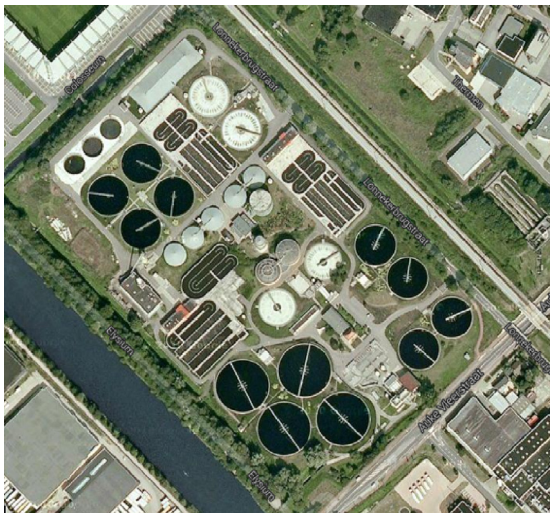


Fig. 1. A bird's eye view picture of the sewage treatment facility in Enschede, the Netherlands. The picture is retrieved using Google Maps.

storage capacity. In such cases, the sewerage system of the community overflows and waste water is spilt on the streets. Recently, this happened in the city of Enschede, the Netherlands [1–3] and caused hindrance to citizens and traffic.

This paper investigates under which circumstances flooding occurs and what can possibly be done to reduce the probability of such flooding. For this purpose we have employed data like the capacity of tanks and the average residence time of water in the different cleaning stages from the treatment facility in the city of Enschede. A bird's-eye view of this facility is shown in Fig. 1. This information is used to model the operation of the treatment facility as a Hybrid Petri-net with general one-shot transitions (HPnG) per different failure scenario.

The modelling formalism of HPnGs has recently been introduced for the analysis of fluid critical infrastructures [4], and efficient algorithms have been introduced for their analysis [5]. This paper is an extension of the work in [6], by allowing more than one general transition. Currently, the exact analysis algorithms of HPnGs are restricted to models with a single general one-shot transition. However, this paper shows how models with multiple general transitions can be analysed by discretizing all general transitions but one. Since the analysis of HPnG models with a single general one-shot transition is extremely quick, this is a feasible approach and the resulting computation times are still reasonable. Clearly, the results obtained with discretization are no longer exact; however, given the speed of the method a very fine grained discretization can be chosen which limits the introduced error. Also, given the lack of analytical methods in this area, the proposed method is very useful from the application point of view, since it allows to evaluate a combination of different failures, which has not been possible before.

Survivability [7–9] is defined as the probability that a system recovers to a predefined service level in a timely manner. It is mostly evaluated for so-called "Given the Occurrence Of Disaster" (GOOD) models. In such models, as the name suggests, the occurrence of a disaster is assumed to happen at a certain time of the day, instead of trying to predict the probability of a disaster using risk management. The focus then lies on the effect, the handling and the recovery of the disaster, once it has happened.

Survivability properties can be expressed for HPnGs using the syntax and semantics of Stochastic Time Logic (STL), that has been introduced in [10], together with algorithms to efficiently check STL properties on HPnGs. Especially the analysis of the path-based until formula is suitable, e.g., to evaluate how well the system performs in the presence of bad weather conditions or failures at the intake to the treatment facility.

In this paper, the HPnG formalism has been extended by two new features, i.e., **guard arcs** and **dynamic transitions**, since they have been shown to be essential when modelling waste water treatment facilities. Guard arcs combine test and inhibitor arcs, as previously present in the formalism, but additionally allow to control discrete events of the system based on the values of continuous variables. Dynamic transitions are continuous transitions, where the rate depends on the rate of other continuous transitions. As will be explained in this paper, both extensions can be incorporated in the analysis and model checking algorithms without increasing their complexity.

Computing measures of interest for the HPnG model of the treatment facility for a wide range of parameters, then allows us investigate how and where the community and the treatment facility could invest best, e.g., by installing larger buffers or more pumping equipment, to reduce the residence time of waste water in the treatment phases in order to decrease the probability of spilling waste water in the streets.

To the best of our knowledge the quantitative evaluation of effects of failures or very bad weather conditions is not usually performed for waste water cleaning facilities in particular or in civil engineering in general. The common way of dimensioning such systems is to use static models and calculations [11]. Risk assessment is generally performed for civil

engineering facilities in various ways [12], however, it is not able to predict the consequences of failures in a quantitative way. Another approach that is commonly used is simulation [13], which is, however, very time consuming and does not allow to analyse a wide range of parameter settings quickly.

Similar to e.g., Fluid Stochastic Petri nets [14,15], Dynamical Systems having Piecewise-Constant Derivatives [16], and Piece-wise Deterministic Markov Processes [17,18], Hybrid Petri Nets form a highly restricted subclass of stochastic hybrid models (SHMs) [19]. They are based on Hybrid Petri nets [20] and can be mapped to hybrid automata [21,22] if no stochastic behaviour is considered. Hybrid Petri nets are analysed by partitioning their underlying state-space into regions, in the spirit of the computation of the underlying state-space of Timed Automata [23,24] and Dense-Time Reactive Systems [25,26]. For a more in-depth treatment of the related work we refer to [4,5,10].

This paper is organized as follows, Section 2 introduces a modified version of the definition and modelling formalism of HPnG with addition of the new features of guard arcs and dynamic transitions. Section 3 revisits region-based analysis and the idea of the partitioning of the state space. A brief definition of STL is provided, and discretization method is introduced for handling more than one general transition. Section 4 demonstrates the use of region-based analysis and STL for a simple control example. In Section 5, using the new features of guard arcs and dynamic transitions, a new component for modelling overflow places in real world water treatment facilities is introduced. In Section 6, the case study for modelling and analysis of the sewage treatment facility in the city of Enschede is investigated. Finally, Section 7 concludes the paper.

2. Model definition

In the following we extend the HPnG definition from [4] by introducing dynamic transitions and guard arcs connecting fluid places and discrete transitions. Dynamic transitions are a special form of continuous transitions where the outflow may depend on the flow of other continuous transitions, as explained later. This can be used to model, for example, overflow places. Test and inhibitor arcs have been used before to control the enabling of discrete or continuous transitions via the content of a connected discrete place. Guard arcs, as introduced in this paper, combine the functionality of test and inhibitor arcs and additionally allow to control the enabling of a transition by comparing the content of a continuous place with a given boundary condition. This allows to model typical control examples. This section repeats the model definition in Section 2.1, the graphical representation of a HPnG in Section 2.2 and briefly discusses the model evolution in Section 2.3. For a more detailed description of HPnGs and their evolution, we refer to [27].

2.1. Model

As before, an HPnG is defined as a tuple $(\mathcal{P}, \mathcal{T}, \mathcal{A}, \mathbf{m}_0, \mathbf{x}_0, \Phi)$, where $\mathcal{P} = \mathcal{P}^D \cup \mathcal{P}^C$ is a set of *places* that can be divided into two disjoint sets $\mathcal{P}^D$ and $\mathcal{P}^C$ for the discrete and continuous places, respectively. The discrete marking $\mathbf{m}$ is a vector that represents the number of tokens $m_P \in \mathbf{N}$ for each discrete place $P \in \mathcal{P}^D$ and the continuous marking $\mathbf{x}$ is a vector that represents the non-negative level of fluid $x_P \in \mathbf{R}_0^+$ for each continuous place P. The initial marking is given by $(\mathbf{m}_0, \mathbf{x}_0)$.

Four types of *transitions* are possible, as follows. The set of immediate transitions, the set of deterministically timed transitions, the set of general transitions, and the set of continuous transitions together form the finite set of transitions $\mathcal{T} = \mathcal{T}^I \cup \mathcal{T}^D \cup \mathcal{T}^G \cup \mathcal{T}^C$. The set of continuous transitions, $\mathcal{T}^C$, itself consists of two disjoint sets: static and dynamic transitions, denoted by $\mathcal{T}^{Dy}$ and $\mathcal{T}^{St}$, respectively. Static continuous transitions are the same as the previous continuous transitions, i.e., they have constant nominal firing rates. Note that, due to rate adaptation (see below) the nominal rate of a static transition may change and is then called the actual rate. In contrast to static transitions, where the nominal rates are always constant, the nominal rate of dynamic transitions may depend on the actual rate of any other static transition in the HPnG at hand.

The set of *arcs* $\mathcal{A}$ consists of three sets: The set of discrete input and output arcs $\mathcal{A}^D$ connects discrete places and discrete transitions and the set of continuous input and output arcs $\mathcal{A}^C$ connects continuous places and continuous transitions. The set of *guard arcs* $\mathcal{A}^G$ connects discrete places to all kinds of transitions, and also continuous places to all but continuous transitions. These arcs ensure that a transition is only enabled in case the number of tokens (in case of a discrete place) or the amount of fluid (in case of a continuous place) fulfils a certain condition that is specified on the guard arc.

The tuple $\Phi = (\phi_b^{\mathcal{P}}, \phi_p^{\mathcal{T}}, \phi_d^{\mathcal{T}}, \phi_{St}^{\mathcal{T}}, \phi_{Dy}^{\mathcal{T}}, \phi_g, \phi_w^{\mathcal{A}}, \phi_u^{\mathcal{A}}, \phi_s^{\mathcal{A}}, \phi_p^{\mathcal{A}})$ contains 10 *functions*. Function $\phi_b^{\mathcal{P}} : \mathcal{P}^C \to \mathbf{R}^+ \cup \infty$ assigns an upper bound to each continuous place. In contrast to the definition of HPnG in [4] in the following $\phi_p^{\mathcal{T}} : \mathcal{T}^D \cup \mathcal{T}^I \to \mathbf{N}$ specifies a *unique priority* to each immediate and deterministic transition to resolve firing conflicts, as in [28]. Deterministic transitions have a constant firing time defined by $\phi_d^{\mathcal{T}} : \mathcal{T}^D \to \mathbf{R}^+$ and continuous static transitions have a constant nominal flow rate defined by $\phi_{St}^{\mathcal{T}} : \mathcal{T}^{St} \to \mathbf{R}^+$. Mapping $\phi_{Dy}^{\mathcal{T}} : \mathcal{T}^{Dy} \to f$, assigns a function $f : \mathbf{R}^{|\mathcal{T}^{St}|} \to \mathbf{R}^+$ to each dynamic continuous transition, which determines how its nominal flow rate depends on the continuous static transitions rates. The general transition is associated with a random variable S, representing its firing time, according to a cumulative distribution function (CDF) $\phi_g(s)$, and its probability density function (PDF) is denoted $g(s)$. We assign a weight to all discrete input and output arcs: $\phi_w^{\mathcal{A}} : \mathcal{A}^D \to \mathbf{N}$ which defines the number of tokens that is taken from or added to connected places upon the firing of the transition. $\phi_u^{\mathcal{A}} : \mathcal{A}^G \to \{(\triangleright, \mathbf{R})\}$, with $\triangleright = \{\geq, <\}$ assigns a tuple which consists of a comparison operator and a real number to all guard arcs. The functions $\phi_s^{\mathcal{A}}$, $\phi_p^{\mathcal{A}}$ specify the share and the priority of a static continuous transition, as will be explained later.

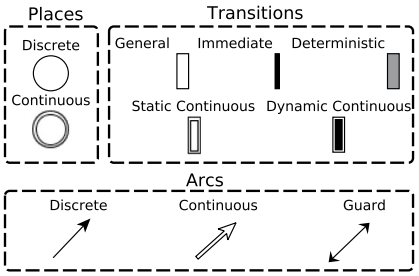


Fig. 2. Graphical representation of primitives of HPnG.

2.2. Graphical representation

The primitives of the hybrid Petri net formalism with general one-shot transitions are shown in Fig. 2. A discrete place is graphically represented by a single circle and a fluid place is represented by two concentric circles. A general transition is drawn as an empty rectangle, a deterministic transition is drawn as a grey rectangle, a continuous static transition is shown as an empty rectangle with double lines, a continuous dynamic transition is shown by a double lined solid rectangle and an immediate transition is a thick black bar. The discrete input and output arcs are drawn as single arrows and fluid input and output arcs are represented with double lines. Guard arcs are drawn with two triangular arrowheads.

2.3. System evolution

Markings are collected in two different vectors, the discrete marking $\mathbf{m} = (m_1, \ldots, m_{|\mathcal{P}^D|})$ and the continuous marking $\mathbf{x} = (x_1, \ldots, x_{|\mathcal{P}^C|})$. The initial marking is composed of a discrete part $\mathbf{m}_0$ that describes the initial amount of tokens in all discrete places and a continuous part $\mathbf{x}_0$ that describes the initial amount of fluid in all continuous places.

The *state* of an HPnG is defined by $\Gamma = (\mathbf{m}, \mathbf{x}, \mathbf{c}, \mathbf{d}, \mathcal{G})$, where vector $\mathbf{c} = (c_1, \ldots, c_{|\mathcal{T}^D|})$ contains a clock c_i for each deterministic transition that represents the time that T_i^D has been enabled. When a transition is disabled the clocks do not evolve, but the clock value is preserved until the transition is enabled again. Clocks are only reset when the corresponding deterministic transition fires. If the general transition has not fired yet, it can be considered as a deterministic transition whose firing time is sampled from the corresponding general firing time distribution. This sampling happens only once per model execution, and it occurs when the general transition becomes enabled for the first time. Vector $\mathbf{d} = (d_1, \ldots, d_{|\mathcal{P}^C|+|\mathcal{T}^D|})$ indicates the drift (speed of change) of all continuous variables present in the model. For continuous places it shows the change of fluid per time unit, and for deterministic transitions it is the clock drift which is either one or zero, if the transition is enabled or disabled, respectively. Note that even though the vector $\mathbf{d}$ is determined uniquely by x, m, and the weights of the guard arcs, it is included in the definition of a state for ease of notations and analysis. The general transition is only allowed to fire once, hence, the flag $\mathcal{G} \in \{0, 1\}$ indicates whether the general transition has already fired ($\mathcal{G} = 1$), or not ($\mathcal{G} = 0$). So, the initial state of the system is $\Gamma_0 = (\mathbf{m}_0, \mathbf{x}_0, \mathbf{0}, \mathbf{d}_0, 0)$. A system state can be seen as a snapshot of the system evolution at a specific time, and assumed general transition firing time; this is elaborated in more detail in the next section.

The firing rules of deterministic, general, immediate and fluid transitions differ. Whether a transition is allowed to fire depends (1) on the structure and the current marking of the Petri net (*concession*) and (2) on the type of the transitions [29], as follows.

Continuous transitions that have concession are always enabled, and continuously transport fluid along fluid arcs. Conflicts in the distribution of fluid occur when a continuous place reaches one of its boundaries. To prevent overflow, the fluid input has to be reduced to match the output, and to prevent underflow the fluid output has to be reduced to match the input, respectively. The firing rate of both, static and dynamic continuous transitions is then adapted according to the share $\phi_s^{\mathcal{A}} : \mathcal{A}^C \to \mathbf{R}^+$ and priority $\phi_p^{\mathcal{A}} : \mathcal{A}^C \to \mathbf{N}$ that is assigned to the continuous arcs that connect the transition to the place. This is done by distributing the available fluid over all continuous arcs. Those with highest priority are considered first and if there is enough fluid available, all transitions with the highest priority can still fire at their nominal speed. Otherwise, their actual fluid rates are adapted according to the firing rate of the connected transitions and the share of the arc, according to [29]. The adaptation of fluid rates in these cases results in a piecewise constant fluid derivative per continuous place.

Non-fluid transitions that have concession may be enabled, depending on their type. If an immediate transition has concession the marking is said to be *vanishing*, otherwise the marking is said to be *tangible*. Immediate transitions have precedence over deterministic and general transitions. In a vanishing marking, deterministic and general transitions are disabled and cannot fire. The clock of each enabled deterministic transition T_i^D evolves with time at rate $dc_i/d\tau = 1$ and when a clock reaches its firing time, i.e., $c_i = \phi_d^{\mathcal{T}}(T_i)$, transition T_i^D fires. Similarly, the enabling time of the enabled general transition, that has not fired yet, evolves with time at rate 1. The general transition then fires with probability $\phi_g(\tau + \Delta\tau) - \phi_g(\tau) = \int_\tau^{\tau+\Delta\tau} g(s)ds$ in any time interval $[\tau, \tau + \Delta\tau]$.

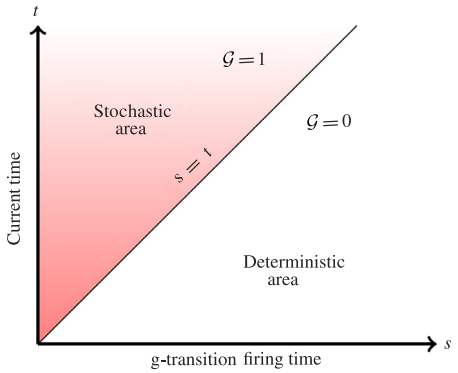


Fig. 3. Generic presentation of STD.

Whenever a non-fluid transition fires, the marking evolves according to a firing rule, depending on the type of the transition. All discrete transition types, i.e., immediate, deterministic and general, change the discrete part of the marking **m** in a similar way.

3. Analysis

We recall how the underlying state space of an HPnG is partitioned and how this can be used for efficient analysis in Section 3.1. Section 3.2 explains the basic idea behind model checking HPnGs and how to formulate STL properties. Finally, Section 3.3 explains how an HPnG model with multiple general transitions can be evaluated, using discretization.

3.1. Stochastic time diagram

The Stochastic Time Diagram (STD) as introduced in [5] provides a convenient visual representation of the evolution of a HPnG for a given initial state. The STD is built based on the idea that for an initial state of an HPnG and a predefined value for the firing time of the general transition, denoted s, for all future time instances t we can determine the state of the system. The STD is a two-dimensional diagram with t and s as its vertical and horizontal axis, respectively. Each point in this diagram is associated with a unique HPnG state, which is denoted by $\Gamma(s, t)$. Note that, for a fixed value of s, the evolution over time is fully deterministic and associates with a vertical line in the STD. A generic version of this diagram is shown in Fig. 3. Two main areas can be distinguished in this figure. The area above the line $t = s$, called *stochastic area*, contains all the HPnG states for which the general transition has fired ($t > s$), whereas the area below the line $t = s$, called *deterministic area*, includes those states for which the general transition has not yet fired ($t < s$). To compute measures of interest for HPnGs, the STD needs to be deconditioned with the probability density function $g(s)$.

The main idea behind the method proposed in [5], is that instead of dealing with infinitely many points in the ts-plane, we can partition it into several *regions*. These regions exist, because the state of the system does not change until an *event* occurs. In each system state, three types of potential events can occur:

1. A continuous place reaching its lower/upper boundary, which imposes a change in the drift of the continuous place, due to rate adaptation [29].
2. A continuous place reaches the weight of the guard arc connected to it, which will enable or disable a transition. In case of an immediate transition, it will fire and alter the discrete marking immediately, and if it is a deterministic transition its clock drift will be set to one, therefore changing a continuous variable.
3. An enabled transition, either deterministic or general fires, which alters the discrete marking or the general transition flag.

Hence, in overall, an event may cause a change in the discrete marking, a change in drift (either for clocks or fluid levels) or a change in a general transition flag. We define a region as a set of states that while the system remains in them no event occurs, i.e., the discrete marking, the drift of continuous variables and the general transition flag remain unchanged. Moreover, at the occurrence of an event the system enters another region. This leads to the following definition.

Definition 1. A region $\mathcal{R}$ is a set of (s, t) points in a given STD, for which we have:

$$\forall (s_1, t_1), (s_2, t_2) \in \mathcal{R}: \begin{cases} \Gamma(s_1, t_1).\mathbf{m} = \Gamma(s_2, t_2).\mathbf{m}, \\ \Gamma(s_1, t_1).\mathbf{d} = \Gamma(s_2, t_2).\mathbf{d}, \\ \Gamma(s_1, t_1).\mathcal{G} = \Gamma(s_2, t_2).\mathcal{G}. \end{cases}$$

In which, while $\Gamma(s,t).\mathbf{m}$ is used to refer to the vector of discrete markings, $\Gamma(s,t).m_P$ is used to refer to the discrete marking of a specific place P. A similar notation is used for the continuous marking.

In this definition vector **d**, contains drifts of continuous places and clock drifts for deterministic transitions, which are either one or zero. The reason for this is that, by introducing guard arcs, a deterministic transition can be enabled or disabled for the same discrete marking, due to a change in the continuous marking. This contradicts the idea of grouping system states into regions.

An example partitioning of the state space into regions is shown in Fig. 4 together with probability density function $g(s)$. In [5] it is shown that, inside a region all continuous variables, i.e., the amount of fluid and the clock valuations, can be represented by simple linear equations in s and t. Intuitively, this is because in a region all continuous places are associated with a constant drift and clocks also have a constant drift (of one). Using this we infer that the boundaries between regions, which represent the occurrence of an event, are characterized by linear functions of s and t. Hence, each region in the STD is a polygon, where the exact shape of these regions depends on the structure of the model. Note that the introduction of dynamic fluid transitions does not change this fact, because their nominal rates depend on the actual rates of other static continuous transitions, which are all constant *within* each region. Hence, we can safely treat dynamic transitions as static transitions. Having the partitioning of the state space, to compute the probability to be in a specific system state at time τ, it suffices to find all regions intersecting the horizontal line $t = \tau$ that correspond to the specific system state and integrate $g(s)$ over the intersection. This idea is also illustrated for a given partitioning in Fig. 4.

Even though reachability computations on the STD are always performed for a given and finite time bound, there is still a possibility of having an infinite number of regions in the STD before the finite time bound. This happens whenever an infinite sequence of vanishing markings occurs. This problem is well-known for all Petri nets formalism that allow immediate transitions. However, if we require that models have to be bounded, infinite sequences of vanishing markings can only take place in the form of cycles of vanishing markings, which can be detected and removed [30]. This ensures that we always reach a tangible marking in a finite number of steps and the number of regions in the STD before a finite time bound is also finite. Hence, for a bounded model and a finite time bound the algorithm always terminates.

3.2. Stochastic Time Logic

Stochastic Time Logic (STL) introduced in [10], allows us to represent and evaluate path-based formula, such as time-bounded reachability, to answer whether a certain property is reached, within a certain time, while another given condition holds. STL is basically an extended version of the state-based logic in [5], with an until operator. STL is used to reason about the underlying state space of an HPnG, i.e., it is possible to reason whether an STL formula holds for a certain system state $\Gamma(s,t)$. Note that STL reasons on the conditioned state-space of an HPnG, that is, on the regions of an STD, which means that the distribution of the general transition is *not* yet taken into account.

Definition 2 (*Stochastic Time Logic*)**.** An STL formula Ψ is defined as

$$\Psi := \mathtt{tt} \mid x_P \geq c \mid m_P = a \mid \neg\Psi \mid \Psi \wedge \Psi \mid \Psi \; \mathcal{U}^{[T_1,T_2)} \; \Psi,$$

where $T_1, T_2 \in \mathbb{R}^+, x \geq c$ and $m = a$, with $a \in \mathbb{N}, c \in \mathbb{R}^+$, are called continuous and discrete atomic properties, respectively.

Note that although the above definition allows nested until formula, we only consider non-nested until formula so far, as in [10].

In the following two different satisfaction relations are introduced. First, the relation $\models^{s,t}$ between a single system state $\Gamma(s,t)$ and an STL formula Ψ, which is intuitively true if the system state at that certain point satisfies the formula.

Definition 3 (*Satisfaction On System States*)**.**

$$\begin{aligned}
&\Gamma(s,t) \models^{s,t} \mathtt{tt} && \forall t, s,\\
&\Gamma(s,t) \models^{s,t} m_P = a && \text{iff } \Gamma(s,t).m_P = a,\\
&\Gamma(s,t) \models^{s,t} x_P \geq c && \text{iff } \Gamma(s,t).x_P \geq c,\\
&\Gamma(s,t) \models^{s,t} \neg\Psi && \text{iff } \Gamma(s,t) \not\models^{s,t} \Psi,\\
&\Gamma(s,t) \models^{s,t} \Psi_1 \wedge \Psi_2 && \text{iff } \Gamma(s,t) \models^{s,t} \Psi_1 \wedge \Gamma(s,t) \models^{s,t} \Psi_2,\\
&\Gamma(s,t) \models^{s,t} \Psi_1 \mathcal{U}^{[T_1,T_2]} \Psi_2 && \text{iff } \exists \tau \in [t+T_1, t+T_2]:\\
& && \Gamma(s,\tau) \models^{s,t} \Psi_2 \wedge (\forall \tau' \in [t,\tau]: \; \Gamma(s,\tau') \models^{s,t} \Psi_1).
\end{aligned}$$

For the STL until operator $\Psi_1 \mathcal{U}^{[T_1,T_2]} \Psi_2$ and a system state $\Gamma(s,t)$, we have to check for a fixed value of s and starting time point t, whether the evolution of the system is such that a time point τ exists at which Ψ_2 holds and before which Ψ_1 holds continuously. Recall that, for the system state $\Gamma(s,t)$ and a fixed sample s, the evolution over time is deterministic and coincides with a vertical line in the STD, starting at point (s,t). Hence, the analysis of the STL until operator for a given

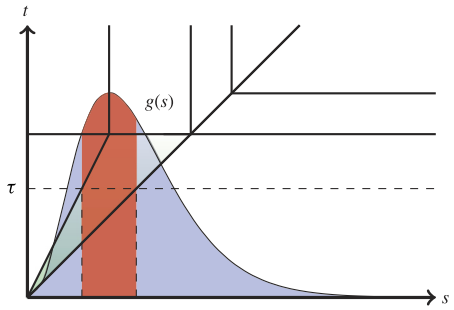


Fig. 4. Deconditioning according to the probability density function $g(s)$.

system state requires us to check whether this line only intersects with regions where Ψ_1 holds until a region is hit where Ψ_2 holds within the defined time interval. For details, we refer to [10].

Next we introduce a satisfaction relation $\models^t$ for intervals on the support of the distribution of the general transition, denoted $I_\psi \subseteq \mathbb{R}_{\geq 0}$, and STL formula Ψ, at time t. This allows for a more efficient model checking procedure than checking each system state individually. The different indices on the satisfaction relations are used to stress their dependencies on s and t.

Definition 4 (*Satisfaction On Intervals*)**.**

$$I_\psi \models^t \Psi \quad \text{iff } \forall s \in I_\psi : \ \Gamma(s,t) \models^{s,t} \Psi.$$

Definition 5. The set of satisfaction intervals $Sat^t(\Psi)$ is defined as the set of all intervals satisfying Ψ at time t, i.e., $Sat^t(\Psi) = \{I_\psi : I_\psi \models^t \Psi\}$.

While the explicit dependency on s (or sets of s-values) is used for the efficient computation of properties, in the end we want to know whether a given STL formula holds at time t for the HPnG model of interest with a certain probability. Hence, we introduce a probability operator $\mathbb{P}_{\sim p}(\Psi)$ which is wrapped around an STL formula, where $p \in [0, 1]$ is a real number and $\sim \in \{\leq, <, >, \geq\}$ a comparison operator. It abstracts from the possible values of s by deconditioning with the probability density function $g(s)$, as follows.

Definition 6. Let $\Gamma(t) = \{\Gamma(s,t) | s > 0\}$ be the set of possible system states at time t, then the satisfaction relation for the probability operator $\mathbb{P}_{\sim p}$ is defined as:

$$\Gamma(t) \models \mathbb{P}_{\sim p}(\Psi) \quad \text{iff } Prob(\Psi, t) \sim p,$$

where

$$Prob^t(\Psi) = \sum_{I_\Psi \in Sat^t(\Psi)} \int_{I_\Psi} g(s)ds.$$

Model checking algorithms for the STL logic, involves computational geometry, especially polygon clipping algorithms. A detailed description of the algorithms is given in [10].

3.3. Multiple general transitions: discretization

Recall that the exact computation of measures for HPnGs is only possible for models with a single general one-shot transition, which guarantees the existence of exactly one stochastic variable in the underlying state-space. Additional firings of the same or a different general transition result in additional stochastic variables, which add to the dimensionality of the STD, hence, to the complexity of the analysis algorithms [31]. In fact, the number of regions in the STD will grow exponentially with the number of stochastic variables, hence dealing with such models becomes practically infeasible.

To circumvent this difficulty, we propose to discretize all general transitions, except for one. The idea is to select a set of firing times (discretization points) for each new general transition, and treat them as deterministic transitions for those firing times, and later generate an STD for the first general transition (the one left to be continuous), for each of these discretization points. We elaborate on this for the case of 2 general transitions. More specifically, let S_2 be the real valued random variable representing the firing time of the second general transition, i.e., the time that it will fire relative to its enabling time. Moreover let the set $\mathcal{S}_2$ be a discretization of the support of S_2, on which the probability measure P_{S_2} is defined. Then the probability that a formula Ψ holds at time t can be computed using the law of total probability, as:

$$Prob^t(\Psi) = \sum_{s_i \in \mathcal{S}_2} Prob^t(\Psi | S_2 = s_i) P_{S_2}(S_2 = s_i), \tag{1}$$

where $Prob^t(\Psi|S_2 = s_2)$ is the probability that Ψ is satisfied at time t, conditioned on the second general transition to fire at time s_2. Although for simplicity of notation, we have included only two general transitions; one can handle more than two general transitions by a recursive application of the above equation.

If the second transition follows a discrete probability distribution with bounded support, it is possible to compute $Prob^t(\Psi)$ exactly, since the probability measure $P_{S_2}(S_2 = s_i)$ is already defined. In such case, each element of $\mathcal{S}_2$ represents a possible firing time of the second general transition. However, if the support of the firing time is unbounded, the summation has to be truncated for a certain value $s_2^{\max} \in \mathcal{S}_2$. In case of a continuous probability distribution, we need to construct $\mathcal{S}_2$ by discretizing the support of the firing times of the second general transition. Moreover, we define the probability measure $P_{S_2}(S_2 = s_i)$ as the total mass of probability between two consecutive discretization points:

$$P_{S_2}(S_2 = s_i) = \int_{s_i}^{s_{i+1}} g_{S_2}(s)ds, \tag{2}$$

in which g_{S_2} is the continuous probability distribution of the second general transition. Both truncation and discretization clearly result in an approximate computation of the desired probability.

The discretization can be done in two different ways, namely using a predefined discretization step, or using the Monte Carlo method [32]. For the former we use discretization step Δs_2 to build the set $\mathcal{S}_2$. However, in the Monte Carlo method, we choose the discretization points based on the probability distribution of S_2, i.e., those points with higher probability are more likely to be chosen. This is indeed more reasonable as we are choosing points which are *likely* to contribute more probability to the final probability. We will investigate and compare the effect of these two methods in Section 6.3.2.

Algorithm 1 shows the process of computing the probability given by (1) for the case of two general transitions. As input it receives property Ψ, time point τ for which we want to know whether the property holds, and the support of $\mathcal{S}_2$. Through lines 2–4, we iterate over all possible values of firing times of the second general transition, s_2, and generate an STD for each value of s_2. Function $generateSTD(s_2)$, schematically, describes the process of generating an STD for which it is assumed that the second general transition is going to fire at time s_2. In other words, the second general transition, for this case, can be replaced by a deterministic transition with firing time at s_2. In line 4, the probability of Ψ holding at time τ for the generated STD, is computed through function $Prob^\tau(std, \Psi)$. Finally, this probability is added to the final probability with weight $P_{S_2}(S_2 = s_2)$, according to (1).

Algorithm 1 *ComputeProb*(Ψ, τ, $\mathcal{S}_2$)

Require: Property Ψ, investigation time τ, and domain of second g-trans firing time $\mathcal{S}_2$.
Ensure: Probability of holding formula Ψ
1: $\pi^\tau \leftarrow 0$
2: **for all** s_2 in $\mathcal{S}_2$ **do**
3: $\quad std \leftarrow generateSTD(s_2)$
4: $\quad \pi^\tau \leftarrow \pi^\tau + Prob^\tau(std, \Psi)P(S_2 = s_2)$
5: **return** π^τ

By using this approximate solution we clearly lose the advantage of exactness; however, we are able to overcome the restriction to one general transition for the analysis of HPnGs. Section 6 illustrates how this method can effectively be used for the analysis of a HPnG model with two general transitions.

4. Control example

In this section we discuss a well-known control example, where the amount of fluid in a tank is supposed to stay between certain bounds. Such an example can be modelled using guard arcs. Section 4.1 introduces a failure to the input pump, resulting in a single general transition, and Section 4.2 introduces a failure at the actuator which is responsible for switching between input and demand.

4.1. One general transition

In the example as shown in Fig. 5, a tank denoted P_m with the capacity of 11 litres, is connected to two pumps. Either, the producer pump T_p fills the tank with rate 1 litre/minute, or the consumer pump T_d takes out the fluid with rate 2 litre/minute. For control purposes, the amount of fluid in the reservoir needs to be between 1 and 10 avoiding both underflow and overflow. We also assume that the overall flow from the place P_m, cannot be stopped, i.e., the two pumps T_p and T_d cannot be off at the same time. Two switches can turn the pumps on and off, both with a delay of 2 min, which is modelled by two deterministically timed transitions. Transition T_a, with firing time of 2 is connected to the reservoir via a guard arc with condition $(\geq, 8)$. Hence, when the amount of fluid is greater or equal to 8, it will be enabled and after 2 min it will fire. As a result the pump T_p becomes disabled and pump T_d becomes enabled. Also, transition T_b, with firing time of 2, is connected to the reservoir via a guard arc with condition $(<, 5)$, so when the amount of fluid is smaller than 5, the transition will be enabled and fires after two minutes.

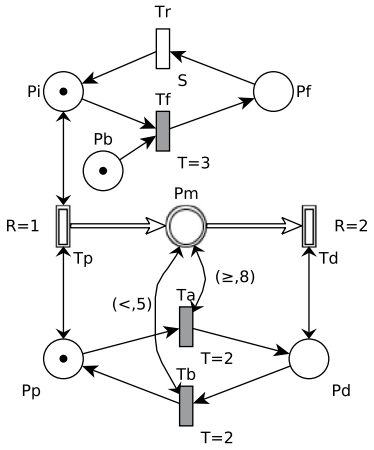


Fig. 5. A simple control example. The amount of fluid in the reservoir P_m is supposed to remain between 1 and 10.

Additionally, the producer pump may fail at different points in time α, which is modelled by the deterministic timed transition T_f, with firing time α. Whenever this transition fires, the general transition T_r becomes enabled. This transition models the repair procedure, which is stochastically distributed according to any arbitrary given probability distribution. Note that the input arc between place P_b and transition T_f ensures that only one failure is possible.

Now suppose, we want to check whether even after a failure of a pumps the fluid level in the reservoir always stays between one and ten. For this purpose we model check the following STL formula at the starting time:

$$\Phi = tt\ \mathcal{U}^{[\alpha,\alpha+T]} \quad (x \leq 1 \vee x \geq 10). \tag{3}$$

In this formula variable x represents the amount of fluid in the reservoir, α is the time of occurrence of the failure, and T is the maximum time of analysis, within which we want to check that a state with less than 1 amount of fluid or more than 10 amounts of fluid is only reached with a very small probability.

Fig. 6 shows the STD of the control example, for the case of $\alpha = 3$. Green regions are representing those regions in which the condition $(x \leq 1 \vee x \geq 10)$ does not hold, and blue regions are the first ones reached after the time boundary $\alpha + T$. It can be seen that for all the possible values of s (x-axis), which represents the general transition firing time, it is impossible to reach a region in which the condition $(x \leq 1 \vee x \geq 10)$ holds within the maximum time T (as the system always remain in green regions). In other words, there is no value of s for which the formula is satisfied. Hence, if we integrate over all possible values of s, as described in Section 3.2, the property holds with probability zero.

4.2. Two general transitions

To illustrate the evaluation procedure for two general transitions, we introduce another stochastic failure, namely at the actuator T_b that is responsible for turning the demand off and the input pump on, when there is less than 5 litres of water in the reservoir.

This new possible failure is represented by the general one-shot transition T_{af} and is only enabled when there is a token in place P_d, i.e., when the demand pump is working. This models for example a valve that is stuck open. In the model this failure is followed by a deterministic repair after three quarters (0.75) of a minute, and transition T_b is enabled again. For this case we investigate the same STL formula as before which is given in Eq. (3). Moreover, we assume the same deterministic failure time at input, i.e., $\alpha = 3$ (see Fig. 7).

Recall that in order to analyse an HPnG with two general transitions, one of these has to be discretized and for each possible realization of the random variable a new STD has to be computed. This is illustrated in Fig. 8 which shows four different STDs for different firing times of the general transition T_{af}, namely for a realization of the failure 0.5, 1.5, 2.5 and 3.5 min after the demand has been turned on. Note that we start the analysis with 6 litres of water in the tank.

Again regions in which $(x \leq 1 \vee x \geq 10)$ does not hold and that are within the time interval $[\alpha, \alpha + T]$ are outlined in green. Moreover, red regions indicate that the formula $(x \leq 1 \vee x \geq 10)$ holds, before reaching the maximum time, $T = 10$. It is not desirable to be in a red region before the maximum time $T = 10$, because in such regions the control condition is violated. Hence, in practice one would check whether formula Φ holds with a very small probability.

When the actuator T_b fails 0.5 min after the demand has been turned on, i.e., the actuator T_a has fired, and P_m is now decreasing at rate 2, the amount of water in the tank at the time of the failure is 9 litres and since the repair only takes 0.75 min, the failure will be repaired when the amount of water in the tank is 7.5 litres $(9 - 2 \times 0.75)$. Since this is still more than 5 litres which is the condition for switching off the demand and the input back on, the failure has no impact on the system, hence, the probability that Φ holds, given a failure of T_b after 0.5 min, is zero. This can be seen in the corresponding

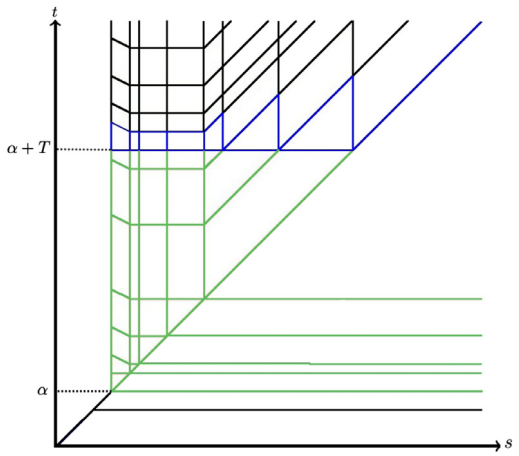


Fig. 6. Stochastic time diagram of control example, for $\alpha = 3$. Regions in which ($x \leq 1 \vee x \geq 10$) does not hold are outlined in green, and regions in which the time boundary $T + \alpha$ is reached, are outlined in blue. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

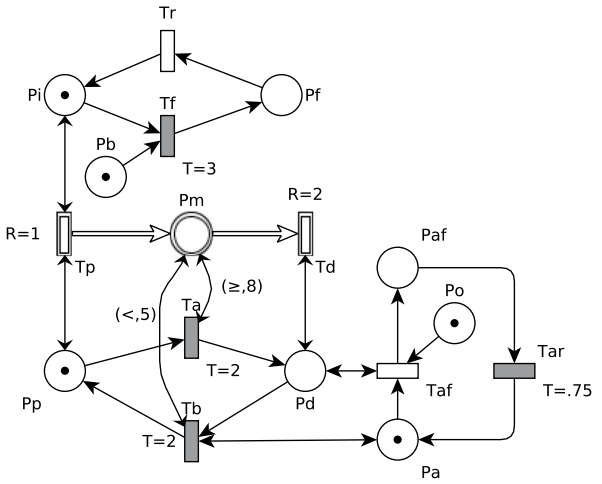


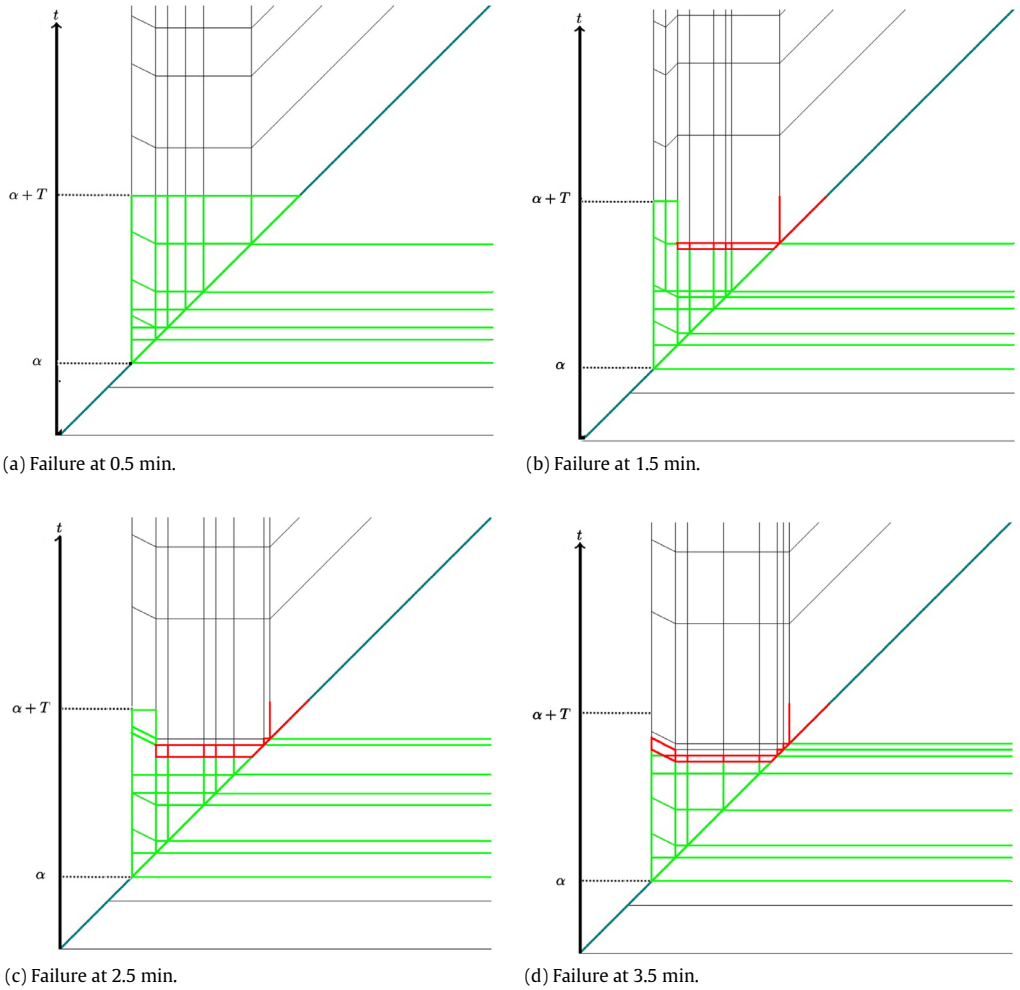
Fig. 7. Control example extended with possible failure in actuator T_b.

STD, as shown in Fig. 8(a), as it is not possible to reach a red region in the described time interval. Note that this scenario is independent of the other failure that switches off the input directly in three minutes after the start of analysis.

However, the impact of a failure that occurs later, i.e., 1.5 min after the demand has been turned on, depends on how quickly the failure at the input is repaired (modelled by general transition T_r). In case the failure at the input has not been repaired the amount of water in the tank is 6 litres at the moment of the failure at T_b. During the repair of T_b the amount of water further decreases to 4.5 litres ($6 - 2 \times 0.75$) and when T_b becomes enabled and fires after 2 min the amount of water has dropped to 0.5 liter ($4.5 - 2 \times 2.0$), which violates the control condition and hence property Φ holds. However, in case the failure at the input is repaired quickly, the amount of water in the tank is larger at the moment T_b fails and this prevents a violation of the control condition. The corresponding STD for failure of the actuator at 1.5 min is shown in Fig. 8(b). One can observe that for small values of the repair time at the input (firing time of T_r), the system reaches the maximum time without hitting red regions, while for larger values does hit the red regions.

A similar reasoning holds for a failure 2.5 min after the demand has been turned on. As shown in Fig. 8(c), again, the right part of the system evolutions, representing a late repair of the input, ends in a red region in the specified time interval.

A failure that takes place 3.5 min after the demand is switched on inevitably leads to a violation of the control condition and Φ holds with probability 1. This means that the system will always violate the control condition no matter how fast the failure at the input is repaired. This is due to the fact that the failure at actuator T_b happens when there is very little water in the tank and the input should be switched on quickly. However, due to the repair delay of 0.75 min, we surely have less than 1 litre water in the tank, once T_b fires. This can be seen in Fig. 8(d), as there is no possibility of reaching the maximum time, $T + \alpha$, without passing a red region.



(a) Failure at 0.5 min.

(b) Failure at 1.5 min.

(c) Failure at 2.5 min.

(d) Failure at 3.5 min.

Fig. 8. Four instances of STD for different firing times of the second general transition T_{af}. Green and red regions demonstrate desirable and undesirable regions according to Formula (3), respectively. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

The shown STDs, illustrate the idea of discretizing the firing times of the second general transition. Once all the STDs for all possible firing times are obtained, the probability that a given formula Φ holds can be computed according to Eq. (1), as described in Algorithm 1. The feasibility of the overall idea for more complex cases will be demonstrated in Section 6.3.

5. Overflow places

In the HPnG formalism as presented before, rate adaptation prevents both overflow and underflow of a reservoir. However, for modelling real systems, especially water treatment and sewage facilities, sometimes we need to allow places to overflow. This was impossible before adding dynamic transitions and guard arcs to the definition of HPnGs. In case of a full continuous place, the overflow is defined as the difference of the actual rate of the inflow and the outflow of that place. This is the reason for adding dynamic transitions to the definition of HPnG. More formally an overflow place is a structure in which when the continuous place reaches its upper boundary a dynamic transition becomes enabled with the rate that equals the difference of the actual rate of all the incoming and outgoing transitions.

Note that the rate adaptation algorithm has no influence on an overflow place. This is because at the moment of reaching the upper boundary the state of the system is changed by the enabling of the dynamic transition. As a result, the drift of the place becomes zero and rate adaptation is not necessary anymore.

As can be seen in Fig. 9, where an overflow place is shown, whenever the continuous place reaches a certain boundary B, the immediate transition connected via the guard arc will fire, and as a result the dynamic transition becomes enabled. Note that the rate of this transition is adapted according to the inflow and outflow of the main continuous place. Also, whenever the fluid level in the continuous place is below the boundary B, due to a change either in inflow or outflow, the connected

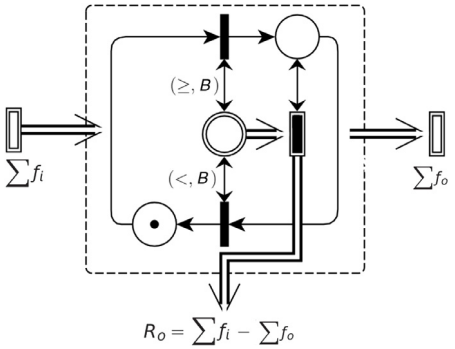


Fig. 9. Modelling of an overflow place using HPnG primitives.

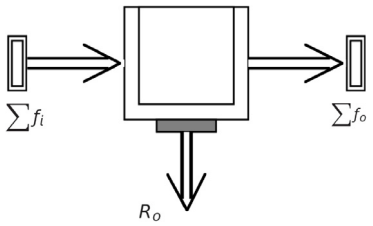


Fig. 10. Overflow place component.

immediate transition will fire and the dynamic transition is disabled. For ease of modelling, we represent an overflow place graphically as follows, in Fig. 10.

In Fig. 10, the sum of inflow and outflow rates are denoted by $\sum f_i$ and $\sum f_o$, respectively. Moreover, the overflow rate is shown by $R_o = \sum f_i - \sum f_o$, and distinguished from ordinary outgoing transitions by a shaded rectangle connected to the overflow place.

6. The Enschede sewage treatment facility

In this section we analyse the survivability of a sewage treatment facility, inspired by the operational facility located in Enschede, the Netherlands (Fig. 1). Section 6.1 introduces the HPnG model for the sewage treatment facility and Section 6.2 analyses the system evolution in the presence of two separate scenarios, namely bad weather conditions and a failure at the main water intake. Section 6.3 then analyses both scenarios in combination by applying the approximate method, as introduced earlier.

6.1. System and model

Recall from the introduction, Section 1, that sewage treatment facilities are by design limited in capacity. In the past this has resulted in flooding the streets. We investigate under which circumstances flooding of the street occurs and which parameters of the system need to be changed in order to prevent this.

This case study models the various stages of the sewage treatment process in an abstract fashion. We are mainly interested in the capacity of each phase and the average amount of time the waste water stays in the different phases. We, however, do not aim at modelling the physical, chemical and biological processes in detail. Then, for a given failure of the system at a certain time, we analyse the survivability of the system for changing weather conditions. Fixing the failure to a specific time of the day results in a so-called Given the Occurrence Of Disaster (GOOD) model. Since our evaluation method is so quick, it is easily possible to parametrize the failure time, hence, analyse the system thoroughly.

The main goal of waste water treatment is to separate the input into water that can be safely released into the environment and into thickened sludge which is either used as fertilizer [33] or can be safely disposed [34]. This is done in several stages, where the primary stage mostly involves physical purification, the secondary stage involves chemical and biological treatment, and finally the sludge treatment phase aims at reducing the amount of sludge.

The HPnG model of the case study is depicted in Fig. 11; volumes of tanks (continuous places) are indicated in 1000 m^3, pump rates (continuous transitions) in 1000 m^3/h, and delays (timed transitions) in hours. The capacity of the community sewerage system is modelled by an overflow place denoted P_c, which has input rates that depend on the weather conditions. From this tank the water is pumped into the treatment facility with a maximum rate 12 and in case the input exceeds the capacity of the place and the intake of the treatment facility, the waste water flows into (overflow) place P_o which models the amount of water in the streets. The primary stage of the sewage treatment consists of two phases, namely the sand

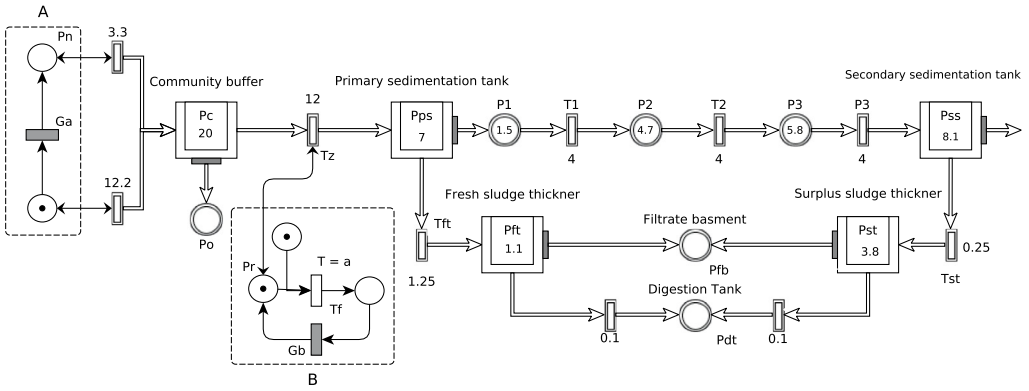


Fig. 11. The abstract HPnG model of the sewage system in Enschede, Netherlands.

interceptor and the primary sedimentation tank. The first, as the naming suggests, is responsible for filtering solids like sand from the water. Then the sewage flows in a large tank, which is used to settle the sludge, while the lighter materials, like oils, rise on the surface and are removed, and the remaining overflows. In the model the sand interceptor is abstracted through pump T_z, and the primary sedimentation tank is modelled through overflow place P_{ps}.

A sedimentation tank physically separates suspended solids from water using gravity [34]. While the dirt settles at the ground, cleaned water is forwarded to the second cleaning stage. This stage consists of several phases for removing chemical and biological contaminations, modelled by a sequence of continuous transitions and places, before a secondary sedimentation tank separates the biological material from the now environment friendly sewage water, that can safely be disposed to surface water. The second sedimentation tank is modelled by overflow place P_{ss}. The sludge that settles at the primary and secondary sedimentation tank is accumulated and forwarded to the sludge treatment stage. There it is thickened to reduce its volume for easier off-site transport. The sludge from the primary tank is pumped out and forwarded to the fresh sludge thickener. This is also modelled by an overflow place, denoted P_{ft}. Sludge is pumped out of the place with a small rate and discharged to the digestion tank which is considered a very large tank. The overflow is directed to the filtrate basement. The same procedure is repeated for the accumulated sludge in the second sedimentation tank.

6.2. Evaluations

In the following, we analyse the model in two different ways, namely, by changing the rate of the produced waste water after a random amount of time, and by introducing a stochastic failure at the sand interceptor T_z, which according to the plant operators is one of the most vulnerable components of the whole process. In Fig. 11, the first scenario is depicted by the dashed box A and the second scenario is shown in box B. Note that we either analyse scenario A or B, but never both at the same time, due to restriction to single general one-shot transition. We analyse the influence of several system parameters on the measures of interest. Also, note that we start the analysis assuming that all tanks in the treatment facility (fluid places and overflow places) are full but the overflow place modelling the community sewage system, P_c, is empty.

Scenario A

For the first scenario, as depicted by box A, we assume that the analysis starts at rainy weather condition, i.e., the production rate of waste water is 12.2, which is slightly more than the capacity of the system, which is 12. Hence, if it continues to rain long enough, the capacity of the community sewage is exceeded and waste water will flood the streets. We assume that the duration of the rain is normally distributed. This is modelled by the general transition G_a, which will fire according to the given normal distribution. When G_a fires, the production sewage will switch to normal weather condition, i.e., a production rate of 3.3.

We would like to analyse how long it may continue to rain without having water in the streets. Using the logic STL, as explained in Section 3.2, we want to ensure that the amount of water in the streets is very low until the rain stops, i.e.,

$$\Phi_A = (x_{P_o} < \epsilon)\ \mathcal{U}^{[0,30]}\ (m_{P_n} = 1), \tag{4}$$

[1]

where $m_{P_n} = 1$ means that the rain has stopped and we are back to the normal conditions. Formula Φ_A is a typical expression of a survivability measure: the first term, before the Until operator is called the *safety condition*, whereas the one after the Until operator is called the *recovery condition*. In other words, as defined in Definition 3, for a specific system evolution, Φ_A is satisfied if and only if the safety condition holds until we reach the recovery condition, before the given time bound. We

[1] For computational purpose, we use ϵ as positive close to zero constant.

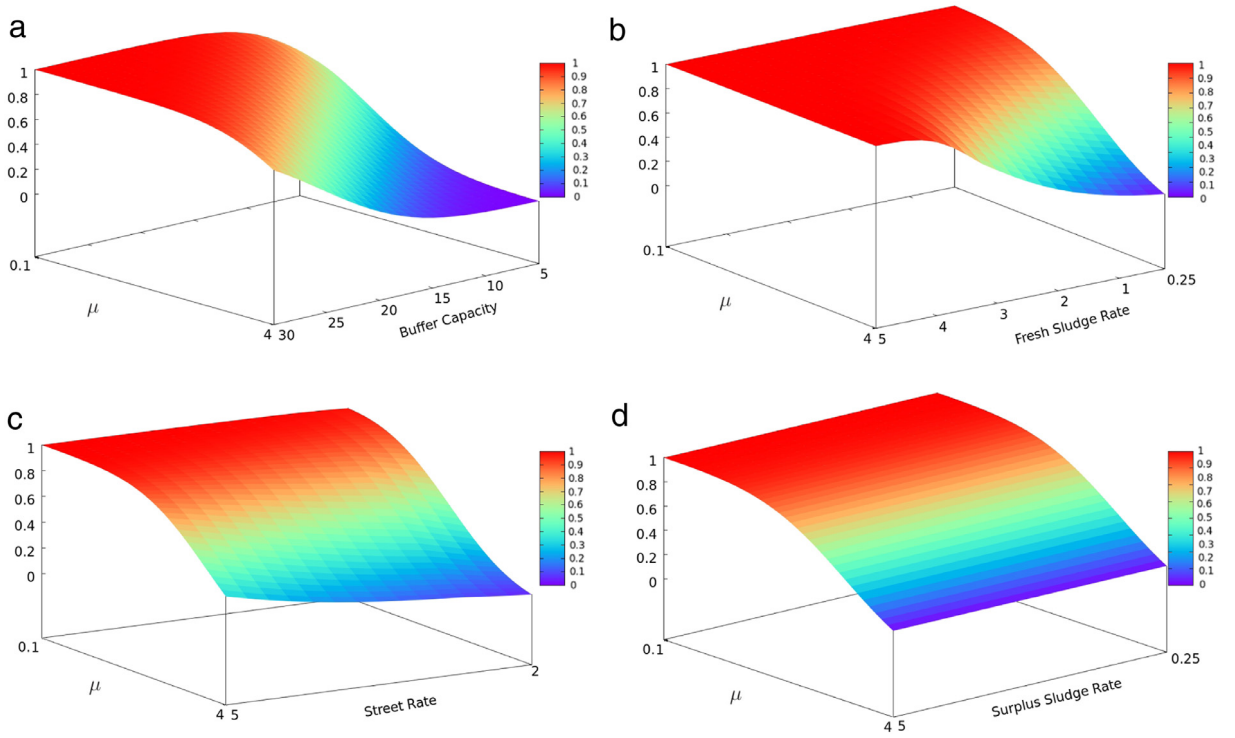


Fig. 12. Probability for the survivability property Φ_A to hold, while varying (a) the capacity of community buffer (P_c), (b) the rate of fresh sludge pump (T_{ft}), (c) the rate of the cleaning street pumps (T_1, T_2 and T_3), and (d) the rate of surplus sludge pump (T_{st}); μ is the mean rain duration, distributed according to a truncated Normal distribution with variance one.

have chosen time bound 30, which is considered to be big enough for this analysis, since it is reasonable that the rain stops within 30 h.

In the following we investigate the influence of four different parameters, all for varying average duration of rain μ; we consider the capacity of the community buffer (P_c), the rate of the fresh sludge pump (T_{ft}), the rate of the cleaning street (T_1, T_2 and T_3) and the rate of the surplus sludge pump (T_{st}). Fig. 12 shows the probability that Φ_A holds for varying mean durations of rain between 0.1 h (6 min) and 4 h. This parameter is the same for all four 3D-plots, and is depicted on the x-axis), while the parameters on the y-axis are different, as mentioned above. All the other characteristics of the model, which are not explicitly parametrized, keep their values according to Fig. 11.

Fig. 12(a) shows the influence of the capacity of the community buffer (P_c), by varying its value from 5 to 30 (from right to left on the y-axis). By increasing this capacity, the probability that formula Φ_A holds increases. We observe that this increase is non-linear, especially for larger values of the capacity, we see a faster improvement. Furthermore, we observe that for long rain duration, even if we increase the buffer capacity to 30, still we have more than 20% probability of not satisfying the survivability property Φ_A. This means that enlarging the buffer capacity alone is not enough for avoiding the flood in the area.

Fig. 12(b) shows how the system survivability depends on the rate of the fresh sludge thickener pump, T_{ft} which is parametrized from 0.25 to 5 (from right to left on the y-axis). It can be seen that, by increasing the rate of this pump, the probability for Φ_A to hold increases. Specially for long rain duration ($\mu = 4$) this increase can be observed well. The increase is steeper than in Fig. 12(a), hence, this pump plays a significant role; for larger values of its rate, e.g., larger than 3, even if it rains for more than four hours, formula Φ_A holds with probability one. The reason for this is that the overflowed sewage from the primary tank, P_{ps} could be handled with a rate of at most 4, the rate of intake into cleaning street. So, the more we pump out of the primary tank, the more sewage intake the system can handle. However, since increasing the rate of this pump means pumping out sludge with more mixed water, this could be a disadvantage or even an obstacle for the next stage, i.e., sludge treatment.

Fig. 12(c) shows the importance of the cleaning street pump rates all together, i.e., pumps T_1, T_2, T_3, of which we vary the rate from 2 to 5 (right to left). These pumps play a similar role as pump T_{ft}, but with lower impact. As can be seen, for long rain duration ($\mu = 4$) the survivability probability remains low. Like in the previous case, also here increasing the pump rates could be problematic, because raising the rates involves pumping out water mixed with more dirt, since there may not have been enough time for the dirt to settle down in the primary sedimentation tank.

Finally, Fig. 12(d), shows the influence of surplus sludge thickener pump T_{st}. As can be seen, the rate of this pump has no effect on the survivability probability of the system. This can be explained by the fact that this pump plays a secondary role

Table 1
Overall computation time for generating results depicted in Fig. 12.

	Number of points	Computation time (ms)
Fig. 12(a)	1000	522
Fig. 12(b)	800	573
Fig. 12(c)	600	269
Fig. 12(d)	800	459

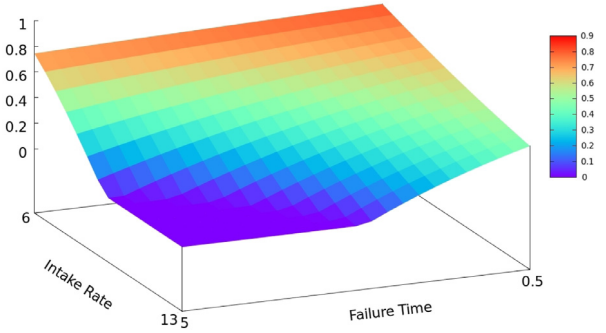


Fig. 13. Probability of survivability property Φ_B to hold for varying intake rate (x-axis) and failure time (y-axis).

comparing to the cleaning street pumps. Since the rates of pumps T_1, T_2, T_3 are constant, increasing any pump rate which is placed *after* them does not change the overall capacity of the system.

In order to generate each diagram in Fig. 12, for each combination of parameters, one STD has to be generated, followed by a model checking procedure. Table 1, shows the number of points in each diagram in Fig. 12 and the overall computation time for producing that diagram, i.e., generating all the STDs and the model checking. As can be seen, even for this big case study, generating and model checking 1000 STDs takes less than a second! This clearly shows the value and efficiency of our method.

Scenario B

For the second scenario, shown by the dashed box B, we consider a failure in the sand interceptor pump, T_z, modelled by the deterministic transition T_f, firing at time α, which again could be parametrized for any arbitrary value. After the occurrence of a failure, a repair crew will repair the pump with a duration distributed according to an exponential distribution, with mean 2 h. For this case we investigate almost the same formula as before, only now the recovery condition is that the pump should be repaired:

$$\Phi_B = (x_{P_0} < 0.01)\ \mathcal{U}^{[\alpha,\alpha+30]}\ (m_{P_r} = 1), \tag{5}$$

where, $m_{P_r} = 1$, means that the sand interceptor pump is repaired. Here, we have chosen the time bound $[\alpha, \alpha + 30]$ for the Until operator, since the pump is supposed to be repaired within 30 h after its failure.

For this scenario, we consider two parameters, the time of failure and the intake rate. The result is shown in Fig. 13. On the x-axis the intake rate is parametrized from 6 to 13, and the y-axis represents different times of failure, from 30 min to 5 h (right to left). As expected, for larger rates of the intake, the probability for survivability property Φ_B to hold decreases. However, it is interesting that for a late occurrence of failure, the probability is lower, especially for high intake rates. The reason for this is that the capacity of the system is equal to the sum of the cleaning street rate pumps (4) and the fresh sludge thickener pump rate (1.25), which is 5.25. Therefore, the buffer is filling up for intake rates greater than 5.25, and a late failure will cause a quicker violation of the safety condition. On the other hand, for early failures, we have a non-zero survivability probability, even for high intake rates.

Fig. 14 provides a better understanding of this case. Each curve in this figure represents the survivability probability (y-axis) for a given fixed intake rate (colour) to the system; the horizontal axis depicts the failure times. The time that the probability hits zero is the very moment that the community buffer has become full, hence, if the failure occurs at any time after that, the surrounding area will be flooded immediately. This is the reason that this probability equals zero for any time of failure after this point.

The two last figures show the importance of fast maintenance in bad weather conditions, otherwise, soon after the occurrence of a failure, flooding of the surrounding area is inevitable.

In addition to these results, the model checking algorithms can compute the set of satisfaction intervals, as explained in [10]. For the current case-study these satisfaction intervals represent the time intervals in which the repair can be done without violating the STL property Φ_B. These intervals arise naturally from the model checking procedure and are used to compute the probability for a certain STL property to hold, as explained in Definition 6.

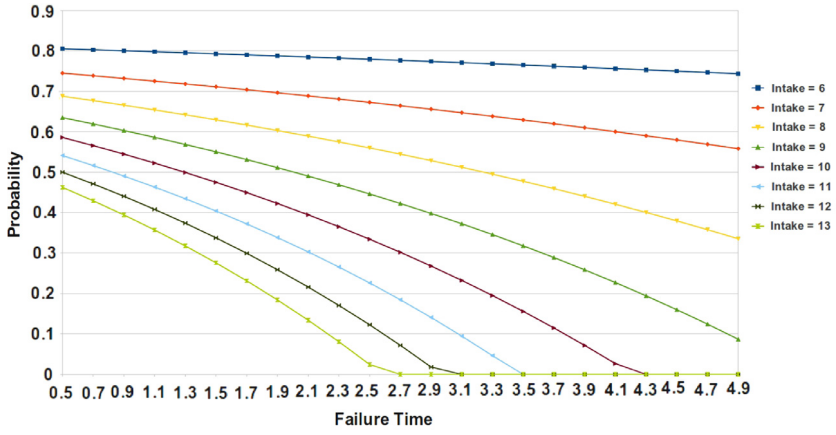


Fig. 14. Probability for survivability property Φ_B to hold. Each curve represents a specific intake rate, and the horizontal axis depicts the failure occurrence time. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 2
Satisfaction intervals for different failure occurrence times α and intake rate f_i.

α	f_i							
	6	7	8	9	10	11	12	13
1	[0, 3.210]	[0, 2.609]	[0, 2.158]	[0, 1.807]	[0, 1.526]	[0, 1.296]	[0, 1.105]	[0, 0.943]
2	[0, 3.085]	[0, 2.359]	[0, 1.814]	[0, 1.390]	[0, 1.051]	[0, 0.774]	[0, 0.542]	[0, 0.347]
3	[0, 2.960]	[0, 2.109]	[0, 1.470]	[0, 0.973]	[0, 0.576]	[0, 0.251]	Ø	Ø
4	[0, 2.835]	[0, 1.859]	[0, 1.126]	[0, 0.557]	[0, 0.101]	Ø	Ø	Ø
5	[0, 2.710]	[0, 1.609]	[0, 0.782]	[0, 0.140]	Ø	Ø	Ø	Ø

Table 2 shows these intervals for different intake rates, f_i, and failure times, α. Note that these intervals are given relative to the failure time α. This means, for instance, if there is a failure in system at $\alpha = 3$, while the intake rate is 7, the repair crew will have 2.109 h to repair the pump, otherwise the safety condition in formula Φ_B is violated and the water is spilt on the street. Moreover, as can be seen, as the intake rate increases the repair crew will have less time to conduct the repair, especially for later failure times, if at all. This knowledge is specifically useful for scheduling repair crews.

6.3. Multiple general transitions

In this section we exploit the idea of discretization for the analysis of HPnG models with more than one general transitions to analyse a setting where both scenarios, namely A and B, are present simultaneously. Firstly, we consider an overall investigation of the system to show the feasibility and efficiency of the proposed method, and secondly we will discuss discretization of each general transition separately to provide an insight on how we can decide discretization of which one of the general transition leads to better performance. Moreover, as we mentioned in Section 3.3 we will investigate performance of different methods of discretization, namely naive discretization and Monte Carlo method.

6.3.1. Feasibility

For this part, we start the system analysis in bad weather conditions (sewage production with rate 12.2) and the sand interceptor pump, T_z, fails at a certain point in time α, followed by a stochastic repair. Again, we want to know the probability that Φ_A (Eq. (4)) holds, since we are interested to not have sewage water in the streets during bad weather.

For this case study, we discretize the general transition G_b which models the stochastic repair process, with discretization step 0.1. We investigate the influence of two parameters, i.e., the mean duration of rain, and the failure time of the sand interceptor which is modelled by the firing of the deterministic transition T_f. Again we assume that the duration of the rain is distributed according to the normal distribution with parameter $\sigma = 1$ and μ between 0.1 and 4, as in Scenario A. The repair process is modelled by a Gamma distribution with parameters, $K = 4$ and $\theta = 1/2$, which means that the repair process has a mean duration of 2 h. The result of the analysis is shown in Fig. 15.

Fig. 15, like scenario B, shows that the later the failure is, the more vulnerable is the system. The reason for this is that the community buffer fills up quickly in case of bad weather, and hence late failures often have to cope with an already full community buffer. Moreover, it can be seen that slightly after 4 h, the failure time has no impact on the probability. This is because at that time the sewage has already slipped into the streets, since the community buffer is already filled by the increased sewage production of rain.

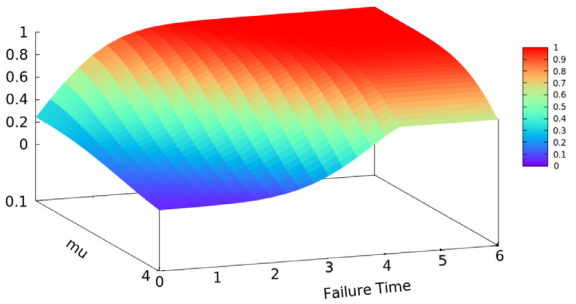
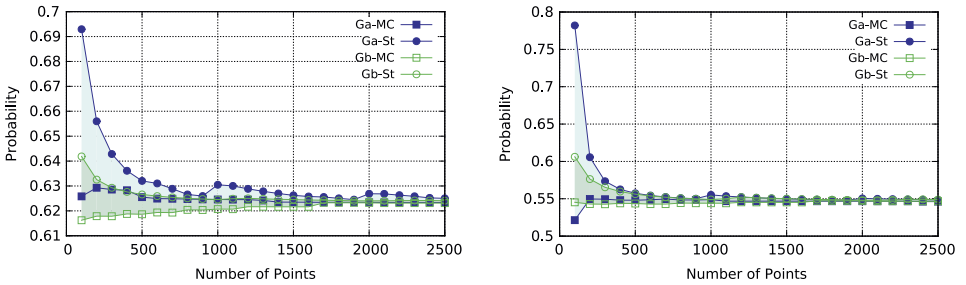


Fig. 15. Probability of survivability property Φ_A to hold for varying mean duration rain μ (x-axis) and failure time (y-axis).



(a) $G_a \sim Gamma(2, 1)$. (b) $G_a \sim Gamma(2^7, 2^{-6})$.

Fig. 16. Comparison of discretization methods, constant steps and Monte Carlo (abbreviated as St and MC in the figures repeatively), by computing the probability that property Φ_A holds.

This diagram contains 960 points, for each of which the discretization of the second general transition is performed. This means that for each point, 400 STDs (discretization step of 0.1 and maximum analysis time of 40) are generated and model checked. Hence, for the generation of this diagram, $960 \times 400 = 38\,000$ times the process of STD generation and model checking is performed. However, the total computation time does not exceed 50 s, on a laptop equipped with a 2.0 GHz intel® CORE™ i7 processor, and 4 GB of RAM. This clearly shows the power of the proposed method.

6.3.2. Discretization methods

In this section we show how using the Monte Carlo (MC) method [32] for discretizing the support of general transitions increases the performance of the proposed method. We also investigate how one can choose one general transition over another, and how the shape of probability distributions influences the convergence rate.

For this section we again start the analysis in bad weather conditions (sewage production with rate 12.2), but with an already failed sand interceptor pump, T_z, i.e., T_f fires at time zero. As in the previous section we compute the probability for Φ_A to hold. For this section we assume that G_a fires according to a Gamma distribution, with parameters defined later, in order to show how the shape of probability distribution influences the convergence rate. Also G_b fires according to a Chi-squared probability distribution with 2 degrees of freedom.[2]

Fig. 16(a) and (b) each shows four different discretizations results. In each figure both G_a and G_b are discretized once with naive steps and once with the MC method. We have used a Gamma distribution with different parameters for G_a, namely $Gamma(2, 1)$ (Fig. 16(a)) and $Gamma(2^7, 2^{-6})$ (Fig. 16(b)), in which the first and second parameters are shape and scale parameters, resulting in the same average of 2 for both distributions. The horizontal axis represents the number of discretization points, and the vertical axis shows the probability that formula Φ_A holds. One can see that in both cases the MC method results in faster convergence to the real value of the probability for both general transitions. However, we can also observe that discretizing G_b performs better in comparison to G_a, as it is closer to the final value, for each number of discretization points. In order to understand this we need to take a closer look at the structure of the discretized area of integration.

As mentioned earlier, when we are discretizing the support of a general transition, say G_a, we treat it as a deterministic transition firing at each discretization point. Then for each of these points, we find the satisfaction interval for the other general transition G_b. According to Eqs. (1) and (2), when we have computed a satisfaction interval for one discretization point, say s_i, we assume that all points between s_i and the next discretization point, s_{i+1}, have the same satisfaction intervals as s_i. This means that we are approximating the final integration area (for computing the probability of a given formula) with

[2] The probability distributions are chosen to show the effect of their shape on the analysis, however, any probability distribution with positive support can replace them.

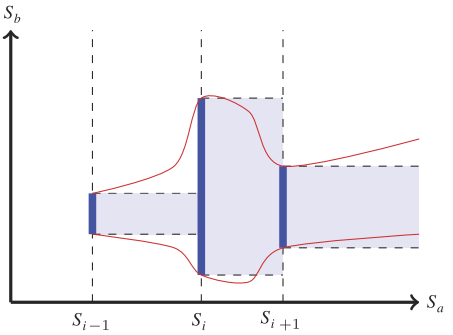
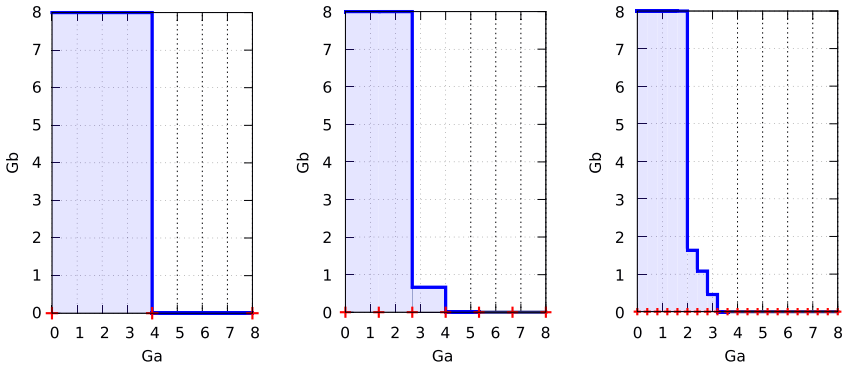


Fig. 17. Showing the approximation of integration area by discretizing S_a. Satisfaction interval for each discretization point is shown by a thick blue line. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)



(a) 10 discretization points. (b) 30 discretization points. (c) 100 discretization points.

Fig. 18. Resulting areas by discretizing support of G_a.

a set of rectangles. This is illustrated schematically in Fig. 17, by showing the approximated rectangular areas (blue) and the real area (red). One can observe that by increasing the number of discretization points we will have a better approximation of the real area.

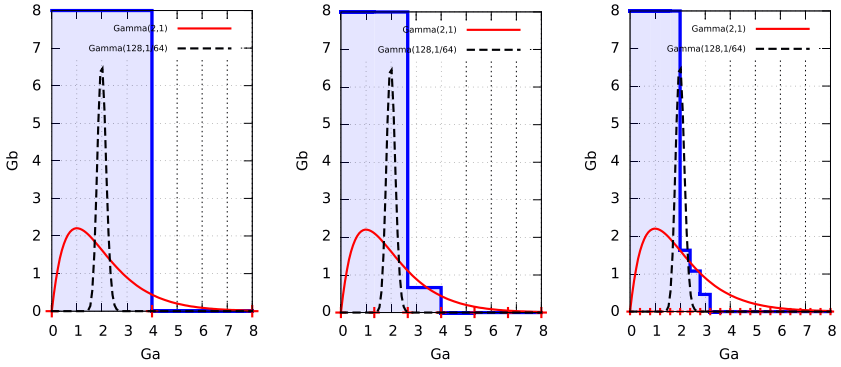
Fig. 18 shows the approximation of integration area, for the case-study, for three different numbers of discretization points of G_a. In each figure the discretization points are indicated by red crosses at the x-axis.[3] For instance, in Fig. 18(a) at point 0, i.e., G_a firing at time 0, the satisfaction interval is the entire support of G_b, while at the next points, i.e., 4 and above, the satisfaction intervals are empty. As can be seen by increasing the number of discretization points in Fig. 18(b) and (c), the resulting area shrinks to a more precise approximation of the real area. Moreover, one can observe that when G_a fires after, approximately, time point 3, the satisfaction intervals of G_b are empty, and G_a has a smaller area of influence in comparison to G_b. This is the reason that discretizations of G_a for small numbers of discretization points result in poor performance, since we need more points, i.e., a small discretization step, to observe this small area.

Knowing the above fact, we can conclude that if the probability distribution associated with G_a takes most of its mass before time point 3, the convergence is fast. This is the case when we use $Gamma(2^7, 2^{-6})$ (Fig. 16(b)). On the other hand, if we use $Gamma(2, 1)$, since considerable amount of probability concentrates after time point 3, the rate of convergence is slow (Fig. 16(a)). However, since the MC method chooses points before time 3 with higher probability, because of the shape of the density function $Gamma(2, 1)$, it results in a better convergence rate. Fig. 19 illustrates these facts by showing both approximated integration areas (as have already been computed and shown in Fig. 18) plus the considered probability density functions $Gamma(2, 1)$, and $Gamma(2^7, 2^{-6})$.

This provides us with useful insight in order to decide which transition to choose for discretization, prior to the start of analysis. For this, one can generate the approximated integration area with few discretization points as in Fig. 18. Then based on the structure of the area and the shape of the associated probability distributions, choose which general transition to discretize. This is indeed why, in Section 6.3.1, we have chosen G_b for discretization.

Finally, it is important to note that the MC method will not always lead to better performance. This can be observed in Fig. 16(a), for discretization of G_b, since the MC method does not show an obvious increase in performance. The MC method guarantees that we are maximizing the term $P_{S_2}(S_2 = s_i)$ in Eq. (1), however, the other term, i.e., $Prob^t(\Psi | S_2 = s_i)$, depends on the resulting satisfaction intervals by fixing firing time S_2 for the second general transition. Fig. 20, is schematically

[3] The maximum time of analysis for this case is 40, however, we are showing the range of general transitions up to 8, for sake of visibility.



(a) 10 discretization points. (b) 30 discretization points. (c) 100 discretization points.

Fig. 19. Demonstrating how the shape of the probability density function and the number of discretization points can influence on the performance of the method.

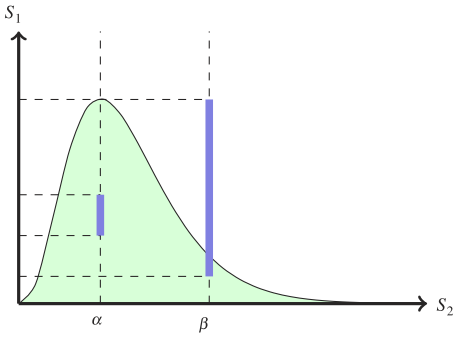


Fig. 20. Showing the satisfaction intervals for two different values of S_2.

demonstrating this fact by showing the corresponding satisfaction intervals of S_1 by fixing S_2 to two values of firing times. As can be seen, although for firing time $S_2 = \alpha$ we have the highest probability (according to the depicted probability distribution of the second general transition), the resulting satisfaction interval is smaller compared to the satisfaction interval of firing of second general transition at $S_2 = \beta$. This may lead to lower total probability (in the deconditioning) for β in spite of being the larger interval.

7. Conclusions

This paper evaluates the survivability of a waste water sewage facility using Hybrid Petri nets with one or two general one-shot transitions. We investigate how the system behaves in very bad weather conditions and in the presence of a failure at the main water intake.

In order to capture the behaviour of a sewage facility, two new concepts have been added to the modelling formalism, namely guard arcs and continuous dynamic transitions. These concepts allow us to express the dependency of the system evolution on the amount of fluid in a continuous place and on the rates of continuous transitions, respectively. Together, this allows us to model so-called overflow places that can be used to model sedimentation tanks as overflow places, where the clean water is separated from the sludge by flowing over the boundaries of the tank.

Using the underlying stochastic time diagram and recent algorithms for model checking the logic STL, it is possible to analyse the survivability of the system for two different scenarios and a wide range of parameter settings. We were able to estimate the entire capacity and performance of the system, for different intake rates. Moreover, we evaluated the importance of several components of the system, and provided suggestions for tuning their characteristics. Furthermore, we provide the set of intervals, for a given survivability formula, where the system can be repaired without violating the survivability requirements. This allows to schedule optimal repair strategies.

Moreover, we introduced an approximate method to circumvent the restriction of having only one general transition in the system. By conditioning the system evolution on all possible values of all but one general transition, all corresponding state space representations and the corresponding conditioned probability that a certain property holds have to be computed. Then, in a deconditioning step the overall probability can be computed by the law of total probability. For discrete probability distributions with bounded support this can be done exactly. However, in case of a continuous distribution and/or an unbounded support, discretization and/or truncation is inevitable.

We discussed and compared the performance of two methods of discretization of general transitions, namely constructing the discretization set using either naive constant steps or based on the Monte Carlo method. We showed how the later can lead to better performance. We also investigated how one can decide to choose a general transition over another for discretization, based on the structure of the integration area and shape of corresponding probability distributions.

This new method then allows to investigate the combination of the two scenarios mentioned above, i.e., a failure at the main water intake which takes place in very bad weather conditions. Our results indicate that it is very important to maintain the system especially before bad weather conditions. This is due to the fact that the time to repair becomes very short for failures that manifest late during bad weather conditions.

Finally, this real world case study, clearly shows the strength of HPnGs for both expressiveness and efficiency of computations. The combination of the two scenarios also shows the feasibility of the proposed approximate method for handling multiple general transitions.

References

[1] RTV Oost. Overijssel Vandaag, July 2013. URL: http://www.rtvoost.nl/tv/uitzendinggemist.aspx?uid=290892.
[2] TV Enschede FM. TV Enschede Nieuws, June 2013. URL: http://www.youtube.com/watch?v=DRIB6JTNvhA.
[3] UT Nieuws. Wanneer kun je kanon op de Auke Vleerstraat? July 2013. URL: http://www.utnieuws.nl/studenten/wanneer-kun-je-kanoen-op-de-auke-vleerstraat.
[4] M. Gribaudo, A. Remke, Hybrid Petri nets with general one-shot transitions for dependability evaluation of fluid critical infrastructures, in: 2010 IEEE 12th International Symposium on High Assurance Systems Engineering, IEEE CS Press, 2010, pp. 84–93.
[5] H. Ghasemieh, A. Remke, B. Haverkort, M. Gribaudo, Region-based analysis of hybrid Petri nets with a single general one-shot transition, in: Formal Modeling and Analysis of Timed Systems, in: Lecture Notes in Computer Science, vol. 7595, Springer, Berlin, Heidelberg, 2012, pp. 139–154.
[6] H. Ghasemieh, A. Remke, B.R. Haverkort, Analysis of a sewage treatment facility using hybrid petri nets, in: Proceedings of the 7th International Conference on Performance Evaluation Methodologies and Tools, ValueTools '13, ICST, 2013, pp. 165–174.
[7] L. Cloth, B. Haverkort, Model checking for survivability!, in: Proceedings of the Second International Conference on the Quantitative Evaluation of Systems, 2005, IEEE, 2005, pp. 145–154.
[8] P.E. Heegaard, K.S. Trivedi, Network survivability modeling, Comput. Netw. 53 (8) (2009) 1215–1234.
[9] J.C. Knight, K. Sullivan, On the definition of survivability, Tech. rep, University of Virginia, 2000.
[10] H. Ghasemieh, A. Remke, B. Haverkort, Survivability evaluation of fluid critical infrastructures using hybrid Petri nets, in: 19th IEEE Pacific Rim International Symposium on Dependable Computing, 2013.
[11] Dynamic simulation software for biological wastewater treatment modelling. URL: http://holinger.com/index.php?id=748&L=10&type=98.
[12] M. Faber, M. Stewart, Risk assessment for civil engineering facilities: critical overview and discussion, Reliab. Eng. Syst. Saf. 80 (2) (2003) 173–184.
[13] J. Derco, L. Cernochova, L. Krcho, A. Lalai, Dynamic simulations of waste water treatment plant operation, Chem. Pap. 65 (6) (2011) 813–821.
[14] G. Horton, V. Kulkarni, D. Nicol, K. Trivedi, Fluid stochastic Petri nets: Theory, applications, and solution techniques, European J. Oper. Res. 105 (1) (1998) 184–201.
[15] M. Gribaudo, M. Sereno, A. Bobbio, Fluid stochastic Petri nets: An extended formalism to include non-Markovian models, in: Proceedings 8th International Workshop on Petri Nets and Performance Models, IEEE-CS Press, 1999.
[16] E. Asarin, O. Maler, Reachability analysis of dynamical systems having piecewise-constant derivatives, Theoret. Comput. Sci. 138 (1) (1995) 35–65.
[17] M. Davis, Piecewise-deterministic Markov processes: A general class of non- diffusion stochastic models, J. R. Stat. Soc. 46 (3) (1984) 353–388.
[18] S. Strubbe, A. van der Schaft, Compositional modeling of stochastic hybrid systems, Control Engrg. Ser. 24 (2006) 47–78.
[19] G. Pola, M. Bujorianu, J. Lygeros, M.D. Benedetto, Stochastic hybrid models: An overview, in: Int. Conf. on Intelligent Control Systems and Signal Processing, (IFAC'03), Elsevier, 2003, pp. 45–50.
[20] H. Alla, R. David, Continuous and hybrid Petri nets, J. Syst. Circuits Comput. 8 (1) (1998) 159–188.
[21] R. David, H. Alla, On hybrid Petri nets, J. Discret. Event Dyn. Syst. Theory and Appl. 11 (2001) 9–40.
[22] T. Henzinger, The theory of hybrid automata, in: Proceedings Eleventh Annual IEEE Symposium on Logic in Computer Science, IEEE, 1996, pp. 278–292.
[23] R. Alur, C. Courcoubetis, T. Henzinger, Hybrid automata: An algorithmic approach to the specification and verification of hybrid systems, Hybrid Syst. 736 (1993) 209–229.
[24] R. Alur, D. Dill, A theory of timed automata, Theor. comput. Sci. 126 (2) (1994) 183–235.
[25] E. Vicario, Static analysis and dynamic steering of time-dependent systems, IEEE Trans. Softw. Eng. 27 (8) (2001) 728–748.
[26] E. Vicario, L. Sassoli, L. Carnevali, Using stochastic state classes in quantitative evaluation of dense-time reactive systems, IEEE Trans. Softw. Eng. 35 (5) (2009) 703–719.
[27] M. Gribaudo, A. Remke, Hybrid Petri nets with general one-shot transitions: model evolution, Tech. rep, University of Twente, 2010.
[28] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, G. Franceschinis, Modelling with Generalized Stochastic Petri Nets, first ed., John Wiley & Sons, Inc., 1994.
[29] R. David, H. Alla, Discrete, Continuous, and Hybrid Petri Nets, second ed., Springer, Berlin, Heidelberg, 2010.
[30] M. Ajmone Marsan, G. Balbo, G. Conte, S. Donatelli, G. Franceschinis, Modelling with Generalized Stochastic Petri Nets, John Wiley & Sons, Inc., 1995.
[31] H. Ghasemieh, A. Remke, B.R. Haverkort, Hybrid Petri nets with multiple stochastic transition firings, in: 8th Int. Conf. on Performance Evaluation Methodologies and Tools, 2014.
[32] N. Metropolis, S. Ulam, The monte carlo method, J. Amer. Statist. Assoc. 44 (247) (1949) 335–341.
[33] S.M. Khopkar, Environmental Pollution Monitoring and Control, New Age International, 2004.
[34] Primer for municipal wastewater treatment systems. URL: http://www.epa.gov/npdes/pubs/primer.pdf.

Hamed Ghasemieh Since 2011 Hamed Ghasemieh is a Ph.D. candidate in the group Design and Analysis of Communication Systems of University of Twente. He is graduated in 2011 from the M.Sc. program in mathematical department of Sharif university of technology, Iran. In 2007 he has received his B.Sc. in software engineering from University of Tehran.

Anne Remke Since October 2014 Anne Remke is professor for computer science at the WWU Münster. She still holds a position at the Design and Analysis of Communication Systems group, at the University of Twente, The Netherlands, where she worked as assistant professor, before. She holds a Ph.D. degree (2008) from the University of Twente and a M.Sc degree (2004) from the RWTH Aachen, both in Computer Science.

As a researcher, her focus is on dependability in critical 7 × 24 infrastructures, such as electrical power systems and their infrastructure and telecommunication and ICT infrastructures (i.e. SCADA). Her interest is currently focused on the dependability analysis of water treatment facilities in the context of her NWO Veni project on 'Dependability Evaluation of fluid critical infrastructures with hybrid stochastic models'.

Boudewijn R. Haverkort (Fellow IEEE) received his M.Sc. and Ph.D. in Computer Science from the University of Twente, the Netherlands, in 1986 and 1991 respectively. He has been a lecturer at the University of Twente (1991–1995), before he was appointed professor for Distributed Systems at the RWTH Aachen, Germany (1995–2002). In 2003 he was appointed as full professor for Design and Analysis of Communication Systems at the University of Twente. In the period 2008–2009 he was chairman of the Department of Computer Science. From March 2009 through December 2012, he was scientific director and chairman of the Embedded Systems Institute in Eindhoven, the Netherlands, a public–private partnership for applied research on embedded systems engineering. Boudewijn Haverkort has written well over 150 papers on performance and dependability modelling and evaluation in computer and communication systems. He also published a textbook on this topic, as well as a number of conference proceedings. He has been host (general chair and PC chair) of a large number of international conferences. He is a Fellow of the IEEE, and member of the ACM and GI.