Faculty of Informatics
Masaryk University
Czech Republic

Stochastic Real-Time Systems: Parameter Synthesis and Games

Habilitation Thesis
(Collection of Articles)

Vojtěch Řehák

November 2018

Abstract

This thesis surveys the author's contributions to the area of models for stochastic real-time systems. Two fundamental concepts meet in these models — probability and real-time. The probabilistic behavior is here deeply connected with time in (continuous) probability distributions that are used to specify random waiting times in states of a model. Moreover, we study not only the analysis of such models but also the synthesis of some unspecified parameters. Hence, we extend the models with the third concept — nondeterminism. In other words, instead of checking whether the model is correct (i.e., satisfies a given specification), we are computing particular model parameters such that the final model with these parameters is (nearly) optimal. Some of the results are delivered for game extensions where part of the nondeterminism is solved by synthesis and the remaining part is considered to be driven by an antagonistic opponent.

The thesis is structured as a collection of ten conference papers and one workshop paper, and an accompanying commentary. The commentary aims to highlight the most important results and to explain the "research flow" with the significant connections between the results. The contribution of the thesis author to the particular papers in the collection is expressed in the list of the included papers at the end of the commentary part.

Abstrakt

Habilitační práce přehledně popisuje autorův vědecký přínos v oblasti modelů pravděpodobnostních systémů se spojitým časem. V těchto modelech se setkávají dva základní modelovací koncepty — pravděpodobnost a spojitý čas. V našem případě jsou tyto dva koncepty velmi úzce propojeny, protože uvažujeme systémy, kde pravděpodobnost je definována na časech setrvání v jednotlivých stavech modelu. Navíc tyto modely pouze neanalyzujeme, ale naším cílem je i syntetizovat parametry, které nejsou přesně určeny. Tím se v modelech vyskytuje i třetí koncept — nedeterminismus. Jinými slovy, namísto kontroly, zda je model správný (tj. splňuje danou specifikaci), vypočítáváme konkrétní hodnoty pro neurčené parametry modelu tak, aby výsledný model byl (téměř) optimální. Některé výsledky přinášíme pro herní rozšíření, kde část neurčených parametrů je řešena syntézou optimálních hodnot a zbývající část je mimo naši kontrolu. V takovém případě se tradičně uvažuje nejhorší možný případ, kdy si představujeme, že zbývající parametry nastavuje nepřítel, který se nám snaží uškodit.

Práce je strukturována jako komentář a soubor deseti konferenčních článků a jednoho seminárního článku. Cílem komentáře je zdůraznit nejdůležitější výsledky a vysvětlit postup výzkumných prací s případnými vazbami mezi jednotlivými výsledky. Přínos autora habilitační práce k dosažení prezentovaných výsledků je vyjádřen v seznamu přiložených článků na konci komentáře.

Acknowledgments

First of all, I would like to thank all my colleagues for constant willingness to discuss any problem. I really appreciate fruitful and long-lasting collaboration, where the main focus is concentrated on obtaining correct and valuable results. Special thanks go to co-authors of the papers included in this collection, namely to Christel Baier, Tomáš Brázdil, Clemens Dubslaff, Adrian Farmadin, Holger Hermanns, Ľuboš Korenčiak, Jan Krčál, Jan Křetínský, Antonín Kučera, and Petr Novotný. Once again I thank Ľuboš Korenčiak for reading a draft of this thesis.

Last but not least, I thank my wife Kateřina, daughter Barbora, and all the family for their constant support, tolerance, infinite patience, and love.

Contents

Part I

Commentary

Chapter 1

Introduction

For successful application of up-to-date mathematical techniques to a real-life problem, we usually need to build a mathematical model of the reality. On this model, the mathematical computation is proceeded, and the obtained results are consequently interpreted back in the original setting of the real-life problem. To acquire reasonable outcomes, we need to work with mathematical models that reflect the crucial parameters of the modeled reality. Here, we are facing the essential problem of modeling. On the one hand, the more accurate (closer to reality) mathematical model we are using, the more relevant results we obtain. On the other hand, the more accurate models we produce, the more demanding computations we need to apply. Hence, the goal is to balance between the traps lurking in the extremes — the too imprecise models that lead to easily achievable results with low practical relevance, and the too complex models that yield precise but hardly computable results.

In this thesis, we focus on stochastic real-time models. The probabilistic modeling is useful when we have some quantified uncertainty about the modeled system. Usually, the probabilistic models are used as a compact representation of a large system, the assigned probabilities correspond to expected frequencies of particular types of behavior. E.g., instead of modeling millions of customers with individual requests, we build one probabilistic request generator. Lots of systems also operate in real-time and thus have to work properly under various time constrains. E.g., an industrial manufacturing robot works in an uncertain environment (possible delays in the preceding processes) with a varying time of its own processes (due to changes in the temperature or the material quality) but still it has to satisfy certain conditions required by the subsequent product line robots. Typical models for such systems studied in the computer science are timed automata extended with a stochastic behavior, or stochastic processes supplemented with real-time constraints. In some models, the probabilistic and real-time aspects are structurally separated, e.g., the probabilities are assigned to transitions leading to subsequent states but the time aspect of the behavior in the particular states of the model is fully deterministic (i.e., non-stochastic). We consider systems where the timing is stochastic. In more detail, there are probability distributions on the waiting times in the model states. We are especially interested in models where both discrete and continuous distributions on waiting times occur.

Another feature that is often present in real-word examples is nondeterminism, i.e., uncertainty without any statistical information. Nondeterminism also naturally occures when the aim is to synthesize an efficient controller of a system. Nondeterminism in stochastic systems is studied in Markov decision processes or in stochastic games. Stochastic real-time systems with nondeterminism are modeled by stochastic games on timed automata or continuous-time stochastic games with timed-automata objectives.

1.1 Motivation Example

Even if it is not usual, let us start with my personal experience that truly motivated my research in this field of stochastic real-time systems. A few years ago, one of my colleagues (working on a research position in a multinational company) brought up a simple task to be solved.

Example 1.1.1 *Let us have an unreliable communication between an air traffic control center and an aircraft. A request message is sent to the aircraft, and the air traffic control center waits for the corresponding response. As the communication is unreliable, there has to be a timeout after which the message is considered to be lost, and subsequent action has to be taken by the air traffic control center. The crucial task for a communication protocol designer is to find the best delay time for the timeout when taking into account all technical parameters of the communication, namely a probability distribution on the response time.*

One would expect that this problem can be more or less easily solved for a given particular instance when all the system parameters are known and given. Unfortunately, the company did not want to disclose their data and other details of the model. They were interested in a tool solving all such questions. Therefore, we reformulated this specific example into a more general computer science problem:

Example 1.1.2 *Let us have a finite-state event-driven system where the events arise after randomly distributed delays. In the system, we allow using both continuous and discrete probability distributions on delays. First, we are interested in the system analysis. Later we would like to algorithmically synthesize (near-)optimal parameters for selected probability distributions concerning a given objective.*

We realized that this had been an interesting open problem that is fairly challenging. We spent around eight years solving it, and this thesis summarizes all the results already achieved during the research.

1.2 Outline

The remaining part of the commentary is divided into four chapters. In the first chapter, all the modeling formalisms used in the collected papers are presented in a unified way. To improve understanding of the relevant results, equivalent or closely relevant formalisms are demonstrated in Chapter 3. In Chapter 4, all the results of the collected papers are readily overviewed. The last chapter lists the citation records of all the papers of the collection and comments my contribution to these papers.

Chapter 2

Preliminaries

The systems we are interested in can be naturally described by modeling formalisms of a large class of event-driven systems. An event-driven system passively waits in one of its states and reacts to events that are emitted by an environment. The first emitted event changes the state of the system, what can retroactively lead to a change of the environment, e.g., when entering the new state, some events of the environment could be enabled or disabled. In our case, we consider the stochastic behavior of events, i.e., for each event, there is a probability distribution specifying the time it needs to be enabled before it is emitted. In what follows, we consider *discrete-state event-driven systems* (DES) that have finitely many states and evolve in stochastic time [CL08]. DES can be described in a large number of formalisms. First, we introduce the formalisms in a context of (Markov) processes that we are using in most of our papers. In the next chapter, we explain how these classes correspond to other related formalisms modeling DES.

2.1 Generalized Semi-Markov Processes

The most general concept of (Markov) processes is a *generalized semi-Markov process* (GSMP) [Mat62] that very precisely corresponds to DES. In GSMP, we have a set of *states* and a set of *events*. To each state, there is assigned a subset of events that are *enabled* in the state. Each event has an *event-time distribution* specifying the probability on time the event needs to be enabled before it occurs. The time evolves in a state until the first event occurs. Note that several events may occur at the same time.[1] Once a set of events E occurs in a state s, the process traverses a *transition* to a subsequent state. The subsequent state is chosen randomly according to a *transition distribution* specifying the probability on the subsequent states. The transition distribution depends only on s and E. Hence, the transition can be drawn as a

[1]This happens with zero probability if the event-time distributions are continuous.

hyperedge that starts from s, is labeled with E, and leads to all states with positive probability in the corresponding transition distribution.

The dynamics of GSMP starts in an initial state s_0 (that is either explicitly specified or given by an initial distribution on states). Immediately, an occurrence time is assigned to each of the events enabled in s_0 according to its event-time distribution. Let t be the minimal time assigned, and E be the set of all events to which the time t was assigned. The process stays in the initial state for time t, i.e., until the occurrence of the first event(s). Then the process moves to a state s' that is chosen randomly according to the transition distribution for s_0 and E. In s', the occurrence times of events are updated in the following way:

- for the *old* events — that were enabled in s but they are not enabled in s' — the times are discarded,

- the *inherited* events — that are enabled in both s and s' (excluding those of E) — remain scheduled to the same time point, i.e., the times to their occurrences are reduced by subtracting t,

- the *new* events — the remaining events enabled in s' — are newly scheduled, i.e., their times to occurrences are freshly chosen according to their event-time distributions.

Now again, the minimal occurrence time and the set of minimal events are selected, and the process goes similarly ahead. We call a *configuration* the state of GSMP supplemented by the times to occurrences assigned to all the enabled events.

To deal with GSMP rigorously, one has to impose some restrictions on the event-time distributions. In what follows, two special types of events are often used — exponential and fixed-delay. *Fixed-delay events* have constant event-time distributions, i.e., they occur after a given constant time with probability one, and so they come in useful when modeling timeouts. *Exponential events* have exponential event-time distributions, hence, due to the memoryless property of exponential distribution [Nor98], they can be newly scheduled after each move (even if they are inherited) without any effect on the model behavior. Now, we can define that a configuration is *regenerative* if and only if the occurrence times for all enabled non-exponential events are newly scheduled, i.e., there is no "truly-inherited" event.

2.2 GSMP Subclasses

In the following, we define some useful subclasses of GSMP.

Markov regenerative processes (MRP) are GSMP where, from each reachable configuration, a regenerative configuration is reached with probability one [Smi55].

Semi-Markov processes (SMP) are GSMP with regenerative configurations only, i.e., all occurrence times of enabled events can be newly scheduled according to the event-time distributions after each move between states [Mat62, LHK01].

Continuous-time Markov chains (CTMC) are GSMP where all event-time distributions are exponential distributions [Nor98].

fixed-delay CTMC (fdCTMC) are GSMP with exponential or fixed-delay events only [KKŘ14].

CTMC with alarms (ACTMC) are GSMP where at most one event with a non-exponential event-time distribution is assigned to each state [BDK+17a].

one-fixed-delay CTMC (1-fdCTMC) are fdCTMC where at most one fixed-delay event is assigned to each state [BKK+15]. Note that this is the common part of fdCTMC and ACTMC.

Discrete-time Markov chains (DTMC) are GSMP where all events are fixed-delay events with the same delay time and each event is enabled in at most one state. Note that any GSMP with only discrete event-time distributions can be equivalently expressed as a DTMC [Nor98].

Hierarchy of the above-mentioned subclasses with respect to their expressive power is depicted in Figure 2.1. For more details see, e.g., [Kor17].

2.3 Performance Measures and Rewards

This section comes up with a short overview of basic properties and measures that we are studying on stochastic systems. For more complex properties, we refer to model-checking results for continuous stochastic logic (CSL) [ASSB00, BHHK03].

There are two types of analysis: transient and long-run. The basic properties of transient analysis are *reachability property* expressing the probability that a given target state is reached and *time-bounded reachability* expressing the probability that it is reached in a given amount of time. In the long-run analysis, the focus is concentrated on the infinite behavior of the

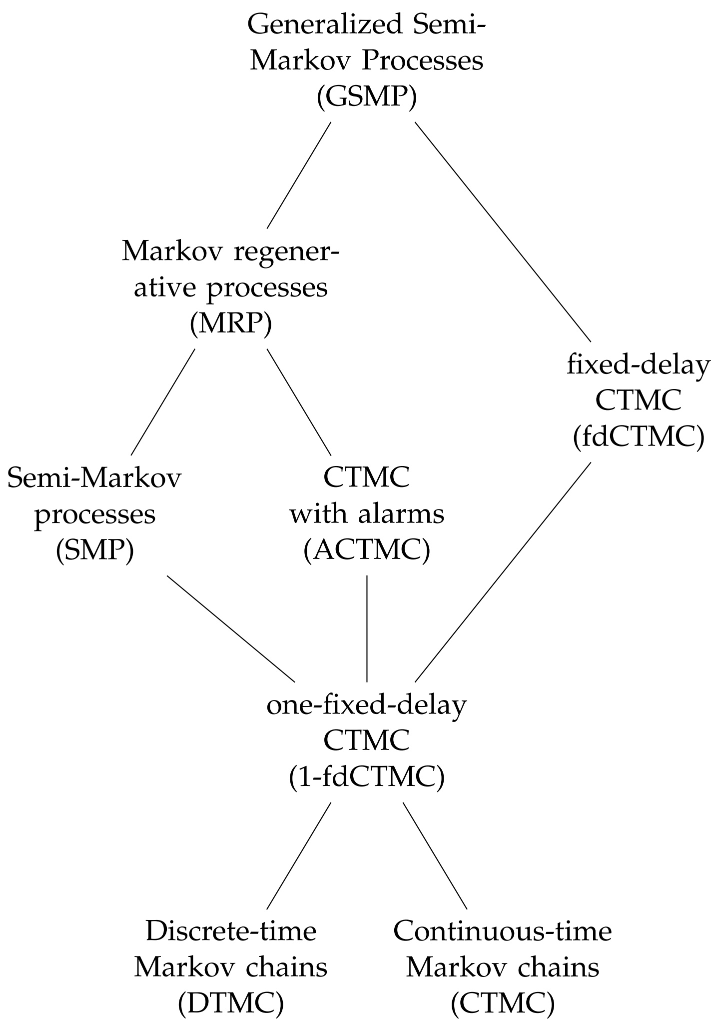


Figure 2.1: Hierarchy of GSMP subclasses with respect to their expressive power.

analyzed system, and the results refer to frequencies of visits to particular states. Technically, for each state s, we can define the discrete frequency of visits $\mathbf{d}_s$ and the timed frequency of visits $\mathbf{c}_s$ by

$$\mathbf{d}_s = \lim_{n\to\infty} \frac{\text{number of visits to } s \text{ in the first } n \text{ steps}}{n}$$

and

$$\mathbf{c}_s = \lim_{t\to\infty} \frac{\text{time spent in } s \text{ up to time } t}{t}.$$

For CTMC these limits are known to be almost-surely well-defined. We show that for some fdCTMC models these frequencies are almost-surely undefined. Such models are called *unstable* and it has no sense to compute the above frequencies for them. More details are explained in Section 4.1.

To obtain a single-number characterization, we use the concept of *rewards* (also called costs when the reward values should be negative). Rewards are values assigned to particular transitions and states. The transition reward is counted whenever the transition is traversed. The state reward is understood as a reward for a time unit; hence, when being counted, they are multiplied by the time spent in the visited state. The transient analysis usually computes the *average reward accumulated before a target state is reached*. Typical measure studied in the long-run analysis is the *expected reward obtained per time unit* on an infinite run.

2.4 Decision Processes and Games

It may also happen that some transition distributions or some event-time distributions are not known. E.g., the events can model reactions with an environment that is out of our control, or the model design is not completed and the remaining distributions are the subject of further construction. In the latter case, the goal is to synthesize the unknown distributions such that they will maximize our profit (depending on what transient or long-run property we are interested in). Contrary, if the unknown part of the model is completely out of our control, we are assuming adversarial control and compute the profit of the worst case scenario. Both of these systems are traditionally called decision processes (or one-player games). When some unspecified parts are under our control and some of them are considered to be adversarial, we apply the game theoretical approach and look for game equilibria.

This context allows for a unified view of the wide range of game modifications of the DES modeling formalisms.

2.4.1 Unspecified Transition Distributions ("where")

Let us have a model where some for some states the transition distributions are unknown. We call such states *decision states*. Note that it would be too powerful to be able to assign an arbitrary transition distribution to a decision state (this will, for example, allow for immediate skip to an arbitrary state of the model). Hence, the transition distributions are not completely unknown in the *decision states*. The are *actions* — candidate transition distributions. Hence, what is unspecified in a decision state is just which of the actions will be chosen. In the game terminology, we are looking for *strategies* — functions that choose a particular action in each decision state. Strategies could be *pure* (when we are choosing one action each time) or *mixed* (when a probabilistic choice on actions is allowed). Strategies could also be *memoryless* (if it is a function of the current state only) or *history-dependent* (if it can depend on all the history of the current execution). Note that strategies depend just on states (not on configurations where the timing of future events is stored) [HHK16].

Now, we can define a *Markov decision process* (MDP) [Kal97, Put94] as a DTMC with decision states. Note that if we commit to some (memoryless) strategy in an MDP, all the transition distributions are specified, and we obtain a DTMC. Our contribution to the synthesis of optimal MDP strategies is described in Section 4.5.

Similarly, we can introduce decision states to CTMC. In a decision state, there are actions instead of events and transition distributions. Whenever a decision state is visited, an action has to be chosen according to a strategy. The chosen action then determines to what state the run goes on. Note that in the decision states there is no time-flow and so the decision is considered to be taken immediately when the decision state is visited. By this we obtain *continuous-time Markov decision process* (CTMDP) [Put94, GHL09][2].

When the decision states of CTMDP are divided between two players, we obtain a *continuous-time stochastic game* (CTG) [Bel57, FV96, BFK$^+$13]. Finally, we define *generalized semi-Markov game* (GSMG) [BKK$^+$10] as a GSMP extended with decision states of two players. Our contribution to solving GSMG is explained in Section 4.2.

2.4.2 Unspecified Event-Time Distributions ("when")

Note that in all above-mentioned approaches the strategies decide in zero time between finitely many options where to go next. Now we would like to discuss situations where the event-time distributions are the subject of the decision. An easy example of such a decision process is the delay time

[2] An alternative definition of CTMDP is that the actions are events and the player chooses what events are active in the decision state. Here, the actions are stochastic transitions to subsequent states, where the events are again active.

synthesis task from our motivation examples of Section 1.1, i.e., we assume GSMP where delay times of some fixed-delay events would be synthesized. Our results solving this problem are discussed in Section 4.4.

A bit more complicated exemplification of unspecified event-time distributions is based on a compositional approach that is natively present in labeled transition systems with synchronization. The unspecified event can be understood as a transition synchronized with an environment. Labeled transition systems are combined with CTMC in *interactive Markov chains* (IMC) [HK09]. IMC is a CTMC where some event-time distributions are not specified, and such events represent labeled transitions waiting for synchronization with an external IMC component running in parallel. Hence, the strategy resolving the unspecified delay-time distributions is the newly constructed external IMC component. We report on our contribution to this field in Section 4.3.

To sum up, in this chapter, we have introduced GSMP, GSMG, IMC, fdCTMC, ACTMC, and MDP, that are all the modeling formalisms we are studying in the collected paper.

Chapter 3

Related Modeling Formalisms

The area of results related to the studied topic is very wide due to a long time of interest and various approaches to modeling and analyzes of probabilistic systems. To show a clear (but definitely not complete) overview of the known results, we introduce other modeling formalisms that are in several contexts used to model DES. We also note how the formalisms relate to the subclasses of GSMP introduced in the previous chapter.

3.1 Queues

The simplest model of the event-driven systems is the classical model of *queues*. The goal of this model is to manage randomly arriving *tasks* that are supposed to be processed by one or more *servers* dedicated to a queue. The tasks are arriving according to an exactly specified probability distribution. When a task arrives, it is stored in the queue and waits to a free server to be assigned to. When assigned, the task is processed by the server. The processing takes a random time. The basic parameters of a queue are specified by *Kendall notation* [Ken53] **A/S/n/B/K** where the particular parameters are:

A – inter-arrival time distribution
(**D** - deterministic, **M** - exponential, **G** - general);

S – service time distribution
(**D** - deterministic, **M** - exponential, **G** - general);

n – number of servers ($\mathbf{1}, \mathbf{2}, \ldots, \infty$);

B – queue size ($\mathbf{1}, \mathbf{2}, \ldots, \infty$), the default value is ∞; and

K – the overall number of tasks ($\mathbf{1}, \mathbf{2}, \ldots, \infty$), the default value is ∞.

For example, **D/M/1/5** identifies a queue with a constant inter-arrival time, say 15 seconds, exponentially distributed service time, say with a rate $\lambda = 0.1$, one server, and five slots for the managed tasks. Hence, every 15 seconds a new task comes. If there is a free space, the task is placed in the queue. If all five slots are occupied, the new-coming task is ignored. The very first task in the queue is being processed by the server what takes a random exponentially-distributed time (the expected service time is 10 seconds, if $\lambda = 0.1$). When the task is done, it is taken out of the queue, and all the tasks waiting in the queue are shifted forward.

Queues can be connected and form a *gueueing network* (QN) where done tasks of a queue can be sent as an input to some subsequent queue(s). Bounded queues are expressible in GSMP with finitely many states, while the unbounded queues require countably many states in GSMP [GSTH08]. Depending on the used distribution types (**D**, **M**, or **G**), the queueing networks are expressible in particular subclasses of GSMP allowing fixed-delay events, exponential events, or general events, respectively.

Thanks to the massive application of QN in practice, even very specific results are of great interest. The most important analysis of queues (as well as QN) are focused on long-run properties like the expected queue length, the utilization factor (portion of the server working time vs. idle time), expected waiting time of a task in the queue, and the probability of ignoring a task (when a bounded queue is full). For more results see, e.g., [GSTH08, BGdMT06, CY01].

3.2 Stochastic Variants of Petri Nets

Petri net (PN) is a well established formalism [Pet62] with a long lasting history and a large number of applications. Formally, Petri net is a bipartite directed multigraph with two type of nodes — *places* and *transitions*[1]. Places represent sources and are usually depicted by circles. Transitions represent operations and are depicted by bars. Each edge connects a place and a transition. Places from where there are edges to a particular transition are called input places of the transition. Similarly, each transition has its output places. The dynamics of a Petri net is modeled by *tokens* that are assigned to the places and moved according to the transitions. The assignments of tokens to places are called *markings*. A Petri net starts in an explicitly defined *initial marking*. A transition, say t, is called *enabled* if each of its input places has at least as many tokes as the number of edges leading from them to t. If a transition is enabled, it can be *fired* what removes tokens from the input places and puts them to the output places (multiplicity of edges corresponds to the number of effected tokens).

[1]Note that the transitions are not edges but special nodes here.

Petri nets are native models of concurrency. When there are more transitions enabled in a marking, the one that fires is chosen nondeterministically. In our context of stochastic-time behavior, the nondeterminism is solved by stochastic times assigned to transitions. Each enabled transition spends a random *heating* time before it is fired. If all the heating times are assigned according to continuous distributions, the nondeterminism is solved with probability one. Depending on the types of assigned heating-time distributions, we obtain different variants of Petri nets.

Stochastic Petri nets (SPN) were independently introduced in [Mol82, Nat80, Sym80] as timed Petri nets where all the heating times are exponentially distributed. It is not surprising that this formalism is equivalent[2] to CTMC. Later, *Generalized stochastic Petri nets* (GSPN) [MCB84] were defined as SPN extended with immediate transitions that are practical when creating a comprehensive model. On the other hand, immediate transitions do not increase the expressive power of SPN, as GSPN was shown to remain equivalent to CTMC [MCB84]. *Deterministic and stochastic Petri nets* (DSPN) [MC86] are GSPN extended with transitions of fixed-delay heating times, hence, they correspond to the fdCTMC class. Omitting all restrictions yields to *extended stochastic Petri nets* (ESPN) [DTGN84] where distributions of arbitrary types are allowed for firing times. ESPN is the counterpart to GSMP. As this class is too general for any reasonable analysis, *Markov regenerative SPN* (MRSPN) [DTGN84] are immediately defined in the same paper as a counterpart to MRP. To complete the list, there are also *Generalized Timed Petri Nets* (GTPN) [Mol85, HV85] where only fixed firing times are allowed. GTPN correspond to GSMP with fixed-delay events only; hence, they closely relate to DTMC.

For more details concerning stochastic Petri net models and their correspondence to Markov processes, we refer to [Mol81, Mar88, Haa02, Krč14, Kor17]. For an up-to-date overview of model-checking results for the class of MRSPN, see [PHV16].

3.3 Stochastic Timed Automata

Another approach to modeling DES is to take the classical *timed automata* (TA) [AD94] and extend them with a stochastic behavior.

Timed automata are finite-state automata extended with special variables called *clocks*. Clocks evolve continuously and synchronously in time. A run of a timed automaton starts in a given *initial state* and all clocks are

[2] Note that in Petri nets, the set of reachable markings could be easily infinite. Hence, to be precise, we need to restrict ourselves to bounded Petri nets (those with finitely many reachable markings). The same holds for the following discussed PN classes and their relations to the corresponding subclasses of finite-state GSMP.

set to 0. The states are changed by transitions[3] that can be, on the one hand, constrained to particular clock values and, on the other hand, they can cause reset of some clocks. In more detail, each transition (with a source state, a label, and a target state) is assigned a clock guard and a set of clocks R. The clock guards are conjunctions of lower or upper bounds on the clock values, and the transition can be proceed only when its clock guards are fulfilled. If the transition is proceeded, all clocks of the assigned set R are reset. Reset of a clock is an assignment that sets the clock value to 0. Note that there are two sources of nondeterminism — delays and transition choices, i.e., for how long time the automaton will stay in a state and what transition it will proceed.

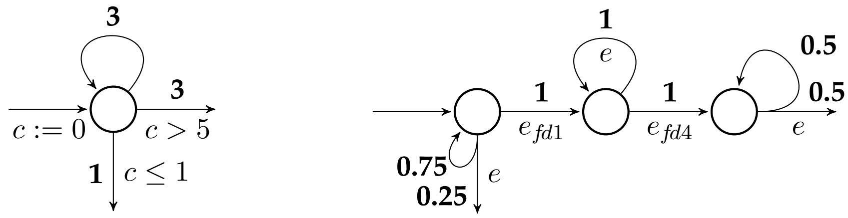
Stochastic timed automata (STA) are TA where both delays and transition choices are made randomly [KNSS00, BBB+07, BBB+14]. Intuitively, for a state, we will first randomly choose a delay among all possible delays, then we will randomly choose a transition among those which are enabled after the delay. There are some natural (but very technical) restrictions on the delay distributions, e.g., they have to be positive (only) on times when there is some outgoing transition enabled. The distribution among enabled transitions is resolved by assigned weights; the probability then respects the relation of weights assigned to the enabled transitions.

The class of STA roughly corresponds to GSMP. The relation is not as straightforward as in the case of ESPN. Note that in STA a clock can constrain many subsequent transitions without any reset. Contrary, when an event occurs in a GSMP (and initiates a transition), it is rescheduled or reset in the subsequent state. Moreover, the transition distribution in STA is defined by the weights of the enabled transitions after the delay time, i.e., it does not depend on the winning event, but on the delay time. Hence, to mimic a clock of an STA in a GSMP, we sometimes need more states and more events. Let us demonstrate that on the example of Figure 3.1 where, for one STA state with a clock c, we need three states and three events in the corresponding GSMP. Note that in the STA the transition going down is enabled only for $c \leq 1$, the transition going right is enabled only for $c > 5$, and the self-loop is always enabled. In the GSMP counterpart, we need three states representing separately the behavior in the time intervals $(0, 1)$, $(1, 5)$, and $(5, \infty)$. The transitions between the three states are triggered by fixed-delay events e_{fd1} and e_{fd4} that are set to delays 1 and 4, respectively[4]. The event-time distribution function of the event e equals the delay distribution of the STA state.

Similarly to unstable GSMP, also STA have some unintended behavior. First, there are *Zeno runs* that were considered already on (non-stochastic)

[3]Here, the transitions are again labeled edges in the graph representation of the automaton.

[4]In this example, we assume that the delay distribution function in the STA state is continuous. Otherwise, we need to discuss the cases when $c = 1$ and $c = 5$.



(a) A state of STA with a clock c. Bold numbers on edges are weights.

(b) Three states of GSMP with events e_{fd1}, e_{fd4}, and e that mimic the state of Figure 3.1a. Bold numbers are transition probabilities.

Figure 3.1: GSMP states representing an STA state.

TA [GB07]. A Zeno run executes infinitely many transitions in a finite period of time. Intuitively, this happens when the delay times get shorter and shorter such that their infinite sum is finite. Zeno runs are also in STA, but they have probability 0 (unless the underlying timed automaton is inherently Zeno) [BBB+14]. Other intended property of runs is *fairness*. A run is fair if every transition which is enabled infinitely often is taken infinitely often. STA with unfair runs of positive probability are studied in [BBB+08]. Fairness and non-Zeno behavior of STA with connections to the corresponding properties in GSMP and ESPN are shown in [BBBC18].

Various restrictions on STA are introduced to obtain subclasses where the runs are almost-surely fair. For example, *one-clock STA* [BBBM08, BBB+08] with only one clock variable, or *reactive STA* [BBJM12] that can have arbitrary many clocks but the distributions on delays have positive density on all non-negative real values. One player and two player games on reactive STA with exponentially distributed delays are studied in [BF09, BS12]. Up-to-date results concerning model checking of various STA subclasses are summarized in a unified notation in [BBB+14, BBBC18]. The compositional design based on STA was recently studied in [BBCM16].

Concerning other stochastic extensions of TA, there are also *probabilistic timed automata* [KNSS99, KNSS02] where discrete probability distributions are assigned to the transitions leaving particular states. Such distributions effect which of the enabled transitions is executed. Contrary to STA, the time-respective behavior remains nondeterministic and is considered to be controlled by an adversarial player. Up-to-date results for synthesis and games on probabilistic timed automata can be found in [FKNT16, JKNP17].

3.4 Process Algebraic Approach

Many other stochastic formalisms arise from *process algebras* (PA) [BPS01] such as Milner's CCS [Mil89] or Hoare's CSP [Hoa85]. The key feature of the PA approach is compositionality, hence, it is natural that IMC (discussed in Section 2.4.2) belong to this family of formalisms defined on the algebraic base. The most used formalism of this family is *performance evaluation PA* (PEPA) [CGHT07] that has exponentially distributed action delays. Each PEPA model can be translated to an equivalent CTMC, hence, all algorithms for CTMC can be successfully applied on PEPA. General distribution delays are discussed in generalized semi-Markov process algebra [BBG98], calculus for interactive GSMP [BG02], prioritized stochastic automata [BD04], stochastic process algebra [DK05b], stochastic automata [DK05a, DGHS18], and the modeling and description language for stochastic timed systems MoDeST [BDHK06]. To sum up, the stochastic process algebras can be viewed as high-level specification formalisms corresponding to CTMC, GSMP, and their nondeterministic extensions.

Chapter 4

Thesis contribution

In the collection, the papers are ordered chronologically from [P1] published in 2010 to [P11] published in 2017. Here, we explain the results rearranged to five logical blocks. In the first one, we comment the results published in [P2], [P3], [P5] and [P6], that relates to GSMP analysis and discuss stability of the GSMP models. This is the only block with fully stochastic models, in all the others we study decision processes or games. Section 4.2 is devoted to GSMG with objectives specified by deterministic timed automata [P1]. Then we focus on compositional approach and study games on IMC published in [P4]. In Section 4.4, we demonstrate our results on parameter synthesis in CTMC extended with some non-Markovian events. Such contributions were presented in [P7], [P8], [P9], and [P10]. Finally, we comment construction of so-called resilient strategies for MDP models introduced in [P11].

4.1 Analysis of Generalized Semi-Markov Processes

In this section, we focus on the analysis of GSMP, i.e., the most general fully stochastic class. In [P3], we show fundamental instability that can occur in GSMP models. It was a surprise that allowing only two fixed-delay events and one variable-delay event may cause an unstable behavior of a GSMP. In particular, in an unstable GSMP there are states for which both $\mathbf{d}_s$ and $\mathbf{c}_s$ frequencies may not be defined for almost all runs. Note that there had been already presented approximation algorithms computing these quantities on GSMP [ACD91, ACD92] or some approximation techniques were proposed to analyze GSMP without questioning existence of a result, e.g., [DTGN84, Lin93, GL94, LS96, LRT99, HTT00, ZFGH00, ZFH01, SDP03, CGV09].

In more detail, we realized that the traditional region-graph representation of reachable configurations of the unstable GSMP models fails in two fundamental principles:

- It is not true that with probability one each run ends in one of the bottom strongly connected components of the region-graph.
- It is no true that with probability one each run visiting a bottom strongly connected component will visit all nodes of the component infinitely often.

We demonstrate this on two simple examples, and so we disprove the correctness of the verification algorithms presented in [ACD91, ACD92].

The problem of non-stable GSMP models lies in fixed-delay events that can immediately schedule themselves whenever they occur; such an event can occur periodically like ticking of clocks. Hence, we define a syntactically restricted GSMP class, called *one-ticking GSMP*. Roughly speaking, in one-ticking GSMP we allow to have at most one periodically ticking sequence of fixed-delay events. In the context of the hierarchy depicted in Figure 2.1, we can note that all MRP models are one-ticking GSMP, but a one-ticking GSMP model is not necessarily regenerative.

We show that all one-ticking GSMP models are stable, i.e., the frequencies $\mathbf{d}_s$ and $\mathbf{c}_s$ are almost surely well defined for each state s. We also show that the frequencies can be computed not only for the one-ticking GSMP states but also for states of a deterministic TA "observer" of the one-ticking GSMP. Finally, we provide algorithms for approximation of these frequencies.

The above-mentioned results were published in [P3]. The positive results were previously published in a weaker variant for SMP in [P2]. We also reformulated the negative result for the community working with Petri nets and published it as a short paper in [P5]. Our stability results are also closely relevant to *almost-sure fairness* in STA (every edge which is enabled infinitely often is taken infinitely often almost surely) [BBB+08] and to *decisiveness* (almost surely every run either reaches a given target state or a state from which the target state is no longer reachable) studied on infinite-state Markov chains in [AHM07]. Concerning up-to-date results, we refer to [BBBC18] where, among others, the positive results of [P3] are presented in context of *stochastic transition systems* that represent a unified view on STA, GSMP, and ESPN. Finally, it is worth to note that deciding whether a GSMP is stable (or STA is almost-surely fair) is still an open problem.

4.1.1 Phase-type Fitting

Let us discuss whether it makes sense to develop analysis techniques for GSMP when one can use a phase-type fitting technique [Neu81] to approximate generally distributed events by CTMC models, i.e., every GSMP can be approximated by a CTMC.

In [P6], we focused on distributions that are known to require an excessive number of states to reach a reasonably precise approximation by the phase-type technique. Typical examples of such distributions are uniform, discrete, and shifted distributions. The shifted distribution is, for example, a packet-delivery time — there is some physical bound on the delivery time; hence, the packet is delivered earlier than this bound with zero probability. Note that all the hardly approximated distributions contain intervals with zero density. Addressing this class of distributions, we suggest an alternative approximation. We generalize phase-type fitting by using also fixed-delay events, i.e., we are approximating the models by fdCTMC instead of CTMC. Thanks to fixed-delay events, we split the density function into multiple intervals and, within each interval, we then approximate the density with standard phase-type fitting. We call this technique *interval phase-type* (IPH) approximation. We provide experimental evidence that our IPH method requires only a moderate number of states to approximate distributions that contain regions of zero density.

The fdCTMC resulting from IPH can be analyzed by state-of-the-art techniques (e.g., the subordinated Markov chain method originally published for the equivalent DSPN class [Lin93]). Let us recall that some fdCTMC models can be unstable and so the analysis techniques are usually applicable only to subclasses such as MRP, ACTMC, or 1-fdCTMC. Thus, our result promises an efficient approach to the analysis of non-Markovian models and motivates further research on MRP, ACTMC, and fdCTMC.

4.2 Generalized Semi-Markov Games

Our first contribution to this field is the definition of the generalized semi-Markov game (GSMG) [P1], the two-player game extension of GSMP. It was introduced as a generalization of continuous-time stochastic games [BFK+13, RS10], whose event-time distributions are only exponentially distributed.

Here, we describe GSMG in more detail as a GSMP that after each event-driven transition goes to a game state, called *control*, belonging to one of the two players. In each control, the particular player chooses one of the available actions that randomly lead to subsequent GSMP states. The controls and the actions are considered to be performed instantly, i.e., the time passes in the GSMP states only. We also assume that all event-time distributions are continuous with positive density on one interval of time values. As the fixed-delayed events are not allowed in here, we are not afflicted with the instability discussed in the previous section.

On GSMG, we study game objectives specified by a *deterministic timed automaton* (DTA) [ACD92]. Intuitively, a timed automaton "observes" a play of a given GSMG and checks whether certain timing constraints are satisfied, or not. Player I wins all plays that are accepted by the timed automaton, and Player II wins the others.

Namely, we show that in this setting Player I does not need to have an optimal strategy. However, if Player I has some almost-sure winning strategy, then she also has an almost-sure winning strategy which can be encoded by a deterministic timed automaton $\mathcal{A}$. The automaton $\mathcal{A}$ reads the history of a play, and the decisions depend only on the regions of the resulting configuration entered by $\mathcal{A}$. Further, we provide an exponential-time algorithm that decides the existence of such a strategy and constructs it if it exists.

Our constructions and proofs are combinations of standard techniques (used for timed automata and finite-state games) and some new non-trivial observations that are specific to the considered model. We also adapt some ideas presented in [ACD92] (in particular, we use the concept of δ-separation). Although the results of [P1] consider only reachability acceptance of the DTA observer, they were easily extended to DTA with a Büchi acceptance condition in the PhD thesis of Jan Krčál [Krč14].

4.3 Interactive Markov Chains

Let us recall (see Section 2.4.2) that Interactive Markov chains (IMC) are compositional behavioral models extending both labeled transition systems and continuous-time Markov chains. In fact, they are CTMC where some event-time distributions are not specified, and such events are rather seen as labels ready for synchronization with an environment. The compositional approach assumes that the labels are synchronized with the corresponding labels in some components running in parallel.

The main advantage of IMC is the compositionality, which allows for comfortable hierarchical design and analysis of systems. When a label is considered only for synchronization among internal subcomponents of an IMC, it can be hidden for external synchronization using a hiding operator. An IMC where all labels are hidden is called *closed*. Let us denote by *internal interactions* all the transitions representing internal synchronization based on labels that are hidden for external components. *External interactions* will stand for communication with other components. Based on this, we can introduce the *maximal-progress assumption* governing the interplay of event delays and labeled interactions of an IMC component: Internal interactions are assumed to happen instantaneously and therefore take precedence over delay transitions. This does not hold for the external interactions that stand for synchronization with other components; hence, they could be delayed. Note that a closed IMC is not necessarily fully stochastic, there still could be more internal interactions leading from a state, what we call the *internal nondeterminism*. The nondeterminism caused by an external interaction is called *external nondeterminism*.

In [P4] we analyze open IMC, i.e., those that are not necessarily closed.

In particular, we introduce the problem of synthesizing optimal control for time-bounded reachability in an IMC interacting with an unknown environment, provided no state has both internal and external interactions. In our game based analysis of an open IMC C, we assume that the internal nondeterminism of C is resolved by Player I, who controls C. Player II constructs an IMC E representing the environment (synchronized by common non-hidden labels with C) and resolves the nondeterminism caused by external interactions. Concretely, assume we are given an IMC C which contains some internal nondeterministic interactions and also offers some external interactions for synchronization to an unknown environment. Our goal is to synthesize a scheduler controlling the internal transitions which maximizes the probability of reaching a set G of goal states, in time T no matter what and when the environment E decides to synchronize with the non-hidden actions. Hence, the environment E ranges over all possible IMC able to synchronize with the non-hidden actions of C.

To get a principal understanding of the complications faced, we need to consider a restricted setting, where C does not enable internal and external interactions at the same state. We provide an algorithm which approximates the probability in question up to a given precision $\varepsilon > 0$ and also computes an ε-optimal scheduler. The algorithm consists of two steps. First, we reduce the problem to a game where the environment is not an IMC but can decide to execute external interactions at nondeterministically chosen time instances. In a second step, we solve the resulting game on C using discretization. Our discretization is based on the same approach as the algorithm of [NZ10]. However, the algorithm as well as its proof of correctness is considerably more complicated due to the nondeterministic choices of Player II controlling the environment E. We finally discuss what happens if we allow internal and external transitions to be enabled at the same time.

To the best of our knowledge, in [P4] we present the first analysis that is focused on *open* IMC. The games we consider exploit special cases of the games studied in [BF09] and in [P1]. However, both papers prove decidability only for qualitative reachability problems and do not discuss compositionality issues. Further, while systems of [RS11, BFK+09] are very similar to ours, the structure of the environment is fixed there and the verification is thus not compositional. The same holds for [Spr11, HNP+11], where time is under the control of the components.

Follow-up results were published in [HKK13], where the IMC component representing the environment is restricted by a modal continuous time automaton. This allows to omit our constraint forbidding common internal and external interactions, and thus it enables the first truly compositional verification. The results were also applied in dynamic fault tree analysis [KK15, BBH+16]. Our result uses the approximation scheme of [NZ10] that was subsequently improved in [HH15]. A distributed synthesis for more

IMC components running in parallel was studied in [HKV16]. Up-to-date research on compositional stochastic real-time systems is currently also developed in the SBIP (Stochastic real-time Behavior, Interaction, Priority) framework [NBB+15, MNB+18]. Based on our results for open IMC, the compositional design of stochastic systems was also studied on stochastic timed automata [BBCM16].

4.4 Synthesis of Delays in fdCTMC

In the section, we consider the parametric version of 1-fdCTMC where the delay times of fixed-delay events are specified by parameters, rather than concrete values. Our goal is to synthesize the values of these parameters that optimize the specified objective. As an objective, we first deal with expected reward accumulated before reaching a given target, then we study long-run average reward optimization.

4.4.1 Delays for Expected Accumulated Reward Objective

In [P7] we published an algorithm solving the synthesis problem by reduction to a finite MDP whose actions correspond to discretized (i.e., rounded to a finite mesh) delay times in the individual states. On such an MDP we apply standard polynomial time algorithm for the synthesis of the optimal delays. The non-trivial part is to prove that the delays may be discretized, i.e., for each $\varepsilon > 0$, we can compute a sufficiently small discretization step which guarantees that the optimal solution of the finite MDP is an ε-optimal solution for the original fdCTMC. We show that naive computation of the discretization step from the maximal slope is not possible. Even a very small change of the parameter can cause an arbitrarily high change in the expected reward. This happens when the delays are near zero. Our solution, based on rather non-trivial insights into the structure of 1-fdCTMC models, shows that optimal delays do not need to be close to zero. Hence, we identify "safe" delays that may be rounded with an error bounded (exponentially) in the size of the system. This leads to an exponential time algorithm for solving the optimization problem.

We experimentally implemented the proposed technique in our repository branch of the PRISM model checker [KNP11] and evaluated it on some examples. The results were published in [P8]. During the experiments, we realized that most of the computation time was spent by the construction of the discretized MDP, and even for some very small examples, the discretized MDP exceeded our 448 GiB RAM. The problem was not in the number of states of the MDP but in the huge number of actions that correspond to suitably discretized values of the delays. Hence, we designed a symbolic synthesis algorithm which avoids the explicit construction of the large action spaces. Instead, the algorithm computes small sets of "promising" candidate actions on demand. The candidate actions are selected by minimizing a certain objective function. Technically, this is done by computing the symbolic derivative of the objective function and extracting a high-degree univariate polynomial whose roots are precisely the points where the derivative takes zero value. Since roots of univariate polynomials can be isolated very efficiently using modern mathematical software (such as Maple [B+12]), we achieve not only drastic memory savings but also speedups by three orders of magnitude compared to the previous method. We demonstrated that our algorithm (experimentally implemented in PRISM with an external use of Maple) can synthesize delays for non-trivial models of large size (with more than 20,000 states). This significant improvement was published in [P9].

4.4.2 Delays for Long-Run Average Reward Objective

Here we concentrate on long-run average reward optimization. Our approach is also based on a reduction to the problem of finding optimal strategies in a symbolically represented decision process, but it is not a straightforward extension of the previous work. We need to use semi-Markov Decision Processes instead of MDP. The discretization bounds for the expected accumulated reward cannot be directly employed here, as the long-run average objectives rely on fractions of expected accumulated rewards and timings.

The results are published in [P10] where instead for 1-fdCTMC it is formulated for (more general) ACTMC models. The event-time distributions of the optimized alarms have to satisfy four abstractly formulated criteria. We show that these criteria are fulfilled, e.g., for fixed-delay events, uniformly distributed alarms (where the beginning is fixed and we synthesize the end of the time interval), and Weibull distributions (where the shape k is fixed to any natural number and we synthesize the scale parameter λ). Note that exponential distribution is a special case of the Weibull distribution, where the fixed shape constant k is 1. Our experimental evaluation shows the applicability of the method on a case study where the goal is to minimize the power consumption of a disk drive [P10].

4.4.3 Follow-up Work

An extended version of [P10] was recently accepted for publication in ACM Transactions on Modeling and Computer Simulation. In the journal version, we prolong the list of supported non-exponential alarm distributions in ACTMC and also discuss solutions for ACTMC with non-localized alarms. Alarms are non-localized whenever we want to synthesize the same delay-time distribution for an event no matter in what state it is scheduled. This approach requires to use partially observable semi-MDP and appropriate methods to find optimal strategies for them. The non-localized delays in the context of expected accumulated rewards are also briefly discussed in [P7]. For more details, we refer to [Kor17] where all the results of this section are presented for ACTMC with both localized and non-localized alarms.

4.5 MDP with Resilient Control

The last contribution [P11] is devoted to models of resilient systems represented by MDP. In resilient systems, there are repair mechanisms that have to return the system to an operational state when an error occurs.

Usually, constraints on the repair mechanisms are imposed, e.g., concerning the time or resources required (such as energy consumption or other kinds of costs). For systems modeled by MDP, we introduce the concept of resilient schedulers, which represent control strategies guaranteeing that these constraints are always met with probability greater then a given number. Technically, for a given resource bound R and a probability threshold p, we call a scheduler *resilient* if it ensures with probability at least p that each error is repaired with at most R resources. Assigning rewards to the operational states of the system, we then aim towards resilient schedulers which maximize the long-run average reward, i.e., the expected mean payoff. We present a pseudo-polynomial (polynomial when R is encoded in unary) algorithm that decides whether a resilient scheduler exists and if so, yields an optimal resilient scheduler. We also show that already the decision problem asking whether there exists a resilient scheduler is PSPACE-hard.

The key technical ingredients of our results are non-trivial observations about the structure of resilient schedulers, which connect the studied problems to the existing works on MDPs with multiple objectives and optimal strategy synthesis [Kal97, EKVY08, BBC$^+$14]. The PSPACE-hardness result is obtained by a simple reduction of the cost-bounded reachability problem in acyclic MDPs [HK15].

Chapter 5

Papers in Collection

[P1] T. Brázdil, J. Krčál, J. Křetínský, A. Kučera, and V. Řehák. Stochastic real-time games with qualitative timed automata objectives. In *CONCUR 2010 - Concurrency Theory, 21th International Conference, CONCUR 2010, Paris, France, August 31-September 3, 2010. Proceedings*, volume 6269 of *Lecture Notes in Computer Science*, pages 207–221. Springer, 2010. [BKK+10]

- Author's contribution: participating mainly on discussions concerning the problem formulation and possible solutions.

[P2] T. Brázdil, J. Krčál, J. Křetínský, A. Kučera, and V. Řehák. Measuring performance of continuous-time stochastic processes using timed automata. In *Proceedings of the 14th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2011, Chicago, IL, USA, April 12-14, 2011*, pages 33–42. ACM, 2011. [BKK+11]

- Author's contribution: participating mainly on discussions concerning the problem formulation and possible solutions.

[P3] T. Brázdil, J. Krčál, J. Křetínský, and V. Řehák. Fixed-delay events in generalized semi-Markov processes revisited. In *CONCUR 2011 - Concurrency Theory - 22nd International Conference, CONCUR 2011, Aachen, Germany, September 6-9, 2011. Proceedings*, volume 6901 of *Lecture Notes in Computer Science*, pages 140–155. Springer, 2011. [BKKŘ11]

- Author's contribution: participating on discussions, bringing the unstable models for Theorem 1 and Theorem 2, participating on writing.

[P4] T. Brázdil, H. Hermanns, J. Krčál, J. Křetínský, and V. Řehák. Verification of open interactive Markov chains. In *IARCS Annual Conference* *on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, volume 18 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 474–485. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012. [BHK+12]

- Author's contribution: bringing the topic, originating discussions with Holger Hermanns, participating on problem formulation and partially also on all other phases.

[P5] T. Brázdil, Ľ. Korenčiak, J. Krčál, J. Křetínský, and V. Řehák. On time-average limits in deterministic and stochastic Petri nets. In *ACM/SPEC International Conference on Performance Engineering, ICPE'13, Prague, Czech Republic - April 21 - 24, 2013*, pages 421–422. ACM, 2013. (Poster paper.) [BKK+13]

- Author's contribution: participating on all phases including poster presentation.

[P6] Ľ. Korenčiak, J. Krčál, and V. Řehák. Dealing with zero density using piecewise phase-type approximation. In *Computer Performance Engineering - 11th European Workshop, EPEW 2014, Florence, Italy, September 11-12, 2014. Proceedings*, volume 8721 of *Lecture Notes in Computer Science*, pages 119–134. Springer, 2014. [KKŘ14]

- Author's contribution: supervising the research, significant contribution to the experimental evaluation.

[P7] T. Brázdil, Ľ. Korenčiak, J. Krčál, P. Novotný, and V. Řehák. Optimizing performance of continuous-time stochastic systems using timeout synthesis. In *Quantitative Evaluation of Systems, 12th International Conference, QEST 2015, Madrid, Spain, September 1-3, 2015, Proceedings*, volume 9259 of *Lecture Notes in Computer Science*, pages 141–159. Springer, 2015. [BKK+15]

- Author's contribution: bringing the topic, participating on all phases.

[P8] Ľ. Korenčiak, V. Řehák, and A. Farmadin. Extension of PRISM by synthesis of optimal timeouts in fixed-delay CTMC. In *Integrated Formal Methods - 12th International Conference, IFM 2016, Reykjavik, Iceland, June 1-5, 2016, Proceedings*, volume 9681 of *Lecture Notes in Computer Science*, pages 130–138. Springer, 2016. [KŘF16]

- Author's contribution: supervising the research, significant participation on all phases.

[P9] Ľ. Korenčiak, A. Kučera, and V. Řehák. Efficient timeout synthesis in fixed-delay CTMC using policy iteration. In *24th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, MASCOTS 2016, London, United Kingdom, September 19-21, 2016*, pages 367–372. IEEE, 2016. [KKŘ16]

- Author's contribution: supervising the research, significant participation on all phases.

[P10] C. Baier, C. Dubslaff, Ľ. Korenčiak, A. Kučera, and V. Řehák. Mean-payoff optimization in continuous-time Markov chains with parametric alarms. In *Quantitative Evaluation of Systems - 14th International Conference, QEST 2017, Berlin, Germany, September 5-7, 2017, Proceedings*, volume 10503 of *Lecture Notes in Computer Science*, pages 190–206. Springer, 2017. [BDK+17a]

- Author's contribution: supervising the research, significant participation on all phases, writing some technical proofs.

[P11] C. Baier, C. Dubslaff, Ľ. Korenčiak, A. Kučera, and V. Řehák. Synthesis of optimal resilient control strategies. In *Automated Technology for Verification and Analysis - 15th International Symposium, ATVA 2017, Pune, India, October 3-6, 2017, Proceedings*, volume 10482 of *Lecture Notes in Computer Science*, pages 417–434. Springer, 2017. [BDK+17b]

- Author's contribution: proportional participation on all phases.

Bibliography

[ACD91] R. Alur, C. Courcoubetis, and D.L. Dill. Model-checking for probabilistic real-time systems. In *Proceedings of ICALP'91*, volume 510 of *Lecture Notes in Computer Science*, pages 115–126. Springer, 1991.

[ACD92] R. Alur, C. Courcoubetis, and D.L. Dill. Verifying automata specifications of probabilistic real-time systems. In *REX 1991: Real-Time: Theory in Practice*, volume 600 of *Lecture Notes in Computer Science*, pages 28–44. Springer, 1992.

[AD94] R. Alur and D.L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126(2):183–235, 1994.

[AHM07] P.A. Abdulla, N.B. Henda, and R. Mayr. Decisive Markov chains. *Logical Methods in Computer Science*, 3(4), 2007.

[ASSB00] A. Aziz, K. Sanwal, V. Singhal, and R.K. Brayton. Model-checking continuous-time Markov chains. *ACM Trans. on Comp. Logic*, 1(1):162–170, 2000.

[B^+12] L. Bernardin et al. *Maple 16 Programming Guide*. Maplesoft, Waterloo, Canada, 2012.

[BBB^+07] C. Baier, N. Bertrand, P. Bouyer, T. Brihaye, and M. Größer. Probabilistic and topological semantics for timed automata. In *FSTTCS 2007: Foundations of Software Technology and Theoretical Computer Science, 27th International Conference, New Delhi, India, December 12-14, 2007, Proceedings*, volume 4855 of *Lecture Notes in Computer Science*, pages 179–191. Springer, 2007.

[BBB^+08] C. Baier, N. Bertrand, P. Bouyer, T. Brihaye, and M. Größer. Almost-sure model checking of infinite paths in one-clock timed automata. In *Proceedings of the Twenty-Third Annual IEEE Symposium on Logic in Computer Science, LICS 2008, 24-27 June 2008, Pittsburgh, PA, USA*, pages 217–226. IEEE Computer Society, 2008.

[BBB+14] N. Bertrand, P. Bouyer, T. Brihaye, Q. Menet, C. Baier, M. Größer, and M. Jurdzinski. Stochastic timed automata. *Logical Methods in Computer Science*, 10(4), 2014.

[BBBC18] N. Bertrand, P. Bouyer, T. Brihaye, and P. Carlier. When are stochastic transition systems tameable? *J. Log. Algebr. Meth. Program.*, 99:41–96, 2018.

[BBBM08] N. Bertrand, P. Bouyer, T. Brihaye, and N. Markey. Quantitative model-checking of one-clock timed automata under probabilistic semantics. In *Fifth International Conference on the Quantitative Evaluaiton of Systems (QEST 2008), 14-17 September 2008, Saint-Malo, France*, pages 55–64. IEEE Computer Society, 2008.

[BBC+14] T. Brázdil, V. Brožek, K. Chatterjee, V. Forejt, and A. Kučera. Two views on multiple mean-payoff objectives in Markov decision processes. *Logical Methods in Computer Science*, 10(1), 2014.

[BBCM16] P. Bouyer, T. Brihaye, P. Carlier, and Q. Menet. Compositional design of stochastic timed automata. In *Computer Science - Theory and Applications - 11th International Computer Science Symposium in Russia, CSR 2016, St. Petersburg, Russia, June 9-13, 2016, Proceedings*, volume 9691 of *Lecture Notes in Computer Science*, pages 117–130. Springer, 2016.

[BBG98] M. Bravetti, M. Bernardo, and R. Gorrieri. Towards performance evaluation with general distributions in process algebras. In *CONCUR '98: Concurrency Theory, 9th International Conference, Nice, France, September 8-11, 1998, Proceedings*, volume 1466 of *Lecture Notes in Computer Science*, pages 405–422. Springer, 1998.

[BBH+16] O. Bäckström, Y. Butkova, H. Hermanns, J. Krčál, and P. Krčál. Effective static and dynamic fault tree analysis. In *Computer Safety, Reliability, and Security - 35th International Conference, SAFECOMP 2016, Trondheim, Norway, September 21-23, 2016*, volume 9922 of *Lecture Notes in Computer Science*, pages 266–280. Springer, 2016.

[BBJM12] P. Bouyer, T. Brihaye, M. Jurdzinski, and Q. Menet. Almost-sure model-checking of reactive timed automata. In *Ninth International Conference on Quantitative Evaluation of Systems, QEST 2012, London, United Kingdom, September 17-20, 2012*, pages 138–147. IEEE Computer Society, 2012.

[BD04] M. Bravetti and P.R. D'Argenio. Tutte le algebre insieme: Concepts, discussions and relations of stochastic process algebras with general distributions. In *Validation of Stochastic Systems - A Guide to Current Research*, volume 2925 of *Lecture Notes in Computer Science*, pages 44–88. Springer, 2004.

[BDHK06] H.C. Bohnenkamp, P.R. D'Argenio, H. Hermanns, and J.-P. Katoen. MODEST: A compositional modeling formalism for hard and softly timed systems. *IEEE Trans. Software Eng.*, 32(10):812–830, 2006.

[BDK+17a] C. Baier, C. Dubslaff, Ľ. Korenčiak, A. Kučera, and V. Řehák. Mean-payoff optimization in continuous-time Markov chains with parametric alarms. In *Quantitative Evaluation of Systems - 14th International Conference, QEST 2017, Berlin, Germany, September 5-7, 2017, Proceedings*, volume 10503 of *Lecture Notes in Computer Science*, pages 190–206. Springer, 2017.

[BDK+17b] C. Baier, C. Dubslaff, Ľ. Korenčiak, A. Kučera, and V. Řehák. Synthesis of optimal resilient control strategies. In *Automated Technology for Verification and Analysis - 15th International Symposium, ATVA 2017, Pune, India, October 3-6, 2017, Proceedings*, volume 10482 of *Lecture Notes in Computer Science*, pages 417–434. Springer, 2017.

[Bel57] R. Bellman. *Dynamic Programming*. Princeton University Press, 1957.

[BF09] P. Bouyer and V. Forejt. Reachability in stochastic timed games. In *Automata, Languages and Programming, 36th Internatilonal Colloquium, ICALP 2009, Rhodes, Greece, July 5-12, 2009, Proceedings, Part II.*, volume 5556 of *Lecture Notes in Computer Science*, pages 103–114. Springer, 2009.

[BFK+09] T. Brázdil, V. Forejt, J. Krčál, J. Křetínský, and A. Kučera. Continuous-time stochastic games with time-bounded reachability. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2009, December 15-17, 2009, IIT Kanpur, India*, volume 4 of *LIPIcs*, pages 61–72. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2009.

[BFK+13] T. Brázdil, V. Forejt, J. Krčál, J. Křetínský, and A. Kučera. Continuous-time stochastic games with time-bounded reachability. *Inf. Comput.*, 224:46–70, 2013.

[BG02] M. Bravetti and R. Gorrieri. The theory of interactive generalized semi-Markov processes. *Theor. Comput. Sci.*, 282(1):5–32, 2002.

[BGdMT06] G. Bolch, S. Greiner, H. de Meer, and K.S. Trivedi. *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. Wiley, Hoboken, New Jersey, USA, 2nd edition, 2006.

[BHHK03] C. Baier, B.R. Haverkort, H. Hermanns, and J.-P. Katoen. Model-checking algorithms for continuous-time Markov chains. *IEEE Trans. Software Eng.*, 29(6):524–541, 2003.

[BHK+12] T. Brázdil, H. Hermanns, J. Krčál, J. Křetínský, and V. Řehák. Verification of open interactive Markov chains. In *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, volume 18 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 474–485. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2012.

[BKK+10] T. Brázdil, J. Krčál, J. Křetínský, A. Kučera, and V. Řehák. Stochastic real-time games with qualitative timed automata objectives. In *CONCUR 2010 - Concurrency Theory, 21th International Conference, CONCUR 2010, Paris, France, August 31-September 3, 2010. Proceedings*, volume 6269 of *Lecture Notes in Computer Science*, pages 207–221. Springer, 2010.

[BKK+11] T. Brázdil, J. Krčál, J. Křetínský, A. Kučera, and V. Řehák. Measuring performance of continuous-time stochastic processes using timed automata. In *Proceedings of the 14th ACM International Conference on Hybrid Systems: Computation and Control, HSCC 2011, Chicago, IL, USA, April 12-14, 2011*, pages 33–42. ACM, 2011.

[BKK+13] T. Brázdil, Ľ. Korenčiak, J. Krčál, J. Křetínský, and V. Řehák. On time-average limits in deterministic and stochastic Petri nets. In *ACM/SPEC International Conference on Performance Engineering, ICPE'13, Prague, Czech Republic - April 21 - 24, 2013*, pages 421–422. ACM, 2013.

[BKK+15] T. Brázdil, Ľ. Korenčiak, J. Krčál, P. Novotný, and V. Řehák. Optimizing performance of continuous-time stochastic systems using timeout synthesis. In *Quantitative Evaluation of Systems, 12th International Conference, QEST 2015, Madrid, Spain, September 1-3, 2015, Proceedings*, volume 9259 of *Lecture Notes in Computer Science*, pages 141–159. Springer, 2015.

[BKKŘ11] T. Brázdil, J. Krčál, J. Křetínský, and V. Řehák. Fixed-delay events in generalized semi-Markov processes revisited. In *CONCUR 2011 - Concurrency Theory - 22nd International Conference, CONCUR 2011, Aachen, Germany, September 6-9, 2011. Proceedings*, volume 6901 of *Lecture Notes in Computer Science*, pages 140–155. Springer, 2011.

[BPS01] J.A. Bergstra, A. Ponse, and S.A. Smolka, editors. *Handbook of Process Algebra*. Elsevier, 2001.

[BS12] N. Bertrand and S. Schewe. Playing optimally on timed automata with random delays. In *Formal Modeling and Analysis of Timed Systems - 10th International Conference, FORMATS 2012, London, UK, September 18-20, 2012. Proceedings*, volume 7595 of *Lecture Notes in Computer Science*, pages 43–58. Springer, 2012.

[CGHT07] A. Clark, S. Gilmore, J. Hillston, and M. Tribastone. Stochastic process algebras. In *Formal Methods for Performance Evaluation, 7th International School on Formal Methods for the Design of Computer, Communication, and Software Systems, SFM 2007, Bertinoro, Italy, May 28-June 2, 2007, Advanced Lectures*, volume 4486 of *Lecture Notes in Computer Science*, pages 132–179. Springer, 2007.

[CGV09] L. Carnevali, L. Grassi, and E. Vicario. State-density functions over DBM domains in the analysis of non-Markovian models. *IEEE Trans. Software Eng.*, 35(2):178–194, 2009.

[CL08] C.G. Cassandras and S. Lafortune. *Introduction to Discrete Event Systems*. Springer, New York, NY, USA, 2nd edition, 2008.

[CY01] H. Chen and D.D. Yao. *Fundamentals of Queueing Networks: Performance, Asymptotics, and Optimization*. Springer, New York, NY, USA, 2001.

[DGHS18] P.R. D'Argenio, M. Gerhold, A. Hartmanns, and S. Sedwards. A hierarchy of scheduler classes for stochastic automata. In *Foundations of Software Science and Computation Structures - 21st International Conference, FOSSACS 2018, Thessaloniki, Greece, April 14-20, 2018, Proceedings*, volume 10803 of *Lecture Notes in Computer Science*, pages 384–402. Springer, 2018.

[DK05a] P.R. D'Argenio and J.-P. Katoen. A theory of stochastic systems part I: stochastic automata. *Inf. Comput.*, 203(1):1–38, 2005.

[DK05b] P.R. D'Argenio and J.-P. Katoen. A theory of stochastic systems. part II: process algebra. *Inf. Comput.*, 203(1):39–74, 2005.

[DTGN84] J.B. Dugan, K.S. Trivedi, R. Geist, and V.F. Nicola. Extended stochastic Petri nets: Applications and analysis. In *Performance '84, Proceedings of the Tenth International Symposium on Computer Performance Modelling, Measurement and Evaluation, Paris, France, 19-21 December, 1984*, pages 507–519. North-Holland, 1984.

[EKVY08] K. Etessami, M.Z. Kwiatkowska, M.Y. Vardi, and M. Yannakakis. Multi-objective model checking of Markov decision processes. *Logical Methods in Computer Science*, 4(4), 2008.

[FKNT16] V. Forejt, M. Kwiatkowska, G. Norman, and A. Trivedi. Expected reachability-time games. *Theor. Comput. Sci.*, 631:139–160, 2016.

[FV96] J. Filar and K. Vrieze. *Competitive Markov Decision Processes*. Springer, 1996.

[GB07] R. Gómez and H. Bowman. Efficient detection of zeno runs in timed automata. In *Formal Modeling and Analysis of Timed Systems, 5th International Conference, FORMATS 2007, Salzburg, Austria, October 3-5, 2007, Proceedings*, volume 4763 of *Lecture Notes in Computer Science*, pages 195–210. Springer, 2007.

[GHL09] X. Guo and O. Hernández-Lerma. *Continuous-Time Markov Decision Processes*, volume 62 of *Stochastic Modelling and Applied Probability*. Springer, Berlin, Heidelberg, 2009.

[GL94] R. German and C. Lindemann. Analysis of stochastic Petri nets by the method of supplementary variables. *Perform. Eval.*, 20(1-3):317–335, 1994.

[GSTH08] D. Gross, J.F. Shortle, J.M. Thompson, and C.M. Harris. *Fundaments of Queueing Theory*. Wiley, Hoboken, New Jersey, USA, 4th edition, 2008.

[Haa02] P.J. Haas. *Stochastic Petri Nets: Modelling, Stability, Simulation*. Springer, New York, NY, USA, 2002.

[HH15] H. Hatefi and H. Hermanns. Improving time bounded reachability computations in interactive Markov chains. *Sci. Comput. Program.*, 112:58–74, 2015.

[HHK16] A. Hartmanns, H. Hermanns, and J. Krčál. Schedulers are no prophets. In *Semantics, Logics, and Calculi - Essays Dedicated to Hanne Riis Nielson and Flemming Nielson on the Occasion of Their 60th Birthdays*, volume 9560 of *Lecture Notes in Computer Science*, pages 214–235. Springer, 2016.

[HK09] H. Hermanns and J.-P. Katoen. The how and why of interactive Markov chains. In *Formal Methods for Components and Objects - 8th International Symposium, FMCO 2009. Revised Selected Papers*, volume 6286 of *Lecture Notes in Computer Science*, pages 311–337. Springer, 2009.

[HK15] C. Haase and S. Kiefer. The odds of staying on budget. In *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 234–246. Springer, 2015.

[HKK13] H. Hermanns, J. Krčál, and J. Křetínský. Compositional verification and optimization of interactive Markov chains. In *CONCUR 2013 - Concurrency Theory - 24th International Conference, CONCUR 2013, Buenos Aires, Argentina, August 27-30, 2013. Proceedings*, volume 8052 of *Lecture Notes in Computer Science*, pages 364–379. Springer, 2013.

[HKV16] H. Hermanns, J. Krčál, and S. Vester. Distributed synthesis in continuous time. In *Foundations of Software Science and Computation Structures - 19th International Conference, FOSSACS 2016, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2016, Eindhoven, The Netherlands, April 2-8, 2016*, volume 9634 of *Lecture Notes in Computer Science*, pages 353–369. Springer, 2016.

[HNP+11] E.M. Hahn, G. Norman, D. Parker, B. Wachter, and L. Zhang. Game-based abstraction and controller synthesis for probabilistic hybrid systems. In *Eighth International Conference on Quantitative Evaluation of Systems, QEST 2011, Aachen, Germany, 5-8 September, 2011*, pages 69–78. IEEE Computer Society, 2011.

[Hoa85] C.A.R. Hoare. *Communicating Sequential Processes*. Prentice-Hall, 1985.

[HTT00] C. Hirel, B. Tuffin, and K.S. Trivedi. SPNP: stochastic Petri nets. version 6.0. In *Computer Performance Evaluation / TOOLS*, volume 1786 of *Lecture Notes in Computer Science*, pages 354–357. Springer, 2000.

[HV85] M.A. Holliday and M.K. Vernon. A generalized timed Petri net model for performance analysis. In *International Workshop on Timed Petri Nets, Torino, Italy, July 1-3, 1985*, pages 181–190. IEEE Computer Society, 1985.

[JKNP17] A. Jovanovic, M. Kwiatkowska, G. Norman, and Q. Peyras. Symbolic optimal expected time reachability computation and controller synthesis for probabilistic timed automata. *Theor. Comput. Sci.*, 669:1–21, 2017.

[Kal97] O. Kallenberg. *Foundations of Modern Probability*. Probability and Its Applications. Springer, 1997.

[Ken53] D.G. Kendall. Stochastic Processes Occurring in the Theory of Queues and their Analysis by the Method of the Imbedded Markov Chain. *Ann. Math. Statist.*, 24(3):338–354, 09 1953.

[KK15] J. Krčál and P. Krčál. Scalable analysis of fault trees with dynamic features. In *45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks, DSN 2015, Rio de Janeiro, Brazil, June 22-25, 2015*, pages 89–100. IEEE Computer Society, 2015.

[KKŘ14] Ľ. Korenčiak, J. Krčál, and V. Řehák. Dealing with zero density using piecewise phase-type approximation. In *Computer Performance Engineering - 11th European Workshop, EPEW 2014, Florence, Italy, September 11-12, 2014. Proceedings*, volume 8721 of *Lecture Notes in Computer Science*, pages 119–134. Springer, 2014.

[KKŘ16] Ľ. Korenčiak, A. Kučera, and V. Řehák. Efficient timeout synthesis in fixed-delay CTMC using policy iteration. In *24th IEEE International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems, MASCOTS 2016, London, United Kingdom, September 19-21, 2016*, pages 367–372. IEEE, 2016.

[KNP11] M. Kwiatkowska, G. Norman, and D. Parker. PRISM 4.0: Verification of probabilistic real-time systems. In *Proc. 23rd International Conference on Computer Aided Verification (CAV'11)*, volume 6806 of *LNCS*, pages 585–591. Springer, 2011.

[KNSS99] M.Z. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Automatic verification of real-time systems with discrete probability distributions. In *Formal Methods for Real-Time and Probabilistic Systems, 5th International AMAST Workshop, ARTS'99, Bamberg, Germany, May 26-28, 1999. Proceedings*, volume 1601 of *Lecture Notes in Computer Science*, pages 75–95. Springer, 1999.

[KNSS00] M.Z. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Verifying quantitative properties of continuous probabilistic timed automata. In *CONCUR 2000 - Concurrency Theory, 11th International Conference, University Park, PA, USA, August 22-25, 2000, Proceedings*, volume 1877 of *Lecture Notes in Computer Science*, pages 123–137. Springer, 2000.

[KNSS02] M.Z. Kwiatkowska, G. Norman, R. Segala, and J. Sproston. Automatic verification of real-time systems with discrete probability distributions. *Theor. Comput. Sci.*, 282(1):101–150, 2002.

[Kor17] Ľ. Korenčiak. *Parameter Synthesis in Continuous-Time Stochastic Systems.* PhD thesis, Faculty of Informatics, Masaryk University, Brno, 2017.

[Krč14] J. Krčál. *Formal Analysis of Discrete-Event Systems with Hard RealTime Bounds*. PhD thesis, Masaryk University, Faculty of Informatics, 2014.

[KŘF16] Ľ. Korenčiak, V. Řehák, and A. Farmadin. Extension of PRISM by synthesis of optimal timeouts in fixed-delay CTMC. In *Integrated Formal Methods - 12th International Conference, IFM 2016, Reykjavik, Iceland, June 1-5, 2016, Proceedings*, volume 9681 of *Lecture Notes in Computer Science*, pages 130–138. Springer, 2016.

[LHK01] G.G.I. López, H. Hermanns, and J.-P. Katoen. Beyond memoryless distributions: Model checking semi-Markov chains. In *Process Algebra and Probabilistic Methods, Performance Modeling and Verification: Joint International Workshop, PAPM-PROBMIV 2001, Aachen, Germany, September 12-14, 2001, Proceedings*, volume 2165 of *Lecture Notes in Computer Science*, pages 57–70. Springer, 2001.

[Lin93] C. Lindemann. An improved numerical algorithm for calculating steady-state solutions of deterministic and stochastic Petri net models. *Perform. Eval.*, 18(1):79–95, July 1993.

[LRT99] C. Lindemann, A. Reuys, and A. Thümmler. The DSPNexpress 2.000 performance and dependability modeling environment. In *Digest of Papers: FTCS-29, The Twenty-Ninth Annual International Symposium on Fault-Tolerant Computing, Madison, Wisconsin, USA, June 15-18, 1999*, pages 228–231. IEEE Computer Society, 1999.

[LS96] C. Lindemann and G.S. Shedler. Numerical analysis of deterministic and stochastic Petri nets with concurrent deterministic transitions. *Perform. Eval.*, 27/28(4):565–582, 1996.

[Mar88] M.A. Marsan. Stochastic Petri nets: an elementary introduction. In *Advances in Petri Nets 1989, covers the 9th European Workshop on Applications and Theory in Petri Nets, held in Venice, Italy in June 1988, selected papers*, volume 424 of *Lecture Notes in Computer Science*, pages 1–29. Springer, 1988.

[Mat62] K. Matthes. Zur Theorie der Bedienungsprozesse. *Transactions of the Third Prague Conference on Information Theory, Statistical Decision Functions, Random Processes*, pages 513–528, 1962.

[MC86] M.A. Marsan and G. Chiola. On Petri nets with deterministic and exponentially distributed firing times. In *Advances in Petri Nets 1987, covers the 7th European Workshop on Applications and Theory of Petri Nets, Oxford, UK, June 1986*, volume 266 of *Lecture Notes in Computer Science*, pages 132–145. Springer, 1986.

[MCB84] M.A. Marsan, G. Conte, and G. Balbo. A class of generalized stochastic Petri nets for the performance evaluation of multiprocessor systems. *ACM Trans. Comput. Syst.*, 2(2):93–122, May 1984.

[Mil89] R. Milner. *Communication and concurrency*. PHI Series in computer science. Prentice Hall, 1989.

[MNB$^+$18] B.L. Mediouni, A. Nouri, M. Bozga, M. Dellabani, A. Legay, and S. Bensalem. *S* BIP 2.0: Statistical model checking stochastic real-time systems. In *Automated Technology for Verification and Analysis - 16th International Symposium, ATVA 2018, Los Angeles, CA, USA, October 7-10, 2018, Proceedings*, volume 11138 of *Lecture Notes in Computer Science*, pages 536–542. Springer, 2018.

[Mol81] M.K. Molloy. *On the integration of delay and throughput measures in distributed processing models*. PhD thesis, Univ. of California, Los Angeles, USA, 1981.

[Mol82] M.K. Molloy. Performance analysis using stochastic Petri nets. *IEEE Trans. Computers*, 31(9):913–917, 1982.

[Mol85] M.K. Molloy. Discrete time stochastic Petri nets. *IEEE Trans. Software Eng.*, 11(4):417–423, 1985.

[Nat80] S. Natkin. *Les réseaux de Petri Stochastiques et leur applications à l'évaluation des systèmes informatiques*. PhD thesis, CNAM, Paris, France, 1980.

[NBB+15] A. Nouri, S. Bensalem, M. Bozga, B. Delahaye, C. Jégourel, and A. Legay. Statistical model checking qos properties of systems with SBIP. *International Journal on Software Tools for Technology Transfer*, 17(2):171–185, 2015.

[Neu81] M.F. Neuts. *Matrix-geometric Solutions in Stochastic Models: An Algorithmic Approach*. Courier Dover Publications, 1981.

[Nor98] J.R. Norris. *Markov Chains*. Cambridge University Press, New York, NY, USA, 1998.

[NZ10] M.R. Neuhäußer and L. Zhang. Time-bounded reachability probabilities in continuous-time Markov decision processes. In *QEST 2010, Seventh International Conference on the Quantitative Evaluation of Systems, Williamsburg, Virginia, USA, 15-18 September 2010*, pages 209–218. IEEE Computer Society, 2010.

[Pet62] C.A. Petri. Kommunikation mit Automaten. Dissertation, Schriften des IIM 2, Rheinisch-Westfälisches Institut für Instrumentelle Mathematik an der Universität Bonn, Bonn, 1962.

[PHV16] M. Paolieri, A. Horváth, and E. Vicario. Probabilistic model checking of regenerative concurrent systems. *IEEE Trans. Software Eng.*, 42(2):153–169, 2016.

[Put94] M. L. Puterman. *Markov Decision Processes*. Wiley, Hoboken, NJ, USA, 1994.

[RS10] M.N. Rabe and S. Schewe. Optimal time-abstract schedulers for CTMDPs and Markov games. In *Eighth Workshop on Quantitative Aspects of Programming Languages, QAPL'10*, pages 144–158, 2010.

[RS11] M.N. Rabe and S. Schewe. Finite optimal control for time-bounded reachability in CTMDPs and continuous-time Markov games. *Acta Inf.*, 48(5-6):291–315, 2011.

[SDP03] M. Scarpa, S. Distefano, and A. Puliafito. A parallel approach for the solution of non-Markovian Petri nets. In *PVM/MPI*, volume 2840 of *Lecture Notes in Computer Science*, pages 196–203. Springer, 2003.

[Smi55] W.L. Smith. Regenerative stochastic processes. *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, 232(1188):6–31, 1955.

[Spr11] J. Sproston. Discrete-time verification and control for probabilistic rectangular hybrid automata. In *Eighth International Conference on Quantitative Evaluation of Systems, QEST 2011, Aachen, Germany, 5-8 September, 2011*, pages 79–88. IEEE Computer Society, 2011.

[Sym80] F.J.W. Symons. Introduction to numerical Petri nets, a general graphical model of concurrent processing systems. *Australian Telecommunications Research*, 14(1):28–33, 1980.

[ZFGH00] A. Zimmermann, J. Freiheit, R. German, and G. Hommel. Petri net modelling and performability evaluation with TimeNET 3.0. In *Computer Performance Evaluation / TOOLS*, volume 1786 of *Lecture Notes in Computer Science*, pages 188–202. Springer, 2000.

[ZFH01] A. Zimmermann, J. Freiheit, and G. Hommel. Discrete time stochastic Petri nets for the modeling and evaluation of real-time systems. In *Proceedings 15th International Parallel and Distributed Processing Symposium. IPDPS 2001*, pages 1069–1074. IEEE Computer Society, 2001.