Probabilistic Analysis

Dorin Maxim, Liliana Cucu-Grosjean, and Robert I. Davis

Contents

Abstract

The classical model of a real-time system consists of a number of tasks, each of which has an execution time which is upper bounded by a constant, referred to as the worst-case execution time (WCET). Further, jobs of each task execute periodically or sporadically, subject to some minimum inter-arrival time. Task

D. Maxim
University of Lorraine, Nancy, France

L. Cucu-Grosjean
Inria, Paris, France
e-mail: liliana.cucu@inria.fr

R. I. Davis (✉)
University of York, York, UK
e-mail: rob.davis@york.ac.uk

Y. Tian, D. C. Levy (eds.), *Handbook of Real-Time Computing*,
https://doi.org/10.1007/978-981-4585-87-3_9-1

execution is controlled by a real-time scheduler that determines, at any given time, which of the ready jobs the processor will execute. For such a model, schedulability analysis provides an a priori mathematical verification indicating whether or not all of the jobs of each task can be guaranteed to meet their deadlines under the particular scheduling policy used. This analysis is typically achieved by determining the worst-case scenario that leads to the worst-case response time (from the release to the completion of any job of the task), calculating the worst-case response time, and comparing it with the task's deadline. Probabilistic real-time systems differ from this classical model in two main ways. Firstly, at least one parameter of the tasks (e.g., execution time) is modeled as a random variable, i.e., described by a probability distribution. Secondly, rather than requiring an absolute guarantee that all deadlines must be met, timing constraints are specified in terms of a threshold on the acceptable probability of a deadline miss for each task. This chapter focuses on research into scheduling and specifically schedulability analysis for probabilistic real-time systems.

1 Introduction

Real-time systems are characterized not only by the need for functional correctness but also by the need for timing correctness. Classically, applications have been categorized as either *hard* real-time, when failure to meet a deadline constitutes a failure of the application, or *soft* real-time, where completion beyond the deadline leads only to a degraded quality of service.

Determining timing correctness for a hard real-time system typically requires two steps:

- *Timing Analysis* is used to determine the maximum amount of time which each software task can take to execute on the hardware platform, referred to as the worst-case execution time (WCET) (Wilhelm et al. 2008).
- *Schedulability Analysis* is then used to determine the worst-case response time (WCRT) of each task, taking into account the scheduling policy and thus any interference between the tasks. This analysis typically assumes that every job of a task executes for its WCET. The WCRT is then compared to the task's deadline to determine if it is schedulable (Davis 2014).

The concept of a *probabilistic real-time system* differs from the classical model in two main ways. Firstly, at least one parameter of the tasks (e.g., execution time) is modeled as a random variable, i.e., described by a probability distribution with distinct probabilities associated with each possible discrete value for the parameter. Secondly, rather than requiring an absolute guarantee that all deadlines must be met, timing constraints are specified in terms of a threshold on the acceptable probability of a deadline miss for each task.

Determining the timing correctness of a probabilistic real-time system typically also requires two steps:

- *Probabilistic Timing Analysis* is used to determine the probabilistic worst-case execution time (pWCET) distribution for each task. This may be obtained either via analytical techniques referred to as *static probabilistic timing analysis (SPTA)* (Cazorla et al. 2013; Davis et al. 2013; Altmeyer and Davis 2014; Altmeyer et al. 2015; Lesage et al. 2015, 2018) or via statistical methods referred to as *measurement-based probabilistic timing analysis (MBPTA)* (Cucu-Grosjean et al. 2012; Wartel et al. 2013; Santinelli et al. 2014, 2017; Lima et al. 2016; Lima and Bate 2017).
- *Probabilistic Schedulability Analysis* is then used to determine the probabilistic worst-case response time (pWCRT) distribution of each task, taking into account the scheduling policy and thus any interference between the tasks (Maxim and Cucu-Grosjean 2013). The pWCRT distributions are then compared to the deadlines to determine if the tasks can be guaranteed to meet their timing requirements, described in terms of acceptable deadline miss probabilities.

The remainder of this section introduces the key concepts, terminology, and notation needed to describe probabilistic real-time systems. The following sections present the state-of-the-art probabilistic schedulability analysis techniques for the commonly used fixed priority preemptive scheduling policy. Section 2 presents schedulability analysis for single processor systems with task execution times described by random variables. Section 3 presents results on efficient priority assignment policies which can determine an optimal priority assignment, ensuring that all tasks will meet their timing constraints whenever there is some priority assignment that can provide such a guarantee. Section 4 considers the complexity of probabilistic schedulability analysis and discusses practical methods of improving the efficiency of the analysis. In a brief chapter such as this, detailed information can necessarily only be provided on specific results; however, Sect. 5 complements this via a brief overview of prior work in the field. Section 6 concludes with a discussion of open problems.

1.1 Probabilistic Terminology and Notation

This subsection introduces the basic notation for random variables and operations upon them.

A random variable $\mathscr{X}$ has an associated probability function (PF) $f_{\mathscr{X}}(.)$ with $f_{\mathscr{X}}(x) = P(\mathscr{X} = x)$. The possible values $X^0, X^1, \cdots, X^k$ of $\mathscr{X}$ belong to the interval $[X^{min}, X^{max}]$, where k is the number of possible values of $\mathscr{X}$. (Note discrete random variables are assumed.)

Probabilities are associated with the possible values of a random variable $\mathscr{X}$ using the following notation:

$$\mathscr{X} = \begin{pmatrix} X^0 = X^{min} & X^1 & \cdots & X^k = X^{max} \\ f_{\mathscr{X}}(X^{min}) & f_{\mathscr{X}}(X^1) & \cdots & f_{\mathscr{X}}(X^{max}) \end{pmatrix}, \tag{1}$$

where $\sum_{j=0}^{k} f_{\mathscr{X}}(X^j) = 1$. A random variable may also be specified using its cumulative distribution function (CDF) $F_{\mathscr{X}}(x) = \sum_{z=X^{min}}^{x} f_{\mathscr{X}}(z)$. For example, the random variable $\mathscr{X} = \begin{pmatrix} 1 & 2 & 5 \\ 0.9 & 0.05 & 0.05 \end{pmatrix}$ has a cumulative distribution function

$$F_{\mathscr{X}}(x) = \begin{cases} 0.9, & \text{if } x = 1; \\ 0.95, & \text{if } x = 2; \\ 1, & \text{otherwise} \end{cases}.$$

Throughout this chapter, cursive characters are used to denote random variables.

Definition 1 Two random variables $\mathscr{X}$ and $\mathscr{Y}$ are (probabilistically) **independent** if they describe two events where the result of one of the events has no effect on the other.

For example, if the execution time observed for one job of a task has no impact on the probability of obtaining any particular execution time for the next (or subsequent) job of the task, then the execution times of the jobs are said to be independent. (Note that in practice the execution times of jobs are typically dependent.)

Note that for independent random variables, the conditional probability of $\mathscr{X} = x$ given that $\mathscr{Y} = y$ is simply the probability of $\mathscr{X} = x$ i.e., $P(\mathscr{X} = x | \mathscr{Y} = y) = P(\mathscr{X} = x)$, and similarly, the conditional probability of $\mathscr{Y} = y$ given $\mathscr{X} = x$ is simply the probability of $\mathscr{Y} = y$, i.e., $P(\mathscr{Y} = y | \mathscr{X} = x) = P(\mathscr{Y} = y)$.

Definition 2 The sum $\mathscr{Z}$ of two independent random variables $\mathscr{X}$ and $\mathscr{Y}$ is given by their **convolution** $\mathscr{X} \otimes \mathscr{Y}$ where $P(\mathscr{Z} = z) = \sum_{k=-\infty}^{k=+\infty} P(\mathscr{X} = k) P(\mathscr{Y} = z - k)$.

For example, the convolution of $\mathscr{X} = \begin{pmatrix} 3 & 7 \\ 0.1 & 0.9 \end{pmatrix}$ and $\mathscr{Y} = \begin{pmatrix} 0 & 4 \\ 0.9 & 0.1 \end{pmatrix}$ is equal to

$$\mathscr{Z} = \begin{pmatrix} 3 & 7 \\ 0.1 & 0.9 \end{pmatrix} \otimes \begin{pmatrix} 0 & 4 \\ 0.9 & 0.1 \end{pmatrix} = \begin{pmatrix} 3 & 7 & 11 \\ 0.09 & 0.82 & 0.09 \end{pmatrix}$$

Definition 3 The **coalescence** of two partial random variables, denoted by the operator $\oplus$, represents the combination of the two partial random variables into a single (partial) random variable so that values that appear multiple times are kept only once gathering the summed probability mass of the respective values. (Note a *partial* random variable has probabilities that sum to less than 1.)

For example, coalescing two partial random variables $\mathscr{A}_1 = \begin{pmatrix} 5 & 8 \\ 0.18 & 0.02 \end{pmatrix}$ and $\mathscr{A}_2 = \begin{pmatrix} 5 & 6 \\ 0.72 & 0.08 \end{pmatrix}$ is equal to

$$\begin{pmatrix} 5 & 8 \\ 0.18 & 0.02 \end{pmatrix} \oplus \begin{pmatrix} 5 & 6 \\ 0.72 & 0.08 \end{pmatrix} = \begin{pmatrix} 5 & 6 & 8 \\ 0.9 & 0.08 & 0.02 \end{pmatrix}$$

Definition 4 (Diaz et al. 2004; López et al. 2008) Let $\mathscr{X}_1$ and $\mathscr{X}_2$ be two random variables. The variable $\mathscr{X}_2$ **is greater than or equal to** $\mathscr{X}_1$, denoted by $\mathscr{X}_2 \succeq \mathscr{X}_1$, if $F_{\mathscr{X}_1}(x) \leq F_{\mathscr{X}_2}(x), \forall x$. Stated otherwise, the CDF of $\mathscr{X}_1$ is never above that of $\mathscr{X}_2$.

Note the relation $\succeq$ between two random variables is not total, i.e., for two random variables $\mathscr{X}_3$ and $\mathscr{X}_4$ it is possible that $\mathscr{X}_3 \not\succeq \mathscr{X}_4$and $\mathscr{X}_4 \not\succeq \mathscr{X}_3$.

1.2 Probabilistic Task Model

This subsection defines a probabilistic real-time task model with task parameters described by random variables.

Let τ be a task set comprising n tasks $\{\tau_1, \tau_2, \ldots, \tau_n\}$, where each task τ_i generates a potentially unbounded number of successive jobs $J_{i,j}$, with $j = 1, \ldots, \infty$.

Definition 5 The **probabilistic execution time (pET)** of a specific job of a task describes the probability that the execution time of the job is equal to a given value.

For example, the jth job $J_{i,j}$ of a task τ_i may have a pET as follows:

$$\mathscr{C}_i^j = \begin{pmatrix} 2 & 3 & 5 & 6 & 105 \\ 0.7 & 0.2 & 0.05 & 0.04 & 0.01 \end{pmatrix} \tag{2}$$

If $f_{\mathscr{C}_i^j}(2) = 0.7$, then the execution time of the job $J_{i,j}$ has a probability of 0.7 of being equal to 2.

Note that the pET of a job typically depends on the set of input values for that specific job.

Definition 6 The **probabilistic worst-case execution time (pWCET)** $\mathscr{C}_i$ of a task is a tight upper bound on the pET of all possible jobs of that task. The pWCET can be described by the relation $\succeq$ where $\mathscr{C}_i \succeq \mathscr{C}_i^j, \forall j$. The CDF of the pWCET is defined by taking the point-wise minimum values from the CDFs of the pETs of

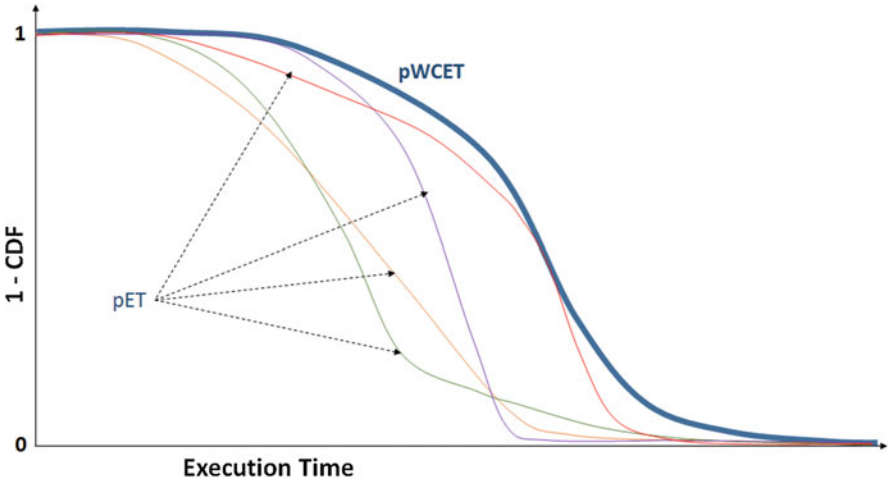

Fig. 1 The pWCET of a task is an upper bound on the pETs of all the jobs

all of the jobs. Equivalently, the 1 – CDF of the pWCET is defined by taking the point-wise maximums from the 1 – CDFs of all of the jobs.

The probabilistic worst-case execution time $\mathscr{C}_i$ of task τ_i can be written as:

$$\mathscr{C}_i = \begin{pmatrix} C_i^0 = C_i^{min} & C_i^1 & \cdots & C_i^{k_i} = C_i^{max} \\ f_{\mathscr{C}_i}(C_i^{min}) & f_{\mathscr{C}_i}(C_i^1) & \cdots & f_{\mathscr{C}_i}(C_i^{max}) \end{pmatrix}, \tag{3}$$

where $\sum_{j=0}^{k_i} f_{\mathscr{C}_i}(C_i^j) = 1$.

For example, a task τ_i can have a pWCET of $\mathscr{C}_i = \begin{pmatrix} 2 & 3 & 25 \\ 0.5 & 0.45 & 0.05 \end{pmatrix}$; then $f_{C_i}(2) = 0.5$, $f_{C_i}(3) = 0.45$ and $f_{C_i}(25) = 0.05$.

The relation between the pWCET of a task and the pETs of its jobs is illustrated in Fig. 1. On this graph of 1 – CDF, the pWCET $\mathscr{C}_i$ is greater than or equal to $\mathscr{C}_i^j$, $\forall j$.

Note that in practice, a precise (tight) pWCET may not necessarily be obtained; however, any upper bound (in terms of the 1 – CDF) on all pETs is valid; the tighter the bound the less pessimism there will be in the subsequent analysis. In the remainder of this chapter, pWCET is used to refer to a valid upper bound.

It is important to note that the random variables describing the pWCETs $\mathscr{C}_1$ and $\mathscr{C}_2$ of two tasks τ_1 and τ_2 are independent due to the definition of the pWCETs as upper bounds. By contrast, the pETs of two jobs of the same or different tasks are typically dependent.

A task is referred to as *periodic* if releases of its jobs occur with a fixed interval of time between them. Alternatively, a task is referred to as *sporadic* if job releases

are separated by some minimum inter-arrival time but may also be released with a larger separation.

A probabilistic real-time task τ_i can therefore be defined by a tuple $(\mathscr{C}_i, T_i, D_i)$ where the random variable $\mathscr{C}_i$ gives the pWCET of the task, T_i is the minimum inter-arrival time or period, and D_i is the relative deadline.

Note that when the pWCET distribution is degenerate (i.e., only has a single value), then the model effectively reduces to the classical periodic or sporadic task model for hard real-time systems.

1.3 Probabilistic Real-Time Constraints

The previous subsection defined the parameters of probabilistic real-time tasks. This subsection defines the corresponding probabilistic time constraints.

In classical hard real-time systems, the response time of a job is the time between its release and completion of its execution, while the worst-case response time of a task is the longest response time of any of its jobs. This is compared to the relative deadline of the task to determine if it is schedulable.

In a probabilistic real-time system, the probabilistic response time (pRT) of a job and the probabilistic worst-case response time (pWCRT) of a task are described by random variables.

Definition 7 The **probabilistic Response Time (pRT)** of a job $J_{i,j}$ of task τ_i, denoted by $\mathscr{R}_{i,j}$, describes the probability distribution of the response time of that job.

Definition 8 The **probabilistic worst-case response time (pWCRT)** of a task τ_i, denoted by $\mathscr{R}_i$, is an upper bound on the pRTs of all of its jobs $\mathscr{R}_{i,j}, \forall j$ described by the relation $\succeq$ with $\mathscr{R}_i \succeq \mathscr{R}_{i,j}, \forall j$. Graphically, this implies that the 1 – CDF of $\mathscr{R}_i$ is never below the 1 – CDF of $\mathscr{R}_{i,j}, \forall j$.

Probabilistic real-time constraints are expressed in the form of a *threshold* ρ_i specifying the maximum acceptable probability of a deadline miss for task τ_i with relative deadline D_i. Typically, the value of the threshold is very small e.g., 10^{-4} to 10^{-9}, since it is expected that deadline failures should be rare events.

In the literature, there are two ways in which the probability of a deadline miss may be calculated for a task:

- The Deadline Miss Probability (DMP) for a task is calculated by taking the average of the probability of a deadline miss for its jobs over some long interval of time; typically the least common multiple (LCM) of the task periods (Diaz et al. 2004; López et al. 2008).
- The Worst-Case Deadline Failure Probability (WCDFP) of a task is upper bounded by directly comparing the pWCRT distribution of the task (valid for any job) with its deadline (Maxim and Cucu-Grosjean 2013).

Note that the latter method potentially introduces some pessimism, since, for example, the relationship between task periods means that not all jobs of a task may be subject to the maximum interference from other tasks and so have a pRT distribution that equates to the pWCRT distribution of the task; however, it provides a valid upper bound on the probability of deadline misses.

Definition 9 The **deadline miss probability for a job** $J_{i,j}$, denoted by $DMP_{i,j}$, is the probability that the jth job of task τ_i misses its deadline and is given by:

$$DMP_{i,j} = P(\mathscr{R}_{i,j} > D_i). \tag{4}$$

where $\mathscr{R}_{i,j}$ is the pRT distribution for the jth job of the task τ_i.

If the tasks studied are periodic, then the deadline miss probability for a task is equal to the average of the deadline miss probabilities of all its jobs activated during the Least Common Multiple of task periods.

Definition 10 The **deadline miss probability for a periodic task** τ_i and a time interval $[a, b]$ equating to the LCM of task periods, denoted by $DMP_i(a, b)$, is given by:

$$DMP_i(a, b) = \frac{P(\mathscr{R}_i^{[a,b]} > D_i)}{n_{[a,b]}} = \frac{1}{n_{[a,b]}} \sum_{j=1}^{n_{[a,b]}} DMP_{i,j} \tag{5}$$

where $n_{[a,b]}$ is the number of jobs of task τ_i activated during the interval $[a, b]$.

Note that the above definition is only valid for tasks that are periodic. Sporadic behavior of higher priority tasks, resulting in intervals between jobs that exceed the minimum inter-arrival time, can, in some cases, result in a higher deadline miss probability for the task under analysis.

Definition 11 The **worst-case deadline failure probability for a task** τ_i, denoted by $WCDFP_i$, is an upper bound on the probability that the task misses its deadline. It is computed directly from the pWCRT and the deadline of the task and is given by:

$$WCDFP_i = P(\mathscr{R}_i > D_i) \tag{6}$$

where $\mathscr{R}_i$ is the pWCRT distribution for task τ_i, and D_i is its relative deadline.

2 Schedulability Analysis for Probabilistic Real-Time Tasks

This section describes the state-of-the-art probabilistic response time analysis for tasks which have probabilistic worst-case execution times (pWCETs). It is a simplified form of the analysis derived by Maxim and Cucu-Grosjean (2013).

The system is assumed to comprise n tasks $\{\tau_1, \tau_2, \ldots, \tau_n\}$ scheduled on a single processor according to a fixed priority preemptive scheduling policy. Each task is assumed to have a unique priority. Without loss of generality, τ_i is assumed to have a higher priority than τ_j for $i < j$. Further, $hp(i)$ is used to denote the set of tasks with higher priorities than τ_i. The tasks are sporadic and thus may all be released at the same time (assumed to be time $t = 0$).

Task τ_i is represented by a tuple $(\mathscr{C}_i, T_i, D_i, \rho_i)$, where $\mathscr{C}_i$ is its pWCET, T_i is its minimum inter-arrival time, D_i is its relative deadline, and ρ_i is the threshold giving the maximum acceptable deadline failure probability. The deadline is assumed to be *constrained*; hence $D_i \leq T_i$, for all tasks.

At runtime, it is assumed that any job that reaches its deadline without completing is aborted.

Maxim and Cucu-Grosjean (2013) proved that the *critical instant*, which yields the largest response time distribution for any job of a task, occurs when all the tasks are released simultaneously. (Here, largest is defined with respect to the relation $\succeq$.) Since the response time distribution of the first job upper bounds the response time distribution of any other job of the same task, it therefore gives the pWCRT distribution for the task ($\mathscr{R}_i = \mathscr{R}_{i,1} \succeq \mathscr{R}_{i,j}\ \forall j$). The pWCRT distribution $\mathscr{R}_i$ of the task can then be compared with its deadline to obtain the worst-case deadline failure probability $WCDFP_i$, which can be compared with the threshold ρ_i to determine if the task is schedulable.

2.1 Probabilistic Response Time Analysis

The following analysis computes the worst-case response time distribution for a given task τ_i.

The worst-case response time distribution for task τ_i is first initialized to:

$$\mathscr{R}_i^0 = \mathscr{B}_i \otimes \mathscr{C}_i \tag{7}$$

where the backlog $\mathscr{B}_i$ at the release of τ_i is given by:

$$\mathscr{B}_i = \bigotimes_{j \in hp(i)} \mathscr{C}_j \tag{8}$$

The worst-case response time is then updated iteratively for each preemption as follows:

$$\mathscr{R}_i^m = (\mathscr{R}_i^{m-1,head} \oplus (\mathscr{R}_i^{m-1,tail} \otimes \mathscr{C}_k^{pr})) \tag{9}$$

Here, m is the index of the iteration. $\mathscr{R}_i^{m-1,head}$ is the part of the distribution $\mathscr{R}_i^{m-1}$ that is not affected by the preemption under consideration (i.e., it only contains values $\leq t_m$ where t_m is the time of the preemption). $\mathscr{R}_i^{m-1,tail}$ is the remaining part of the distribution $\mathscr{R}_i^{m-1}$ that may be affected by the preemption. Finally, $\mathscr{C}_k^{pr}$ is the pWCET distribution of the preempting task τ_k.

Iteration ends when there are no releases left from jobs of higher priority tasks at time instants smaller than the largest value in the response time distribution currently obtained. Iteration may also be terminated once any new preemptions are beyond the deadline of the task.

Once iteration is complete, the worst-case deadline failure probability valid for any job of task τ_i is given by:

$$WCDFP_i = P(\mathscr{R}_i > D_i) \tag{10}$$

The task is then deemed schedulable if the worst-case deadline failure probability does not exceed the required threshold.

$$WCDFP_i \leq \rho_i \tag{11}$$

Hypothesis of (probabilistic) independence Equations (7) and (9) are based on the operation of convolution $\otimes$ that requires probabilistic independence between $\mathscr{C}_i, \forall i$. For this reason, it is important that the probability distributions used for $\mathscr{C}_i$ are upper bound pWCET distributions, and not pET distributions which typically would not be independent.

2.2 Detailed Example

The example below illustrates the operation of probabilistic response time analysis.

Example 1 Assume a task set $\Gamma = \{\tau_1, \tau_2\}$, with task τ_1 defined by $(\begin{pmatrix} 1 & 2 & 3 \\ 0.6 & 0.3 & 0.1 \end{pmatrix}, 5, 5, 1)$ and task τ_2 by $(\begin{pmatrix} 4 & 5 \\ 0.7 & 0.3 \end{pmatrix}, 12, 12, 0.005)$. Note that task τ_1 is required to always meet it's deadline ($\rho_1 = 1$), while task τ_2 has an acceptable threshold of $\rho_2 = 0.005$ on deadline failure.

The response time computation for task τ_2 starts by initializing the response time distribution with the pWCET of the task under analysis. ($\mathscr{R}_i^j$ denotes the current response time distribution of task τ_i at step j of the analysis.)

$$\mathscr{R}_2^0 = \begin{pmatrix} 4 & 5 \\ 0.7 & 0.3 \end{pmatrix} \tag{12}$$

Then the interference from higher priority tasks at $t = 0$ is included to account for the synchronous release of jobs of all tasks:

$$\mathscr{R}_2^1 = \mathscr{R}_2^0 \otimes \begin{pmatrix} 1 & 2 & 3 \\ 0.6 & 0.3 & 0.1 \end{pmatrix} = \begin{pmatrix} 5 & 6 & 7 & 8 \\ 0.42 & 0.39 & 0.16 & 0.03 \end{pmatrix} \tag{13}$$

Once the interference due to synchronous releases has been taken into account, the preemptions can be included and the response time distribution updated. As task τ_1 has an arrival at $t = 5$, then the current response time distribution is split into two parts, one containing values less than or equal to 5, which is referred to as the head of the distribution $\mathscr{R}_2^{1,head}$:

$$\mathscr{R}_2^{1,head} = \begin{pmatrix} 5 \\ 0.42 \end{pmatrix} \tag{14}$$

and another part containing values strictly larger than 5, which is referred to as the tail of the distribution $\mathscr{R}_2^{1,tail}$:

$$\mathscr{R}_2^{1,tail} = \begin{pmatrix} 6 & 7 & 8 \\ 0.39 & 0.16 & 0.03 \end{pmatrix} \tag{15}$$

The head of the distribution contains stable response time values and associated probabilities that are not modified in the subsequent steps of the analysis. The tail of the distribution is updated to take into account the preemption at $t = 5$. After the tail is updated, it is coalesced with the head to once again form a complete distribution $\mathscr{R}_2^2$ which can subsequently be split at the appropriate point to account for further preemptions:

$$\begin{aligned} \mathscr{R}_2^2 &= \mathscr{R}_2^{1,head} \oplus \mathscr{R}_2^{1,tail} \otimes \begin{pmatrix} 1 & 2 & 3 \\ 0.6 & 0.3 & 0.1 \end{pmatrix} \\ &= \mathscr{R}_2^{1,head} \oplus \begin{pmatrix} 7 & 8 & 9 & 10 & 11 \\ 0.234 & 0.213 & 0.105 & 0.025 & 0.003 \end{pmatrix} \\ &= \begin{pmatrix} 5 & 7 & 8 & 9 & 10 & 11 \\ 0.42 & 0.234 & 0.213 & 0.105 & 0.025 & 0.003 \end{pmatrix} \end{aligned} \tag{16}$$

Similarly, task τ_2 may be preempted by task τ_1 at $t = 10$. The current response time distribution $\mathscr{R}_2^2$ is split into the head $\mathscr{R}_2^{2,head}$, containing values less than or equal to 10, and the tail $\mathscr{R}_2^{2,tail}$ containing values larger than 10. The tail part is then updated to include the second preemption from τ_1:

$$\mathscr{R}_2^{2,head} = \begin{pmatrix} 5 & 7 & 8 & 9 & 10 \\ 0.42 & 0.234 & 0.213 & 0.105 & 0.025 \end{pmatrix} \tag{17}$$

$$\mathscr{R}_2^{2,tail} = \begin{pmatrix} 11 \\ 0.003 \end{pmatrix} \tag{18}$$

$$\begin{aligned}\mathscr{R}_2^3 &= \mathscr{R}_2^{2,head} \oplus \mathscr{R}_2^{2,tail} \otimes \begin{pmatrix} 1 & 2 & 3 \\ 0.6 & 0.3 & 0.1 \end{pmatrix} \\ &= \mathscr{R}_2^{2,head} \oplus \begin{pmatrix} 12 & D_2^+ \\ 0.0018 & 0.0012 \end{pmatrix} \\ &= \begin{pmatrix} 5 & 7 & 8 & 9 & 10 & 12 & D_2^+ \\ 0.42 & 0.234 & 0.213 & 0.105 & 0.025 & 0.0018 & 0.0012 \end{pmatrix} \end{aligned} \tag{19}$$

Note D_2^+ collects the probability mass for all values beyond the task deadline.

Since the deadline of task τ_2 is 12, and there are no further preemptions before $t = 15$, which is in any case beyond the end of the response time distribution, iteration can stop at this point. The WCDFP corresponds to the probability mass of the response time distribution $\mathscr{R}_2^2$ that exceeds 12, which is 0.0012. Since this value is less than the threshold $\rho_2 = 0.005$, then task τ_2 is schedulable; it meets its probabilistic timing constraints.

3 Optimal Priority Assignment

For the classical real-time task model, it is well-known that *rate-monotonic* (Liu and Layland 1973) and *deadline-monotonic* (Leung and Whitehead 1982) priority assignment are optimal for task sets with implicit and constrained deadlines, respectively. As shown by Maxim et al. (2011), this is not however the case for task sets with parameters described by random variables and time constraints given as thresholds on acceptable deadline failure probabilities.

3.1 Priority Assignment Example

A simple example suffices to show that neither rate-monotonic nor deadline-monotonic priority assignments are optimal for systems with parameters described by random variables and timing constraints given by thresholds on acceptable deadline failure probabilities.

Consider the following set of two sporadic tasks, which may share a common release time at $t = 0$.

Let $\Gamma = \{\tau_1, \tau_2\}$ be a task set such that each task is characterized by $(\mathscr{C}, T, D, \rho)$. Recall that ρ is the threshold on the acceptable deadline miss probability for the task. Thus τ_1 is defined by $\left(\begin{pmatrix} 2 & 3 \\ 0.5 & 0.5 \end{pmatrix}, 8, 6, 0.7\right)$ and τ_2 by $\left(\begin{pmatrix} 3 & 5 \\ 0.5 & 0.5 \end{pmatrix}, 10, 7, 0.2\right)$.

According to deadline-monotonic priority assignment, τ_1 has the highest priority and τ_2 the lowest priority. In this case the response time of task τ_1 is equal to $\mathscr{R}_1 = \begin{pmatrix} 2 & 3 \\ 0.5 & 0.5 \end{pmatrix}$ and the probability of a deadline miss is zero.

The response time of task τ_2 is equal to $\mathscr{R}_2 = \begin{pmatrix} 5 & 6 & 7 & D_2^+ \\ 0.25 & 0.25 & 0.25 & 0.25 \end{pmatrix}$, having a worst-case deadline failure probability $WCDFP_2 = 0.25$, which is greater than the threshold $\rho_2 = 0.2$. This means that the priority assignment is not feasible.

The alternative priority assignment has τ_2 at the highest priority and τ_1 at the lowest priority. In this case the response time of task τ_2 is equal to $\mathscr{R}_2 = \begin{pmatrix} 3 & 5 \\ 0.5 & 0.5 \end{pmatrix}$, and the probability of a deadline miss is zero.

The response time of task τ_1 is equal to $\mathscr{R}_1 = \begin{pmatrix} 5 & 6 & D_1^+ \\ 0.25 & 0.25 & 0.5 \end{pmatrix}$, having a worst-case deadline failure probability $WCDFP_2 = 0.5$, which is less than the threshold $\rho_1 = 0.7$. This means that the priority assignment is feasible.

This simple example shows that neither rate-monotonic (the same result is obtained with the task periods set equal to the deadlines) nor deadline-monotonic priority assignment is optimal for task sets with parameters described by random variables and time constraints given as thresholds on acceptable deadline miss probabilities.

3.2 Optimal Priority Assignment Using Audsley's Algorithm

Davis and Burns (2011) proved three conditions for the applicability of Audsley's algorithm (Audsley 2001) with a schedulability test S:

1. The schedulability of a task may, according to test S, be dependent on the set of higher-priority tasks, but not on the relative priority ordering of those tasks.
2. The schedulability of a task may, according to test S, be dependent on the set of lower-priority tasks, but not on the relative priority ordering of those tasks.
3. When the priorities of any two tasks of adjacent priority are swapped, the task being assigned the higher priority cannot become unschedulable according to test S, if it was previously schedulable at the lower priority. (As a corollary, the task being assigned to the lower priority cannot become schedulable according to test S, if it was previously unschedulable at the higher priority.)

These conditions may be lifted to the problem of tasks with parameters described by random variables. In this case, the concept of a task being *schedulable* corresponds to meeting its probabilistic time constraints, i.e., having a WCDFP that is below the acceptable threshold for the task.

The schedulability test given in Sect. 2.1 meets both Conditions 1 and 2, since there is no dependency on the order of lower- or higher-priority tasks. Further,

Maxim and Cucu-Grosjean (2013) showed that the pWCRT distribution for a task τ_h at a higher priority is greater than that of a task τ_i at a lower priority (i.e., $\mathscr{R}_h \succeq \mathscr{R}_i$). It follows that Condition 3 also holds.

This means that for task systems analyzed using the schedulability test given in Sect. 2.1, Audsley's algorithm can be used to find an optimal priority assignment with respect to that test. The algorithm guarantees to find a priority ordering that is schedulable according to the test if such an ordering exists. Further, for a set of n tasks, it does so in at most $n(n+1)/2$ task schedulability tests; a large improvement on having to potentially check all $n!$ possible priority orderings.

Algorithm 1 sets out Audsley's optimal priority assignment algorithm for this problem.

Algorithm 1: Audsley's Optimal Priority Assignment algorithm. The function *feasibility* verifies that for task τ_i, $WCDFP_i < p_i$

Input: $\Gamma = \{\tau_i, i \in 1..n\}$ /* initial set of tasks */
Output: Φ /* ordered set of tasks */

```
Φ ← () ;
for l ∈ n..1 do
    assignment ← FALSE ;
    for τ_i ∈ Γ' do
        /* feasibility function such that WCDFP_i < p_i */;
        if feasible(τ_i, Φ) then
            Φ ← Φ.τ_i ;
            Γ' ← Γ'\{τ_i} ;
            assignment ← TRUE ;
            break;
    if assignment = FALSE then
        /* no task is suitable for this priority level */;
        break;
```

Proof that deadline-monotonic priority assignment is not optimal for this problem and that Audsley's algorithm is applicable was first given by Maxim et al. (2011).

4 Complexity of Probabilistic Schedulability Analyses

Compared to classical response time analysis for tasks with deterministic parameters, probabilistic response time analysis for tasks with execution times described by random variables, i.e., pWCET distributions, may have much higher computational complexity. This is due to two factors, the additional information in the pWCET distributions and the effects of the convolution operator $\otimes$.

When convolving two distributions that have m and n values, respectively, the resulting distribution can have up to $m \times n$ values. This is true when the two

distributions that are convolved are very different from one another, for example, the gaps between each pair of values in one distribution are larger than the maximum value in the other distribution. In other cases, for example, when the distributions are dense with all values separated by 1, then the resulting distribution can have no more than $m + n - 1$ values.

In general, probabilistic response time analysis could produce a pWCRT distribution which contains the largest value equal to the deterministic response time that would be obtained by considering the largest value in each pWCET distribution (the so-called *limit condition*) and nearly all values below it. This distribution could easily be too large to handle efficiently in practice.

One way of dealing with this complexity problem is through *resampling* (Maxim et al. 2012). Resampling can be used to reduce the number of values within the pWCET distributions of the tasks and also within the intermediate distributions used in the pWCRT calculation.

Definition 12 (Sound resampling) Let $\mathscr{C}_i$ be a distribution with n values describing the pWCET of a task τ_i. The process of resampling involves the approximation of $\mathscr{C}_i$ by some other distribution $\mathscr{C}'_i$ that has $k < n$ values and is greater than or equal to $\mathscr{C}_i$, i.e., $\mathscr{C}'_i \succeq \mathscr{C}_i$.

Sound resampling ensures that if $\mathscr{C}'_i$ is used in place of $\mathscr{C}_i$ in probabilistic response time analysis, then the resulting pWCRT distribution $\mathscr{R}'_i$ obtained will be an upper bound on the pWCRT distribution $\mathscr{R}_i$ obtained using $\mathscr{C}_i$ (Diaz et al. 2004).

Many forms of sound resampling are possible, since a sound resampling simply moves probability mass from smaller to larger values. Maxim et al. (2012) explored a number of different resampling strategies, the most effective of which is *domain quantization*.

Domain quantization not only reduces the number of values in each distribution; it also reduces the number of values in the resulting distribution after convolution. The idea is to quantize the values to some multiple of a base quantum. The approach is best illustrated via an example. Assume there are two tasks with pWCET distributions as follows:

$$\mathscr{C}_1 = \begin{pmatrix} 2 & 3 & 6 & 8 & 9 \\ 0.1 & 0.2 & 0.3 & 0.1 & 0.3 \end{pmatrix}$$

$$\mathscr{C}_2 = \begin{pmatrix} 10 & 11 & 12 & 17 & 19 & 20 \\ 0.1 & 0.25 & 0.35 & 0.15 & 0.10 & 0.05 \end{pmatrix}$$

Convolving these two distributions gives the following distribution:

$$\mathscr{R}_2 = \begin{pmatrix} 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 \\ 0.01 & 0.045 & 0.085 & 0.07 & 0.03 & 0.075 & 0.115 & 0.07 & 0.14 & 0.115 & 0.025 & 0.055 \\ 25 & 26 & 27 & 28 & 29 \\ 0.045 & 0.06 & 0.01 & 0.035 & 0.015 \end{pmatrix}$$

Applying domain quantization with a quantum of 3 gives:
$\mathscr{C}'_1 = \begin{pmatrix} 3 & 6 & 9 \\ 0.3 & 0.3 & 0.4 \end{pmatrix}$ and $\mathscr{C}'_2 = \begin{pmatrix} 12 & 18 & 21 \\ 0.7 & 0.15 & 0.15 \end{pmatrix}$

Note that the probability mass is collected at the next value which is a multiple of the quantum (i.e., a multiple of 3), including in the case of $\mathscr{C}'_2$ a value of 21 which is larger than the maximum in the original distribution. Convolving these two distributions gives: $\mathscr{R}'_2 = \begin{pmatrix} 15 & 18 & 21 & 24 & 27 & 30 \\ 0.21 & 0.21 & 0.325 & 0.09 & 0.105 & 0.06 \end{pmatrix}$

Note that $\mathscr{R}'_2 \succeq \mathscr{R}_2$. Further, as all of the values in $\mathscr{R}'_2$ are multiples of the quantum, subsequent convolution with distributions that have been resampled via domain quantization with a quantum of 3 can only produce values that are also multiples of the quantum, limiting the increase in the number of values in the distribution.

Choosing the quanta to be used is an important problem, since it determines the number of samples to be kept per distribution, scaling a distribution with a large number of values to a large quanta means that few values are kept out of the initial number, and so the loss in precision is potentially large; on the other hand, scaling a large distribution to a small quanta results in keeping too many values, which makes the resampling inefficient. This problem can be solved by taking advantage of the fact that convolution is commutative, so, when there are multiple distributions to be convolved with each other, which is often the case in probabilistic response time analysis, first the small distributions (representing tasks with relatively short execution times) are convolved among themselves until they become bigger, and they can be convolved with larger distributions. To facilitate this, Maxim et al. (2012) recommend setting the quanta for each distribution to the smallest power of 2 (e.g., 1, 2, 4, 8...) that results in at most k samples.

Resampling to a smaller number of values trades off between analysis precision and runtime complexity. Note that with a sound resampling, the pWCRT distributions obtained are always upper bounds, and so the computed values for the worst-case deadline failure probability are valid but potentially pessimistic.

5 Review of Prior Work

This section briefly reviews research on probabilistic response time analysis. Note other forms of probabilistic schedulability analysis also exist, for example, (i) for systems where servers are used to manage task execution (Abeni and Buttazzo 1998, 1999; Abeni et al. 2012; Palopoli et al. 2012; Frias et al. 2017), (ii) based on real-time queuing theory (Lehoczky 1996; Hansen et al. 2002), and (iii) where the response time distribution is obtained directly via statistical methods based on measurements (Lu et al. 2010, 2012; Maxim et al. 2015). These areas are not covered in detail here.

Woodbury and Shin (1988) provided analysis that computes the probability of deadline failure for periodic tasks. They assumed that each task has multiple paths

each with a fixed execution time and a probability of occurrence. They computed the response time distribution for each job over the hyperperiod and hence the deadline miss probability for each task.

Tia et al. (1995) proposed a *probabilistic time-demand analysis* (PTDA) based on the time-demand analysis technique given for the simpler case of deterministic execution times by Lehoczky et al. (1989). At each scheduling point, the cumulative probability distribution is computed for all job releases up to that point, via convolution. This enables a bound to be computed on the probability that the task can meet its deadline.

Gardner and Liu (1999) presented *stochastic time-demand analysis* (STDA) which computes a lower bound on the probability that jobs of a task will meet their deadlines under fixed priority scheduling. They note an issue with the prior work of Tia et al. (1995) in that it is only valid if there is no backlog at the deadline of a task. Gardner and Liu (1999) solve this problem by considering busy periods and the backlog present at subsequent releases of each job.

Diaz et al. (2002) introduced a method of computing the response time distribution for all of the jobs in the hyperperiod for a set of periodic tasks scheduled using fixed priorities or EDF. They note that earlier work (Tia et al. 1995; Gardner and Liu 1999) assumes that the worst case occurs for a job in the first busy period following synchronous release; however, this is not necessarily correct when the worst-case utilization exceeds 1. Diaz et al. (2002) show that the *backlog* at the start of each hyperperiod is stationary provided that the *average utilization* is less than 1. They give a method to find this stationary backlog and hence compute the worst-case response time distribution for each job in the hyperperiod.

Diaz et al. (2004) introduced the concept of *greater than or equal to* between random variables $\mathscr{X} \succeq \mathscr{Y}$. They note that any approximations in the analysis must result in distributions that are *greater than or equal to* the exact distribution in order to ensure soundness. Diaz et al. (2004) also highlighted and addressed issues with their previous work (Diaz et al. 2002) in relation to the tractability of the backlog computation. They also provided a sketch proof that the priority assignment algorithm of Audsley (2001) is optimal when execution times are described by random variables. This was later confirmed by the work of Maxim et al. (2011).

López et al. (2008) extended earlier work (Diaz et al. 2004), providing a set of transformations that can be made to the parameters of a system which are guaranteed to result in a response time distribution greater than or equal to (i.e., $\succeq$) that for the original system.

Kim et al. (2005) built upon the analysis framework of Diaz et al. (2002, 2004). They discussed methods for obtaining the stationary backlog, including an exact solution which has a very high computational cost, and two approximate solutions.

Cucu and Tovar (2006) introduced a method of computing the probabilistic worst-case response time distribution for tasks with constant execution times but inter-arrival times modeled via random variables. Kaczynski et al. (2007) later addressed the more complex model where tasks have both execution times and arrival times modeled via random variables.

Ivers and Ernst (2009) presented analysis that accounts for the effect of unknown statistical dependencies between the execution times of jobs of the same task and jobs of different tasks, with the execution times modeled as random variables.

Cucu-Grosjean (2013) considered different types of *independence* in the context of probabilistic real-time systems. A key aspect of this work is the discussion covering the definition of and the differences between probabilistic execution time distributions (pET) and probabilistic worst-case execution time distributions (pWCET).

Maxim and Cucu-Grosjean (2013) introduced probabilistic response time analysis for tasks which may have their worst-case execution times, inter-arrival times, and deadlines all described by random variables.

Tanasa et al. (2015) studied the problem of determining probabilistic worst-case response time distributions for a set of periodic tasks with execution times described by random variables. This work differs from prior publications in that it describes the distributions via continuous functions and tightly approximates them with polynomial functions.

Ben-Amor et al. (2016) derived probabilistic schedulability analysis for tasks with precedence constraints and execution times described by random variables, scheduled under EDF.

Chen and Chen (2017) considered the complexity involved in repeated use of the convolution operator in probabilistic response time analysis. They proposed a more efficient way of computing the probability of deadline misses, based on the *moment generating function* of random variables, and Chernoff bounds for the probability that the sum of a number of random variables (e.g., the execution times of multiple jobs) exceeds some bound (e.g., the deadline). The evaluation shows that this method is effective in determining slightly pessimistic bounds on the probability of deadline misses without the need to derive the whole response time distribution, which can be very inefficient.

Criticality is a designation of the level of assurance needed against failure. A mixed criticality system is a system that contains tasks of two or more criticality levels. Draskovic et al. (2016) examined fixed priority preemptive scheduling of mixed criticality periodic tasks with execution times described by random variables. They employed the method of Diaz et al. (2002) to compute the probability of a deadline miss for every job in the hyperperiod.

Maxim et al. (2016, 2017) adapted probabilistic response time analysis (Maxim and Cucu-Grosjean 2013) to scheduling of mixed criticality systems using the Adaptive Mixed Criticality (AMC) and Static Mixed Criticality (SMC) schemes (Baruah et al. 2011). Abdeddaim and Maxim (2017) derived probabilistic response time analysis for mixed criticality tasks under fixed priority preemptive scheduling, allowing for multiple criticality levels.

6 Conclusions and Open Problems

This chapter presented the key concepts underpinning schedulability analysis for probabilistic real-time systems, including probabilistic worst-case execution time (pWCET) distributions and probabilistic worst-case response time (pWCRT) distributions. Deadline miss probabilities (DMP) for jobs and worst-case deadline failure probabilities (WCDFP) for tasks were also defined.

Section 2 presented probabilistic response time analysis for tasks with execution times modeled as independent random variables via a pWCET distribution, scheduled using fixed priority preemptive scheduling. This analysis computes the pWCRT distribution valid for any job of the task. Comparing this distribution with the task's deadline enables its WCDFP to be computed. Section 3 discussed priority assignment for probabilistic real-time systems, showing that policies which are optimal for conventional task models, such as rate-monotonic and deadline-monotonic, are no longer optimal in this case. However, Audsley's optimal priority assignment algorithm can be applied. Section 4 discussed the complexity of probabilistic response time analysis and ways in which it can be reduced in practice via resampling. Finally, Sect. 5 gave a brief overview of related research.

Recent results have begun to extend probabilistic schedulability analysis to mixed criticality task models. Other avenues for future research include extensions to multiprocessor scheduling.

Appendix: Task Set Generation

This appendix details a simple approach to generating task sets with probabilistic parameters that are suitable for empirical assessment of the performance of different scheduling algorithms and probabilistic schedulability analyses.

STEP 1 Generate the worst-case utilizations ($U_i = C_i^{max}/T_i$) for each of the n tasks using the UUnifast algorithm (Bini and Buttazzo 2005) to give an unbiased distribution of maximum utilization values.

STEP 2 Generate the task periods according to a log-uniform distribution (Emberson et al. 2010). For example, the range of task periods may span two orders of magnitude, e.g., from 10 to 1000 ms.

STEP 3 Obtain the worst-case execution time of each task from its utilization and period as follows: $C_i^{max} = U_i T_i$.

STEP 4 The best case execution time of each task may be obtained by using a fixed multiplier on the worst-case execution time, $C_i^{min} = SF \cdot C_i^{max}$, where SF is the scaling factor.

STEP 5 Task deadlines can be *implicit*, i.e., equal to the task period or *constrained*, i.e., no larger than the period. Constrained deadlines may be chosen from a uniform distribution in the range $[C_i^{max}, T_i]$.

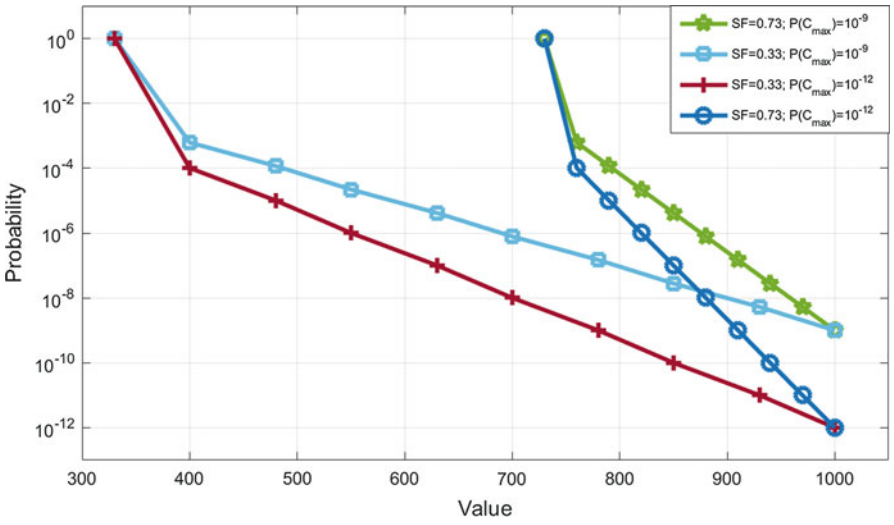

Fig. 2 Example of possible pWCET distributions

STEP 6 The size of the pWCET distribution is given as an input parameter to the probabilistic real-time task generator. If the size is 1, then the distribution has a single value, i.e., $C_i^{max} = C_i^{min}$, with probability equal to 1.

STEP 7 The probability associated with C_i^{max} can also be given as input to the task generator. It is expected that this value is small, for example, in the range $[10^{-6}, 10^{-12}]$, since it is expected that the probability of extreme execution times is very small (Cucu-Grosjean et al. 2012).

The pWCET distribution for each task can then be generated via extrapolation from the C_i^{min} and C_i^{max} parameter values, using the probability for the maximum value, and assuming that the distribution has an exponential tail. Thus the 1-CDF of the pWCET, plotted on an exceedance graph with probabilities given on a log scale, is as depicted in Fig. 2. Each line ends with the right most point at C_i^{max} and connects the intermediate points via a straight line (exponential tail). The left most point, at C_i^{min}, collects the remaining part of the distribution so that the probability mass sums to 1. (Note the longer lines are for a scaling factor of $SF = 0.33$ and thus show more execution time variation than the shorter lines which are for $SF = 0.73$.)

STEP 8 Task priorities may be set using the algorithm presented in Sect. 3.

References

Y. Abdeddaim, D. Maxim, Probabilistic schedulability analysis for fixed priority mixed criticality real-time systems, in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, 2017

L. Abeni, G. Buttazzo, Integrating multimedia applications in hard real-time systems, in *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, Dec 1998, pp. 4–13. https://doi.org/10.1109/REAL.1998.739726

L. Abeni, G. Buttazzo, Qos guarantee using probabilistic deadlines, in *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, 1999, pp. 242–249. https://doi.org/10.1109/EMRTS.1999.777471

L. Abeni, N. Manica, L. Palopoli, Efficient and robust probabilistic guarantees for real-time tasks. J. Syst. Softw. **85**(5), 1147–1156 (2012). ISSN:0164-1212. https://doi.org/10.1016/j.jss.2011.12.042

S. Altmeyer, R.I. Davis, On the correctness, optimality and precision of static probabilistic timing analysis, in *Proceedings of the Conference on Design, Automation and Test in Europe (DATE)*, 2014, pp. 26:1–26:6. ISBN:978-3-9815370-2-4. http://dl.acm.org/citation.cfm?id=2616606.2616638

S. Altmeyer, L. Cucu-Grosjean, R.I. Davis, Static probabilistic timing analysis for real-time systems using random replacement caches. Springer Real-Time Syst. **51**(1), 77–123 (2015). ISSN:1573-1383. https://doi.org/10.1007/s11241-014-9218-4

N. Audsley, On priority assignment in fixed priority scheduling. Info. Process. Lett. **79**(1), 39–44 (2001). ISSN:0020-0190. https://doi.org/10.1016/S0020-0190(00)00165-4. http://www.sciencedirect.com/science/article/pii/S0020019000001654

S.K. Baruah, A. Burns, R.I. Davis, Response-time analysis for mixed criticality systems, in *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)* (IEEE, 2011), pp. 34–43

S. Ben-Amor, D. Maxim, L. Cucu-Grosjean, Schedulability analysis of dependent probabilistic real-time tasks, in *Proceedings of the International Conference on Real-Time Networks and Systems (RTNS)* (ACM, 2016), pp. 99–107. ISBN:978-1-4503-4787-7. https://doi.org/10.1145/2997465.2997499

E. Bini, G. Buttazzo, Measuring the performance of schedulability tests. Real-Time Syst. **30**(1–2), 129–154 (2005)

F.J. Cazorla, E. Quiñones, T. Vardanega, L. Cucu, B. Triquet, G. Bernat, E. Berger, J. Abella, F. Wartel, M. Houston, L. Santinelli, L. Kosmidis, C. Lo, D. Maxim, Proartis: probabilistically analyzable real-time systems. ACM Trans. Embed. Comput. Syst. **12**(2s), 94:1–94:26 (2013). ISSN:1539-9087. https://doi.org/10.1145/2465787.2465796

K.H. Chen, J.J. Chen, Probabilistic schedulability tests for uniprocessor fixed-priority scheduling under soft errors, in *Proceedings of the IEEE International Symposium on Industrial Embedded Systems (SIES)*, June 2017, pp. 1–8. https://doi.org/10.1109/SIES.2017.7993392

L. Cucu, E. Tovar, A framework for the response time analysis of fixed-priority tasks with stochastic inter-arrival times. SIGBED Rev. **3**(1), 7–12 (2006). ISSN:1551-3688. https://doi.org/10.1145/1279711.1279714

L. Cucu-Grosjean, Independence a misunderstood property of and for probabilistic real-time systems, in *Real-Time Systems: The Past, the Present and the Future*, 2013, pp. 29–37

L. Cucu-Grosjean, L. Santinelli, M. Houston, C. Lo, T. Vardanega, L. Kosmidis, J. Abella, E. Mezzetti, E. Quinones, F.J. Cazorla, Measurement-based probabilistic timing analysis for multi-path programs, in *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, July 2012, pp. 91–101. https://doi.org/10.1109/ECRTS.2012.31

R.I. Davis, A review of fixed priority and EDF scheduling for hard real-time uniprocessor systems. ACM SIGBED Rev. **11**(1), 8–19 (2014)

R.I. Davis, A. Burns, Improved priority assignment for global fixed priority pre-emptive scheduling in multiprocessor real-time systems. Real-Time Syst. **47**(1), 1–40 (2011)

R.I. Davis, L. Santinelli, S. Altmeyer, C. Maiza, L. Cucu-Grosjean, Analysis of probabilistic cache related pre-emption delays, in *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, July 2013, pp. 168–179. https://doi.org/10.1109/ECRTS.2013.27

J.L. Diaz, D.F. Garcia, K. Kim, C.-G. Lee, L.L. Bello, J.M. Lopez, S.L. Min, O. Mirabella, Stochastic analysis of periodic real-time systems, in *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, 2002, pp. 289–300. https://doi.org/10.1109/REAL.2002.1181583

J.L. Diaz, J.M. Lopez, M. Garcia, A.M. Campos, K. Kim, L.L. Bello, Pessimism in the stochastic analysis of real-time systems: concept and applications, in *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, Dec 2004, pp. 197–207. https://doi.org/10.1109/REAL.2004.41

S. Draskovic, P. Huang, L. Thiele, On the safety of mixed-criticality scheduling, in *Proceedings of Workshop on Mixed Criticality (WMC)*, 2016

P. Emberson, R. Stafford, R.I. Davis, Techniques for the synthesis of multiprocessor tasksets, in *Proceedings 1st International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS 2010)*, 2010, pp. 6–11

B. Frias, L. Palopoli, L. Abeni, D. Fontanelli, Probabilistic real-time guarantees: there is life beyond the i.i.d. assumption, in *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, Apr 2017

M.K. Gardner, J.W.S. Liu, *Analyzing Stochastic Fixed-Priority Real-Time Systems* (Springer, Berlin/Heidelberg, 1999), pp. 44–58. ISBN:978-3-540-49059-3. https://doi.org/10.1007/3-540-49059-0_4

J.P. Hansen, J.P. Lehoczky, H. Zhu, R. Rajkumar, Quantized EDF scheduling in a stochastic environment, in *Proceedings of the 16th International Parallel and Distributed Processing Symposium*, IPDPS'02 (IEEE Computer Society, Washington, DC, 2002), p. 279. ISBN:0-7695-1573-8. http://dl.acm.org/citation.cfm?id=645610.660905

M. Ivers, R. Ernst, *Probabilistic Network Loads with Dependencies and the Effect on Queue Sojourn Times* (Springer, Berlin/Heidelberg, 2009), pp. 280–296. ISBN:978-3-642-10625-5. https://doi.org/10.1007/978-3-642-10625-5_18

G.A. Kaczynski, L.L. Bello, T. Nolte, Deriving exact stochastic response times of periodic tasks in hybrid priority-driven soft real-time systems, in *Proceedings of the IEEE Conference on Emerging Technologies Factory Automation (ETFA)*, Sept 2007, pp. 101–110. https://doi.org/10.1109/EFTA.2007.4416759

K. Kim, J.L. Diaz, L. Lo Bello, J.M. Lopez, C.-G. Lee, S.L. Min, An exact stochastic analysis of priority-driven periodic real-time systems and its approximations. IEEE Trans. Comput. **54**(11), 1460–1466 (2005). ISSN:0018-9340. https://doi.org/10.1109/TC.2005.174

J.P. Lehoczky, Real-time queueing theory, in *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, Dec 1996, pp. 186–195. https://doi.org/10.1109/REAL.1996.563715

J. Lehoczky, L. Sha, Y. Ding, The rate monotonic scheduling algorithm: exact characterization and average case behavior, in *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, Dec 1989, pp. 166–171. https://doi.org/10.1109/REAL.1989.63567

B. Lesage, D. Griffin, S. Altmeyer, R.I. Davis, Static probabilistic timing analysis for multi-path programs, in *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, Dec 2015, pp. 361–372. https://doi.org/10.1109/RTSS.2015.41

B. Lesage, D. Griffin, S. Altmeyer, L. Cucu-Grosjean, R.I. Davis, On the analysis of random replacement caches using static probabilistic timing methods for multi-path programs. *Real-Time Syst.* Apr 2018, **54**(2), 307–388. https://doi.org/10.1007/s11241-017-9295-2

J.Y.-T. Leung, J. Whitehead, On the complexity of fixed-priority scheduling of periodic, real-time tasks. Perform. Eval. **2**(4), 237–250 (1982). ISSN:0166-5316. https://doi.org/10.1016/0166-5316(82)90024-4. http://www.sciencedirect.com/science/article/pii/0166531682900244

G. Lima, I. Bate, Valid application of evt in timing analysis by randomising execution time measurements, in *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, Apr 2017

G. Lima, D. Dias, E. Barros, Extreme value theory for estimating task execution time bounds: a careful look, in *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, July 2016

C.L. Liu, J.W. Layland, Scheduling algorithms for multiprogramming in a hard-real-time environment. J. ACM **20**(1), 46–61 (1973). ISSN:0004-5411. https://doi.org/10.1145/321738.321743

J.M. López, J.L. Díaz, J. Entrialgo, D. García, Stochastic analysis of real-time systems under preemptive priority-driven scheduling. Springer Real-Time Syst. **40**(2), 180–207 (2008). ISSN:1573-1383. https://doi.org/10.1007/s11241-008-9053-6

Y. Lu, T. Nolte, J. Kraft, C. Norstrom, Statistical-based response-time analysis of systems with execution dependencies between tasks, in *Proceedings of the IEEE International Conference on Engineering of Complex Computer Systems (ICECCS)*, Mar 2010, pp. 169–179. https://doi.org/10.1109/ICECCS.2010.55

Y. Lu, T. Nolte, I. Bate, L. Cucu-Grosjean, A statistical response-time analysis of real-time embedded systems, in *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, Dec 2012, pp. 351–362. https://doi.org/10.1109/RTSS.2012.85

D. Maxim, L. Cucu-Grosjean, Response time analysis for fixed-priority tasks with multiple probabilistic parameters, in *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, 2013

D. Maxim, O. Buffet, L. Santinelli, L. Cucu-Grosjean, R. Davis, Optimal priority assignments for probabilistic real-time systems, in *Proceedings of the International Conference on Real-Time Networks and Systems (RTNS)*, 2011

D. Maxim, M. Houston, L. Santinelli, G. Bernat, R.I. Davis, L. Cucu-Grosjean, Re-sampling for statistical timing analysis of real-time systems, in *Proceedings of the International Conference on Real-Time Networks and Systems (RTNS)*, 2012

D. Maxim, F. Soboczenski, I. Bate, E. Tovar, Study of the reliability of statistical timing analysis for real-time systems, in *Proceedings of the International Conference on Real-Time Networks and Systems (RTNS)*, 2015, pp. 55–64. ISBN:978-1-4503-3591-1. https://doi.org/10.1145/2834848.2834878

D. Maxim, R.I. Davis, L. Cucu-Grosjean, A. Easwaran, Probabilistic analysis for mixed criticality scheduling with SMC and AMC, in *Proceedings of Workshop on Mixed Criticality (WMC)*, 2016

D. Maxim, R.I. Davis, L. Cucu-Grosjean, A. Easwaran, Probabilistic analysis for mixed criticality systems using fixed priority preemptive scheduling, in *Proceedings of the International Conference on Real-Time Networks and Systems (RTNS)* (ACM, 2017), pp. 237–246

L. Palopoli, D. Fontanelli, N. Manica, L. Abeni, An analytical bound for probabilistic deadlines, in *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, July 2012, pp. 179–188. https://doi.org/10.1109/ECRTS.2012.19

L. Santinelli, J. Morio, G. Dufour, D. Jacquemart, On the sustainability of the extreme value theory for WCET estimation, in *Proceedings of the Workshop on Worst-Case Execution Time Analysis (WCET)*, 2014, pp. 21–30. https://doi.org/10.4230/OASIcs.WCET.2014.21

L. Santinelli, F. Guet, J. Morio, Revising measurement-based probabilistic timing analysis, in *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, Apr 2017

B. Tanasa, U.D. Bordoloi, P. Eles, Z. Peng, Probabilistic response time and joint analysis of periodic tasks, in *Proceedings of the Euromicro Conference on Real-Time Systems (ECRTS)*, July 2015, pp. 235–246. https://doi.org/10.1109/ECRTS.2015.28

T.S. Tia, Z. Deng, M. Shankar, M. Storch, J. Sun, L.C. Wu, J.W.S. Liu, Probabilistic performance guarantee for real-time tasks with varying computation times, in *Proceedings of the IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, May 1995, pp. 164–173. https://doi.org/10.1109/RTTAS.1995.516213

F. Wartel, L. Kosmidis, C. Lo, B. Triquet, E. Quinones, J. Abella, A. Gogonel, A. Baldovin, E. Mezzetti, L. Cucu, T. Vardanega, F.J. Cazorla, Measurement-based probabilistic timing analysis: lessons from an integrated-modular avionics case study, in *Proceedings of the IEEE International Symposium on Industrial Embedded Systems (SIES)*, June 2013, pp. 241–248. https://doi.org/10.1109/SIES.2013.6601497

R. Wilhelm, J. Engblom, A. Ermedahl, N. Holsti, S. Thesing, D. Whalley, G. Bernat, C. Ferdinand, R. Heckmann, T. Mitra, F. Mueller, I. Puaut, P. Puschner, J. Staschulat, P. Stenström, The worst-case execution-time problem overview of methods and survey of tools. ACM Trans. Embed. Comput. Syst. **7**(3), 36:1–36:53 (2008). ISSN:1539-9087. https://doi.org/10.1145/1347375.1347389

M.H. Woodbury, K.G. Shin, Evaluation of the probability of dynamic failure and processor utilization for real-time systems, in *Proceedings of the IEEE Real-Time Systems Symposium (RTSS)*, Dec 1988, pp. 222–231. https://doi.org/10.1109/REAL.1988.51117