

An Analytical Framework to Deal with Changing Points and Variable Distributions in Quality Assessment

Dario Bruneo, Salvatore Distefano, Francesco Longo
and Marco Scarpa

Abstract Nonfunctional properties such as dependability and performance have growing impact on the design of a broad range of systems and services, where tighter constraints and stronger requirements have to be met. This way, aspects such as dependencies or interference, quite often neglected, now have to be taken into account due to the higher demand in terms of quality. In this chapter, we associate such aspects with operating conditions for a system, proposing an analytical framework to evaluate the effects of condition changing to the system quality properties. Starting from the phase type expansion technique, we developed a fitting algorithm able to catch the behavior of the system at changing points, implementing a codomain memory policy forcing the continuity of the observed quantity when operating conditions change. Then, to also deal with the state-space explosion problem of the underlying stochastic process, we resort to Kronecker algebra providing a tool able to evaluate, both in transient and steady states, nonfunctional properties of systems affected by variable operating conditions. Some examples from different domains are discussed to demonstrate the effectiveness of the proposed framework and its suitability to a wide range of problems.

D. Bruneo (✉) · F. Longo · M. Scarpa
Dipartimento di Ingegneria, Università degli Studi di Messina,
C.da di Dio, 98166 Messina, Italy
e-mail: dbruneo@unime.it

F. Longo
e-mail: flongo@unime.it

M. Scarpa
e-mail: mscarpa@unime.it

S. Distefano
Higher Institute for Information Technology and Information Systems
Kazan Federal University, 35 Kremlevskaya Street, 420008 Kazan, Russia
e-mail: sdistefano@kpfu.ru; sdistefano@unime.it

S. Distefano
Dipartimento di Scienze Matematiche e Informatiche, Scienze Fisiche e Scienze della Terra,
Università degli Studi di Messina, Viale F. Stagno d'Alcontres, 31, 98166 Messina, Italy

1 Introduction

Current research trends move toward new technologies and solutions that have to satisfy tight requirements raising the quality standards and calling for adequate techniques, mechanisms, and tools for measuring, modeling, and assessing nonfunctional properties and parameters. Aspects so far coarsely approximated or neglected altogether must now be taken into account in the quantitative evaluation of modern systems. This encompasses a wide range of phenomena, where it is often necessary to consider the relationships among different quantities in terms of aggregated metrics (e.g., dependability, performability, sustainability, and quality of service).

A particularly interesting aspect that often emerges while dealing with the continuously growing complexity of modern systems is the fallouts on observed quantities due to *changes*. The notion of change that we consider in the present work takes into account both the interdependencies among internal quantities, parts, or subparts, and the external environment interactions. Workload fluctuations, energy variations, environmental phenomena, user-related changing behaviors, and changing system operating conditions are only few examples of such aspects that can no longer be neglected in quantitative assessment [1, 2].

When non-exponentially distributed events are associated with system quantities, an important issue to address is the *memory* representation. In fact, in presence of changes, a wide class of phenomena requires that the quantity distribution may change its behavior still preserving its continuity. More specifically, the stochastic characterization at changing points, where the distributions governing the underlying phenomena affecting the observed quantity could change, is strategic to have a good and trustworthy representation of the actual system through a model. Stochastic processes more complex and powerful than Markov models are necessary in order to model this kind of memory, such as *semi-Markov processes* [3] or those deriving from *renewal theory* [4], but these techniques are sometimes not able to cover some specific aspects. Other modeling formalisms, such as fluid models [5], can deal with more complex phenomena but their analytic solution cannot always be easily automated, thus often requiring simulation [6].

In this chapter, we collect the results obtained in some previous works dealing with the above discussed aspects and phenomena, thus proposing a detailed problem analysis, a well-established analytical framework, as well as a comprehensive set of case studies with examples coming from ICT areas (network, Cloud, IoT) and science (physics).

The key aspects of the proposed modeling technique are: (i) the stochastic characterization of the observed quantity in different working conditions; (ii) the specification of the working condition process governing the corresponding variations; (iii) the modeling of the behavior of the observed quantity at changing points.

In particular, at changing points, the quantity distribution may change its behavior, preserving, under specific assumptions, its continuity. In order to implement the continuity constraints imposed by the changing working conditions in the presence

of non-exponential distributions, we propose a solution technique based on phase type distributions, providing an ad hoc fitting algorithm based on codomain discretization.

The chapter is organized as follows. Section 2 introduces the scenario considered also presenting background details that can be useful in order to understand the proposed approach. Section 3 presents our ad hoc fitting algorithm, while Sect. 4 explains how to adopt this algorithm in the evaluation of a system affected by variable operating conditions through stochastic models. Section 5 shows how the proposed framework can be exploited to deal with a variety of application domains. Section 6 concludes the work with some final remarks.

2 Preliminary Concepts

2.1 Problem Rationale

Let us consider a continuous random variable X defined on the sample space Ω stochastically characterizing a generic event. Moreover, let us assume such an event is also characterized and influenced by a specific *condition* belonging to a numerable set of conditions $c_X^i \in \mathbf{C}_X$ identified by $i \in \mathbb{N}$. Thus, the values of $X \in \Omega$ also depend on the condition in a way that X in the i th specific condition is characterized by the cumulative distribution function (CDF) $F_X^i(x) = \Pr\{X \leq x \mid \text{The condition is } c_X^i\}$ and the probability density function (PDF) $f_X^i(x) = d(F_X^i(x))/dx$. Let us assume $F_X^i(x) \in \mathbf{F}_X$ is *continuous* and *strictly increasing* for all i .

To stochastically characterize conditions, let us consider another continuous random variable $Y_{i,j}$ on Ω . Specifically, $Y_{i,j}$ triggers the switching of X from condition c_X^i to c_X^j , with $c_X^i \neq c_X^j \in \mathbf{C}_X$. Assuming that $Y_{i,j}$ is characterized by the $F_{Y_{i,j}}(y)$ CDF (and by the $f_{Y_{i,j}}(y)$ PDF), we can identify a set $\mathbf{Y}$ of condition switching random variables and a set $\mathbf{F}_Y$ of CDFs. Moreover, considering the following CDF stochastically characterizing the event related to X when condition c_X varies:

$$F_X(x, c_X) = \Pr\{X \leq x \wedge \text{The condition is } c_X \in \mathbf{C}_X\}$$

since the condition changes are triggered by the corresponding random variables in $\mathbf{Y}$, such a CDF can be treated as a joint CDF of X and $\mathbf{Y}$, $F_X(x, c_X) = F_{X,Y}(x, \mathbf{y})$. We also impose that $F_X(x, c_X)$ is a continuous function with respect to c . This choice reflects the behavior of many physical systems where quantities under study have no sudden changes. The main consequence of this continuity is that at the condition changing points there is no probability mass for $F_X(x, c)$.

Our aim is to express $F_X(x, c)$ (or equivalently, $F_{X,Y}(x, \mathbf{y})$) in terms of its marginal distributions $F_X^i(x) \in \mathbf{F}_X$ and $F_{Y_{i,j}}(y) \in \mathbf{F}_Y$. This is an undefined problem if no further hypotheses are assumed. To proceed with our discussion, let us consider that just two conditions for X are possible and that they are characterized by the CDFs $F_X^1(x)$ and $F_X^2(x)$ respectively. Moreover, let us assume we start with condition c_1 and then to

switch to condition c_2 . Thus, we are considering just one change point from c_1 to c_2 with $F_{Y_{1,2}}(y) = F_Y(y)$. We can express $F_{X,Y}(x, y)$ as

$$F_{X,Y}(x, y) = \begin{cases} F_X^1(x) & x \leq y \\ F_X^2(x + \tau) & x > y \end{cases} \quad (1)$$

where $\tau \in \Omega$ is a constant depending on the change point such that $x, y, x + \tau \in \Omega$ and $F_{X,Y}(x, y)$ is continuous in y . To quantify τ , we need to understand what happens at the change point. Given that $F_{X,Y}(x, y)$ must be continuous and strictly increasing (and thus invertible) by Eq. (1),

$$F_X^1(y) = F_X^2(y + \tau) \Rightarrow \tau = F_X^{2(-1)}(F_X^1(y)) - y \quad (2)$$

where $F_X^{2(-1)}(\cdot)$ is the inverse function of $F_X^2(\cdot)$. Finally, a simple application of the law of total probability allows us to decondition $F_{X,Y}(x, y)$, obtaining

$$\begin{aligned} F_{X,Y}(t) &= \int_{-\infty}^{+\infty} \Pr(X \leq t | Y = y) f_Y(y) dy \\ &= \int_0^t \Pr(X \leq t | Y = y) f_Y(y) dy \\ &\quad + \int_t^{+\infty} \Pr(X \leq t | Y = y) f_Y(y) dy \\ &= \int_0^t (1 - \Pr(X > t | Y = y)) f_Y(y) dy \\ &\quad + F_X^1(t) (F_Y(y)|_t^\infty) \end{aligned} \quad (3)$$

This way, considering $y \leq t$, $\Pr(X > t | Y = y) = 1 - F_X^2(t + \tau) = 1 - F_X^2(t + F_X^{2(-1)}(F_X^1(y)) - y)$ and thus by Eq. (3)

$$\begin{aligned} F_{X,Y}(t) &= F_X^1(t) (1 - F_Y(t)) \\ &\quad + \int_0^t F_X^2(t + F_X^{2(-1)}(F_X^1(y)) - y) f_Y(y) dy \end{aligned} \quad (4)$$

Note that Eq. (4) has been deduced by simply applying probability theory. Therefore, it is valid whenever the above model and related assumptions (continuity and invertibility of marginal CDFs, and no probability mass at changing point for the joint CDF) are satisfied. We can associate several physical meanings with such an equation. All of them are related to *conservation laws* in different application areas, such as stochastic modeling (conservation of reliability-Sedyakin model [2], survival models, duration models), physics (e.g., conservation of energy, momentum), economics, sociology, history, biology, and so on.

2.2 Phase Type

Phase type distributions were first used by Erlang [7] in his work on congestion in telephone systems at the beginning of this century. His *method of stages*, that has been further generalized since then, represents a non-exponentially distributed state sojourn time by a combination of *stages*, each of which is exponentially distributed. In this way, once a proper stage combination has been found to represent or approximate a distribution, the whole process becomes Markovian and can be easily analyzed through well-known and established approaches. Note that the division into stages is an operational device. Thus, stages have no physical significance in general.

Later, Neuts [8, 9] formally defined the class of phase type distributions. A non-negative random variable T (or its distribution function) is said to be of phase type (PH) when T is the time until absorption of a finite-state Markov chain [9]. When continuous time Markov chains are considered, we talk about *continuous phase-type* (CPH) distributions. In particular, we say that T is distributed according to a CPH distribution with representation $(\alpha, \mathbf{G})$ and of order n if α and $\mathbf{G}$ are such that $\pi(0) = [\alpha, \alpha_{n+1}]$ is the $n + 1$ dimensional *initial probability vector* in which $\alpha_{n+1} = 1 - \sum_{i=1}^n \alpha_i$ and $\hat{\mathbf{G}} \in \mathbb{R}^{n+1} \times \mathbb{R}^{n+1}$ is the *infinitesimal generator matrix* of the underlying continuous time Markov chain in which

$$\hat{\mathbf{G}} = \left[\begin{array}{c|c} \mathbf{G} & \mathbf{U} \\ \hline \mathbf{0} & 0 \end{array} \right]$$

Matrix $\mathbf{G} \in \mathbb{R}^n \times \mathbb{R}^n$ describes the transient behavior of the CTMC and $\mathbf{U} \in \mathbb{R}^n$ is a vector grouping the transition rates to the absorbing state. Matrix $\hat{\mathbf{G}}$ must be stochastic so $\mathbf{U} = -\mathbf{G}\mathbf{1}$, where $\mathbf{1} \in \mathbb{R}^n$ is a vector of 1. Thus, the CDF of T , $F_T(t)$, i.e. the probability of reaching the absorbing state of the CPH, is given by:

$$F(t) = 1 - \alpha e^{\mathbf{G}t} \mathbf{1}, \quad t \geq 0 \quad (5)$$

One of the main drawbacks of PH distributions is the state-space explosion. In fact, expanding each non-exponential distribution with a set of phases greatly increases the number of model states. In order to provide a compact representation of the resulting CTMC, Kronecker algebra is usually exploited [9–11], providing several advantages: (i) it is not necessary to physically generate nor store the expanded state space as a whole; (ii) the memory cost of the representation of the expanded state space grows linearly with the dimension of the PH distributions instead of geometrically; (iii) specific algorithms developed for similar cases can be used for the model solution [12].

The method of stages allows one to code memory through Markov models. Each phase represents a particular condition reached by the stochastic process during its evolution. Usually, this condition is associated to the time. CPHs are specified with the aim of coding the time instant at which a specific event occurs, by keeping memory of the phase reached so far. This allows resumption of the evolution of an

activity from the saved point if the activity is interrupted. This mechanism works if the observed events are not changing their nature. On the other hand, if the nature of the observed event changes preserving memory of the time instant may not be enough.

In such cases, as described in Sect. 2.1, since the distribution characterizing the observed phenomenon changes according to some event, it becomes a bi-variate function of both the observed event and the external one. Moreover, it is generally difficult or even impossible to obtain the joint CDF by only knowing the marginal CDFs.

As an example, let us consider that the observed event is a component failure characterized by the component lifetime CDF in a changing environment. In such a context, a way to adequately address the problem may be to characterize the lifetime CDFs of the observed system both in isolation or in the initial working condition (without dependencies or in the baseline environment), and in the new environment or after the dependency application. By representing the two lifetime CDFs as CPHs in this way, it would be necessary to specify how the system jumps from one CPH to the other one and viceversa. Such a “jump” process is usually governed by a law regulating the underlying system. For example, the conservation of reliability principle states that reliability should be preserved in the transition between the two conditions. Of course, other possible examples are the conservation of momentum, of age, of battery charge, and so on. In general, such quantities can be expressed as a function of time. Thus, the quantity to preserve is not necessarily the time but a (usually monotonic, nonincreasing) function of the time.

In several previous works, [13–23], we presented a method based on CPHs and Kronecker algebra to manage such kind of memory, without incurring in the complexity of the problem described in Sect. 2.1. The proposed method is able to manage many changing points affecting the reliability, availability, or other performability indexes that can change many times during the system evolution. The technique requires some assumptions on the CPHs structure in order to be easily implemented. In particular, it is necessary that the CPHs coding the functions associated to the changing indexes in different operating states have the same number of phases and initial probability vector. In order to deal with this issue, we propose an ad hoc fitting algorithm. The basic idea to represent the conservation of the considered quantity in terms of CPHs is to discretize the probability codomain associating to each stage of the CPH a specific value of the function. Thus, in order to jump from a CPH representing the behavior of the system under specific environmental conditions to the one representing the system in different environmental conditions, it is only necessary to save the phase reached, thus implementing the conservation of the considered quantity. In other words, a specific phase encodes the value reached by the quantity. Then, if we represent the two functions in terms of CPHs with the same number of phases n , the i th phase of the two CPHs, with $i \leq n$, corresponds to the same function value.

In this chapter, we give an overview of the technique also providing a set of simplified examples taken from previous works. The aim is to show that our approach can deal with a wide class of phenomena and to give the reader an overall idea of our work on such a topic.

3 PH Codomain-Fitting Algorithm

The goal of the fitting algorithm we propose is to associate a specific meaning, a memory tag or value, with the phases of the CPH, while approximating the original distribution through a CPH. The idea is to characterize the CPH phases with some information encoding the probability codomain, in order for each phase to represent a value of probability or better, an interval of probability values in $\mathbf{K}_i = [a_i, b_i] \subseteq [0, 1]$. The $\mathbf{K}_i$ intervals must be mutually exclusive ($\mathbf{K}_i \cap \mathbf{K}_j = \emptyset \forall i \neq j$) and cover the whole $[0, 1]$ probability codomain ($\bigcup_i \mathbf{K}_i = [0, 1]$).

The first step to perform the proposed fitting algorithm is to discretize the codomain of the CDF to be approximated through continuous phase type. For example, applying this algorithm to the CDF shown in Fig. 1a, $n = 10$ contiguous intervals $\mathbf{K}_i = [i/n, (i+1)/n) \in [0, 1] \subset \mathbb{R}$, with $i = 0, \dots, n-1 \subset \mathbb{N}$, have been identified, characterizing $n+1$ endpoints in total. The proposed technique associates with each of such endpoints a phase of the corresponding CPH as reported in Fig. 1b for the CDF given as example, where phase 0 corresponds to probability 0, 1 to 0.1, 2 to 0.2, and so on. In general, the j th phase corresponds to the j/n probability with $j = 0, \dots, n$.

The transition between the states i and j is strictly related to the event characterizing the underlying process. Transitions are allowed either to the next phase of the CPH model or to the absorbing state, as shown in Fig. 1b. If, while in the i th interval, the event occurs a transition between the i th and the last n th phase is performed, otherwise, if the event does not occur, the CPH transits from the i th phase to the $(i+1)$ st.

A generic CPH obtained by applying the proposed technique is reported in Fig. 2, where $n+1$ phases labeled $0, \dots, n$ are identified, subdividing the probability codomain into n contiguous intervals $[i/n, (i+1)/n) \in [0, 1] \subset \mathbb{R}$ with $i = 0, \dots, n-1 \subset \mathbb{N}$. As stated above, two are the possible transitions outgoing from the i th phase: one incoming to the next $i+1$ phase and the other to the absorbing n th phase. To evaluate the rates to associate with such transitions, we start from the $i \rightarrow i+1$ transition. Assuming the considered event is not occurred before t_i and does not occur in the interval $[t_i, t_{i+1})$, the sojourn time in phase i depends only on the time spent in the i th state due to the Markov property of CTMCs. This sojourn time can be characterized by the random variable τ_i with mean value $E[\tau_i] = t_{i+1} - t_i$. Since τ_i has to be approximated by a random sojourn time characterized by the negative exponential rate λ_i , we have that that:

$$\lambda_i = \frac{1}{E[\tau_i]} = \frac{1}{t_{i+1} - t_i} \quad (6)$$

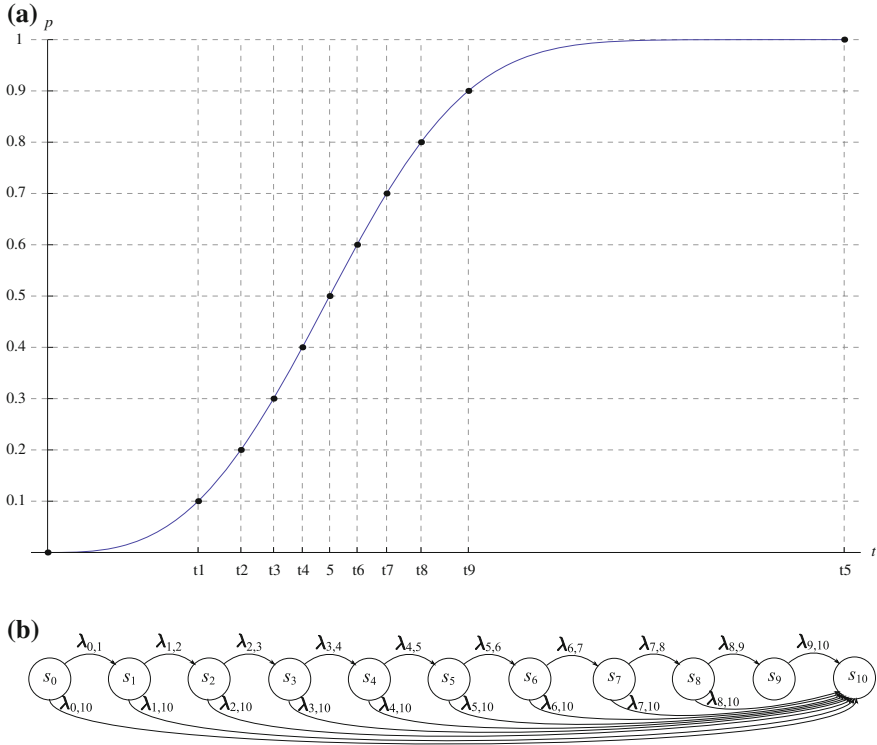


Fig. 1 A CDF Codomain Sampling (a) and corresponding CPH (b)

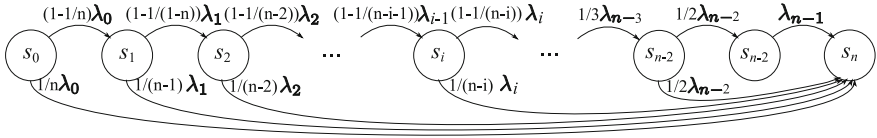


Fig. 2 A generic CPH of the proposed approach

It is worth noting that Eq. (6) is the rate of the CPH approximating the sojourn time in state i if the observed event has not occurred before t_i and does not occur in the interval $[t_i, t_{i+1})$. Thus, we have to identify the probability of transiting to state $i + 1$ and n from i . The $i \rightarrow i + 1$ probability $p_{i,i+1}$ is

$$\begin{aligned}
 p_{i,i+1} &= \Pr\{X > t_{i+1} | X > t_i\} = 1 - \Pr\{X \leq t_{i+1} | X > t_i\} \\
 &= 1 - \frac{\Pr\{t_i < X \leq t_{i+1}\}}{\Pr\{X > t_i\}} = 1 - \frac{F(t_{i+1}) - F(t_i)}{1 - F(t_i)} \\
 &= 1 - \frac{1/n}{1 - i/n} = 1 - \frac{1}{n - i}
 \end{aligned} \tag{7}$$

where X is the observed random variable and $F(t)$ is its corresponding CDF to be approximated by CPH. Since the approximation algorithm split the codomain into intervals of size $1/n$, we have that $F(t_{i+1}) - F(t_i) = 1/n$ and $F(t_i) = i/n$. On the other hand, the $i \rightarrow n$ probability $p_{i,n}$ is the complement of the Eq. (7):

$$\begin{aligned} p_{i,n} &= Pr\{X \leq t_{i+1} | X > t_i\} = 1 - Pr\{X > t_{i+1} | X > t_i\} \\ &= 1 - p_{i,i+1} = \frac{1}{n-i}. \end{aligned} \quad (8)$$

This way, the rate specified by Eq. (6) is split between the two possible transitions such that:

$$\lambda_{i,i+1} = p_{i,i+1} \lambda_i = \left(1 - \frac{1}{n-i}\right) \lambda_i \quad (9)$$

$$\lambda_{i,n} = p_{i,n} \lambda_i = \frac{\lambda_i}{n-i} \quad (10)$$

as shown in Fig. 2. The $n-1 \rightarrow n$ transition has only one rate λ_n that, since a CDF asymptotically reaches 1 at $t = \infty$, should be ideally 0. For the CTMC of Fig. 2 to be a CPH, there exists only one absorbing state. Therefore, in building the CPH, the last rate λ_n is characterized by a low value, approximating with only one phase the long asymptotic tail of the CDF considered, for example by a fraction of the CPH lowest rate λ_k . The proposed algorithm has some problems exhibits some inaccuracy when fitting long-tail distributions and in general of distributions with constant or quasi-constant segments. A possible solution or workaround could be to increase n , especially in those segments, adopting a kind of “resolution” trimming or tuning on them, to be taken into account in the retrieval of the codomain value associated with the phase.

4 Phase-Type Distributions to Model Variable Conditions

The fundamentals introduced in the previous section together with the fitting algorithm presented in Sect. 3 aim to more easily manage the phenomena of changing conditions. In fact, the method of stages is a way to code memory through a Markov model, since a phase of a CPH represents a particular condition reached by the underlying stochastic process during its evolution. In classical methods [24–26], the condition usually is associated with time in order to remember the elapsed time since the enabling of an event. In other words, such CPHs are specified to save the time instant in which a specific event occurs, by just saving the phase reached and to continue their evolution from the saved point. Such mechanism works if the observed events are not changing their nature. In the context where the component dynamics are subject to change, preserving time memory is useless due to the change in the distribution characterizing the firing time. In such cases, we need a specific represen-

tation of the change event in order to preserve the memory as expressed by Eq. (2) in Sect. 2.1. According to that relation the quantity to preserve is no longer the time but the probability to fire at the enabling time y . To directly apply Eq. (2) implies a search of the time shift τ and the computation of Eq. (4) that is numerically inapplicable in real cases where more than one changing event can happen at a state change.

Our proposal is to represent the event firing times in terms of CPHs obtained as result of the fitting algorithm of Sect. 3; in that case, the firing probability is discretized and a specific value of probability is associated to each stage of the CPH. Thus, in order to jump from a CPH T_1 representing the behavior of the system under specific environmental conditions to the one (T_2) representing the system in different environmental conditions, it is only necessary to save the phase reached, thus implementing the conservation of the firing probability. In other term, a specific phase must encode the value of reached firing probability, and, by representing the two CDF in terms of CPHs with the same number of phases n , the probability encoded by the i -th phase of a CPH, where $i \leq n$, is the same encoded in the i -th phase of the other CPH. Of course, T_1 and T_2 must have the same number of phases; their representation is different due to the different dynamic of the events into the two operating conditions.

Jumping from T_1 to T_2 maintaining the phase reached implements the behavior described by Eq. (2). In fact, the CDF value is maintained by construction and the time shift is a consequence of a different rate assigned in CPH T_2 . The fact that at a given time y the process switches from a phase k of T_1 to phase k of T_2 implies that the event in the new condition is as enabled by a different time with respect to the dynamic described through T_1 . In this way, we take into account of the time shift τ of Eq. (2). This behavior is depicted in Fig. 3.

When the events are coded through CPHs, expansion methods can be used to adequately study the stochastic process also in the case of non-exponentially distributed events. The memory conservation at the switching condition time is taken

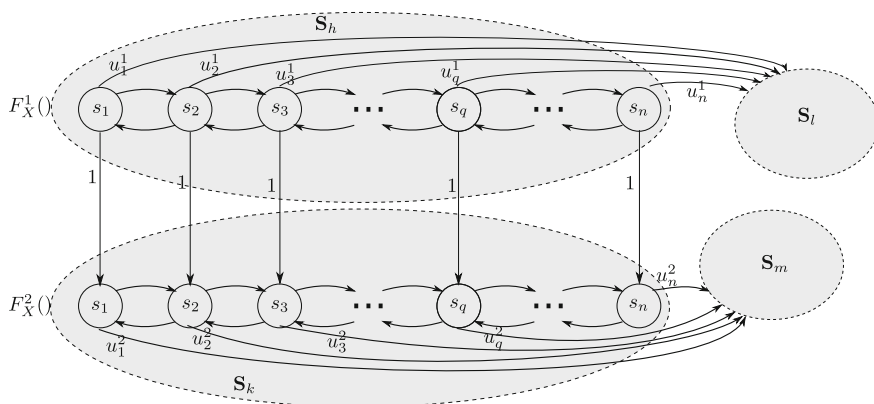


Fig. 3 CPH switching due to operating condition changing

into account using the CPH coding given above and associating to the changing event the classical memory policy managed by the expansion methods. These methods are very memory intensive. Thus, in order to provide a compact representation of CTMCs resulting from phase-type expansion, the Kronecker algebra is used as shown in [12, 25, 27–29].

4.1 Symbolic Representation

Let us consider a discrete-state discrete-event model and let S be the system state space and ε the set of CPH distributed system events. Following the state-space expansion approach [9], the stochastic process can be represented by an expanded CTMC. Such CTMC is composed of $||S||$ macro-states and it is characterized by a $||S|| \times ||S||$ block infinitesimal generator matrix $\mathbf{Q}$ in which

- the generic diagonal block $\mathbf{Q}_{n,n}$ ($1 < n < ||S||$) is a square matrix that describes the evolution of the CTMC inside the macro-state related to state n , which depends on the possible events that are enabled in such state;
- the generic off-diagonal block $\mathbf{Q}_{n,m}$ ($1 < n < ||S||$, $1 < m < ||S||$) describes the transition from the macro-state related to state n to the one related to state m , which depends on the events that occur in state n and on the possible events that are still able to occur in state m .

By exploiting Kronecker algebra, matrix $\mathbf{Q}$ does not need to be generated and stored as a whole, but can be symbolically represented through Kronecker expressions and algorithmically evaluated on-the-fly when needed with consequent extreme memory saving [25, 28]. Moreover, this approach is particularly suitable in models where the memory conservation at the switching condition time has to be applied. Going into details of matrix $\mathbf{Q}$, its blocks have the following form:

$$\mathbf{Q}_{n,n} = \bigoplus_{1 < e < ||\varepsilon||} \mathbf{Q}_e \quad (11)$$

$$\mathbf{Q}_{n,m} = \bigotimes_{1 < e < ||\varepsilon||} \mathbf{Q}_e \quad (12)$$

In other words, the diagonal blocks are computed as Kronecker sums whereas off-diagonal blocks are computed as Kronecker products of a series of matrices $\mathbf{Q}_e$, each associated to a CPH representing system events. In the case of switching conditions incurred during a state change, the CPHs in the two states are different and this is reflected in Eq. (11) in the fact that matrix $\mathbf{Q}_e$ related to the same event is different in the two states. Thanks to the fitting algorithm of Sect. 3, the structure of CPHs in different conditions can be modeled with the same number of phases n where each

phase codes a level of probability to fire. As a consequence, the conservation of the phase reached between the two CPHs the contribution of event e to the equation Eq. (12) is the rank n identity matrix $\mathbf{I}_n$.

5 Examples

In this section we aim at highlighting the generality of the proposed framework in modeling a wide range of systems. To this purpose, four examples are proposed dealing with both ICT and physics' systems and demonstrating the effectiveness of the approach in a variety of application fields.

5.1 Network Data Performability

In this example, we investigate both performance and reliability of data transferred through a connection-oriented virtual circuit network (VCN) at datalink or network layer such as X.25, ATM, Frame Relay, GPRS, or MPLS. To send data in a layer 2–3 VCN, a virtual circuit between the two endpoints must be established first. This means that data transmissions in such networks always follow the same path among nodes. Although this implies several benefits (bandwidth reservation, low overhead, and simple switching), the main drawback of such an approach is the virtual channel reliability, especially in wireless networks. Indeed, a channel failure usually breaks the virtual circuit forcing creation of a new connection. Several mechanisms and countermeasures, such as fault tolerance and redundancy policies, have been adopted by virtual circuit protocols to mitigate these failure effects. For example, bandwidth overprovisioning sends packet replicas to increase the delivery ratio. This often implies bandwidth overbooking to improve the channel utilization and the provider outcome. However, it impacts both the channel and the data transmission reliability, which become sensitive to network traffic fluctuations.

In this example, we focus on both reliability and performance of packet transmission in VCN, taking into account different traffic conditions in the evaluation of the packet time-to-failure and time-to-delivery r.v. Packet lifetime includes all the causes of delivery failure (channel failure, congestion, hardware and software faults, etc.). In other words, we aim at evaluating the probability the channel is reliable when and while a packet is sent, given that the channel is available at sending time.

As discussed above, we assume that both the packet time-to-failure X and time-to-delivery T also depend on the traffic, i.e. on the bandwidth available for transmission. To this end, we characterize two traffic conditions, *average* and *high*, assuming knowledge of the CDFs of the packet lifetime and time-to-delivery in such conditions, namely $F_X^A(t)$, $F_T^A(t)$, $F_X^H(t)$ and $F_T^H(t)$, respectively.

This way, the trigger events associated with traffic fluctuations are characterized by two CDFs $F^I(t)$ and $F^D(t)$ corresponding to the switching from average to high traffic

(traffic increase – I) and vice-versa (traffic decrease – D), respectively. Assuming periodic fluctuations between the two types of traffic, a bursty traffic pattern, a Markov modulated Poisson process can be adopted in the modeling if $F^I(t)$ and $F^D(t)$ are exponentially distributed.

The resulting state-space model is composed of four main states identifying the reliable channel both in the average (S_{T_A}) and in high (S_{T_H}) traffic conditions, the successful packet delivery (S_{P_D}), and the failed transmission (S_{P_F}), as shown in Fig. 4a. If the channel is available at the beginning, the system starts from either state S_{T_A} or S_{T_H} according to the traffic condition. The transitions between these two states model the traffic condition switching and are triggered by the e_{T_I} and e_{T_D} events representing traffic fluctuations characterized by the corresponding CDFs $F^I(t)$ and $F^D(t)$.

From such states, in the case of a failure during message sending, the state S_{P_F} is reached through transitions labeled e_{F_A} or e_{F_H} depending on the traffic condition, associated with $F_X^A(t)$ and $F_X^H(t)$ respectively. Otherwise, for successful delivery, state S_{P_D} is reached through events e_{D_A} or e_{D_H} , according to the traffic, characterized by $F_T^A(t)$ and $F_T^H(t)$ respectively. The goal is to evaluate the CDFs of the time to message delivery ($F_R(t)$), i.e., the probability of reaching state S_{P_D} , and the time to message sending failures ($F_F(t)$), i.e. the probability of reaching state S_{P_F} . Since states S_{P_D} and S_{P_F} are both absorbing, a steady state exists for the model and the values $F_R(\infty)$ and $F_F(\infty)$ represent the steady state probability the packet is delivered or fails, respectively, i.e. the packet/data delivery ratio, with $F_R(\infty) + F_F(\infty) = 1$. It follows that the that $F_R(t)$ and $F_F(t)$ are defective CDFs.

To evaluate the model, we based our analysis on parameters and values taken from the literature. With regard to the time to delivery, we used the data published in [30, 31], providing statistics of two workload conditions on the channel. Therefore, we stochastically characterized the time to delivery by Gaussian CDFs, the average traffic one ($F_T^A(t)$) with mean value 2.5 ms and variance 0.5 ms, the high traffic CDF ($F_T^H(t)$) has higher mean value (11 ms) and variance (3 ms).

The (sending) time to failure CDF has been obtained by the delivery ratio taken from [32], i.e. 0.9 in the average traffic case and 0.6 in the high one. This requires

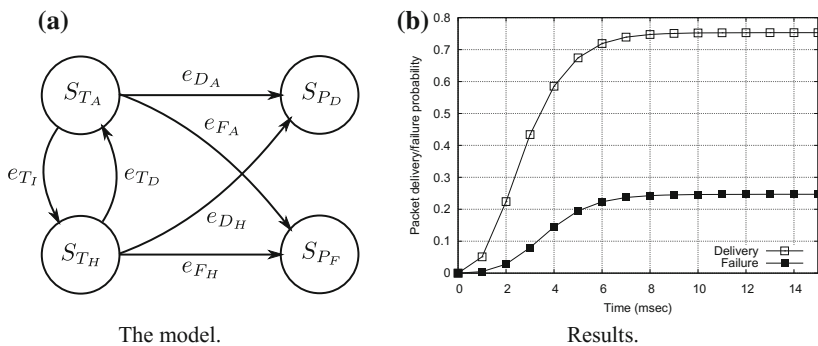


Fig. 4 Network data performability example

further manipulation. The delivery ratio is the steady state probability the packet is delivered in the above model. Knowing the delivery steady state probability in each traffic condition, we have to find a failure CDF such that the steady state probability of delivery in the model of Fig. 4a is equal to the delivery ratio. Assuming the message sending lifetime is represented by a Weibull CDF $(1 - e^{-(\frac{t}{\alpha})^\beta})$ commonly used in reliability, setting $\beta = 5$ through the model we identified $\alpha_A = 22$ ms for the average traffic and $\alpha_H = 4$ ms for the high one. Finally, the traffic switching rate is $\lambda_I = \lambda_D = 500 \text{ s}^{-1}$, i.e. a switch every 2 ms.

Applying the codomain fitting algorithm of Sect. 3 four 100 phase CPHs approximate these distributions. Then we evaluated the packet delivery and failure probability, obtaining the results shown in Fig. 4b. As discussed above, two defective distributions the time to delivery and the lifetime of the packet are identified, corresponding to states S_{P_D} and S_{P_F} , respectively. The steady state is reached at approximately 12 ms, with a delivery ratio of about 75 %. Similar transient trends can be observed. The first knee identifies a change in probability following the underlying CDFs, with initially a very low probability to deliver the packet, then suddenly increasing. The second knee represents the transition to the steady state, where it is very likely either the packet has been already delivered or sending failed.

5.2 IaaS Cloud Performance

Cloud computing systems offer virtualized environments where jobs, in terms of virtual machines (VMs), can be executed without any knowledge on the physical computing infrastructure. In such a scenario, multiple VMs can be allocated in the same physical machine (PM), e.g., a core in a multi-core architecture, in order to respond to specific provider goals such as energy consumption or load burst management. Multiple VMs sharing the same PM can incur in a reduction of the performance mainly due to I/O interference between VMs. In order to meet the stringent QoS level required by service-level agreements (SLAs), cloud providers must find trade-offs between the internal management strategies and the QoS offered. To this end, we propose an application of the proposed technique that could help data center managers to tune system parameters.

We model a system composed of a single PM that is able to concurrently execute up to m VMs. The actual number of VMs is a function of the traffic load as well as of the VM allocation strategy. We model such an aspect through a system composed of m states (see Fig. 5a), where transitions among states are triggered by events e_{C_I} (Increase) and e_{C_D} (Decrease) representing an increase or a decrease in the level of concurrency. Such events can be generally distributed with mean λ and μ , respectively and can be correlated through the *traffic intensity* factor ρ defined as $\rho = \frac{\lambda}{\mu}$. With low values of ρ (< 1), the system will experience a low concurrency level while with values of ρ near to 1 the system will experience a concurrency level near to m .

We are interested in measuring the time needed to execute a job that can be modeled through events $e_{T_1}, e_{T_2}, \dots, e_{T_m}$ that represent the execution times associated with

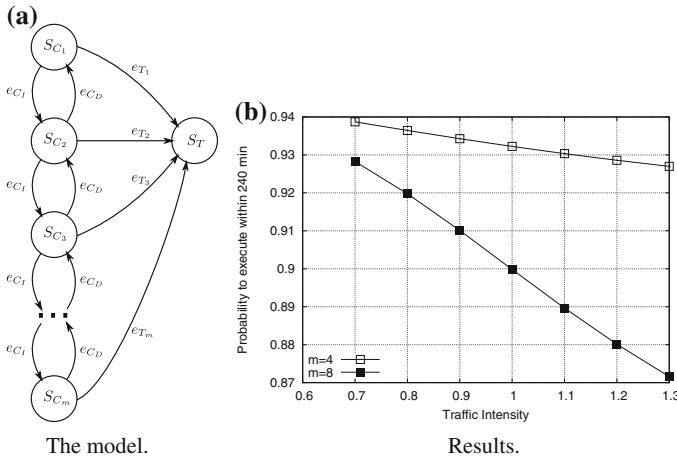


Fig. 5 The Cloud example

the different operating conditions. Events e_{T_i} are generally distributed and depend on the performance degradation due to the concurrent execution. They can be estimated by means of theoretical studies or statistical observations. Finally, the state S_T is the absorbing state, which corresponds to the task completion.

Measures of interest include the mean execution time or the probability to execute the task within a given time deadline τ . Such performance indexes can be investigated by varying the maximum concurrency level m under different load conditions characterized by different values of ρ .

In order to provide a quantitative example, we start defining a job whose duration T_1 can be modeled as a series of cycles composed of a CPU execution (modeled by an exponentially distributed execution time T_{CPU}^1 with mean λ_{CPU}) followed by an I/O access (modeled by an exponentially distributed execution time $T_{I/O}^1$ with mean $\lambda_{I/O}$). To reach the job completion, such events need to be repeated for a certain number of times. We model such an aspect by defining a probability p_f that is the probability of finishing the job at the end of a CPU-I/O cycle. The performance degradation [33] is modeled by varying the CPU and I/O execution times with respect to the concurrency level: $T_{CPU}^i = T_{CPU}^1 \cdot (1 + d_{CPU})^{(i-1)}$, $T_{I/O}^i = T_{I/O}^1 \cdot (1 + d_{I/O})^{(i-1)}$, where d_{CPU} and $d_{I/O}$ are the degradation factors related to CPU and I/O events, respectively. Notice that modeling job duration according to such a technique corresponds to a CPH distributed job completion time that, even without the specific form of Fig. 2, respects the characteristic discussed in Sect. 3, i.e., phases have a physical meaning and preserving memory of the phase is equal to preserving memory of the corresponding physical quantity (in this case the execution time).

Figure 5b shows the results obtained with the parameters $\lambda_{CPU} = 0.33 \text{ min}^{-1}$, $\lambda_{I/O} = 0.2 \text{ min}^{-1}$, $p_f = 0.9$, $d_{CPU} = 0.05$, $d_{I/O} = 0.1$. This figure gives an example of how the proposed technique can be exploited in order to obtain high level information on the cloud system. In particular, from the plotted curves, it is possible to quantify

the probability that a job executes within the threshold defined by an SLA (in this case $\tau = 240$ min). Such a probability can be evaluated under different load conditions and using different concurrency strategies, thus providing useful insights to the system administrator to identify a better set of strategies.

5.3 *Energy Consumption in IoT Mobile Devices*

While talking about mobile environments, the first thing that attracts attention is the comparison between resource-poor devices and fixed stations. Mobile devices are limited in computational power, available memory, and battery life. The presence of wireless connections with different available bandwidth and probability of disconnection and reconnection is another important characteristic of these environments. One of the more interesting topics in this research field is the Internet of Things (IoT), where concerns regarding mobility are even more constrained.

Mobile crowdsensing (MCS) [34] applications are a typical scenario in the IoT world. In such applications, hundreds or even thousands of mobile phones are employed to collect environmental information usually aggregated with user-generated patterns, in order to analyze and forecast crowd behavior and attitude measuring various individual as well as community trends. Individual trends are those pertaining to a single device(s) owner, while collective trends are those inherent to an aggregate of surroundings and not limited to any individual in particular. Examples of individual trends are movement patterns (e.g., running, walking, climbing stairs), commuting modes (e.g., biking, driving, taking a bus, riding the subway), as well as activities in general (e.g., using an ATM, visiting a store, having a conversation, listening to music, making coffee). Collective trends may be exemplified by (air/noise) pollution levels, e.g., in a neighborhood or, even more aptly, by realtime traffic patterns and transit timings, including significant deviations from recurrent trends and averages, like reroutings due to either scheduled (e.g., closures) or sudden (e.g., pot holes) road alterations.

Such large scale, collective trends monitoring is achievable only if a large community of individuals are willing to share their resources by accepting a certain amount of the processing duties. We distinguish between participatory and opportunistic approaches. In participatory scenarios, individuals are actively involved in contributing by mean of devices or (meta-)data (e.g., taking a picture or reporting a road bump). In opportunistic scenarios, sensing is more automatic and user involvement is minimal (e.g., continuous sampling of geo-localized data).

Given that mobile devices are usually equipped with low voltage batteries, energy saving is certainly one of the main issues in such a context. In fact, users usually have full control on their devices and they are allowed to choose when and how they want to contribute to the MCS application. Usually, some kind of policy is exploited, referring to the energy status in terms of battery charge. Experimental results show that the energy consumption of a mobile device depends on the load the device is subjected to. This is related to the specific tasks and operations the

device is performing at a particular time instant. There are tasks that request a higher energy consumption when compared to others, while low consumption statuses, e.g., suspend, idle, can be forced to extend lifetime.

Here, we consider the simple example of a user willing to contribute to an MCS application by allowing its smart phone to periodically share geo-localized data collected through on-board sensors. The user agrees to send such data only if the remaining battery energy is above 20 %. Several works, such as [35], show that the main high-level tasks that affect smart phone energy consumption are: (i) making phone calls, and (ii) browsing the Web, while, in all the other situations, the smart phone is usually maintained in a minimum energy consumption (suspend) mode. For simplicity, let us consider three different usage patterns:

- *generic*—over a single day, about 20 % of time is spent making phone calls while about 10 % of time is spent in browsing the Web;
- *business*—over a single day, about 40 % of time is spent making phone calls while about 10 % of time is spent in browsing the Web;
- *student*—over a single day, about 10 % of time is spent making phone calls while about 40 % of time is spent in browsing the Web.

Finally, let us suppose that the battery duration T_d is approximately equal to 43 h in suspend mode, 10 h while making phone calls, and 5 h while browsing the Web. Our technique allows us to predict the smart phone battery duration and, accordingly, the amount of contribution that the mobile device can provide to the MCS application, according to the different usage patterns.

The state space of the model that can be used to represent our MCS application scenario is depicted in Fig. 6a. States S_S , S_C , and S_B represent the suspend, calling, and web browsing modes respectively. Events e_{S_C} and e_{C_S} represent the switching between suspend and calling modes while events e_{S_B} and e_{B_S} model the switching between suspend and browsing mode. They can be characterized with exponential distributions and their rates must be fixed so that the mean sojourn times in states S_S , S_C , and S_B during a day is set accordingly to the above reported usage patterns. State S_D represents the fully discharged battery state while events e_{D_C} , e_{D_S} , and e_{D_B} represent the battery discharge process.

As described in [17], the linear or nonlinear discharge process of a battery can be modeled, according to the technique described in Sect. 3, through an Erlang CPH with a sufficient number of phases. In our case, we used $n = 100$ phases. The phases of the CPH assume the physical meaning of the battery charge level. The rates of the Erlang distributions are set equal to n/T_d . The discharge time is different in the three states, as described above. However, the battery charge level should be kept while switching among them.

Thanks to the physical meaning of the CPH phases resulting from the codomain fitting algorithm, our technique allows us to compute the probability of having a 80 % discharged battery and, as a consequence, the probability for the mobile device to stop contributing to the IoT mobile crowdsensing service. Figure 6b shows this probability in the three usage scenarios considered.

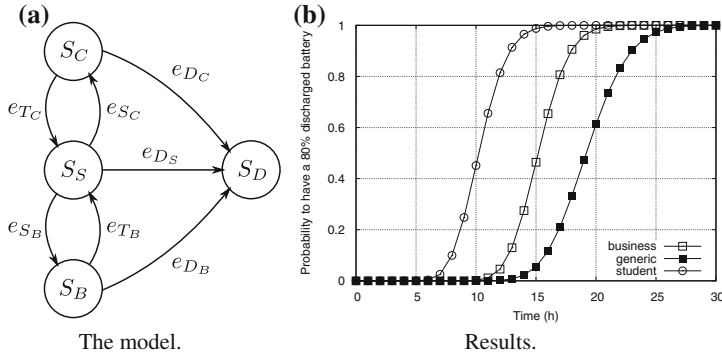


Fig. 6 The energy management in IoT mobile devices example

5.4 Car Travel Planning

As the last example, we consider a car with a fixed quantity of fuel in its tank that should arrive at a defined destination within a certain time. The car has to travel along a path growing at different altitudes. Thus, the road could possess a different slope in different sections. In particular, we consider a road with the elevation graph depicted in Fig. 7a.

The main requirement the driver must meet is to arrive to the stop point by the time of a scheduled appointment. Thus, he/she develops two different driving plans in order to meet this requirement. Two remarks must be considered: (1) even if the driver chooses to travel each path section at a given speed some degree of uncertainty must be considered due to the road condition (holes, traffic congestion, and other unpredictable events potentially occurring during the travel). Thus each section travel time is a r.v. characterized by a mean value (the time planned by the driver) and a nonzero coefficient of variation; (2) the power required to cover a path section depends on the section slope; this means the engine consumes petrol at different rates in each section.

Our method is able to evaluate such problems thanks to its ability to remember an arbitrary quantity following changing operational conditions. In this example, we preserve memory of the level of petrol in the tank, taking into account both the road slope and the changing engine usage.

Based on these considerations, a state model to investigate the car travel plan is depicted in Fig. 7b. Specifically, the model must evaluate the distance the car is able to run and the corresponding time, given an initial quantity of fuel and a travel plan. States S_i , $0 \leq i \leq 5$, represent the car running in section S_i of the path, whereas states S_{E_i} represent the car stopped somewhere in those sections due to empty tank. If the state S_5 is reached the mission is successfully accomplished.

Each plan fixes the expected speeds on each path section and, as a consequence, the required engine power per section is computed by referring to the mechanical

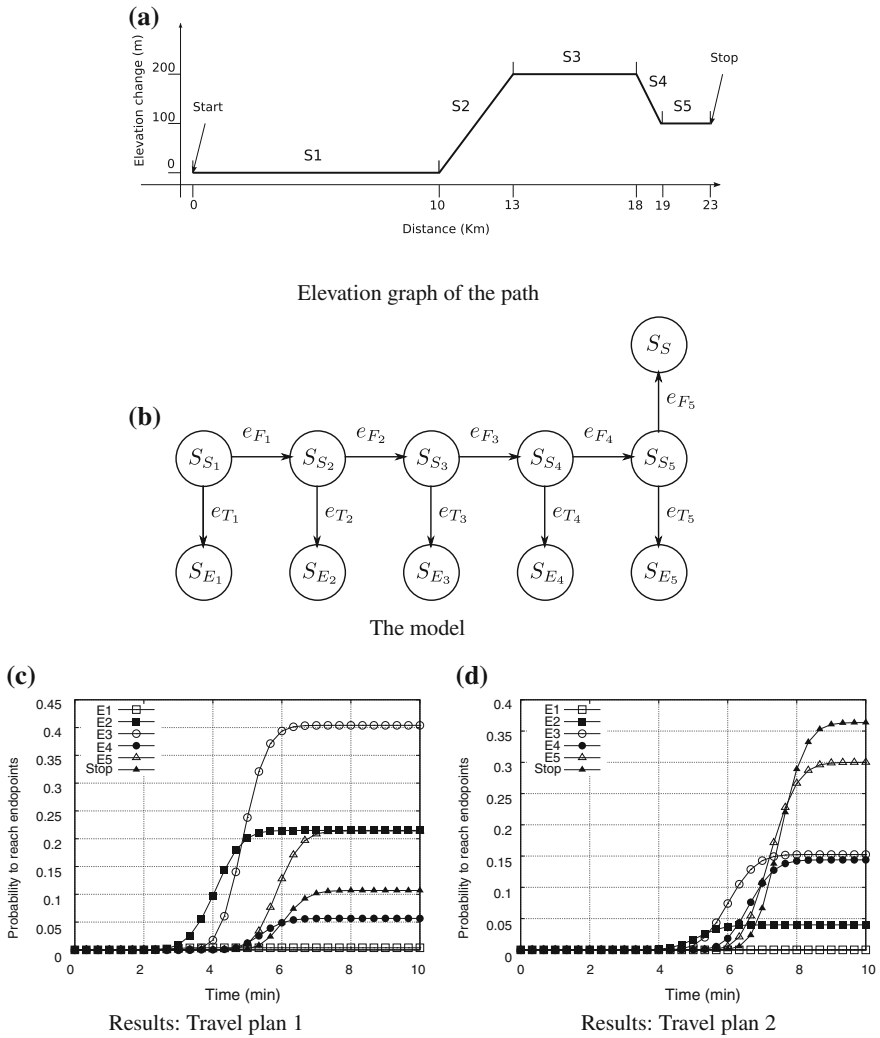


Fig. 7 Car travel planning example

specifications of the car (power vs. engine speed graph, final drive ratio, gear, and tires). In our model, we assume the expected power as the power used along a specific path section.¹ The amount of fuel $a(\tau)$ used during a time interval $[0, \tau]$ depends on both the *engine fuel consumption* c and the power P since $a = c \cdot P \cdot \tau$. Thus, the

¹This is an approximation we made because the speed is not constant but a r.v. Thus, the supplied power is a r.v. too. The approximation introduced is not required by the proposed method and we made it just to simplify the presentation because our purpose is to show the effectiveness of the technique and not to study the mechanical system with high accuracy.

Table 1 Travel plans

Plane n.	Speed (km/h)				
	S1	S2	S3	S4	S5
1	290	80	290	130	290
2	222	80	222	130	222

quantity of fuel in the tank after τ is $a(\tau) = a_0 - c \cdot P \cdot \tau$, where a_0 is the quantity of fuel in the tank when the car starts running.

We use $a(\tau)$ to define events e_{T_i} , as explained in Sect. 3, by taking into account the fact that P changes according to the path sections but the quantity of fuel at changing points must be preserved. The time to run along the path sections S_i (events e_{F_i}) are modeled by Erlang distributions with the mean time depending on the speed provided by the travel plan and a coefficient of variation estimated according to the environmental conditions. The travel plans we considered are reported in Table 1. Moreover, we assumed the car has just 15 l of fuel in its tank. Fuel consumption depends on the power and the speed of the engine. It is important to remark that the fuel consumption may also increase in case of lower speeds. The numerical values are taken from the fuel consumption graph of a BMW M3 E46 car and the engine installed as its standard equipment. For lack of space, we do not show all the mechanical feature graphs used to obtain such values.

Figure 7c, d show the results related to the two travel plans of Table 1. The lesson learned by observing the graphs is that plan 1 does not ensure the car arrives at the end of the path (the *Stop* curve), since it has 0.4 probability to stop at the middle of the overall path and a probability equal to 0.1 to arrive at destination. Plan 2 gives more confidence about the possibility to complete the travel with a probability equal to 0.36. On the other hand, the time required to travel along path 2 is longer than the path 1 one, due to the reduced speed in path sections S1, S3, and S5.

6 Conclusions

In this chapter, we considered a class of systems characterized by different working conditions that alternate and change a system’s stochastic behavior. In such systems, the continuity of particular quantities needs to be preserved and this calls for adequate analytical techniques. The chapter summarizes our work on an analytical framework able to represent such phenomena which is based on phase-type distributions and on an ad hoc fitting algorithm relying on codomain discretization. The framework is general enough to model a wide range of systems and, to show its usefulness, four examples have been presented both for ICT and physical systems. Different quantitative analyses have been conducted in order to provide an overview of the available modeling tools.

References

1. Finkelstein MS (1999) Wearing-out of components in a variable environment. *Reliab Eng Syst Saf* 66(3):235–242
2. Sedyakin N (1966) On one physical principle in reliability theory. *Tekhn Kibernetika* (in Russian—Tech Cybern) 3:80–87
3. Ciardo G, Marie R, Sericola B, Trivedi K (1990) Performability analysis using semi-Markov reward processes. *IEEE Trans Comput* 39:1252–1264
4. Limnios N, Oprisan G (2001) Semi-Markov processes and reliability, ser. Statistics for industry and technology. Birkhäuser, Boston, MA, USA
5. Griboaud M, Bobbio A, Sereno M (2001) Modeling physical quantities in industrial systems using fluid stochastic Petri nets. In: Proceeding of fifth international workshop on performability modeling of computer and communication systems (PMCCS 5), 2001, pp 81–85
6. Ciardo G, Nicol D, Trivedi KS (1999) Discrete-event simulation of fluid stochastic Petri nets. *IEEE Trans Softw Eng* 2:207–217
7. Brokemeyer F, Halstron HS, Jensen A (1948) The life and works of A. K. Erlang. *Trans Danish Acad Tech Sci* 2
8. Neuts MF, Meier KS (1981) On the use of phase type distributions in reliability modelling of systems with two components. *OR Spectr* 2(4):227–234
9. Neuts MF (1981) Matrix-geometric solutions in stochastic models: an algorithmic approach. Johns Hopkins University Press, Baltimore
10. Scarpa M (1999) Non Markovian stochastic Petri nets with concurrent generally distributed transitions. PhD dissertation, University of Turin
11. Pérez-Ocón R, Montoro-Cazorla D (2004) A multiple system governed by a quasi-birth-and-death process. *Reliab Eng Syst Saf* 84(2):187–196
12. Buchholz P, Ciardo G, Donatelli S, Kemper P (2000) Complexity of memory-efficient kronecker operations with applications to the solution of markov models. *Inf J Comput* 12(3):203–222
13. Distefano S, Longo F, Scarpa M (2010) Availability assessment of ha standby redundant clusters. In: SRDS, 2010, pp 265–274
14. Distefano S, Longo F, Scarpa M (2010) Symbolic representation techniques in dynamic reliability evaluation. In: HASE, 2010, pp 45–53
15. Bruneo D, Distefano S, Longo F, Puliafito A, Scarpa M (2010) Reliability assessment of wireless sensor nodes with non-linear battery discharge. In: Wireless Days (WD), 2010 IFIP, Oct 2010, pp 1–5
16. Bruneo D, Longo F, Puliafito A, Scarpa M, Distefano S (2010) Software rejuvenation in the cloud. In: Proceedings of the 5th international ICST conference on simulation tools and techniques, ser. SIMUTOOLS '12. ICST, ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), Brussels, Belgium, Belgium, pp 8–16 [Online]. <http://dl.acm.org/citation.cfm?id=2263019.2263022>
17. Bruneo D, Distefano S, Longo F, Puliafito A, Scarpa M (2012) Evaluating wireless sensor node longevity through Markovian techniques. *Comput Netw* 56(2):521–532
18. Bruneo D, Distefano S, Longo F, Puliafito A, Scarpa M (2013) Workload-based software rejuvenation in cloud systems. *IEEE Trans Comput* 62(6):1072–1085
19. Distefano S, Bruneo D, Longo F, Scarpa M (2013) Quantitative dependability assessment of distributed systems subject to variable conditions. In: Pathan M, Wei G, Fortino G (eds) Internet and distributed computing systems, ser. Lecture notes in computer science, vol 8223. Springer, Berlin, pp 385–398 [Online]. http://dx.doi.org/10.1007/978-3-642-41428-2_31
20. Distefano S, Longo F, Scarpa M (2013) Investigating mobile crowdsensing application performance. In: Proceedings of the third ACM international symposium on design and analysis of intelligent vehicular networks and applications, ser. DIVANet '13. ACM, New York, NY, USA, pp. 77–84 [Online]. <http://doi.acm.org/10.1145/2512921.2512931>

21. Distefano S, Longo F, Scarpa M, Trivedi K (2014) Non-markovian modeling of a blade center chassis midplane. In: Horvath A, Wolter K (eds) *Computer performance engineering*, ser. Lecture notes in computer science, vol 8721. Springer International Publishing, pp. 255–269 [Online]. http://dx.doi.org/10.1007/978-3-319-10885-8_18
22. Longo F, Bruneo D, Distefano S, Scarpa M (2010) Variable operating conditions in distributed systems: modeling and evaluation. *Concurrency and computation: practice and experience*, pp n/a–n/a, 2014 [Online]. <http://dx.doi.org/10.1002/cpe.3419>
23. Longo F, Distefano S, Bruneo D, Scarpa M (2015) Dependability modeling of software defined networking. *Comput Netw* 83(0):280–296 [Online]. <http://www.sciencedirect.com/science/article/pii/S1389128615001139>
24. Cumani A (1985) ESP—a package for the evaluation of stochastic Petri nets with phase-type distributed transition times. In: *PNPM*, pp 144–151
25. Bobbio A, Scarpa M (1998) Kronecker representation of stochastic petri nets with discrete ph distributions. In: *Proceedings of the third IEEE annual international computer performance and dependability symposium (IPDS '98)*
26. Horvath A, Puliafito A, Telek M, Scarpa M (2000) Analysis and evaluation of non-markovian stochastic Petri nets. In: *Tools 2000*. Springer, Schaumburg, IL, USA, March 2000, pp 171–187
27. Buchholz P, Ciardo G, Donatelli S, Kemper P (1997) Complexity of kronecker operations on sparse matrices with applications to the solution of markov models. Technical report
28. Longo F, Scarpa M (2009) Applying symbolic techniques to the representation of non-markovian models with continuous ph distributions. In: *EPEW '09: Proceedings of the 6th European performance engineering workshop on computer performance engineering*. Springer, Berlin, Heidelberg, pp 44–58
29. Longo F, Scarpa M (2015) Two-layer symbolic representation for stochastic models with phase-type distributed events. *Int J Syst Sci* 46(9):1540–1571
30. Franz R, Gradischnig K, Huber M, Stiefel R (1994) Atm-based signaling network topics on reliability and performance. *IEEE J Sel Areas Commun* 12(3):517–525
31. Herrmann C, Lott M, Du Y, Hettich A (1998) A wireless atm lan prototype: test bed implementation and first performance results. In: *Proceedings of the of MTT-S wireless'98 symposium*, pp 145–150
32. Liu H-L, Shooman M (2003) Reliability computation of an ip/atm network with congestion. In: *Annual reliability and maintainability symposium*, pp 581–586
33. Bruneo D (2014) A stochastic model to investigate data center performance and qos in iaas cloud computing systems. *IEEE Trans Parallel Distrib Syst* 25(3):560–569
34. Ganti R, Ye F, Lei H (2011) Mobile crowdsensing: current state and future challenges. *IEEE Commun Mag* 49(11):32–39
35. Carroll A, Heiser G (2010) An analysis of power consumption in a smartphone. In: *Proceedings of the 2010 USENIX conference on USENIX annual technical conference*, ser. *USENIX-ATC'10*. USENIX Association, Berkeley, CA, USA, pp 21–21

Principles of Performance and Reliability Modeling and
Evaluation

Essays in Honor of Kishor Trivedi on his 70th Birthday

Fiondella, L.; Puliafito, A. (Eds.)

2016, XXII, 655 p. 216 illus., Hardcover

ISBN: 978-3-319-30597-4