# Synthesis of Optimal Resilient Control Strategies

Christel Baier[1], Clemens Dubslaff[1(✉)], L'uboš Korenčiak[2(✉)], Antonín Kučera[2], and Vojtěch Řehák[2]

[1] TU Dresden, Dresden, Germany
{christel.baier,clemens.dubslaff}@tu-dresden.de
[2] Masaryk University, Brno, Czech Republic
{korenciak,kucera,rehak}@fi.muni.cz

**Abstract.** Repair mechanisms are important within resilient systems to maintain the system in an operational state after an error occurred. 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 Markov decision processes (MDPs), we introduce the concept of *resilient schedulers*, which represent control strategies guaranteeing that these constraints are always met within some given probability. 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 algorithm that decides whether a resilient scheduler exists and if so, yields an optimal resilient scheduler. We show also that already the decision problem asking whether there exists a resilient scheduler is PSPACE-hard.

## 1 Introduction

Computer systems are resilient when they incorporate mechanisms to adapt to changing conditions and to recover rapidly or at low costs from disruptions. The latter property of resilient systems is usually maintained through repair mechanisms, which push the system towards an operational state after some error occurred. Resilient systems and repair mechanisms have been widely studied in the literature and are an active field of research (see, e.g., [2] for an overview). Errors such as measurement errors, read/write errors, connection errors do not necessarily impose a system error but may be repaired to foster the system to be operational. Examples of repair mechanisms include rejuvenation procedures that face the degradation of software over time [13], the evaluation of checksums to repair communication errors, or methods to counter an attack from outside a security system. The repair of a degraded software system could be achieved, e.g., by clearing caches (fast, very good availability), by running maintenance

The authors are partly supported by the Czech Science Foundation, grant No. 15-17564S, by the DFG through the Collaborative Research Center SFB 912 – HAEC, the Excellence Initiative by the German Federal and State Governments (cluster of excellence cfAED), and the DFG-projects BA-1679/11-1 and BA-1679/12-1.


D. D'Souza and K. Narayan Kumar (Eds.): ATVA 2017, LNCS 10482, pp. 417–434, 2017.
DOI: 10.1007/978-3-319-68167-2_27

methods (more time, less availability, but higher success), or by a full restart (slow, cutting off availability, but guaranteed success). Depending on the situation the system faces, there is a trade-off between these characteristics and a choice has to be made, which of the repair mechanisms should be executed to fulfill further constraints on the repair, which errors should be avoided, and to optimize an overall goal. Usually, finding suitable control strategies performing the choices for repair is done in an ad-hoc manner and requires a considerable engineering effort.

In this paper, we face the question of an automated synthesis of *resilient control strategies* that maximize the long-run average availability of the system. Inspired by the use of probabilistic response patterns to describe resilience [8], we focus on control strategies that are *probabilistically resilient*, i.e., with high probability repair mechanisms succeed within a given amount of time or other kinds of costs. Our formal model we use to describe resilient systems is provided by Markov decision processes (MDPs, see, e.g., [17,19]). That is, directed graphs over states with edges annotated by actions that stand for non-deterministic choices and stochastic information about the probabilistic choices resolved after taking some action. Following [3,16], we distinguish between three kinds of states: error, repair and operational states. Error states stand for states where a disruption of the system is discovered, initiating a repair mechanism modeled by repair states. Operational states are those states where the system is available and no repair is required. To reason about the trade-off between choosing control strategies, we amend error and repair states with cost values, and operational states with payoff values, respectively. Assigned costs formalize, e.g., the time required or the energy consumed for leaving an error or repair state. Likewise, assigned payoff values quantify the benefit of some operational state, e.g., stand for the number of successfully completed tasks while being operational. We define the long-run average availability as the mean-payoff. Control strategies in MDPs are provided by (randomized) schedulers that, depending on the history of the system execution, choose the probability of the next action to fire. When the probabilities for action choices are Dirac, i.e., exactly one action is chosen almost surely, the scheduler is called deterministic. Schedulers which select an action only depending on the current state, i.e., do not depend on the history, are called memoryless. For a given cost bound $R$ and a probability threshold $\wp$, we call a scheduler *resilient* if the scheduler ensures for every error a recovery within at most $R$ costs with probability at least $\wp$.

*Our Contribution.* We show that if the cost bound $R$ is represented in *unary*, the existence of a resilient scheduler is solvable in polynomial time. Further, we show that if there is at least one resilient scheduler, then there also exists an *optimal* resilient scheduler $\mathfrak{R}$ computable in polynomial time. Here, optimality means that $\mathfrak{R}$ achieves the maximal long-run average availability among all resilient schedulers. The constructed scheduler $\mathfrak{R}$ is randomized and uses finite memory. The example below illustrates that deterministic or memoryless randomized schedulers are less powerful. If $R$ is encoded in binary, our algorithms are exponential, and we show that deciding the existence of a resilient scheduler

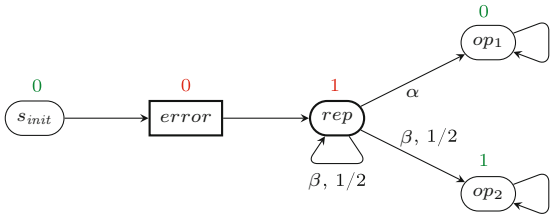


**Fig. 1.** Optimal resilient schedulers might require finite memory and randomization

becomes PSPACE-hard. Let us note that all numerical constants (such as $\wp$ or MDP transition probabilities) except for $R$ are represented as fractions of binary numbers. 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 [7,11,17]. The PSPACE-hardness result is obtained by a simple reduction of the cost-bounded reachability problem in acyclic MDPs [15]. More details are given at appropriate places in Sect. 3. Due to space constraints, full proofs can be found in the full version of this paper [4].

*Example.* As a simple example, consider an MDP model of a resilient system depicted in Fig. 1. Operational states are depicted by thin rounded boxes, error states are shown as rectangles and repair states are depicted by thick-rounded boxes. Assigned cost and payoff values are indicated above the nodes of the MDP. For edges without any action name or probability, we assume one action with probability one. The system starts its execution in the operational state $s_{init}$, from which it reaches the error state *error* and directly invokes a repair mechanism by switching to the repair state *rep*, where either action $\alpha$ or $\beta$ can be chosen. After taking $\alpha$, an operational state $op_1$ is reached that, however, does not grant any payoff. When choosing $\beta$, a fair coin is flipped and either the repair mechanism has to be tried again or the operational state $op_2$ is reached, while providing the payoff value 1 for each visit of $op_2$. Assume that we have given the cost bound $R = 2$ and probability threshold $\wp = 4/5$. The memoryless deterministic strategy always choosing $\beta$ yields the maximal possible mean payoff of 1, but is not resilient as $\wp > 1 - 1/2^R = 3/4$. The memoryless randomized scheduler that chooses $\beta$ with probability $2/\sqrt{5}$ is resilient and achieves the maximal mean payoff of $1/(\sqrt{5} - 1) \approx 0.809$, when ranging over all memoryless randomized schedulers. Differently, the finite-memory randomized scheduler playing $\beta$ with probability $4/5$ in the second step and with probability 1 in all other steps yields the mean payoff of 0.9, which is optimal within all resilient schedulers. As this example shows, optimal resilient schedulers might require randomization and finite memory in terms of remembering the accumulated costs spent so far after an error occurred.

*Related work.* Concerning the analysis of resilient systems, [3] presented algorithms to reason about trade-offs between costs and payoffs using (probabilistic)

model-checking techniques. In [18], several metrics to quantify resiliency and their applications to large scale systems has been detailed.

Synthesis of control strategies for resilient systems have been mainly considered in the non-probabilistic setting. In [16], a game-theoretic approach towards synthesizing strategies that maintain a certain resilience level has been presented. The resilience level is defined in terms of the number of errors from which the system can recover simultaneously. Automatic synthesis of Pareto-optimal implementations of resilient systems were detailed in [10]. Robust synthesis procedures with both, qualitative and mean-payoff objectives have been presented in [6]. In [14], the authors present algorithms to synthesize controllers for fault-tolerant systems compliant to constraints on power consumption.

Optimization problems for MDPs with mean-payoff objectives and constraints on cost structures have been widely studied in the field of constrained Markov decision processes (see, e.g., [19] and [1] for an overview). MDPs with multiple constraints on the probabilities for satisfying $\omega$-regular specifications were studied in [11]. This work has been extended to also allow for (multiple) constraints on the expected total reward in MDPs with rewards in [12]. Synthesis of optimal schedulers with multiple long-run average objectives in MDPs has been considered in [7,9]. All of the mentioned approaches have in common that they adapt well-known linear programs to synthesize optimal memoryless randomized schedulers (see, e.g., [17,19]). We also use combinations of similar techniques to find optimal resilient schedulers. As far as we know, we are the first to consider mean-payoff optimization problems under cost-bounded reachability probability constraints. Although we investigate these problems in the context of resilient systems, they are interesting by its own.

## 2 Notations and Problem Statement

Given a finite set $X$, we denote by $\mathrm{Dist}(X)$ the set of *probability distributions* on $X$, i.e., the set of functions $\mu\colon X \to [0,1]$ where $\sum_{x\in X} \mu(x) = 1$. By $X^\infty$ we denote finite or infinite sequences of elements of $X$. We assume that the reader is familiar with principles about probabilistic systems, logics, and model-checking techniques and refer to [5] for an introduction in these subjects.

### 2.1 Markov Decision Processes

A *Markov decision process (MDP)* is a triple $\mathcal{M} = (S, Act, P, s_{init})$, where $S$ is a finite state space, $s_{init} \in S$ an initial state, $Act$ a finite set of actions, and $P\colon S \times Act \times S \to [0,1]$ a transition probability function, i.e., a function where $\sum_{s'\in S} P(s,\alpha,s') \in \{0,1\}$ for all $s \in S$ and $\alpha \in Act$. For $s \in S$, let $Act(s)$ denote the set of actions $\alpha \in Act$ that are enabled in $s$, i.e., $\alpha \in Act(s)$ iff $P(s,\alpha,\cdot)$ is a probability distribution over $S$. Unless stated differently, we suppose that any MDP does not have any trap states, i.e., states $s$ where $Act(s) = \varnothing$. *Paths* in $\mathcal{M}$ are alternating sequences $s_0\alpha_0 s_1\alpha_1 \ldots \in S \times (Act \times S)^\infty$ of states and actions, such that $P(s_i,\alpha_i,s_{i+1}) > 0$ for all $i \in \mathbb{N}$. The set of all finite paths starting in

state $s \in S$ is denoted by $\mathit{FinPaths}(s)$, where we omit $s$ when all finite paths from any state are issued.

A *(randomized, history-dependent) scheduler* for $\mathcal{M}$ is a function $\mathfrak{S}\colon \mathit{FinPaths} \to \mathrm{Dist}(\mathit{Act})$. A $\mathfrak{S}$*-path* in $\mathcal{M}$ is a path $\pi = s_0\alpha_0 s_1\alpha_1 \ldots$ in $\mathcal{M}$ where for all $n \in \mathbb{N}$ we have that $\mathfrak{S}(s_0\alpha_0 s_1\alpha_1 \ldots \alpha_{n-1}s_n)(\alpha_n) > 0$. We write $\Pr^{\mathfrak{S}}_{\mathcal{M},s}$ for the probability measure on infinite paths of $\mathcal{M}$ induced by a scheduler $\mathfrak{S}$ and starting in $s$. For a scheduler $\mathfrak{S}$ and $\pi \in \mathit{FinPaths}$, $\mathfrak{S} \uparrow \pi$ denotes the *residual scheduler* $\mathfrak{T}$ given by $\mathfrak{T}(\pi') = \mathfrak{S}(\pi;\pi')$ for each finite path $\pi'$ where the first state of $\pi'$ equals the last state of $\pi$. Here ; is used for the concatenation operator on finite paths. $\mathfrak{S}$ is called *memoryless* if $\mathfrak{S}(s) = \mathfrak{S}(\pi)$ for all $s \in S$ and all finite paths $\pi \in \mathit{FinPaths}$ where the last state of $\pi$ is $s$. We abbreviate memoryless (randomized) schedulers as *MR-schedulers*.

### 2.2 Markov Decision Processes with Repair

Let $\mathcal{M} = (S, \mathit{Act}, P, s_{\mathit{init}})$ be an MDP and suppose that we have given two disjoint sets of states $\mathit{Err}, \mathit{Op} \subseteq S$. Intuitively, $\mathit{Err}$ stands for the set of states where an error occurs, and $\mathit{Op}$ stands for the set of states where the system modeled is operational. In all other states, we assume that a repair mechanism is running, triggered directly within the next transition after some error occurred. We formalize the latter assumption by

$$e \models \forall \bigcirc \forall(\neg \mathit{Err}\, \mathsf{W}\, \mathit{Op}) \qquad \text{for all states } e \in \mathit{Err} \tag{$*$}$$

where $\bigcirc$ and $\mathsf{W}$ stand for the standard next and weak-until operator, respectively, borrowed from computation tree logic (CTL, see, e.g., [5]). Assumption (*) also asserts that as soon as a repair protocol has been started, the system does not enter a new error state before a successful repair, i.e., until the system switches to its operational mode.

Further, we suppose that states in $\mathcal{M}$ are amended with non-negative integer values, i.e., we are given a non-negative integer reward function $\mathit{rew}\colon S \to \mathbb{N}$. For an operational state $s \in \mathit{Op}$, the value $\mathit{rew}(s)$ is viewed as the *payoff* value of state $s$, while for the non-operational states $s \in S \setminus \mathit{Op}$, the value $\mathit{rew}(s)$ is viewed as the repairing *costs* caused by state $s$. To reflect this intuitive meaning of the reward values, we shall write $\mathit{payoff}(s)$ instead of $\mathit{rew}(s)$ for $s \in \mathit{Op}$ and $\mathit{cost}(s)$ instead of $\mathit{rew}(s)$ for $s \in S \setminus \mathit{Op}$. Furthermore, we assume $\mathit{payoff}(s) = 0$ if $s \in S \setminus \mathit{Op}$ and $\mathit{cost}(s) = 0$ if $s \in \mathit{Op}$. For a finite path $\pi = s_0\alpha_0 s_1 \ldots \alpha_{n-1}s_n$, let $\mathit{cost}(\pi)$ and $\mathit{payoff}(\pi)$ be $\sum_{i=0}^{n} \mathit{cost}(s_i)$ and $\sum_{i=0}^{n} \mathit{payoff}(s_i)$, respectively.

An *MDP with repair* is formally defined as a tuple $(\mathcal{M}, \mathit{Err}, \mathit{Op}, \mathit{rew})$, where assumption (*) is satisfied and the transition probability function of $\mathcal{M}$ is rational, assuming representation of probabilities as fractions of binary numbers.

### 2.3 Long-Run Availability and Resilient Schedulers

Given an MDP with repair $(\mathcal{M}, \mathit{Err}, \mathit{Op}, \mathit{rew})$ and a scheduler $\mathfrak{S}$ for $\mathcal{M}$, we define the *long-run availability* of $\mathfrak{S}$, denoted by $\mathrm{Avail}^{\mathfrak{S}}_{\mathcal{M},s_{\mathit{init}}}$, as the expected

long-run average (mean-payoff) of the payoff function. That is, for any $s_0 \in S$, $\mathrm{Avail}^{\mathfrak{S}}_{\mathcal{M},s_0}$ agrees with the expectation of the random variable $X$ under $\Pr^{\mathfrak{S}}_{\mathcal{M},s_0}$ that assigns to each infinite path $\zeta = s_0\,\alpha_0\,s_1\,\alpha_1\,s_2\,\alpha_2\ldots$ the value

$$X(\zeta) \;=\; \liminf_{n\to\infty}\; \frac{1}{n}\sum_{i=0}^{n-1} \mathit{payoff}(s_i).$$

Let us further assume that we have given a rational probability threshold $\wp \in (0,1]$ and a cost bound $R \in \mathbb{N}$. The threshold $\wp$ is always represented as a fraction of two binary numbers. The bound $R$ is represented either in binary or in unary, which significantly influences the (computational) complexity of the studied problems.

**Definition 1 (Resilient schedulers).** *A scheduler $\mathfrak{S}$ is said to be* probabilistically resilient *with respect to $\wp$ and $R$ if the following conditions (Res) and (ASRep) hold for all finite $\mathfrak{S}$-paths $\pi$ from $s_{init}$ to an error state $s$:*

$$Pr^{\mathfrak{S}\uparrow\pi}_{\mathcal{M},s}\big(\; \Diamond^{\leqslant R}\, Op \;\big) \geqslant \quad \wp \tag{Res}$$

$$Pr^{\mathfrak{S}\uparrow\pi}_{\mathcal{M},s}\big(\; \Diamond\, Op \;\big) = \quad 1 \tag{ASRep}$$

*Here, $\Diamond\, Op$ denotes the set of infinite paths $\zeta$ for which there exist a finite path $\pi'$ and an infinite path $\varrho$ such that $\zeta = \pi';\varrho$ and the last state of $\pi'$ is in $Op$. Further, $\Diamond^{\leqslant R}\, Op$ denotes the set $\Diamond\, Op$ restricted to paths satisfying cost$(\pi') \leqslant R$.*

The task addressed in this paper is to check the existence of resilient schedulers (i.e., schedulers that are probabilistically resilient w.r.t. $\wp$ and $R$), and if so, construct an *optimal* resilient scheduler $\mathfrak{R}$ that has maximal long-run availability amongst all resilient schedulers, i.e., $\mathrm{Avail}^{\mathfrak{R}}_{\mathcal{M},s_{init}} = \mathrm{Avail}^{\max}_{\mathcal{M},s_{init}}$, where

$$\mathrm{Avail}^{\max}_{\mathcal{M},s_{init}} \;=\; \sup\,\big\{\; \mathrm{Avail}^{\mathfrak{R}'}_{\mathcal{M},s_{init}} \;:\; \mathfrak{R}' \text{ is a resilient scheduler } \big\}.$$

## 3 The Results

In the following, we present and prove our main result of this paper:

**Theorem 1.** *Let $(\mathcal{M}, Err, Op, rew)$ be an MDP with repair, $\wp \in (0,1]$ a rational probability threshold, and $R \in \mathbb{N}$ a cost bound encoded in unary. The existence of a probabilistically resilient scheduler w.r.t. $\wp$ and $R$ is decidable in polynomial time. If such a scheduler exists, then an* optimal *probabilistically resilient scheduler $\mathfrak{R}$ (w.r.t. $\wp$ and $R$) is computable in polynomial time.*

If $R$ is encoded in binary, our algorithms are exponential, and we show that even the existence of a probabilistically resilient scheduler w.r.t. $\wp$ and $R$ becomes PSPACE-hard. The optimal scheduler $\mathfrak{R}$ is randomized and history dependent, which is unavoidable (see the example in the introduction). More precisely, the

memory requirements of $\mathfrak{R}$ are finite with at most $|Err| \cdot R$ memory elements, and this memory is only used in the repairing phase where the scheduler needs to remember the error state and the total costs accumulated since visiting this error state.

For the rest of this section, we fix an MDP with repair $(\mathcal{M}, Err, Op, rew)$ where $\mathcal{M} = (S, Act, P, s_{init})$, a rational probability threshold $\wp \in (0, 1]$, and a cost bound $R \in \mathbb{N}$. We say that a scheduler is *resilient* if it is probabilistically resilient w.r.t. $\wp$ and $R$.

The proof of Theorem 1 is obtained in two steps. First, the MDP $\mathcal{M}$ is transformed into a suitable MDP $\hat{\mathcal{M}}$ where the total costs accumulated since the last error are explicitly remembered in the states. Hence, the size of $\hat{\mathcal{M}}$ is polynomial in the input size if $R$ is encoded in unary. We will show that the problem of computing an optimal resilient scheduler can be safely considered in $\hat{\mathcal{M}}$ instead of $\mathcal{M}$. In the second step, it is shown that there exists an optimal *memoryless* resilient scheduler for $\hat{\mathcal{M}}$ computable in time polynomial in the size of $\hat{\mathcal{M}}$. This is the very core of our paper requiring non-trivial observations and constructions. Roughly speaking, we start by connecting our problem to the problem of multiple mean-payoff optimization, and use the results and algorithms presented in [7] to analyze the limit behavior of resilient schedulers. First, we show how to compute the set of end components such that resilient schedulers can stay only in these end components without loosing availability. We also compute memoryless schedulers for these end components that can safely be adopted by resilient schedulers. Then, we show that the behavior of a resilient scheduler prior entering an end component can also be modified so that it becomes memoryless and the achieved availability does not decrease. After understanding the structure of resilient schedulers, we can compute an optimal memoryless resilient scheduler for $\hat{\mathcal{M}}$ by solving suitable linear programs.

The first step (i.e., the transformation of $\mathcal{M}$ into $\hat{\mathcal{M}}$) is described in Sect. 3.1, and the second step in Sect. 3.2.

### 3.1 Transformation

Let $(\hat{\mathcal{M}}, \hat{Err}, \hat{Op}, \hat{rew})$ be an MDP with repair where $\hat{\mathcal{M}}$ is an MDP $(\hat{S}, \hat{Act}, \hat{P}, s_{init})$ such that $\hat{S} = S \cup Rep$ with

$$Rep \;=\; Err \times S \times \{0, 1, \ldots, R\}.$$

Intuitively, state $\langle e, s, r \rangle \in Rep$ indicates that the system is in state $s$ executing a repair procedure that has been triggered by visiting $e \in Err$ somewhen in the past and with accumulated costs $r$ so far. For technical reasons, we also include triples $\langle e, s, r \rangle$ with $s \in Op$ in which case a repair mode with total cost $r$ has just finished. The sets of error and operational states in $\hat{\mathcal{M}}$ are:

$$\hat{Err} \;=\; Err \quad \text{and} \quad \hat{Op} \;=\; Op \cup \big\{\, \langle e, s, r \rangle \in Rep \;:\; s \in Op \,\big\}.$$

The action set of $\hat{\mathcal{M}}$ is the same as for $\mathcal{M}$. In what follows, we write $\hat{Act}(\hat{s})$ for the set of actions that are enabled in state $\hat{s}$ of $\hat{\mathcal{M}}$. Then, $\hat{Act}(s) = \hat{Act}(\langle e, s, r\rangle) = Act(s)$. Let $s, s' \in S$ and $\alpha \in Act$. Then, $\hat{P}(s, \alpha, s') = P(s, \alpha, s')$ if $s \notin Err$. If $e \in Err$ and $\alpha \in Act(e)$, then

$$\hat{P}\big(e, \alpha, \langle e, s, cost(e)\rangle\big) \;=\; P(e, \alpha, s)$$

For, $e \in Err$, $r \in \{0, 1, \ldots, R\}$, and $\alpha \in Act(s)$ we have:

$$\begin{aligned} \hat{P}\big(\langle e, s, r\rangle, \alpha, \langle e, s', r{+}cost(s)\rangle\big) &= P(s, \alpha, s') \qquad \text{if } r{+}cost(s) \leqslant R \text{ and } s \notin Op \\ \hat{P}\big(\langle e, s, r\rangle, \alpha, s'\big) &= P(s, \alpha, s') \qquad \text{if } r{+}cost(s) > R \text{ or } s \in Op \end{aligned}$$

In all remaining cases, we set $\hat{P}(\cdot) = 0$. The reward function $\hat{rew}$ of $\hat{\mathcal{M}}$ is given by $\hat{cost}(s) = \hat{cost}(\langle e, s, r\rangle) = cost(s)$ and $\hat{payoff}(s) = \hat{payoff}(\langle e, s, r\rangle) = payoff(s)$. Note that assumption (*) ensures that $s \notin Err$ for all states $\langle e, s, r\rangle$.

There is a one-to-one correspondence between the paths in $\mathcal{M}$ and in $\hat{\mathcal{M}}$. More precisely, given a (finite or infinite) path $\hat{\pi}$ in $\hat{\mathcal{M}}$, let $\hat{\pi}|_{\mathcal{M}}$ denote the unique path in $\mathcal{M}$ that arises from $\hat{\pi}$ by replacing each repair state $\langle e, s, r\rangle$ with $s$. Vice versa, each path $\pi$ in $\mathcal{M}$ can be lifted to a path $\pi|^{\hat{\mathcal{M}}}$ in $\hat{\mathcal{M}}$ such that $(\pi|^{\hat{\mathcal{M}}})|_{\mathcal{M}} = \pi$. Next lemmas follow directly from definitions of $\hat{cost}$ and $\hat{payoff}$.

**Lemma 1.** *For each finite path $\hat{\pi}$ in $\hat{\mathcal{M}}$ starting in some state $e \in \hat{Err}$ we have $\hat{cost}(\hat{\pi}) = cost(\hat{\pi}|_{\mathcal{M}})$.*

**Lemma 2.** *For each infinite path $\hat{\zeta}$ in $\hat{\mathcal{M}}$, $\hat{payoff}(\hat{\zeta}) = payoff(\hat{\zeta}|_{\mathcal{M}})$.*

The one-to-one correspondence between the paths in $\mathcal{M}$ and in $\hat{\mathcal{M}}$ carries over to the schedulers for $\mathcal{M}$ and $\hat{\mathcal{M}}$. Given a scheduler $\mathfrak{S}$ for $\mathcal{M}$, let $\mathfrak{S}|^{\hat{\mathcal{M}}}$ denote the scheduler for $\hat{\mathcal{M}}$ given by $\mathfrak{S}|^{\hat{\mathcal{M}}}(\hat{\pi}) = \mathfrak{S}(\hat{\pi}|_{\mathcal{M}})$ for all finite paths $\hat{\pi}$ of $\hat{\mathcal{M}}$. This yields a scheduler transformation $\mathfrak{S} \mapsto \mathfrak{S}|^{\hat{\mathcal{M}}}$ that maps each scheduler for $\mathcal{M}$ to a scheduler for $\hat{\mathcal{M}}$. Vice versa, given a scheduler $\hat{\mathfrak{S}}$ for $\hat{\mathcal{M}}$ there exists a scheduler $\hat{\mathfrak{S}}|_{\mathcal{M}}$ such that $\hat{\mathfrak{S}} = (\hat{\mathfrak{S}}|_{\mathcal{M}})|^{\hat{\mathcal{M}}}$.

Due to assumption (*) we have that $s \notin Err$ for all repair states $\langle e, s, r\rangle$ that are reachable from $e$ in $\hat{\mathcal{M}}$. Thus, with Lemmas 1 and 2, we obtain:

**Lemma 3.** *Let $\mathfrak{S}$ be a scheduler for $\mathcal{M}$ and $\hat{\mathfrak{S}}$ a scheduler for $\hat{\mathcal{M}}$ such that $\mathfrak{S} = \hat{\mathfrak{S}}|_{\mathcal{M}}$. Then:*

*(a) For each state $e \in Err$:* $Pr^{\mathfrak{S}}_{\mathcal{M},e}\big(\Diamond\, Op\big) \;=\; Pr^{\hat{\mathfrak{S}}}_{\hat{\mathcal{M}},e}\big(\Diamond\, \hat{Op}\big)$ *and*

$$Pr^{\mathfrak{S}}_{\mathcal{M},e}\big(\Diamond^{\leqslant R}\, Op\big) \;=\; Pr^{\hat{\mathfrak{S}}}_{\hat{\mathcal{M}},e}\big(\Diamond^{\leqslant R}\, \hat{Op}\big) \;=\; Pr^{\hat{\mathfrak{S}}}_{\hat{\mathcal{M}},e}\big(\bigcirc(Rep \cup Op_e)\big)$$

*where* $Op_e \;=\; \big\{\, \langle e, s, r\rangle \in Rep \;:\; s \in Op \,\big\}$.

*(b)* $\mathrm{Avail}^{\mathfrak{S}}_{\mathcal{M},s_{init}} \;=\; \mathrm{Avail}^{\hat{\mathfrak{S}}}_{\hat{\mathcal{M}},s_{init}}$

**Corollary 1.** $\mathrm{Avail}^{\max}_{\mathcal{M},s_{init}} \;=\; \mathrm{Avail}^{\max}_{\hat{\mathcal{M}},s_{init}}$

*Proof.* The above transformations $\pi \mapsto \pi|^{\hat{\mathcal{M}}}$ and $\mathfrak{S} \mapsto \mathfrak{S}|^{\hat{\mathcal{M}}}$ for paths and schedulers of $\mathcal{M}$ to paths and schedulers of $\hat{\mathcal{M}}$, and the inverse mappings $\hat{\pi} \mapsto \hat{\pi}|_{\mathcal{M}}$ and $\hat{\mathfrak{S}} \mapsto \hat{\mathfrak{S}}|_{\mathcal{M}}$ for paths and schedulers of $\hat{\mathcal{M}}$ to paths and schedulers of $\mathcal{M}$ are compatible with the residual operator for schedulers in the following sense:

$$(\mathfrak{S} \uparrow \pi)|^{\hat{\mathcal{M}}} = (\mathfrak{S}|^{\hat{\mathcal{M}}}) \uparrow (\pi|^{\hat{\mathcal{M}}}) \qquad \text{and} \qquad (\hat{\mathfrak{S}} \uparrow \hat{\pi})|_{\mathcal{M}} = (\hat{\mathfrak{S}}|_{\mathcal{M}}) \uparrow (\hat{\pi}|_{\mathcal{M}})$$

Thus, part (a) of Lemma 3 yields that $\hat{\mathfrak{S}}$ is resilient for $\hat{\mathcal{M}}$ if and only if $\mathfrak{S}$ is resilient for $\mathcal{M}$. Part (b) of Lemma 3 then yields the claim. □

The following mainly technical lemma shows that residual schedulers arising from resilient schedulers maintain the resilience property.

**Lemma 4.** *Let $\mathfrak{S}$ be a resilient scheduler for $\hat{\mathcal{M}}$, and let $s$ be a state of $\hat{\mathcal{M}}$ such that $s \notin Rep$. Let $\mathcal{P}$ be a set of finite $\mathfrak{S}$-paths initiated in $s_{init}$ and terminating in $s$, and let $\mathfrak{S}'$ be a scheduler for $\hat{\mathcal{M}}$ resilient for the initial state changed to $s$. Consider the scheduler $\mathfrak{S}[\mathcal{P}, \mathfrak{S}']$ which is the same as $\mathfrak{S}$ except that for every finite path $w$ such that $w = w'; w''$ where $w' \in \mathcal{P}$ we have that $\mathfrak{S}[\mathcal{P}, \mathfrak{S}'](w) = \mathfrak{S}'(w'')$. Then $\mathfrak{S}[\mathcal{P}, \mathfrak{S}']$ is resilient (for the initial state $s_{init}$).*

### 3.2 Solving the Resilience-Availability Problem for $\hat{\mathcal{M}}$

In this section, we analyze the structure of resilient schedulers for $\hat{\mathcal{M}}$ and prove the following proposition:

**Proposition 1.** *The existence of a resilient scheduler for $\hat{\mathcal{M}}$ can be decided in polynomial time. The existence of some resilient scheduler for $\hat{\mathcal{M}}$ implies the existence of an optimal memoryless resilient scheduler for $\hat{\mathcal{M}}$ computable in polynomial time.*

Note that Theorem 1 follows immediately from Proposition 1 and Corollary 1.

We start by introducing some notions. A *fragment* of $\hat{\mathcal{M}} = (\hat{S}, \hat{Act}, \hat{P}, s_{init})$ is a pair $(F, \mathcal{A})$ where $F \subseteq \hat{S}$ and $\mathcal{A}\colon F \to 2^{\hat{Act}}$ is a function such that $\mathcal{A}(s) \neq \varnothing$ and $\mathcal{A}(s) \subseteq \hat{Act}(s)$ for every $s \in F$. An *MR-scheduler for* $(F, \mathcal{A})$ is a function $\mathfrak{S}_F$ assigning a probability distribution over $\mathcal{A}(s)$ to every $s \in F$. We say that a scheduler $\mathfrak{S}$ for $\hat{\mathcal{M}}$ is *consistent* with $\mathfrak{S}_F$ if for every $\pi \in FinPaths$ ending in a state of $F$ we have that $\mathfrak{S}(\pi) = \mathfrak{S}_F(\pi)$.

An *end component* of $\hat{\mathcal{M}}$ is a fragment $(E, \mathcal{A})$ of $\hat{\mathcal{M}}$ such that

- $(E, \mathcal{A})$ is strongly connected, i.e., for all $s, s' \in E$ there is a finite path $s_0 \alpha_0 s_1 \dots \alpha_{n-1} s_n$ from $s = s_0$ to $s' = s_n$ such that $s_i \in E$ and $\alpha_i \in \mathcal{A}(s_i)$ for all $0 \leq i < n$;
- for all $s \in E$, $\alpha \in \mathcal{A}(s)$, and $s' \in \hat{S}$ such that $\hat{P}(s, \alpha, s') > 0$ we have $s' \in E$.

Let $\mathfrak{S}$ be a scheduler for $\hat{\mathcal{M}}$ (not necessarily resilient). For every infinite path $\zeta$, let $F_\zeta$ be the set of states occurring infinitely often in $\zeta$. For every $s \in F_\zeta$, let $\mathcal{A}_\zeta(s)$ be the set of all actions executed infinitely often from $s$ along $\zeta$. For

a fragment $(F, \mathcal{A})$, let $Path(F, \mathcal{A})$ be the set of all infinite paths $\zeta$ such that $F_\zeta = F$ and $\mathcal{A}_\zeta = \mathcal{A}$, and let $\Pr^{\mathfrak{S}}_{\hat{\mathcal{M}}, s_{init}}(F, \mathcal{A})$ be the probability of all $\zeta \in Path(F, \mathcal{A})$ starting in $s_{init}$. If $(F, \mathcal{A})$ is *not* an end component, then clearly $\Pr^{\mathfrak{S}}_{\hat{\mathcal{M}}, s_{init}}(F, \mathcal{A}) = 0$. Hence, there are end components $(F_1, \mathcal{A}_1), \ldots, (F_m, \mathcal{A}_m)$ such that:

$$\Pr^{\mathfrak{S}}_{\hat{\mathcal{M}}, s_{init}}(F_i, \mathcal{A}_i) > 0 \text{ for all } i \leq m, \text{ and } \sum_{i=1}^{m} \Pr^{\mathfrak{S}}_{\hat{\mathcal{M}}, s_{init}}(F_i, \mathcal{A}_i) = 1$$

We say that $\mathfrak{S}$ *stays* in these end components.

Proposition 1 is proved as follows. We show that there is a set $\mathcal{E}$, computable in time polynomial in $|\hat{\mathcal{M}}|$, consisting of triples of the form $(E, \mathcal{A}, \mathfrak{S}_E)$ such that $(E, \mathcal{A})$ is an end component of $\hat{\mathcal{M}}$ and $\mathfrak{S}_E$ is an MR-scheduler for $(E, \mathcal{A})$, satisfying the following conditions (E1) and (E2):

**(E1)** If $(E, \mathcal{A}, \mathfrak{S}_E), (E', \mathcal{A}', \mathfrak{S}_{E'}) \in \mathcal{E}$, then the two triples are either the same or $E \cap E' = \varnothing$.

**(E2)** Every $(E, \mathcal{A}, \mathfrak{S}_E) \in \mathcal{E}$ is *strongly connected*, i.e., the directed graph $(E, \rightarrow)$, where $s \rightarrow s'$ iff there is some $\alpha \in \mathcal{A}(s)$ such that $\mathfrak{S}_E(s)(\alpha) > 0$ and $\hat{P}(s, \alpha, s') > 0$, is strongly connected. (In this case, $E$ is a bottom strongly connected component of the Markov chain induced by $\mathfrak{S}_E$.)

Further, we can safely restrict ourselves to resilient schedulers whose long-run behavior is captured by some subset $\mathcal{E}' \subseteq \mathcal{E}$ in the following sense:

**Lemma 5.** *Given the set $\mathcal{E}$, for every resilient scheduler $\mathfrak{R}$ there exist a set $\mathcal{E}' \subseteq \mathcal{E}$ and a resilient scheduler $\mathfrak{R}'$ such that*

- *almost all $\mathfrak{R}'$-paths starting in $s_{init}$ visit a state of $\bigcup_{(E,\mathcal{A},\mathfrak{S}_E)\in\mathcal{E}'} E$,*
- *$\mathfrak{R}'$ is consistent with $\mathfrak{S}_E$ for every $(E, \mathcal{A}, \mathfrak{S}_E) \in \mathcal{E}'$,*
- $\text{Avail}^{\mathfrak{R}}_{\hat{\mathcal{M}}, s_{init}} \leq \text{Avail}^{\mathfrak{R}'}_{\hat{\mathcal{M}}, s_{init}}$.

Using Lemma 5, we prove the following:

**Lemma 6.** *Given the set $\mathcal{E}$, there is a linear program $\mathcal{L}$ computable in time polynomial in $|\hat{\mathcal{M}}|$ satisfying the following: If $\mathcal{L}$ is not feasible, then there is no resilient scheduler for $\hat{\mathcal{M}}$. Otherwise, there is a subset $\mathcal{E}' \subseteq \mathcal{E}$ and an MR-scheduler $\mathfrak{S}_F$ for the fragment $(F, \mathcal{A})$ with $F = \hat{S} \setminus \bigcup_{(E,\mathcal{A},\mathfrak{S}_E)\in\mathcal{E}'} E$ and $\mathcal{A}(s) = \hat{Act}(s)$ for every $s \in F$ such that*

- *$\mathcal{E}'$ and $\mathfrak{S}_F$ are computable in time polynomial in $|\hat{\mathcal{M}}|$,*
- *the scheduler $\mathfrak{R}$ consistent with $\mathfrak{S}_F$ and $\mathfrak{S}_E$ for every $(E, \mathcal{A}, \mathfrak{S}_E) \in \mathcal{E}'$ is resilient, and*
- *for every resilient scheduler $\mathfrak{R}'$ we have that $\text{Avail}^{\mathfrak{R}}_{\hat{\mathcal{M}}, s_{init}} \geq \text{Avail}^{\mathfrak{R}'}_{\hat{\mathcal{M}}, s_{init}}$.*

In the next subsections, we show how to compute the set $\mathcal{E}$ satisfying conditions (E1) and (E2) in polynomial time and provide proofs for Lemmas 5 and 6. Note that Proposition 1 then follows from Lemma 6 and the polynomial-time computability of $\mathcal{E}$.

**Constructing the Set** $\mathcal{E}$. For each $e \in Err$, we define the weight function $wgt_e \colon \hat{S} \to \mathbb{Q}$ given by

$$wgt_e(\langle e, s, r\rangle) = 1-\wp \quad \text{if } s \in Op$$
$$wgt_e(\langle e, s, r\rangle) = -\wp \quad \text{if } s \notin Op \text{ and } r+cost(s) > R$$

and $wgt_e(\hat{s}) = 0$ otherwise (in particular, for all states in $\hat{s} \in \hat{S}$ that do not have the form $\langle e, s, r\rangle$). For every scheduler $\mathfrak{S}$, let $\mathrm{MP}_e^{\mathfrak{S}}$ be the expected value (under $\Pr^{\mathfrak{S}}_{\hat{\mathcal{M}}, s_{init}}$) of the random variable $X_e$ assigning to each infinite $\mathfrak{S}$-path $\zeta = s_0\, \alpha_0\, s_1\, \alpha_1\, s_2\, \alpha_2 \ldots$ the value

$$X_e(\zeta) \;=\; \liminf_{n\to\infty} \tfrac{1}{n} \sum_{i=0}^{n-1} wgt_e(s_i).$$

We say that a scheduler $\mathfrak{S}$ for $\hat{\mathcal{M}}$ is *average-resilient* if $\mathrm{MP}_e^{\mathfrak{S}} \geq 0$ for all $e \in Err$. Note that if $\mathfrak{R}$ is a resilient scheduler for $\hat{\mathcal{M}}$, then $X_e(\zeta) \geq 0$ for almost all $\zeta$ (this follows by a straightforward application of the strong law of large numbers). Thus, we obtain:

**Lemma 7.** *Every resilient scheduler for $\hat{\mathcal{M}}$ is average-resilient.*

Although an average-resilient scheduler for $\hat{\mathcal{M}}$ is not necessarily resilient, we show that the problems of maximizing the long-run availability under resilient and average-resilient schedulers are to some extent related. The latter problem can be solved by the algorithm of [7]. More precisely, by Theorem 4.1 of [7], one can compute a linear program $\mathcal{L}_{\hat{\mathcal{M}}}$ in time polynomial in $|\hat{\mathcal{M}}|$ such that:

- if $\mathcal{L}_{\hat{\mathcal{M}}}$ is not feasible, then there is no average-resilient scheduler for $\hat{\mathcal{M}}$;
- otherwise, there is a 2-memory stochastic update scheduler $\mathfrak{H}$ for $\hat{\mathcal{M}}$, constructible in time polynomial in $|\hat{\mathcal{M}}|$, which is average-resilient and achieves the maximal long-run availability among all average-resilient schedulers.

The scheduler $\mathfrak{H}$ almost surely "switches" from its initial mode to its second mode where it behaves memoryless. Hence, there is a set $\mathcal{E}_{\mathfrak{H}}$ (computable in time polynomial in $|\hat{\mathcal{M}}|$) comprising triples $(E, \mathcal{A}, \mathfrak{H}_E)$ that enjoy the following properties (H1) and (H2):

**(H1)** $(E, \mathcal{A})$ is an end component of $\hat{\mathcal{M}}$ and $\mathfrak{H}_E$ is an MR-scheduler for $(E, \mathcal{A})$ achieving the *maximal* long-run availability among all average-resilient schedulers for every initial state $s \in E$.

**(H2)** If $(E, \mathcal{A}, \mathfrak{H}_E), (E', \mathcal{A}', \mathfrak{H}_{E'}) \in \mathcal{E}_{\mathfrak{H}}$, then the two triples are either the same or $E \cap E' = \varnothing$. Further, every $(E, \mathcal{A}, \mathfrak{H}_E) \in \mathcal{E}_{\mathfrak{H}}$ is strongly connected.

We show that for every $(E, \mathcal{A}, \mathfrak{H}_E) \in \mathcal{E}_{\mathfrak{H}}$ and every $s \in E$, the scheduler $\mathfrak{H}_E$ is *resilient* when the initial state is changed to $s$ (see Lemma 10). So, $\mathfrak{H}$ starts to behave like a resilient scheduler after a "switch" to some $(E, \mathcal{A}, \mathfrak{H}_E) \in \mathcal{E}_{\mathfrak{H}}$. However, in the initial transient phase, $\mathfrak{H}$ may violate the resilience condition, which may disallow a resilient scheduler $\mathfrak{R}$ to enter some of the end components

**Algorithm 1.** Computing the set $\mathcal{E}$.

**input** : the transformed MDP $\hat{\mathcal{M}}$
**output**: the set $\mathcal{E}$ satisfying (E1) and (E2)

$\mathcal{Q} := \hat{\mathcal{M}}$, $s := s_{init}$, $\mathcal{E} := \emptyset$
**repeat**
  Compute the linear program $\mathcal{L}_{\mathcal{Q}}$
  **if** $\mathcal{L}_{\mathcal{Q}}$ *is feasible* **then**
    compute the scheduler $\mathfrak{H}$ and the set $\mathcal{E}_{\mathfrak{H}}$ satisfying (H1) and (H2)
    $\mathcal{E} := \mathcal{E} \cup \mathcal{E}_{\mathfrak{H}}$
    $\mathcal{Q} := \mathcal{Q} \ominus \mathcal{E}_{\mathfrak{H}}$
  **else**
    $\mathcal{Q} := \mathcal{Q} \ominus \{s\}$
  **if** $s$ *is not a state of* $\mathcal{Q}$ **then**
    $s :=$ some state of $\mathcal{Q}$
**until** $\mathcal{Q}$ *becomes empty*
**return** $\mathcal{E}$

of $\mathcal{E}_{\mathfrak{H}}$. Thus, a resilient scheduler $\mathfrak{R}$ can in general be forced to stay in an end component that does not appear in $\mathcal{E}_{\mathfrak{H}}$. So, the set $\mathcal{E}$ needs to be *larger* than $\mathcal{E}_{\mathfrak{H}}$, and we show that a sufficiently large $\mathcal{E}$ is computable in polynomial time by Algorithm 1.

Algorithm 1 starts by initializing $\mathcal{Q}$ to $\hat{\mathcal{M}}$, $s$ to $s_{init}$, and $\mathcal{E}$ to $\emptyset$. Then, it computes the linear program $\mathcal{L}_{\mathcal{Q}}$ and checks its feasibility. If $\mathcal{L}_{\mathcal{Q}}$ is not feasible, the initial state $s$ of $\mathcal{Q}$ is removed from $\mathcal{Q}$ in the way described below. Otherwise, the algorithm constructs the scheduler $\mathfrak{H}$, adds $\mathcal{E}_{\mathfrak{H}}$ to $\mathcal{E}$, and "prunes" $\mathcal{Q}$ into $\mathcal{Q} \ominus \mathcal{E}_{\mathfrak{H}}$. If the state $s$ is deleted from $\mathcal{Q}$, some state of $\mathcal{Q}$ is chosen as a new initial state. This goes on until $\mathcal{Q}$ becomes empty. Here, the MDP $\mathcal{Q} \ominus X$ is the largest MDP subsumed by $\mathcal{Q}$ which does not contain the states in $X \subseteq \hat{S}$. Note that when a state of $\mathcal{Q}$ is deleted, all actions leading to this state must be disabled; and if all outgoing actions of a state $s$ are disabled, then $s$ must be deleted. Hence, deleting the states appearing in $\mathcal{E}_{\mathfrak{H}}$ may enforce deleting additional states and disabling further actions. Note that every $(E, \mathcal{A}, \mathfrak{H}_E) \in \mathcal{E}$ is obtained in some iteration of the repeat-until cycle of Algorithm 1 by constructing the scheduler $\mathfrak{H}$ for the current value of $\mathcal{Q}$. We denote this MDP $\mathcal{Q}$ as $\mathcal{Q}_E$ (note that $\mathcal{Q}_E$ is not necessarily connected). The set $\mathcal{E}$ returned by Algorithm 1 indeed satisfies conditions (E1) and (E2). The outcome $\mathcal{E} = \varnothing$ is possible, in which case there is no resilient scheduler for $\hat{\mathcal{M}}$ as the linear program $\mathcal{L}$ of Lemma 6 is not feasible for $\mathcal{E} = \varnothing$.

An immediate consequence of property (H1) is the following:

**Lemma 8.** *Let* $(E, \mathcal{A}, \mathfrak{H}_E) \in \mathcal{E}$ *and* $s \in E$. *Then* $\mathfrak{H}_E$ *achieves the* maximal *long-run availability for the initial state* $s$ *among all average-resilient schedulers for* $\mathcal{Q}_E$.

The next lemma follows easily from the construction of $\mathcal{E}$.

**Lemma 9.** *Let $\mathfrak{S}$ be a scheduler for $\hat{\mathcal{M}}$ (not necessarily resilient) and let $(F, \mathcal{B})$ be an end component where $\mathfrak{S}$ stays with positive probability. Then there is $(E, \mathcal{A}, \mathfrak{H}_E) \in \mathcal{E}$ such that $(F, \mathcal{B})$ is an end component of $\mathcal{Q}_E$ and $F \cap E \neq \varnothing$.*

Let $(E, \mathcal{A}, \mathfrak{H}_E) \in \mathcal{E}$. Since $\mathfrak{H}_E$ is an MR-scheduler, the behavior of $\mathfrak{H}_E$ in an error state $f \in E$ (for an arbitrary initial state $s \in E$) is independent of the history. That is, the resilience condition is either simultaneously satisfied or simultaneously violated for all visits to $f$. However, if the second case holds, $\mathfrak{H}_E$ is not even average-resilient, what is a contradiction. Thus, we obtain:

**Lemma 10.** *Let $(E, \mathcal{A}, \mathfrak{H}_E) \in \mathcal{E}$, and let $s \in E$. Then the scheduler $\mathfrak{H}_E$ is* resilient *when the initial state is changed to $s$. Further, if $\mathfrak{R}$ is a resilient scheduler for $\mathcal{Q}_E$ with the initial state $s$, then $\mathrm{Avail}^{\mathfrak{H}_E}_{\mathcal{Q}_E,s} \geq \mathrm{Avail}^{\mathfrak{R}}_{\mathcal{Q}_E,s}$.*

**Proof of Lemma** 5. Let $\mathfrak{R}$ be a resilient scheduler for $\hat{\mathcal{M}}$. We show that there is another resilient scheduler $\mathfrak{R}'$ satisfying the conditions of Lemma 5. First, let us consider the end components $(F_1, \mathcal{B}_1), \ldots, (F_m, \mathcal{B}_m)$ where $\mathfrak{R}$ stays. For every $(F_i, \mathcal{B}_i)$, let $(E, \mathcal{A}, \mathfrak{H}_E) \in \mathcal{E}$ be a triple with the maximal $\mathrm{Avail}(E)$ such that $F_i \cap E \neq \varnothing$ (such a triple exists due to Lemma 9). We say that $(E, \mathcal{A}, \mathfrak{H}_E)$ is *associated* to $(F_i, \mathcal{B}_i)$. Let $\mathrm{Avail}(F_i, \mathcal{B}_i)$ be the *conditional availability* w.r.t. scheduler $\mathfrak{R}$ under the condition that an infinite path initiated in $s_{init}$ stays in $(F_i, \mathcal{B}_i)$. Given a triple $(E, \mathcal{A}, \mathfrak{H}_E) \in \mathcal{E}$, we use $\mathrm{Avail}(E)$ to denote the availability achieved by scheduler $\mathfrak{H}_E$ for $s$. Note that $\mathrm{Avail}(E)$ is independent of $s$.

**Lemma 11.** $\mathrm{Avail}(F_i, \mathcal{B}_i) \leq \mathrm{Avail}(E)$, *where $(E, \mathcal{A}, \mathfrak{H}_E) \in \mathcal{E}$ is the triple associated to $(F_i, \mathcal{B}_i)$.*

Further, we say that $(F_i, \mathcal{B}_i)$ is *offending* if there is a finite $\mathfrak{R}$-path $\pi$ initiated in $s_{init}$ ending in a state $s \in E$, where $(E, \mathcal{A}, \mathfrak{H}_E)$ is associated to $(F_i, \mathcal{B}_i)$, such that $s \notin \mathit{Rep}$ and the availability achieved by the scheduler $\mathfrak{R} \uparrow \pi$ in $s$ is *strictly larger* than $\mathrm{Avail}(E)$. Note that if no $(F_i, \mathcal{B}_i)$ is offending, we can choose $\mathcal{E}'$ as the set of triples associated to $(F_1, \mathcal{B}_1), \ldots, (F_m, \mathcal{B}_m)$, and redefine the scheduler $\mathfrak{R}$ into a resilient scheduler $\mathfrak{R}'$ as follows: $\mathfrak{R}'$ behaves exactly like $\mathfrak{R}$ until a state $s$ of some $(E, \mathcal{A}, \mathfrak{H}_E) \in \mathcal{E}'$ is visited. Then, $\mathfrak{R}'$ switches to $\mathfrak{H}_E$ immediately. The scheduler $\mathfrak{R}'$ is resilient because $s \notin \mathit{Rep}$ (a visit to a repair state is preceded by a visit to the associated fail state which also belongs to $E$) and hence we can apply Lemma 4. Clearly, $\mathfrak{R}'$ is consistent with every $\mathfrak{H}_E$ such that $(E, \mathcal{A}, \mathfrak{H}_E) \in \mathcal{E}'$. It remains to show that the availability achieved by $\mathfrak{R}'$ in $s_{init}$ is not smaller than the one achieved by $\mathfrak{R}$. This follows immediately by observing that whenever $\mathfrak{R}'$ makes a switch to $\mathfrak{H}_E$ after performing a finite $\mathfrak{R}$-path initiated in $s_{init}$ ending in $s \in E$, the availability achieved by the resilient scheduler $\mathfrak{R} \uparrow \pi$ for the initial state $s$ must be bounded by $\mathrm{Avail}(E)$, because otherwise some $(F_i, \mathcal{B}_i)$ would be offending. So, the introduced "switch" can only increase the availability.

Now assume that $(F_m, \mathcal{B}_m)$ is offending, and let $(E, \mathcal{A}, \mathfrak{H}_E)$ be the triple associated to $(F_m, \mathcal{B}_m)$. We construct a resilient scheduler $\tilde{\mathfrak{R}}$ which stays in $(F_1, \mathcal{B}_1), \ldots, (F_{m-1}, \mathcal{B}_{m-1})$ and achieves availability not smaller than the one achieved by $\mathfrak{R}$. This completes the proof of Lemma 5, because we can then

successively remove all offending pairs. Since $(F_m, \mathcal{B}_m)$ is offending, there is a finite $\mathfrak{R}$-path $\pi$ initiated in $s_{init}$ ending in a state $s \in E$ such that $s \notin \mathit{Rep}$ and the availability $A$ achieved by $\mathfrak{R} \upharpoonright \pi$ in $s$ is larger than $\mathrm{Avail}(E)$. Since $F_m \cap E \neq \varnothing$, there is a state $t \notin \mathit{Rep}$ such that $t \in F_m \cap E$. Note that $\mathfrak{H}_E$ is resilient for the initial state $t$, and almost all infinite paths initiated in $t$ visit the state $s$ under the scheduler $\mathfrak{H}_E$.

Now, we construct a resilient scheduler $\mathfrak{S}_s$ achieving availability at least $A$ in $s$ such that all components where $\mathfrak{S}_s$ stays (for the initial state $s$) are among $(F_1, \mathcal{B}_1), \ldots, (F_{m-1}, \mathcal{B}_{m-1})$. Let $P_m$ be the probability that an infinite path initiated in $s$ stays in $(F_m, \mathcal{B}_m)$ under the scheduler $\mathfrak{R} \upharpoonright \pi$. If $P_m = 0$, we put $\mathfrak{S}_s = \mathfrak{R} \upharpoonright \pi$. Now assume $P_m > 0$. We cannot have $P_m = 1$, because then $A$ is bounded by $\mathrm{Avail}(E)$ (see Lemma 11). Let $B$ be the conditional availability achieved in $s$ by $\mathfrak{R} \upharpoonright \pi$ under the condition that an infinite path initiated in $s$ stays in $(F_1, \mathcal{B}_1), \ldots, (F_{m-1}, \mathcal{B}_{m-1})$. Since $A \leq (1-P_m) \cdot B + P_m \cdot \mathrm{Avail}(E)$ and $A > \mathrm{Avail}(E)$, we obtain $B > A$. For every $\varepsilon > 0$, let $\mathit{\Pi}^\varepsilon$ be the set of all finite $(\mathfrak{R} \upharpoonright \pi)$-paths $\pi'$ initiated in $s$ and ending in $t$ such that the probability of all infinite paths initiated in $t$ staying in $(F_m, \mathcal{B}_m)$ under the scheduler $\mathfrak{R} \upharpoonright (\pi; \pi')$ is at least $1-\varepsilon$. Note that each $(\mathfrak{R} \upharpoonright \pi)$-path initiated in $s$ and staying in $(F_m, \mathcal{B}_m)$ is included in $(\mathfrak{R} \upharpoonright \pi)$-paths starting with a prefix of $\mathit{\Pi}^\varepsilon$. Hence, a smart redirection of the strategy after passing via $\mathit{\Pi}^\varepsilon$ can avoid staying in $(F_m, \mathcal{B}_m)$. We use $P^\varepsilon_m$ to denote the probability (under the scheduler $\mathfrak{R} \upharpoonright \pi$) of all infinite paths initiated in $s$ starting with a prefix of $\mathit{\Pi}^\varepsilon$, and $B^\varepsilon$ to denote the conditional availability achieved in $s$ by $\mathfrak{R} \upharpoonright \pi$ under the condition that an infinite path initiated in $s$ does *not* start with a prefix of $\mathit{\Pi}^\varepsilon$. Since $\lim_{\varepsilon \to 0} P^\varepsilon_m = P_m$ and $\lim_{\varepsilon \to 0} B^\varepsilon = B$, we can fix a sufficiently small $\delta > 0$ where

I. $\delta \cdot M + (1-\delta) \cdot \mathrm{Avail}(E) < A$, where $M$ is the maximal payoff assigned to a state of $\hat{\mathcal{M}}$.
II. conditional bound $B^\delta > A$.

The scheduler $\mathfrak{S}_s$ is defined in the following way, where $\mathit{\Sigma}$ denotes the set of all finite paths $\varrho$ initiated in $t$ and ending in $s$, such that the state $s$ is visited by $\varrho$ only once:

$$\mathfrak{S}_s(\pi') = \begin{cases} \mathfrak{S}_s(\pi'') & \text{if } \pi' = \hat{\pi}; \varrho; \pi'' \text{ where } \hat{\pi} \in \mathit{\Pi}^\delta \text{ and } \varrho \in \mathit{\Sigma}, \\ \mathfrak{H}_E(\pi'') & \text{if } \pi' = \hat{\pi}; \pi'' \text{ where } \hat{\pi} \in \mathit{\Pi}^\delta \text{ and no prefix of } \pi'' \text{ is in } \mathit{\Sigma}, \\ (\mathfrak{R} \upharpoonright \pi)(\pi') & \text{otherwise.} \end{cases}$$

Intuitively, $\mathfrak{S}_s$ simulates $\mathfrak{R} \upharpoonright \pi$ unless a path of $\mathit{\Pi}^\delta$ is produced, in which case $\mathfrak{S}_s$ temporarily "switches" to $\mathfrak{H}_E$ until $s$ is revisited and the simulation of $\mathfrak{R} \upharpoonright \pi$ is restarted. It is easy to verify that $\mathfrak{S}_s$ is a resilient scheduler achieving availability equal to $B^\delta > A$ staying in end components $(F_i, \mathcal{B}_i)$ with $i < m$.

Now we can easily construct the scheduler $\tilde{\mathfrak{R}}$. Let $\mathit{\Xi}^\delta$ be the set of all finite paths $\pi$ initiated in $s_{init}$ and ending in $t$ where the probability of all infinite paths initiated in $t$ staying in $(F_m, \mathcal{B}_m)$ is at least $1-\delta$. The scheduler $\tilde{\mathfrak{R}}$ behaves as $\mathfrak{R}$ unless a path of $\mathit{\Xi}^\delta$ is produced, in which case $\tilde{\mathfrak{R}}$ temporarily switches to

$\mathfrak{H}_E$ until the state $s$ is reached, and then it permanently switches to $\mathfrak{S}_s$. The availability achieved by $\tilde{\mathfrak{R}}$ in $s_{init}$ can be only larger that the availability achieved by $\mathfrak{R}$ due to Conditions I and II above.

**Proof of Lemma 6.** Let $\mathcal{E}$ denote the set of triples computed by Algorithm 1. Due to Lemma 5, we can concentrate on schedulers those paths almost surely reach subsets $\mathcal{E}' \subseteq \mathcal{E}$ and are consistent with the schedulers in $\mathcal{E}'$. Observe that the transient prefix of each path then has no effect on the long-run availability of the path and just influences the reachability probability distribution on $\mathcal{E}$. The resulting availability then is a convex combination of availabilities of the triples in $\mathcal{E}$. Thus, the aim is to find a *resilient* scheduler that maximizes this convex combination. We do so by constructing an MDP $\mathcal{N}$ where the resilient MR-scheduler $\mathfrak{R}_{\mathcal{N}}$ with optimal reachability reward induces optimal resilient scheduler in $\hat{\mathcal{M}}$. We show that $\mathfrak{R}_{\mathcal{N}}$ can be obtained from a slightly modified linear program of [17,19].

Let $\mathcal{N} = (S_{\mathcal{N}}, Act_{\mathcal{N}}, P_{\mathcal{N}}, s_{init})$ be an MDP over the state space

$$S_{\mathcal{N}} \;=\; \hat{S} \cup \big\{\, goal_E \,:\, (E, \mathcal{A}, \mathfrak{H}_E) \in \mathcal{E} \,\big\} \cup \{goal\}$$

and the action space $Act_{\mathcal{N}} = \hat{Act} \cup \{\tau\}$, where $\tau$ is a fresh action symbol. The transition probabilities $P_{\mathcal{N}}$ are defined as for $\hat{\mathcal{M}}$, but with additional $\tau$-transitions for each $(E, \mathcal{A}, \mathfrak{H}_E) \in \mathcal{E}$:

- from each state $\hat{s} \in E \cap \hat{Op}$ to $goal_E$, i.e., $P_{\mathcal{N}}(\hat{s}, \tau, goal_E) = 1$,
- from $goal_E$ to $goal$, i.e., $P_{\mathcal{N}}(goal_E, \tau, goal) = 1$, and
- from $goal$ to $goal$, i.e., $P_{\mathcal{N}}(goal, \tau, goal) = 1$.

The reward function in $\mathcal{N}$ is given by $rew(goal_E) = \mathrm{Avail}(E)$ for each $goal_E \in S_{\mathcal{N}}$ and $rew(s) = 0$ for all the remaining states $s \in \hat{S} \cup \{goal\}$. Given a scheduler $\mathfrak{S}$, the random variable $TR$ assigns to an infinite $\mathfrak{S}$-path $\zeta = s_0\, \alpha_0\, s_1\, \alpha_1\, s_2\, \alpha_2 \ldots$ the total accumulated reward $TR(\zeta) = \sum_{i=0}^{\infty} rew(s_i)$. The expected total accumulated reward from a state $s \in S_{\mathcal{N}}$ is denoted by $\mathbb{E}^{\mathfrak{S}}_{\mathcal{N},s}[TR]$.

**Lemma 12.** *Let $\mathfrak{R}'$ be a resilient scheduler for $\hat{\mathcal{M}}$ such that $\mathfrak{R}'$-paths from $s_{init}$ almost surely reach a subset $\mathcal{E}' \subseteq \mathcal{E}$ and is consistent with the schedulers in $\mathcal{E}'$. Then, there is a resilient scheduler $\mathfrak{R}$ for $\mathcal{N}$ where the $\mathfrak{R}$-paths from $s_{init}$ almost surely reach goal and*

$$\mathrm{Avail}^{\mathfrak{R}'}_{\hat{\mathcal{M}}, s_{init}} = \mathbb{E}^{\mathfrak{R}}_{\mathcal{N}, s_{init}}[TR].$$

From $\mathfrak{R}'$ we can easily construct an equivalent scheduler $\mathfrak{R}$ by redefining $\mathfrak{R}'$ to almost surely perform $\tau$ actions in $E \cap \hat{Op}$ for $(E, \mathcal{A}, \mathfrak{H}_E) \in \mathcal{E}'$. From Lemmas 5 and 12 it follows that if there is no resilient scheduler for $\mathcal{N}$ there is no resilient scheduler for $\hat{\mathcal{M}}$. Let $\mathfrak{R}_{\mathcal{N}}$ be the resilient scheduler that acquires the supremum of the expected total accumulated rewards from $s_{init}$ among all resilient schedulers for $\mathcal{N}$ that reach *goal* almost surely from $s_{init}$. As we shall see bellow, we can safely assume that $\mathfrak{R}_{\mathcal{N}}$ is an MR-scheduler. The technical details for proving the following lemma can be found in [4].

**Lemma 13.** *Let $\mathfrak{R}_{\mathcal{N}}$ be an MR-scheduler that acquires maximal $\mathbb{E}^{\mathfrak{R}'}_{\mathcal{N},s_{init}}[TR]$ within resilient schedulers $\mathfrak{R}'$ for $\mathcal{N}$ such that almost all $\mathfrak{R}'$-paths reach the goal. Let $\mathcal{E}'$ be the set of all $(E,\mathcal{A},\mathfrak{H}_E) \in \mathcal{E}$ such that $goal_E$ is visited from $s_{init}$ with positive probability under $\mathfrak{R}_{\mathcal{N}}$, and let $\mathfrak{S}_e(s) = \mathfrak{R}_{\mathcal{N}}(s)$ for each $s \in F$ where $F = \hat{S} \setminus \bigcup_{(E,\mathcal{A},\mathfrak{S}_E)\in\mathcal{E}'} E$. Moreover, let $\mathfrak{R}$ be the unique scheduler consistent with $\mathfrak{S}_e$ and $\mathfrak{H}_E$ for each $(E,\mathcal{A},\mathfrak{H}_E) \in \mathcal{E}'$. It holds that*

$$\mathrm{Avail}^{\mathfrak{R}}_{\hat{\mathcal{M}},s_{init}} = \mathbb{E}^{\mathfrak{R}_{\mathcal{N}}}_{\mathcal{N},s_{init}}[TR].$$

Note that the scheduler $\mathfrak{R}$ of Lemma 13 simulates the scheduler $\mathfrak{R}_{\mathcal{N}}$ only until a state of $\mathcal{E}'$ is visited (not until $\mathfrak{R}_{\mathcal{N}}$ visits a $goal_E$ state). This is the main subtlety hidden in Lemma 13.

*A resiliency linear program.* To obtain $\mathfrak{R}_{\mathcal{N}}$, let us consider the following linear program clearly constructible in polynomial time in $|\mathcal{N}|$ (and thus also in $|\hat{\mathcal{M}}|$). Intuitively, the variables $y_{t,\alpha}$ stand for the expected number of times an action $\alpha \in Act_{\mathcal{N}}$ is taken from state $t \in S_{\mathcal{N}}$. We set $y_t = \sum_{\alpha\in Act_{\mathcal{N}}(t)} y_{t,\alpha}$ and define

(1) flow equation: for all states $s \in S_{\mathcal{N}} \setminus \{goal\}$

$$y_s \quad = \quad \delta(s,s_{init}) \; + \; \sum\nolimits_{t\in S_{\mathcal{N}}} \sum\nolimits_{\alpha\in Act_{\mathcal{N}}(t)} y_{t,\alpha} \cdot P_{\mathcal{N}}(t,\alpha,s)$$

where $\delta(s,s_{init})$ is 1 if $s = s_{init}$, and 0 otherwise.

(2) non-negativeness: $y_{s,\alpha} \geqslant 0$ for all state-action pairs $(s,\alpha)$.

(3) flow equation for the goal state: $y_{goal} \geqslant 1$.

(4) resiliency constraint: for all $e \in Err$

$$\sum\nolimits_{s\in Op_e} y_s \quad \geqslant \quad \wp \cdot y_e$$

The next lemma is proven by the methods of [17,19] (the only difference distinguishing our case is Constraint (4), which is easy to handle).

**Lemma 14.** *Each feasible solution $(z^*_{s,\alpha})_{s\in S_{\mathcal{N}},\alpha\in Act_{\mathcal{N}}(s)}$ of the linear program (1)–(4) under the objective to maximize $\sum_{(E,\mathcal{A},\mathfrak{H}_E)\in\mathcal{E}} y_{goal_E} \cdot \mathrm{Avail}(E)$, induces an MR-scheduler $\mathfrak{R}_{\mathcal{N}}$ that is resilient in $\mathcal{N}$ and can be computed in time polynomial in $|\mathcal{N}|$. If there is no such solution, there is no resilient scheduler in $\mathcal{N}$.*

*Conversely, let $\mathfrak{R}$ be a resilient scheduler such that $\mathfrak{R}$-paths almost surely reach goal and the expected number of actions executed before reaching goal is finite. Let $z_{s,\alpha}$ denote the expected number of times an action $\alpha \in Act_{\mathcal{N}}$ is taken in a state $s \in S_{\mathcal{N}}$ using $\mathfrak{R}$. Then, values $z_{s,\alpha} = y_{t,\alpha}$ form a solution of the above linear constraints (1)–(4).*

According to the second part of Lemma 14, the scheduler $\mathfrak{R}_{\mathcal{N}}$ achieves the optimal total accumulated reward among all resilient schedulers where the expected number of transitions executed before reaching *goal* is finite. The next lemma shows that $\mathfrak{R}_{\mathcal{N}}$ achieves the optimal total accumulated reward among *all* resilient schedulers, which completes the proof of Lemma 6.

**Lemma 15.** $\mathbb{E}^{\mathfrak{R}_{\mathcal{N}}}_{\mathcal{N},s_{init}}[TR] \geq L$ *with* $L$ *being the supremum over all* $E^{\mathfrak{R}}_{\mathcal{N},s_{init}}[TR]$ *ranging over resilient schedulers* $\mathfrak{R}$ *in* $\mathcal{N}$ *those paths almost surely reach goal.*

*Proof.* First, note that $\mathbb{E}^{\mathfrak{S}}_{\mathcal{N},s_{init}}[TR]$ for an HR-scheduler $\mathfrak{S}$ can be approximated up to an arbitrary small error using a sequence of schedulers $\mathfrak{R}_i$: For each $i \in \mathbb{N}$ we define the scheduler $\mathfrak{R}_i$ by acting as $\mathfrak{S}$ until the $i$-th step and then continuing as $\mathfrak{R}_{\mathcal{N}}$. The expected number of executed actions before reaching the *goal* state is finite for all $\mathfrak{R}_i$. Clearly, $|\mathbb{E}^{\mathfrak{S}}_{\mathcal{N},s_{init}}[TR] - \mathbb{E}^{\mathfrak{R}_i}_{\mathcal{N},s_{init}}[TR]|$ gets arbitrarily small for increasing $i$. Towards a contradiction, assume that $L - \mathbb{E}^{\mathfrak{R}_{\mathcal{N}}}_{\mathcal{N},s_{init}}[TR] > \delta > 0$. Then, there is a sequence of schedulers that approximate $L$ arbitrarily close and there is a scheduler $\mathfrak{R}$ such that $\mathbb{E}^{\mathfrak{R}}_{\mathcal{N},s_{init}}[TR] = K$ with $|L-K| < \delta/2$. Moreover, there is sequence of schedulers $\mathfrak{R}_i$ that approximate $K$ arbitrarily close and have a finite expected number of executed actions before reaching *goal*. Hence, there is some $\mathfrak{R}_i$ such that $\left|L - \mathbb{E}^{\mathfrak{R}_i}_{\mathcal{N},s_{init}}[TR]\right| < \delta$, which is in contradiction with the optimality of $\mathfrak{R}_{\mathcal{N}}$ among all schedulers with a finite expected number of actions executed before reaching *goal*. □

### 3.3 A Lower Complexity Bound

When the bound $R$ is encoded in binary, our algorithms become exponential. Using the PSPACE-hardness result for cost-bounded reachability problems in acyclic MDPs by Haase and Kiefer [15], we show that the question whether there exists a resilient scheduler is PSPACE-hard, even for acyclic MDPs, when $R$ is encoded in binary.

**Lemma 16.** *If* $R$ *is encoded in binary, the problem to check the existence of a resilient scheduler and the decision variant of the resilience-availability problem are PSPACE-hard.*

*Proof.* In [15], the PSPACE-completeness of the following cost-problem has been proven: Given an acyclic MDP $\mathcal{N} = (S, Act, P, s_{init})$ with a cost function and a cost bound $R$, the task is to check whether there is a scheduler $\mathfrak{S}$ for $\mathcal{N}$ such that $\Pr^{\mathfrak{S}}_{\mathcal{N},s_{init}}(\Diamond^{\leqslant R} T) \geqslant \frac{1}{2}$. Here, $T$ denotes the set of trap states in $\mathcal{N}$ and $s_{init} \notin T$.

We now provide a polynomial reduction from the cost-problem à la Haase and Kiefer [15] to the problem to decide the existence of a resilient scheduler and the decision variant of the resilience-availability problem.

Let $\mathcal{M}$ be the MDP resulting from $\mathcal{N}$ by defining $Err = \{s_{init}\}$ and $Op = T$ and adding a fresh action symbol $\tau$ and $\tau$-transitions from the states $t \in T$ to $s_{init}$. That is, $\mathcal{M}$ has the same state space as $\mathcal{N}$, the action set is $Act_{\mathcal{M}} = Act \cup \{\tau\}$ and the $\mathcal{M}$'s transition probability function extends $\mathcal{N}$'s transition probability function by $P(t, \tau, s_{init}) = 1$ and $P(t, \alpha, s) = 0$ for all states $t \in T$, $\alpha \in Act_{\mathcal{M}}$ and $s \in S$ with $(s, \alpha) \neq (s_{init}, \tau)$. $\mathcal{M}$'s cost function is the same as in $\mathcal{N}$ for all states $s \in S$ and $cost(t) = 0$ for all states $t \in T$. Obviously, each scheduler $\mathfrak{S}$ for $\mathcal{N}$ with $\Pr^{\mathfrak{S}}_{\mathcal{N},s_{init}}(\Diamond^{\leqslant R} T) \geqslant \frac{1}{2}$ can be viewed as a memoryless resilient scheduler for $\mathcal{M}$ with respect to the probability threshold $\wp = \frac{1}{2}$ and cost bound $R$. Vice versa, given a resilient scheduler $\mathfrak{S}'$ for $\mathcal{M}$, the decisions of $\mathfrak{S}'$ for the paths from $s_{init}$ to a $T$-state yield a scheduler $\mathfrak{S}$ for $\mathcal{N}$ with $\Pr^{\mathfrak{S}}_{\mathcal{N},s_{init}}(\Diamond^{\leqslant R} T) \geqslant \frac{1}{2}$.

For the decision problem of the resilience-availability problem, we use the same reduction with availability threshold $\vartheta = 0$ and the payoff function that assign 0 to all operational states.

## References


1. Altman, E.: Constrained Markov Decision Processes. Chapman and Hall, Boca Raton (1999)
2. Attoh-Okine, N.: Resilience Engineering: Models and Analysis. Resilience Engineering: Models and Analysis. Cambridge University Press, Cambridge (2016)
3. Baier, C., Dubslaff, C., Klüppelholz, S., Leuschner, L.: Energy-utility analysis for resilient systems using probabilistic model checking. In: Ciardo, G., Kindler, E. (eds.) PETRI NETS 2014. LNCS, vol. 8489, pp. 20–39. Springer, Cham (2014). doi:10.1007/978-3-319-07734-5_2
4. Baier, C., Dubslaff, C., Korenčiak, Ľ., Kučera, A., Řehák, V.: Synthesis of optimal resilient control strategies. CoRR, abs/1707.03223 (2017)
5. Baier, C., Katoen, J.-P.: Principles of Model Checking. MIT Press (2008)
6. Bloem, R., Chatterjee, K., Greimel, K., Henzinger, T.A., Hofferek, G., Jobstmann, B., Könighofer, B., Könighofer, R.: Synthesizing robust systems. Acta Inf. **51**(3), 193–220 (2014)
7. Brázdil, T., Brožek, V., Chatterjee, K., Forejt, V., Kučera, A.: Markov decision processes with multiple long-run average objectives. LMCS **10**(1) (2014)
8. Camara, J., de Lemos, R.: Evaluation of resilience in self-adaptive systems using probabilistic model-checking. In: SEAMS, pp. 53–62 (2012)
9. Chatterjee, K.: Markov decision processes with multiple long-run average objectives. In: Arvind, V., Prasad, S. (eds.) FSTTCS 2007. LNCS, vol. 4855, pp. 473–484. Springer, Heidelberg (2007). doi:10.1007/978-3-540-77050-3_39
10. Ehlers, R., Topcu, U.: Resilience to intermittent assumption violations in reactive synthesis. In: HSCC, pp. 203–212. ACM, New York (2014)
11. Etessami, K., Kwiatkowska, M., Vardi, M.Y., Yannakakis, M.: Multi-objective model checking of Markov decision processes. LMCS **4**(4) (2008)
12. Forejt, V., Kwiatkowska, M., Norman, G., Parker, D., Qu, H.: Quantitative multi-objective verification for probabilistic systems. In: Abdulla, P.A., Leino, K.R.M. (eds.) TACAS 2011. LNCS, vol. 6605, pp. 112–127. Springer, Heidelberg (2011). doi:10.1007/978-3-642-19835-9_11
13. German, R.: Performance Analysis of Communication Systems with Non-Markovian Stochastic Petri Nets. Wiley, Hobokon (2000)
14. Girault, A., Rutten, É.: Automating the addition of fault tolerance with discrete controller synthesis. Form. Methods Syst. Des. **35**(2), 190–225 (2009)
15. Haase, C., Kiefer, S.: The odds of staying on budget. In: Halldórsson, M.M., Iwama, K., Kobayashi, N., Speckmann, B. (eds.) ICALP 2015. LNCS, vol. 9135, pp. 234–246. Springer, Heidelberg (2015). doi:10.1007/978-3-662-47666-6_19
16. Huang, C.H., Peled, D.A., Schewe, S., Wang, F.: A game-theoretic foundation for the maximum software resilience against dense errors. IEEE Trans. Software Eng. **42**(7), 605–622 (2016)
17. Kallenberg, L.: Markov Decision Processes. Lect. Notes, University of Leiden (2011)
18. Longo, F., Ghosh, R., Naik, V.K., Rindos, A.J., Trivedi, K.S.: An approach for resiliency quantification of large scale systems. SIGMETRICS **44**(4), 37–48 (2017)
19. Puterman, M.L.: Markov Decision Processes. Wiley (1994)