What Really is pWCET?
A Rigorous Axiomatic Proposal

Sergey Bozhko[1,2] Filip Marković[1] Georg von der Brüggen[3] Björn B. Brandenburg[1]
[1]Max Planck Institute for Software Systems, Germany
[2]Saarbrücken Graduate School of Computer Science, Saarland University, Germany
[3]Department of Computer Science, TU Dortmund University, Germany

***Abstract*—The concept of a *probabilistic worst-case execution time* (pWCET) has gradually emerged from the work of many authors over the course of 2–3 decades. Intuitively, pWCET is a simplifying model abstraction that safely over-approximates the ground-truth *probabilistic execution time* (pET) of a real-time task. In particular, when analyzing the cumulative processor demand of multiple jobs, the pWCET abstraction is intended to allow for the use of techniques from probability theory that require random variables to be *independent and identically distributed* (IID), even though the underlying ground-truth pET random variables are usually not independent. However, while powerful, the pWCET concept is subtle and difficult to define precisely, and easily misinterpreted. To place the pWCET concept on firm, unambiguous mathematical foundations, this paper proposes the first rigorous, axiomatic definition of pWCET that is suitable for formal proof. In addition, an adequacy property is stated that formally captures the intuitive notion of an "IID upper bound on pET." The proposed pWCET definition is shown to satisfy this adequacy condition, and thereby is the first notion of pWCET for which the IID guarantee is formally established. All definitions and proofs have been verified with the Coq proof assistant.**

I. Introduction

Whether by choice or necessity, interest in probabilistic real-time systems is on the rise. By choice, because there are good reasons to prefer a stochastic perspective (*e.g.*, cost considerations when dealing with soft or "firm" workloads that can tolerate the occasional deadline violation). Or by necessity, because the complexities of today's commodity hardware platforms (such as multi-level caches, speculative execution, or undisclosed component specifications) quite often prevent a meaningful *worst-case execution time* (WCET) analysis, leaving measurement-based approaches as the only available option. Either way, real-world systems—subject to market pressures, technological limitations, or both—commonly fail to meet the prerequisites for traditional worst-case guarantees.

Now, if absolute certainty is unattainable given the circumstances, then the next best guarantee is bounds on the probability of undesirable events (*e.g.*, missed deadlines). However, while the motivation and benefits are clear, the problem of actually obtaining such bounds is far from trivial and, as we review in Sec. II, has been the subject of intense study [18, 19].

In response to this challenge, the notion of *probabilistic worst-case execution time* (pWCET) has emerged over the past two decades as a central concept that is now routinely used in new work in this area. Intuitively, pWCET is a simplifying *model abstraction* that helps overcome two major obstacles commonly encountered in the probabilistic setting.

First, it is obviously extremely difficult to determine the true *probabilistic execution time* (pET) of a given task, especially if the task exhibits nontrivial control flow. It is thus natural to want to over-approximate the amount of required processor service with some margin. This idea of a somewhat pessimistic but safe "upper bound" on the ground-truth pET distribution in any possible scenario is at the heart of pWCET.

Second, since in practice tasks share a common execution environment and interact with each other, pETs are bound to be correlated across tasks and also across jobs (*i.e.*, successive activations) of the same task. In other words, when considering multiple jobs executing in temporal proximity, their pETs are decidedly *not* independent random variables, which unfortunately leads to major analytical complications. The pWCET abstraction promises a convenient way out [16]: by *substituting* all pET random variables with random variables following *suitably chosen* pWCET distributions, one obtains a problem composed only of *independent and identically distributed* (IID) random variables, which opens the door to a wealth of classic techniques from probability theory.

Given these advantages, it is no wonder that pWCET has become a dominant method in the probabilistic toolbox. Upon closer inspection, however, the situation is not as clearcut and settled as it may first appear. Even after decades of development of the idea (reviewed in Sec. II), the pWCET intuition remains subtle and difficult to capture in precise mathematical language. More often than not, key aspects are addressed only in prose. As a result, existing definitions are arguably difficult to interpret and unsuitable for formal proof. In fact, as we illustrate in Sec. III, even the state-of-the-art definition [19] can easily be misinterpreted to provide stronger guarantees than it does.

It is high time to place the pWCET concept on a firm mathematical foundation. To this end, we propose the first rigorous, axiomatic definition of pWCET that is amenable to formal proof. In particular, we have relied extensively on the *Coq* proof assistant [15] both in developing our definition and in validating its adequacy as a "safe upper bound."

Contributions. In this paper, we:

- observe that the currently accepted pWCET definition has come a long way (Sec. II), but that it is also still difficult to interpret and easily misunderstood (Sec. III);

- introduce the first formal semantics of probabilistic real-time systems, suitable for reuse in future research, to lay a precise foundation for formalization (Secs. IV and V);
- propose the first precise notions of pET and pWCET, following the established intuition (Sec. VI); and
- formally state an adequacy property capturing the notion of an "IID upper bound on pET" that, we argue, any reasonable pWCET definition should satisfy, and verify with Coq that our proposal is adequate in this sense (Sec. VII).

II. pWCET: A Short History

The pWCET concept has gradually emerged from the work of many authors over the course of 2–3 decades. Bernat et al. [6] were the first to coin the term "pWCET" in 2003, with key ideas going back to even earlier work. The most recent authoritative definition appears in Davis and Cucu-Grosjean's excellent surveys of probabilistic timing and schedulability analysis [18, 19]. Since these surveys already cover the area in detail, we focus here on the key ideas, twists, and turns that lead to the pWCET notion as it is intuitively understood today.

Historically, work on probabilistic real-time systems has focused on one of two questions that mirror the divide between WCET and schedulability analysis in the classical setting.

Q1 *How to characterize a task's resource needs stochastically, either by means of measurement or by static derivation?*

Q2 *How to exploit a given stochastic characterization of resource needs in schedulability analysis?*

The earliest work on Q2 (surveyed in [18]) precedes the work on Q1 (surveyed in [19]). We thus proceed in the same order.

A. Early Schedulability Perspective (Q2)

In 1993, Heidmann [26] was first to propose a schedulability analysis exploiting stochastic task costs. Focusing on preemptive rate-monotonic (*i.e.*, fixed-priority) scheduling, he made two simplifying assumptions: all tasks exhibit execution times characterized by *Normal* distributions with upper and lower bounds and the execution times of all tasks are *stochastically independent*. In modern terminology, Heidmann placed these assumptions on the pETs of all jobs of each task.

Soon after, Tia et al. [37] proposed an analysis that removes the (very) restrictive assumption of normally distributed execution times. Still targeting fixed-priority scheduling, Tia et al. allowed job execution times to be modeled by an arbitrary finite discrete probability distribution, while still retaining the (clearly unrealistic) independence assumption. From these assumptions, they derived a bound on the *deadline-miss probability* of each task. Crucially, Tia et al. evaluated their proposal in a real system and observed: "*Unfortunately, the computation times of individual requests are not statistically independent. [. . .] As a consequence, the probability of meeting deadlines thus computed may be overly optimistic.*" [37]. In other words, incorrectly assuming that pETs are independent is a soundness issue that demonstrably causes incorrect predictions.

Nevertheless, independence of random variables is a very desirable (or even necessary) property, so it was still assumed in later work. Díaz et al. [21] made an important clarification in this regard in 2002 when they explicitly stated that one must (also) assume independence with respect to "previous instances of the same task" [21]. That is, in order to apply independence-assuming techniques from probability theory to random variables representing pETs, one must also assume independence among pETs among jobs of the same task (and not just other tasks), which was not stated in earlier work.

A conceptually much larger step that brings us closer to the contemporary view was taken by Díaz et al. [22] in 2004, who proposed to over-approximate the distributions of random variables modeling task behavior to obtain a key monotonicity property: "*if pessimistic variables are introduced into the stochastic analysis, the response times provided by the analysis will also be pessimistic. [. . .] The pessimistic analysis is a safe approximation in the sense that the probabilities of deadline misses it provides are guaranteed to be greater than the exact ones*" [22]. This, of course, is what we are looking for in pWCET, and we return to this notion in Sec. VI-B.

To give a precise meaning to the concept of "pessimistic variables," Díaz et al. [22] introduced an order $\preceq$ on random variables that closely resembles first-order stochastic dominance and remains used to this day (we will recall it in Def. 2). Intuitively, $\mathcal{X}_1 \preceq \mathcal{X}_2$ means that, for any given fixed threshold x, the probability of $\mathcal{X}_1$ exceeding x is bounded by the probability of $\mathcal{X}_2$ exceeding x, *i.e.*, $\mathcal{X}_2$ dominates $\mathcal{X}_1$ point-wise.

Díaz et al. [22], however, still retained the IID assumption on job pETs (which they called "exact variables," *i.e.*, the ground-truth behavior of the tasks). In other words, while Díaz et al.'s "pessimistic variables" can be interpreted as a "proto-pWCET" from today's vantage point, their over-approximation guarantee applies only if the underlying pETs are independent.

However, to clarify the desired relationship between pWCET and non-IID pETs, progress on question Q1 was necessary first.

B. Derivation Perspective (Q1)

One of the pioneering papers on pWCET derivation is due to Burns and Edgar [10] in 2000, who recognized the problem of increasingly complex, superscalar architectures that hinder the use of static WCET analysis methods. To overcome this problem, Burns and Edgar proposed a measurement-based approach relying on *extreme value theory* (EVT) to estimate the maximum of the sampled execution-time distribution [10].

Burns and Edgar's work [10] paved the way for a rich literature on *measurement-based probabilistic timing analysis* (MBPTA) [1, 5, 6, 25, 28, 32, 33]; Davis and Cucu-Grosjean provide a comprehensive review [19]. Notably, it is in this line of work that Bernat et al. [6] coined the "pWCET" terminology. EVT remains central to current MBPTA techniques.

In parallel, the area of *static probabilistic timing analysis* (SPTA) emerged alongside MBPTA (*e.g.*, [2, 7, 17, 27]). In contrast to MBPTA, SPTA methods explicitly model sources of randomness in the hardware platform (*e.g.*, random-replacement caches), the software itself (*e.g.*, randomized algorithms), or the environment (*e.g.*, input distributions) and use this information to characterize (or upper-bound) the ground-truth execution-time distribution. Notably, David and Puaut coined the term

"pET" in their work [17] on the static determination of such ground-truth distributions.

However, even with the pWCET and pET concepts in place, it still took several more years before a link emerged.

C. Connection, Confusion, and Consensus

In a 2013 position paper, Cucu-Grosjean [16] was first to offer side-by-side definitions of the ground-truth execution-time distributions (pETs) [16, Def. 1], upper-bounding distributions (pWCETs) [16, Def. 2], and to establish a relationship among the two [16, Def. 3], enabling pWCET to be used in schedulability analyses to over-approximate pET. Going a significant step further than prior work, Cucu-Grosjean argued that no additional IID assumptions were required for probabilistic analysis to be sound, claiming that: "*probabilistic real-time analyses do not have stronger requirements from the task systems than a deterministic real-time analysis [. . .] as long as pWCETs are used*" [16]. In particular, Cucu-Grosjean observed that, when using pWCET for probabilistic analysis, "*the (probabilistic) independence of tasks is implicit and it does not require any new hypothesis*" [16], which obviously provides major benefits (*e.g.*, enabling convolution). Thus, for the first time, pETs could be allowed to be arbitrarily non-IID without preventing the use of IID-assuming analysis techniques, which unsurprisingly proved to be a highly influential observation.

Another important clarification was subsequently provided by Davis et al. [20], who noted that there are two fundamentally different interpretations of pWCET that should not be confused. In the first interpretation, pWCET is a statement about how *confident* one is that a given value bounds the maximum execution time. In this interpretation, no claim is made about the shape of the underlying pET distribution—it is only a statement about the maximum. Consequently, the pWCET distribution does not provide any information about, say, the mean execution time of jobs (*i.e.*, the pET distribution's expected value).

In the second interpretation, pWCET is understood as an *over-approximation* of the underlying pET distributions in the sense of Díaz et al.'s dominance relation $\preceq$ [22]. Here, the pWCET distribution can be seen as retaining some information about the shape of the pET distribution. In particular, the expected value of the pWCET distribution provides an upper bound on the ground-truth expected execution time.

Obviously, the two interpretations are not interchangeable, and neither can be used to infer the other. So which is pWCET?

A consensus view finally emerged in 2019, when Davis and Cucu-Grosjean offered the following definition in their surveys of probabilistic schedulability [18] and timing analysis [19].

Def. 1 (Def. 2 in both [18] and [19])**.** *The probabilistic Worst-Case Execution Time (pWCET) distribution for a program is the least upper bound, in the sense of [Díaz et al.'s dominance relation $\preceq$], on the execution time distribution of the program for every valid scenario of operation, where a scenario of operation is defined as an infinitely repeating sequence of input states and initial hardware states that characterize a feasible way in which recurrent execution of the program may occur.*

Def. 1 follows the dominance interpretation in the tradition of Díaz et al. [22] and Cucu-Grosjean [16], relating the pWCET distribution to "every valid scenario of operation" (*i.e.*, the ground-truth execution-time distribution) resulting from "an infinitely repeating sequence of input states and initial hardware states" (*i.e.*, any possible evolution of the system and its environment). Following Cucu-Grosjean [16], we continue to refer to this notion of "all possible ground-truth execution times" concisely as "pET" and offer a formal definition later in Sec. V.

On the all-important issue of independence, Davis and Cucu-Grosjean [18] echoed Cucu-Grosjean's observation [16], stating w.r.t. pWCET distributions derived via SPTA: "*We note that the actual execution times for a sequence of jobs of a task, which exercise the same or different paths, may well show strong correlations and dependences. It is the* modelling *of the execution times via an appropriate pWCET distribution which enables probabilistic independence to be assumed.*" [18, p. 4:10]. Similarly, when discussing pWCET obtained through MBPTA, they stated: "*Probabilistic independence of the pWCET distribution means that it can be used to characterise the behaviour of any randomly selected job of the task, and also composed using basic convolution to upper bound the interference from multiple jobs in probabilistic schedulability analysis.*" [18, p. 4:11].

In summary, the modern understanding of pWCET (Def. 1) relates an upper-bounding distribution to the ground truth such that pWCET-based schedulability analysis can assume IID execution times, provided the pWCET distributions are suitably derived to be "probabilistically independent" [18].

Nevertheless, while the state of the art has certainly come a long way, we believe that the pWCET concept is still not fully understood and in need of further clarification. First, it is not entirely obvious what mathematical properties a pWCET distribution must satisfy for Def. 1 to formally enable the independence claims quoted above. Second, the definition leaves key elements defined only in prose (*e.g.*, ground-truth behavior, probabilistic independence), requiring much interpretation by the reader. Third, it is consequently ill-suited for formal proof (*e.g.*, using Coq). Indeed, as we illustrate next with an example (Sec. III-A), it is all too easy to mistakenly attribute guarantees to Def. 1 that it does not actually provide.

In general, probabilistic analysis is inherently tricky and an area where intuition can easily deceive. Therefore, we believe that anchoring the pWCET idea in a precise formal foundation is the logical and necessary next step in its evolution.

III. Probabilistic Pitfalls

To motivate a more rigorous approach, we next highlight four subtleties affecting pWCET and probabilistic analysis.

A. Definition 1 Alone Does Not Enable IID Reasoning

The primary reason to adopt pWCET as a model abstraction is to allow for IID reasoning despite non-IID ground-truth execution times, as discussed in Sec. II-A. Clearly, there must be some mathematical relation between ground-truth execution times and pWCET for this to hold. What exactly is this relation?

It is tempting to believe that stochastic dominance in all scenarios of operation is sufficient. That is, when reading Def. 1 in isolation, an unsuspecting reader might come to expect that, if each task's pWCET dominates its ground-truth execution-time distribution, then a response-time distribution provided by an IID-assuming analysis operating on pWCETs will necessarily dominate the ground-truth response-time distribution.

However, this is not the case, which Davis and Cucu-Grosjean [18] also cautioned in their subsequent discussion of Def. 1. As it is a crucial point, we illustrate this potential pitfall with "Program B," one of their examples [19, pp. 3:09–3:10]. Each invocation of Program B executes one of four possible paths p_0, p_1, p_2, and p_3, which the program cycles through in order $(p_i \mapsto p_{(i+1)\%4})$. The ground-truth execution cost of path p_i is $(i+1)\cdot 10 \pm 2$, *i.e.*, the cost comprises a *fixed cost* $10\cdot(i+1)$ and some "random variability" ± 2 (with some unspecified distribution). The path taken by the first invocation of Program B is unknown but may be assumed to be uniformly distributed (*i.e.*, each p_i with probability 0.25 [19]).

The ground-truth execution-time distribution of Program B is $E^B \triangleq \left(\begin{smallmatrix} 10\pm2 & 20\pm2 & 30\pm2 & 40\pm2 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{smallmatrix}\right)$, where $\left(\begin{smallmatrix} c_1 & c_2 & \cdots & c_n \\ p_1 & p_2 & \cdots & p_n \end{smallmatrix}\right)$ denotes a multi-modal distribution with n modes such that mode c_i has occurrence probability p_i. (Again, the distribution of the variable component ± 2 is irrelevant and left unspecified.) Davis and Cucu-Grosjean explain that, "*[for Program B], the pWCET distribution valid for any scenario of operation*" is $F^B \triangleq \left(\begin{smallmatrix} 12 & 22 & 32 & 42 \\ 0.25 & 0.25 & 0.25 & 0.25 \end{smallmatrix}\right)$ [19, p. 3:10]. Distribution F^B clearly satisfies Def. 1: it upper-bounds the random variation ± 2 (irrespective of its distribution) in each scenario of operation and is indeed "the least upper bound" satisfying $E^B \preceq F^B$.

Now consider two periodic tasks $\{\tau_1, \tau_2\}$ under rate-monotonic scheduling, where τ_1 is Program B and τ_2 simply has a fixed cost of 110 (*i.e.*, single path, no variability). Additionally, assume τ_1 has a period and relative deadline equal to 50 and τ_2 has a relative deadline of 200 and a period of 1000. Both tasks release their first job at time zero. What is the probability of the first job of τ_2 missing its deadline?

Ground truth. Since τ_2's job requires 110 time units of processor service by time 200, it misses its deadline iff the first four jobs of τ_1 (*i.e.*, those released during $[0, 200)$) jointly require more than 90 time units of service. We do not know the initial path taken by Program B, but since exactly four jobs of τ_1 execute during $[0, 200)$, the program will execute each path exactly once. Hence, the total fixed cost is $10+20+30+40 = 100$ time units, so that even in the best case (each job exhibiting -2 variability) $100 - 4\cdot 2 = 92$ is the least-possible amount of service required to complete all four jobs of τ_1. Therefore, τ_2's job *certainly* misses its deadline.

pWCET convolution. As discussed, the reason to use pWCET in the first place is to allow for analysis methods rooted in IID assumptions, in particular convolution. Thus, let us approximate the joint demand of the first four jobs of τ_1 by basic convolution of F^B, which we denote as $\bigoplus_{i=1}^{4} F^B$. According to the resulting distribution, the first four jobs of τ_1 exhibit a total cost of less than 90 time units with non-zero probability (*e.g.*, $4\cdot 12 = 48$ with probability 0.25^4), as illustrated in Fig. 1. The convolved pWCET distribution hence incorrectly predicts that τ_2's first job meets its deadline with roughly 0.25 probability.

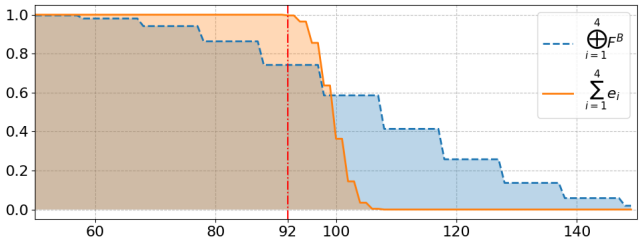

Fig. 1. Survival curves (*i.e.*, exceedance probability) of the total cost of four consecutive invocations of "Program B" $e_1, \ldots, e_4$ (*orange, solid*) and of a convolution of four instances of F^B (*blue, dashed*). A vertical line (*red*) marks the least-possible total cost $\sum_1^4 e_i$, which is 92.

The example shows that Def. 1 *by itself* does not actually guarantee the main feature generally associated with pWCET. Although Davis and Cucu-Grosjean discuss this very issue [18, pp. 4:11–4:12], it (arguably) may still come as a surprise, depending on how one interprets "every valid scenario of operation" in Def. 1. For example, all jobs exhibiting best-case execution times is a "feasible recurrent execution," and clearly not an "invalid" scenario, so one might reasonably wonder, should it not be covered by the criteria set forth in Def. 1?

Ultimately, the prose surrounding the state-of-the-art pWCET definition [18, 19] does much more of the "heavy lifting" than one may first realize. From a formal point of view, Def. 1 is too weak to derive IID guarantees, which limits its suitability for rigorous proof. In general, given two random variables $\mathcal{X}$ and $\mathcal{Y}$ with "upper bounds" $\hat{\mathcal{X}}$ and $\hat{\mathcal{Y}}$ such that $\mathcal{X} \preceq \hat{\mathcal{X}}$ and $\mathcal{Y} \preceq \hat{\mathcal{Y}}$, it is not necessarily the case that $\mathcal{X}+\mathcal{Y} \preceq \hat{\mathcal{X}} \oplus \hat{\mathcal{Y}}$ if $\mathcal{X}$ and $\mathcal{Y}$ are not independent. Thus, stochastic dominance alone cannot enable IID reasoning at the pWCET level when pETs are non-IID. But then what, *exactly*, are the mathematical requirements that a pWCET distribution must satisfy to cover non-IID pETs? We believe this question is central to defining "pWCET" and propose one possible answer in Sec. VI.

B. pWCET is not a Standalone Property

Another point that has received little coverage in the existing literature is that pWCET is not purely a property of the workload. That is, Def. 1 and prior concepts such as Díaz et al.'s "pessimistic variables" [22] are often discussed as if they provide guarantees by themselves. In fact, it is impossible to derive useful guarantees, such as Díaz et al.'s response-time monotonicity (recall Sec. II-A), from pWCET alone.

To demonstrate this aspect, we adapt the well-known fact that non-preemptive fixed-priority scheduling is subject to scheduling anomalies to the probabilistic setting. Consider the example workload in Fig. 2, which exhibits one of two possible schedules in the first 12 time units. If job $J_{1,1}$ executes for 3 time units, then job $J_{3,1}$ commences execution before job $J_{2,1}$ is released, which results in $J_{2,1}$ missing its deadline. Otherwise, if job $J_{1,1}$ executes for 4 time units, then $J_{2,1}$ is not delayed by $J_{3,1}$ and no deadline is missed. The ground-truth deadline-miss probability of $J_{2,1}$ is thus 0.5.

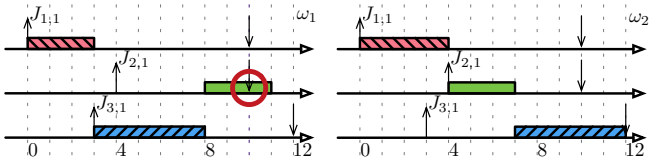

Fig. 2. The two possible scenarios of a periodic workload $\tau \triangleq \{\tau_1, \tau_2, \tau_3\}$ under non-preemptive fixed-priority scheduling, where task τ_1 (*red*) has period 16, deadline 10, offset 0, and a cost of either 3 or 4 with probability 0.5 each, task τ_2 (*green*) has period 16, deadline 6, offset 4, and a fixed cost of 3, and task τ_3 (*blue*) has period 16, deadline 9, offset 3, and a fixed cost of 5.

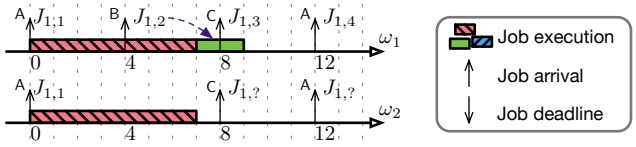

Fig. 3. If a job is omitted in scenario ω_2, which release corresponds to $J_{1,3}$?

Now suppose we (over-)approximate the ground truth with pWCETs as follows: $F_1 = \left(\begin{smallmatrix} 4 & 5 \\ 0.5 & 0.5 \end{smallmatrix}\right)$, $F_2 = \left(\begin{smallmatrix} 3 \\ 1 \end{smallmatrix}\right)$, and $F_3 = \left(\begin{smallmatrix} 5 \\ 1 \end{smallmatrix}\right)$. Clearly, this choice satisfies $\preceq$-dominance for each τ_i and is pessimistic for τ_1. However, if we "replace" the actual job-cost distributions with the stated pWCET distributions—that is, if we enumerate all possible schedules *assuming pWCET distributions in lieu of the ground truth* to approximate the response-time distribution—then we wrongly conclude that it is impossible for job $J_{3,1}$ to start execution before job $J_{2,1}$ (according to F_1, $J_{1,1}$ always takes at least 4 time units to finish, so $J_{2,1}$ has time to arrive). The stated pWCET would cause unsafe under-estimation of the true deadline-miss probability.

Clearly, pWCET is not the right abstraction in the presence of scheduling anomalies. In hindsight, it seems obvious that the pWCET concept is tied to *sustainable* scheduling policies [3, 12], but the existing literature does not dwell much on this constraint [18, 19]. A major advantage of a formal approach like the one we propose in Sec. VI is that such implicit assumptions become obvious and cannot be accidentally overlooked.

C. Incompatible Interpretations of Job Indices

Finally, even something as innocuous as how one counts jobs can produce misleading results. In the (classical) real-time systems literature, it is customary to enumerate a task's jobs in order of their release, so that $J_{i,j}$ (or alternatively also $\tau_{i,j}$) denotes the j-th activation of the i-th task. Unfortunately, this common notation can lead astray in a stochastic context.

For simplicity, we illustrate this point with a contrived setup; however, the issue is nontrivial and affects common models such as sporadic tasks. The core of the problem is that a probabilistic analysis inherently considers *multiple scenarios*, in which the order of jobs may differ and some jobs may not even arrive in some scenarios. Unfortunately, the customary indexed notation $J_{i,1}, J_{i,2}, \ldots$ may then result in cases of mistaken identity, and ultimately in incompatible results.

For example, consider the jobs of a task τ_1 and suppose exactly two scenarios are possible: an "exceptional" scenario ω_1 occurs with probability ε and the "common case" ω_2 occurs with probability $1 - \varepsilon$. Fig. 3 shows the jobs released by τ_1 in these two scenarios. In ω_1, the first job $J_{1,1}$ is released at time 0, executes procedure A, and has cost 7. A second job $J_{1,2}$ is released at time 4, which executes procedure B with cost 2. Suppose $J_{1,2}$ has a tight deadline at time 8: due to $J_{1,1}$'s long execution time, $J_{1,2}$ misses its deadline in ω_1. At times 8 and 12, two more jobs are released, which respectively execute procedures C and then again A, and so on.

Now consider ω_2, which differs from ω_1 only by releasing no job at time 4. This poses a seemingly trivial, but actually deep question: which job is released at time 8 in ω_2?

If we use the conventional consecutive numbering scheme, then the job released at time 8 in ω_2—which executes procedure C—is $J_{1,2}$. In this case, job $J_{1,1}$ finishes before the release of $J_{1,2}$ and $J_{1,2}$ can finish on time. We thus conclude that job $J_{1,2}$ has a *deadline-failure probability* (DFP) of only ε.

In contrast, "the B job" $J_{1,2}$ can be considered *absent* in ω_2—giving it a notion of *identity* other than its index. Now, job $J_{1,2}$'s DFP changes drastically: "the B job" misses a staggering 100% of its deadlines w.r.t. scenarios in which it arrives. For example, this interpretation could be highly relevant if B is an error handler that is executed only in abnormal conditions.

Clearly, these are two *very* different situations. A minuscule DFP is often acceptable in many applications, whereas a 100% DFP is a clear design flaw. Notice, however, that the system did not change at all: only our job-naming convention did.

What is the right interpretation? Ultimately, both conventions are sensible in their own right, depending on application needs and semantics. The concern is, however, that two individually sound analyses using different conventions may produce unsound results when they are unwittingly combined.

Our approach to bring more clarity to this issue is to omit indices altogether, which makes it obvious that our result does not depend on how jobs are named. To this end, we rely on *Leibniz equality* [*e.g.*, 34, Ch. 5]: two jobs J and J' are equal iff there is no predicate p such that $p(J) \wedge \neg p(J')$. As a result, two activations in different scenarios may be considered to be "the same job" even if they occur at different times or positions.

Our index-free notation generalizes all prior conventions: it can be augmented with arbitrary naming schemes to encode either way of indexing—or even completely different ways of relating jobs across scenarios (*e.g.*, by the input processed, code executed, state observed, *etc.*).

D. The Execution Cost of Rare Jobs

Similar issues affect pWCET: what is the pWCET of a sporadic task, say, an "emergency handler," that does not release a job in 99% of the "valid scenarios of operation"? What is its expected execution cost? Is its 99$^{\text{th}}$ percentile execution cost zero? If pWCET distributions are stated only with regard to scenarios in which a task releases jobs, can we still freely convolve this pWCET with that of another task that arrives in some, but not all of the same scenarios?

An informal approach is prone to glossing over such issues, which to our knowledge have not previously been raised in the literature on probabilistic real-time systems. To allow for precise answers, we next introduce a semantics, formalized with

the Coq proof assistant, in which the possibility of alternate traces and the absence of jobs are first-class concepts.

IV. Multi-Evolution Trace Semantics

To introduce a formal definition of pWCET that is mechanizable in Coq, we propose a formal semantics of real-time scheduling capable of expressing a wide variety of assumptions. At a high-level, we generalize Prosa's [4, 11] single-trace semantics to *multiple* scenarios, which we call *evolutions*. Whereas Prosa's single-trace semantics are best suited for classical worst-case analysis, the *multi-evolution semantics* proposed here enable reasoning about all "valid scenario[s] of operation" [18] in a formal manner.

It is important to realize that reasoning about a set of possible evolutions does *not* necessarily imply probabilistic reasoning: **(i)** recognizing that *multiple scenarios are possible* and **(ii)** assigning a *probability of occurrence* to each scenario are two distinct modeling steps. To ensure a clear separation of concerns, we focus exclusively on (i) in this section, and then, in Sec. V, present a stochastic interpretation that augments our semantics with a probabilistic measure. We stress that the probability measure is just *an* extension of the semantics; others are possible. Hence, the distinction between multi-evolution trace semantics and its stochastic extension makes both practical sense, since the former can be reused in other contexts (*e.g.*, sustainability theory [12]), and pedagogical sense, since the distinction allows us to isolate conceptually orthogonal parts.

Evolutions. To define the trace semantics, we let Ω denote the set of *all possible evolutions* of the system under analysis. This set Ω encapsulates everything relevant that may affect the system's dynamic behavior: the evolution of its environment, all observed input values, the timing of all inputs and stimuli, any source of entropy within the system itself (such as a hardware random number generator), *etc.* Environmental disturbances such as electro-magnetic interference (EMI), the occurrence of hardware faults, silicon defects, and so on can be seen as a kind of input for our purposes and are accounted for in Ω. Each $\omega \in \Omega$ is one possible evolution of the system behavior. In particular, fixing a specific $\omega \in \Omega$ completely determines the system's evolution (*i.e.*, the trace of its behavior), leaving no room for uncertainty. For example, any non-deterministic tie-breaking decisions are fixed in each $\omega \in \Omega$. Conversely, if two scenarios differ in some observable aspect (such as how a tie is resolved), then they are distinct elements of Ω.

The set Ω can be viewed as a formal notion of the phrase "all valid scenarios of operation." Importantly, Ω *is a purely theoretical concept*: we assume that such a set exists, in principle, but *not* that it can be derived, *e.g.*, by enumerating all program paths or some other form of *a priori* reasoning.

In the following, consider a system comprised of n *tasks* $\tau = \{\tau_1, \ldots, \tau_n\}$ and let $\mathbb{J}$ be the set of all jobs arriving in any evolution $\omega \in \Omega$. To allow evolutions to be related, we assume that a job can arrive in multiple evolutions (*i.e.*, jobs have *identity*). We let $J \in \mathbb{J}$ denote an arbitrary job and, in a slight abuse of notation, $J_i \in \tau_i$ denotes a job released by task τ_i, interpreting $\tau_i \subseteq \mathbb{J}$ as the set of all jobs produced by some task τ_i. Throughout this paper, we assume a discrete-time model, where the value $\varepsilon > 0$ represents the least indivisible unit of time and $\mathbb{T} \triangleq \{\varepsilon \cdot k \mid k \in \mathbb{N}\} \subset \mathbb{R}$ denotes the time domain. Finally, to differentiate between absolute time $\mathbb{T}$ and the amount of processor service needed to complete a job, $\mathbb{W} \triangleq \{\varepsilon \cdot k \mid k \in \mathbb{N}\}$ denotes the set of workload values.

Foundational properties. Each individual evolution $\omega \in \Omega$ fixes all information relevant to (one specific) dynamic behavior of the system under analysis. To allow statements about evolutions, we introduce functions that express basic properties of the system under analysis in a given evolution ω. For the purposes of this paper, it suffices to introduce only the most basic properties: arrival time and execution cost. In future work, more properties like other resource needs, preemption points, release jitter, self-suspensions, *etc.* can be added as needed.

The *arrival time* is a function $\mathcal{A}_\bullet : \mathbb{J} \times \Omega \to \mathbb{T} \cup \bot$ that maps a job J and an evolution ω to J's arrival time if it arrives in ω, or to $\bot$ if the job does not arrive. Given a job $J \in \mathbb{J}$, we write $\mathcal{A}_J$ to denote $\mathcal{A}_\bullet$ specialized to job J. Vice versa, we write $\mathcal{A}_\bullet(\omega)$ to denote $\mathcal{A}_\bullet$ specialized to a specific $\omega \in \Omega$.

Here we observe the first major departure from single-trace analyses, where one can simply assume that every job has a release time, since otherwise it can safely be ignored. In contrast, in the multi-evolution trace semantics, one must consider that a job may arrive in only a subset of evolutions. Therefore, we must explicitly distinguish between cases in which a job arrives and cases in which it does not (*i.e.*, $\bot$).

We analogously define a function $\mathcal{C}_\bullet : \mathbb{J} \times \Omega \to \mathbb{W} \cup \bot$ that, given a job J and an evolution ω, returns the *execution cost* of job J in evolution ω, or $\bot$ if J does not arrive in ω.

Another indispensable element of a real-time system is the scheduler, which we define as a function $\sigma : \mathbb{T} \times \Omega \to \mathbb{J} \cup \bot$ that maps a time instant t and an evolution ω to a job that is scheduled at time t in evolution ω, or $\bot$ if the processor is idle at time t. For now, we do not assume any specific algorithm behind σ. If the scheduling policy is non-deterministic, then this fact is reflected by the existence of (many) distinct $\omega \in \Omega$ representing different possible choices made by σ.

Derived definitions. From the foundational properties, we derive higher-level concepts. Later in this paper, the notion of a job's response time will be of primary importance, so we focus here on properties needed for defining "response time." In general, however, many other properties can be defined.

The *processor service* that a job J receives up to time t in an evolution ω is defined as $\mathfrak{s}_J(t, \omega) \triangleq \sum_{0 \leq i < t} \mathbb{1}[\sigma(i, \omega) = J]$, where $\mathbb{1}[x]$ denotes the *indicator function* that evaluates to 1 when x is true and 0 otherwise. Completion of a job can be derived from its cost and service: a job J is *complete* at a time instant t in an evolution ω, denoted $\mathfrak{c}_J(t, \omega)$, if it has received enough service, *i.e.*, $\mathfrak{c}_J(t, \omega) \triangleq \mathfrak{s}_J(t, \omega) \geq \mathcal{C}_J(\omega)$.

We let $\mathcal{R}_J(\omega)$ denote the exact *response time* of job J in evolution ω: $\mathcal{R}_J(\omega) \triangleq \inf \{r \mid r \in \mathbb{T} \colon \mathfrak{c}_J(\mathcal{A}_J(\omega) + r, \omega)\}$ if J arrives in ω and $\mathcal{R}_J(\omega) = \bot$ otherwise. If J arrives but never completes, then the set is empty and $\inf$ evaluates to $+\infty$.

Let D_i denote the *relative deadline* of task τ_i. A job $J_i \in \tau_i$ misses its deadline in evolution ω iff $\mathcal{R}_J(\omega) > D_i$. We assume

that $\bot \leq t$ for any $t \in \mathbb{T}$; hence, if a job does not arrive in ω, then by definition it does not miss its deadline.

For brevity, we introduce the notion of an *arrival sequence* $\xi \colon \mathbb{T} \times \Omega \to 2^{\mathbb{J}}$, where $\xi(t, \omega) \triangleq \{J \mid \mathcal{A}_J(\omega) = t\}$. That is, the arrival sequence ξ yields the set of jobs that arrive at a given time t in evolution ω. For convenience, we also define the arrival sequence of a task τ_i as $\xi_i(t, \omega) \triangleq \xi(t, \omega) \cap \tau_i$.

Classical assumptions. Our proposed semantics can be used in a wide variety of contexts. For example, the classical *hard real-time* constraint can be expressed by stating that *not a single* $\omega \in \Omega$ can give rise to a situation where there exists a job $J \in \tau_i$ such that $\mathcal{R}_J(\omega) > D_i$; or stated positively, the goal invariant is $\forall \tau_i \in \tau, \forall J \in \tau_i, \forall \omega \in \Omega \colon \mathcal{R}_J(\omega) \leq D_i$. Other classical concepts can be expressed similarly concisely.

Example 1 (🐓[1]). *The* worst-case execution time *(WCET) of a task τ_i is a constant C_i such that* ***(i)*** $\forall \omega \in \Omega, \forall J \in \tau_i \colon \mathcal{C}_J(\omega) \leq C_i$ *and* ***(ii)*** $\exists\, \omega \in \Omega, \exists J \in \tau_i \colon \mathcal{C}_J(\omega) = C_i$.

Note how similar Example 1 is to the usual way of defining WCET: we simply account for all evolutions $\omega \in \Omega$ instead of assuming a scalar parameter such as the usual per-job execution cost "$c_{i,j}$" (which corresponds to our $\mathcal{C}_J(\omega)$). By dropping clause (ii), we naturally obtain the notion of a WCET *bound*.

Example 2 (🐓). *A constant T_i is a valid bound on the* minimum inter-arrival time *of a task τ_i iff*

$$\begin{aligned}&\forall \omega \in \Omega \colon \forall t_1, t_2 \in \mathbb{T} \colon \forall J, J' \in \tau_i \colon \\ &\quad t_1 \leq t_2 \wedge J \in \xi_i(t_1, \omega) \wedge J' \in \xi_i(t_2, \omega) \wedge J \neq J' \to \\ &\quad t_2 - t_1 \geq T_i.\end{aligned}$$

That is, any two distinct jobs J and J' of τ_i arrive at least T_i time units apart in every possible evolution $\omega \in \Omega$.

In general, single-trace semantics, *e.g.*, as used in Prosa [4, 11], can be embedded in our multi-evolution semantics in a *lossless* manner simply by lifting all properties to $\forall \omega \in \Omega$. The added power of our semantics, however, is that it allows reasoning about many evolutions *simultaneously*. For example, future work on sustainability theory might be interested in analyzing subsets of Ω that do not contain certain scenarios. In this paper, we augment Ω with a probability measure, so we can express that some evolutions are more likely than others.

V. Stochastic Interpretation

To formally define pWCET, we first need to characterize the ground truth. For this purpose, we extend the multi-evolution trace semantics with a probability measure. In preparation, we briefly recall the necessary concepts from probability theory.

A. Probability Primer and Notation

A discrete *probability space* is a pair $(\Omega, \mathbb{P})$, where Ω is a countable, non-empty set of all possible *outcomes*, $\mathbb{P} : 2^{\Omega} \to [0, 1]$ is a *probability function*, and 2^{Ω} is the set of all subsets of Ω. In the discrete case, it is always possible to derive a distribution function $\mu \colon \Omega \to [0, 1]$ such that $\forall \omega, \mu(\omega) \geq 0$ and $\sum_{\omega \in \Omega} \mu(\omega) = 1$, and that, for any *event* $A \subseteq \Omega$, it holds that $\mathbb{P}[A] = \sum_{a \in A} \mu(a)$. We use μ and $\mathbb{P}$ interchangeably.

A *random variable* is a function $\mathcal{X} \colon \Omega \to E$, where E is any set (*i.e.*, we impose no restrictions since Ω is countable). For brevity, the argument of a random variable is often omitted when used in the context of a probability function. For instance, $\mathbb{P}[\mathcal{X} \leq 7]$ is equivalent to $\mathbb{P}[\{\omega \in \Omega \mid \mathcal{X}(\omega) \leq 7\}]$.

The *conditional probability* of an event A, given an event with positive probability B, is $\mathbb{P}[A|B] \triangleq \mathbb{P}[A \cap B] / \mathbb{P}[B]$. We will frequently use the (discrete) *law of total probability* (LTP):

Fact 1 (🐓). *Given a probability space $(\Omega, \mathbb{P})$, an event $A \subseteq \Omega$, and a finite or countably infinite partition $\{B_i\}_i$ of the sample space, that is, for $i \neq j$,* ***(i)*** $B_i \cap B_j = \emptyset$ *and* ***(ii)*** $\forall \omega \in \Omega \colon \mu(\omega) > 0 \implies \exists i \colon \omega \in B_i$*, we have* $\mathbb{P}[A] = \sum_i \mathbb{P}[A \cap B_i]$.

Given a random variable $\mathcal{X} \colon \Omega \to E$ with $E \subseteq \mathbb{R}$, its *cumulative distribution function* (CDF) $\mathbb{F}[\mathcal{X}] : \mathbb{R} \to [0, 1]$ is defined as $\mathbb{F}[\mathcal{X}](x) \triangleq \mathbb{P}[\mathcal{X} \leq x]$. Akin to conditional probability, the *conditional CDF* of $\mathcal{X}$ given an event with positive probability B is defined as $\mathbb{F}[\mathcal{X}|B](x) \triangleq \mathbb{P}[\mathcal{X} \leq x|B]$.

We adopt and slightly generalize the partial order on random variables introduced by Díaz et al. [22]. Let $f, g \colon \mathbb{N} \to \mathbb{R}$ be two arbitrary functions. We say that f is *dominated* by g iff $\forall x \in \mathbb{N}, f(x) \geq g(x)$ and denote this relation as $f \preceq g$ (🐓). Graphically speaking, $f \preceq g$ iff f always stays above g.

Def. 2 (Díaz et al. [22], 🐓). *Let $\mathcal{X}_1 \colon \Omega_1 \to E$ and $\mathcal{X}_2 \colon \Omega_2 \to E$ be two random variables with two not necessarily identical domains and identical codomain $E \subseteq \mathbb{R}$. We say that $\mathcal{X}_1$ is dominated by $\mathcal{X}_2$, denoted as $\mathcal{X}_1 \preceq \mathcal{X}_2$, iff $\mathbb{F}[\mathcal{X}_1] \preceq \mathbb{F}[\mathcal{X}_2]$.*

We next recall the crucial notion of conditional independence.

Def. 3. *Given two events $A, B \subseteq \Omega$ and an event with positive probability $C \subseteq \Omega$, we say that events A and B are* conditionally independent given C *iff* $\mathbb{P}[A \cap B|C] = \mathbb{P}[A|C] \cdot \mathbb{P}[B|C]$.

A set of random variables $\mathcal{X}_1, \ldots, \mathcal{X}_n \colon \Omega \to E$ is *mutually independent* iff $\mathbb{P}\left[\bigcap_{i \in G} \mathcal{X}_i = x_i\right] = \prod_{i \in G} \mathbb{P}[\mathcal{X}_i = x_i]$ for any constants $x_1, \ldots, x_n \in E$ and any subset $G \subseteq \{1, \ldots, n\}$.

Finally, we say that a set of random variables is *IID* if they are mutually independent and all share the same CDF.

B. Probabilistic Real-Time Semantics

Let us now introduce the probabilistic structure. Crucially, we make two common-sense assumptions that greatly simplify the subsequent formalization, not only, but especially, in Coq.

First, we assume there exists a finite *horizon* $H \in \mathbb{T}$ such that the system ceases operation after at most H time units. In practice, H is not a restriction since it can be arbitrarily large. Formally, it means we may assume the total number of jobs to be finite (*i.e.*, $|\mathbb{J}| \leq N$ for some very large $N \in \mathbb{N}$), assuming no task releases an unbounded number of jobs instantaneously.

Second, we assume the system under analysis to have finite memory, *i.e.*, it is a finite-state system, which obviously holds for practical digital computers. As a useful consequence, we

[1] We use the symbol 🐓 to indicate mechanization in Coq. The symbol is clickable (in the PDF) and leads to the corresponding definition or lemma. The full proof, building on Tassarotti's probability theory library [35, 36] and in small parts on Prosa [11, 31], is available online [9].

may conclude that Ω is countable because there are only finitely many evolutions of a finite-state system of length at most H.

We thus interpret the set of all possible evolutions Ω as a countable outcome space. Next, we assume the *existence* (but not necessarily knowledge) of a distribution function $\mu\colon \Omega \to [0,1]$, which as discussed implies a measure $\mathbb{P}\colon 2^{\Omega} \to [0,1]$, thereby yielding the probability space $(\Omega, \mathbb{P})$.

Given $(\Omega, \mathbb{P})$, we can now define the *probabilistic execution time (pET)*: it is simply the cost function $\mathcal{C}_\bullet\colon \mathbb{J} \times \Omega \to \mathbb{W} \cup \bot$ introduced in Sec. IV. Recall that a random variable is a function $\Omega \to E$, for some set E. Seen this way, $\mathcal{C}_\bullet$ maps a job J to a random variable describing its cost $\mathcal{C}_J\colon \Omega \to \mathbb{W} \cup \bot$. Thus, we can use $\mathcal{C}_J$ to reason about the probability of events related to the job's execution time. For example, given a job J, we can state "J's cost is bounded by 12 time units with probability 0.99" simply as $\mathbb{P}\left[\mathcal{C}_J \leq 12\right] = 0.99$.

Analogously to pET, the probability space $(\Omega, \mathbb{P})$ imposes a probabilistic structure on $\mathcal{A}$, ξ, σ, $\mathfrak{c}$, *etc.*, which all become random variables under the stochastic interpretation. As a more involved example, we can express the generally true proposition that the probability of completion increases with time (🐓) as $\forall J \in \mathbb{J}, \forall t_1, t_2 \in \mathbb{T}\colon t_1 \leq t_2 \implies \mathbb{P}[\mathfrak{c}_J(t_1)] \leq \mathbb{P}[\mathfrak{c}_J(t_2)]$. Note that the second argument of $\mathfrak{c}_J(t,\omega)$ is omitted following the usual convention for random variables.

For notational convenience, we overload the arrival sequence notation ξ in the context of events and conditional probability. Consider the set of all arrival sequences $\Xi \triangleq \{\xi(\cdot,\omega) \,|\, \omega \in \Omega\}$. Given a fixed arrival sequence $\xi_o \in \Xi$, we define a corresponding event comprising all evolutions with this fixed arrival sequence ξ_o, formally $\{\omega \in \Omega \,|\, \xi(\cdot,\omega) = \xi_o\}$. In the following, we simply use ξ to refer to this induced event.

The presented formalization is expressive enough to account for the interplay between the "average" and "worst-case" behaviors. One well-known example where this matters is the *deadline failure probability* (DFP). Let J be some job of a task τ_i. If we simply state $\mathbb{P}\left[\mathcal{R}_J > D_i\right]$, it means the DFP of job J assuming we *do not* consider the arrival sequence to be a critical instant, *i.e.*, it is a statement w.r.t. all possible arrival sequences (similarly to [29]). To state the *worst-case DFP* (WCDFP), we actually need to use conditional probability: $\max_{\xi\colon \mathbb{P}[\xi>0]} \max_{J \in \tau_i} \mathbb{P}\left[\mathcal{R}_J > D_i \,|\, \xi\right]$ (*e.g.*, similarly to [30]).

More generally, a significant number of prior analyses consider the arrival sequence either to be fixed [8] or to be formed in a worst-case fashion [13, 14, 38]. With the proposed stochastic multi-evolution semantics, all such assumptions and metrics can be stated clearly and related. For the purposes of this paper, the semantics allow us to precisely define pWCET.

VI. Axiomatic pWCET and its Adequacy

In the following, we develop the main technical contribution of this paper: a formal definition of pWCET called *axiomatic pWCET* (Sec. VI-A). To ensure that axiomatic pWCET captures the intuition laid out in Sec. II, we introduce a formal adequacy requirement (in Secs. VI-B and VI-C), and then prove that our proposal is adequate in this sense (in Sec. VII).

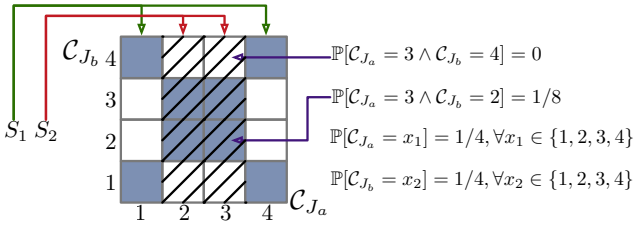

Fig. 4. An example system $(\Omega, \mathbb{P})$. An evolution in this system is a pair $(x_1, x_2) \in \{1,2,3,4\} \times \{1,2,3,4\}$, illustrated as a grid of outcomes. Each blue outcome (*i.e.*, cell in the grid) has a probability of occurrence of $1/8$, and each white cell has probability 0. The two subsets S_1 (*clear*) and S_2 (*dashed*) form a partition $\mathfrak{S}$ of Ω for job J_a (as explained after Def. 7).

A. Axiomatic pWCET via Partitioning

Before we gradually define axiomatic pWCET, let us consider the illustrative example in Fig. 4. The set of evolutions Ω consists of 16 scenarios $\{(x_1, x_2) \,|\, x_1, x_2 \in \{1,2,3,4\}\}$ such that each evolution is a pair $\omega = (x_1, x_2)$. The probability distribution is given by $\mu((x_1,x_2)) = 1/8$ if $(x_1 + x_2 = 5) \vee (x_1 = x_2)$ and 0 otherwise (see Fig. 4 for a graphical representation). For simplicity, there are only two jobs J_a and J_b that both arrive at time 0, with costs $\mathcal{C}_{J_a}((x_1,x_2)) = x_1$ and $\mathcal{C}_{J_b}((x_1,x_2)) = x_2$. Given Ω and μ, pETs $\mathcal{C}_{J_a}$ and $\mathcal{C}_{J_b}$ are clearly dependent random variables. In order to introduce a valid "pWCET," one thus needs a condition stronger than $\preceq$.

This stronger condition is at the heart of our axiomatic pWCET proposal. To define it, we first require three new concepts: a "partition" of all possible evolutions, "partition independence," and "partition dominance." We formally state each of these ideas next and then relate them to Fig. 4.

Recall from Sec. II how Def. 1 requires pWCET to dominate pET for every "valid scenario of operation." In a similar spirit, we introduce the notion of a partition of all possible evolutions.

Def. 4 (🐓). *A partition $\mathfrak{S} \triangleq \{S_l\}_l$ is any finite, or countably infinite, disjoint cover of all positive-probability elements of Ω.*

It follows from Def. 4 that $\mathbb{P}[\bigcup_l S_l] = 1$. Intuitively, a partition $\mathfrak{S}$ can be understood as splitting Ω into subsets of related evolutions representing different "scenarios." Formally, we place no restrictions on how such a "scenario" $S_l \in \mathfrak{S}$ is found or defined. Intuitively, one can think of application-specific criteria such as, for instance, the path taken by a job, the hardware state, or the value of some input to the system.

For axiomatic pWCET, we require not just any partition, but one with a particular feature: it must ensure that the pET of a given job is *conditionally* independent of all other pETs for each $S_l \in \mathfrak{S}$. We call this property "partition independence."

In the below definition, we use a new notation $\vec{c}_\bullet\colon \mathbb{J} \to \mathbb{W} \cup \bot$ to denote a vector of *fixed* job costs. In contrast to the pET $\mathcal{C}_\bullet$, the cost vector $\vec{c}_\bullet$ does not depend on Ω (and hence is *not* a random variable); rather, it is simply a vector of *fixed* values that we use to concisely express one possible "assignment" of job costs. Similarly to pET, we write $\vec{c}_J$ to denote the element of $\vec{c}_\bullet$ corresponding to a given job J.

Def. 5 (🐓). *Given a job $J \in \mathbb{J}$, a fixed arrival sequence ξ, and a partition $\mathfrak{S}$, job J's pET is* partition-independent w.r.t. $\mathfrak{S}$ *iff,*

for any set $G \subseteq \mathbb{J}$ with $J \notin G$ and any fixed cost vector $\vec{c}_\bullet$:

$$\begin{aligned}&\forall S_l \in \mathfrak{S} \text{ s.th. } \mathbb{P}[S_l \wedge \xi] > 0\colon \\ &\quad \mathbb{P}\left[\mathcal{C}_J = \vec{c}_J \wedge \forall J' \in G\colon \mathcal{C}_{J'} = \vec{c}_{J'} | S_l \wedge \xi\right] \\ &\quad = \mathbb{P}\left[\mathcal{C}_J = \vec{c}_J | S_l \wedge \xi\right] \cdot \mathbb{P}\left[\forall J' \in G\colon \mathcal{C}_{J'} = \vec{c}_{J'} | S_l \wedge \xi\right].\end{aligned}$$

In other words, the probability of job J exhibiting a particular fixed cost $\vec{c}_J$ must be *conditionally* independent (given ξ and any $S_l \in \mathfrak{S}$) of any other job J' exhibiting a particular fixed cost $\vec{c}_{J'}$. This matches the definition of conditional independence (Def. 3) with event $\mathcal{C}_J = \vec{c}_J$ being A, event $\forall J' \in G\colon \mathcal{C}_{J'} = \vec{c}_{J'}$ being B, and event $S_l \wedge \xi$ being C. Note that *some* partition ensuring this property always exists, as $\mathfrak{S} = \{\{\omega\} \mid \omega \in \Omega\}$ satisfies Def. 5 (🐓).

Finally, we incorporate Díaz et al.'s dominance relation $\preceq$ on a per-partition basis, which we call *partition dominance*.

Def. 6 (🐓). *Given a job $J \in \mathbb{J}$, a fixed arrival sequence ξ, and a partition $\mathfrak{S}$, a function $F\colon \mathbb{W} \to [0,1]$ $\mathfrak{S}$-dominates $\mathcal{C}_J$ iff*

$$\forall S_l \in \mathfrak{S} \text{ s.th. } \mathbb{P}[S_l \wedge \xi] > 0\colon \quad \mathbb{F}[\mathcal{C}_J | S_l \wedge \xi] \preceq F.$$

Based on Defs. 4–6, we can finally define axiomatic pWCET.

Def. 7 (🐓). *A monotonically increasing function $F_i\colon \mathbb{W} \to [0,1]$ with $F_i(0) = 0$ and $\lim_{t\to\infty} F_i(t) = 1$ is an* axiomatic pWCET *for a task τ_i if, for every $J \in \tau_i$ and every fixed arrival sequence $\xi \in \Xi$, there exists a partition $\mathfrak{S}$ (Def. 4) such that*

1) *$\mathcal{C}_J$ is partition-independent w.r.t. ξ and $\mathfrak{S}$ (Def. 5), and*
2) *F_i $\mathfrak{S}$-dominates $\mathcal{C}_J$ w.r.t. ξ (Def. 6).*

Note that Def. 7 does *not* require there to be only one "global" partition $\mathfrak{S}$—rather, it merely requires a $\mathfrak{S}$ satisfying Defs. 5 and 6 to exist for each job, but no two jobs are required to have the same $\mathfrak{S}$. That is, in the terminology of Def. 1, what is considered a "scenario of operation" can differ across tasks and jobs. In this regard, Def. 7 is considerably weaker than Def. 1. The rationale is that Def. 7's weaker requirement suffices to establish adequacy (Sec. VII). Also unlike Def. 1, axiomatic pWCET does not impose a notion of "*least* upper bound"—Def. 7 ensures soundness, not tightness. In one key point, however, Def. 7 is stronger than Def. 1: partition independence and dominance jointly enable IID reasoning (Sec. VII).

Example. Let us illustrate Defs. 4–7 with the example from Fig. 4. The arrival sequence in the example is fixed and hence not relevant. Suppose J_a is released by a task τ_a. Let us find an axiomatic pWCET for τ_a, which requires the following steps. **(1)** We construct a partition $\mathfrak{S} = \{S_1, S_2\}$ for J_a according to Def. 4, where $S_1 \triangleq \{(x_1, x_2) \mid x_1 \in \{1,4\}\}$ and $S_2 \triangleq \{(x_1, x_2) \mid x_1 \in \{2,3\}\}$, and $x_2 \in \{1,2,3,4\}$ in both cases. **(2)** Geometrically, it is easy to see that $\mathcal{C}_{J_a}$ and $\mathcal{C}_{J_b}$ are partition-independent w.r.t. $\mathfrak{S}$ (Def. 5): $\mathbb{P}[\mathcal{C}_{J_a} = \vec{c}_{J_a} \wedge \mathcal{C}_{J_b} = \vec{c}_{J_b} | S_i] = \mathbb{P}[\mathcal{C}_{J_a} = \vec{c}_{J_a} | S_i] \cdot \mathbb{P}[\mathcal{C}_{J_b} = \vec{c}_{J_b} | S_i]$ for any $\vec{c}_{J_a}, \vec{c}_{J_b} \in \{1,2,3,4\}$ and $S_l \in \mathfrak{S}_0$. For example, conditioned on S_1, the pET $\mathcal{C}_{J_a}$ equals either 1 or 4 with probability 0.5 each, regardless of the value assumed by $\mathcal{C}_{J_b}$. Similarly, it's either 2 or 3 when conditioned on S_2. **(3)** We choose $F_a \triangleq \left(\begin{smallmatrix}2 & 4\\ 0.5 & 0.5\end{smallmatrix}\right)$ and check that it $\mathfrak{S}$-dominates $\mathcal{C}_{J_a}$ (Def. 6): for $\mathcal{C}_{J_a}|S_1$, we have $\left(\begin{smallmatrix}1 & 4\\ 0.5 & 0.5\end{smallmatrix}\right) \preceq \left(\begin{smallmatrix}2 & 4\\ 0.5 & 0.5\end{smallmatrix}\right)$, and for $\mathcal{C}_{J_a}|S_2$, we have $\left(\begin{smallmatrix}2 & 3\\ 0.5 & 0.5\end{smallmatrix}\right) \preceq \left(\begin{smallmatrix}2 & 4\\ 0.5 & 0.5\end{smallmatrix}\right)$. Thus, $\left(\begin{smallmatrix}2 & 4\\ 0.5 & 0.5\end{smallmatrix}\right)$ is a safe axiomatic pWCET distribution for task τ_a (Def. 7).

Let us also revisit "Program B" from Sec. III-A. Let P_i^1 denote the event that the first job executes path p_i. Since the pET of a job depends on the path taken by the previous job, all the way back to P_i^1, we conclude that $\mathfrak{S}_B = \{P_1^1, P_2^1, P_3^1, P_4^1\}$ ensures partition-independence (assuming that the noise term ± 2 does not exhibit correlation itself). The tightest possible pWCET distribution satisfying Def. 6 w.r.t. $\mathfrak{S}_B$ is $\left(\begin{smallmatrix}40\pm 2\\ 1\end{smallmatrix}\right)$, which indeed results in a safe over-approximation. This revised bound also suggests that mostly deterministic execution does not leave much room for pWCET to provide a large benefit, which, however, is a discussion that we leave to future work.

B. pWCET Adequacy: A Minimal Requirement

Having proposed axiomatic pWCET, we now wish to assess whether it is *adequate*, in the sense of allowing the kind of IID-based reasoning sketched in Secs. II and III-A. To this end, we define a class of "system transformations" that are monotone w.r.t. the *probabilistic response time* (pRT) $\mathcal{R}_\bullet(\omega)$ and argue that "replacing pETs with pWCETs" ought to be monotone in this sense for any reasonable notion of "pWCET."

So far in this paper, we have relied heavily on the reader's intuition when saying "we *replace* pETs with pWCETs." Let us now clarify the meaning behind "replacement." Recall the definition of a probabilistic real-time system $(\Omega, \mathbb{P})$ with its foundational properties $\mathcal{A}_\bullet$ and $\mathcal{C}_\bullet$. We call $(\Omega, \mathbb{P}, \mathcal{A}_\bullet, \mathcal{C}_\bullet)$ a *system* and allow its transformation as follows.

Def. 8. *A* system transformation *is a function $\mathfrak{T}$ that maps a given probabilistic real-time system $(\Omega, \mathbb{P}, \mathcal{A}_\bullet, \mathcal{C}_\bullet)$ to a new system $(\Omega^\star, \mathbb{P}^\star, \mathcal{A}_\bullet^\star, \mathcal{C}_\bullet^\star) \triangleq \mathfrak{T}(\Omega, \mathbb{P}, \mathcal{A}_\bullet, \mathcal{C}_\bullet)$.*

With this terminology, the "replacement" of one element with another is simply a mapping of $(\Omega, \mathbb{P}, \mathcal{A}_\bullet, \mathcal{C}_\bullet)$ to a new system $(\Omega^\star, \mathbb{P}^\star, \mathcal{A}_\bullet^\star, \mathcal{C}_\bullet^\star)$. In the context of pWCET, we are interested in a class of system transformations that are monotone w.r.t. the induced change in the pRT of any job.

Def. 9 (🐓). *Given a system $(\Omega, \mathbb{P}, \mathcal{A}_\bullet, \mathcal{C}_\bullet)$, a system transformation $\mathfrak{T}$ is* pRT-monotone *iff the resulting system $(\Omega^\star, \mathbb{P}^\star, \mathcal{A}_\bullet^\star, \mathcal{C}_\bullet^\star) \triangleq \mathfrak{T}(\Omega, \mathbb{P}, \mathcal{A}_\bullet, \mathcal{C}_\bullet)$ is guaranteed to satisfy*

$$\forall J \in \mathbb{J}\colon \mathcal{R}_J \preceq \mathcal{R}_J^\star,$$

where $\mathcal{R}_J$ and $\mathcal{R}_J^\star$ are the pRT of J in systems $(\Omega, \mathbb{P}, \mathcal{A}_\bullet, \mathcal{C}_\bullet)$ and $(\Omega^\star, \mathbb{P}^\star, \mathcal{A}_\bullet^\star, \mathcal{C}_\bullet^\star)$, respectively.

We posit that *pRT-monotone pET substitution* is the minimum adequacy requirement that any reasonable pWCET definition must satisfy: as surveyed in Secs. I and II, when over-approximating pET distributions with pWCET distributions, we expect to obtain *safe* pRT distributions.

Example. To illustrate Defs. 8 and 9, consider a trivial transformation of the system described in Fig. 4 that replaces pETs with (classical) WCETs. Let $(\Omega, \mathbb{P}, \mathcal{A}_\bullet, \mathcal{C}_\bullet)$ be the original system described in the example and $(\Omega^\star, \mathbb{P}^\star, \mathcal{A}_\bullet^\star, \mathcal{C}_\bullet^\star)$ be the result of the following transformation: pETs are mapped to degenerate random variables always returning the corresponding WCETs

(*i.e.*, $\forall \omega' \in \Omega^\star \colon \mathcal{C}^\star_\bullet(\omega') \triangleq \max_{\omega \in \Omega} \mathcal{C}_\bullet(\omega)$) and the rest is left untouched (*i.e.*, $\Omega^\star \triangleq \Omega$, $\mathbb{P}^\star \triangleq \mathbb{P}$, and $\mathcal{A}^\star_\bullet \triangleq \mathcal{A}_\bullet$). Under fully preemptive fixed-priority scheduling (with any priority assignment), the response times of all jobs in all evolutions do not decrease, and thus $\forall J \in \{J_a, J_b\} \colon \mathcal{R}_J \preceq \mathcal{R}^\star_J$.

C. pRT-Monotone pET Substitution

What remains to be clarified is how, formally speaking, pWCETs can be substituted for pETs, since pWCET is not a job-level random variable, but a task-level function resembling a CDF. The formal connection requires elaboration.

At a high level, pET substitution proceeds on a job-by-job basis. For convenience, let $\kappa \colon \{1, \ldots, |\mathbb{J}|\} \to \mathbb{J}$ enumerate the jobs in the order in which they are processed. For now, let us consider just *one step* of the transformation, in which we seek to replace the pET of an arbitrary given job J_o.

Def. 10 (🏃). *Let $(\Omega^s, \mu^s, \mathcal{A}^s_\bullet, \mathcal{C}^s_\bullet)$ be the system after the s-th step, and let $J_o = \kappa(s+1)$ be a job released by a task τ_i. Given the task's pWCET F_i, define the corresponding* probability mass function *(PMF) f_i as $f_i(0) \triangleq F_i(0)$ and $f_i(t+\varepsilon) \triangleq F_i(t+\varepsilon) - F_i(t)$. The $s+1$-th pET substitution step $\mathfrak{T}^{s+1}$ is defined as:*

$$
\begin{aligned}
\Omega^{s+1} &\triangleq \Omega^s \times \mathbb{W} \\
\mu^{s+1}((\omega, c)) &\triangleq \mu^s(\omega) \cdot f_i(c) \\
\forall J, \mathcal{A}^{s+1}_J((\omega, c)) &\triangleq \mathcal{A}^s_J(\omega) \\
\forall J, \mathcal{C}^{s+1}_J((\omega, c)) &\triangleq \begin{cases} \mathcal{C}^s_J(\omega) & \text{if } J \neq J_o \\ c & \text{if } J = J_o \end{cases}
\end{aligned}
$$

Let us consider each element in turn. First, it can be shown that $\sum_c f_i(c) = 1$ and $f_i(c) \geq 0$ for any cost c; hence f_i is indeed the PMF of F_i. Second, the given set of evolutions Ω^s is augmented with an extra dimension $\mathbb{W}$. The new set Ω^{s+1} remains countable by the classic diagonalization argument [*e.g.*, 24, p. 92]. All arrival times remain unchanged. Finally, the new job cost function directly references the new dimension for J_o, and defers to the given cost function for all other jobs.

By construction, Def. 10 ensures two important properties: first, for any $g \in \{s+1, \ldots, |\mathbb{J}|\}$, $\mathcal{C}^g_{J_o}$ is independent of all other properties of the system since it depends only on its dedicated dimension; and second, $\mathcal{C}^g_{J_o}$'s distribution is the pWCET distribution F_i since $\mathcal{C}^g_{J_o} = c$ occurs with probability $\mathbb{P}[\{(\omega, c) \,|\, \omega \in \Omega^s\}] = \sum_{\omega \in \Omega^s} \mu^s(\omega) \cdot f_i(c) = 1 \cdot f_i(c)$. Thus, $\mathcal{C}^g_{J_o}$ can now be thought of as an "pWCET random variable."

Repeating Def. 10 for each job yields the desired result.

Def. 11 (🏃). *Let $\mathfrak{T}^s$ denote the step-wise transformation defined in Def. 10, and let $S = (\Omega, \mu, \mathcal{A}_\bullet, \mathcal{C}_\bullet)$ be the initial system. The* pET substitution $\mathfrak{T}$ *maps the initial system to the system $(\Omega^\star, \mu^\star, \mathcal{A}^\star_\bullet, \mathcal{C}^\star_\bullet) = \mathfrak{T}(S) = \mathfrak{T}^{|\mathbb{J}|}(\ldots(\mathfrak{T}^2(\mathfrak{T}^1(S))))$.*

By construction, the transformation $\mathfrak{T}$ generates a new system in which all job costs are independent and job costs corresponding to jobs of the same task have identical distributions. That is, job costs are provably IID after pET substitution (since each pET is confined to its own dimension), which matches the expected intuition (Sec. II).

Example. To show how transformation $\mathfrak{T}$ works at the low level, we apply it to our running example from Fig. 4. Transformation $\mathfrak{T}$ iterates twice since there are only two jobs in the example system. A safe pWCET used for both jobs is $F = \left(\begin{smallmatrix} 2 & 4 \\ 0.5 & 0.5 \end{smallmatrix}\right)$, with a PMF $f(c) = 1/2$ iff $c = 2 \vee c = 4$ and $f(c) = 0$ otherwise. The sample space is augmented with two additional dimensions: $\Omega^\star = \Omega \times \mathbb{W} \times \mathbb{W}$. The final distribution function becomes $\mu^\star((\omega, c_1, c_2)) = \mu(\omega) \cdot f(c_1) \cdot f(c_2)$. Arrivals simply ignore the newly added dimensions: $\mathcal{A}^\star_\bullet((\omega, c_1, c_2)) = \mathcal{A}_\bullet(\omega)$. Finally, $\mathcal{C}^\star_{J_a}((\omega, c_1, c_2)) = c_1$ and $\mathcal{C}^\star_{J_b}((\omega, c_1, c_2)) = c_2$.

To verify that $\mathcal{C}^\star_{J_a}$ now follows the pWCET distribution F, let us see why $\mathbb{P}[\mathcal{C}^\star_{J_a} = 2] = 1/2$ (the probability of $\mathcal{C}^\star_{J_a} = 4$ is computed analogously). Since $\mathcal{C}^\star_{J_a}((\omega, c_1, c_2)) = 2$ iff $c_1 = 2$, we must consider the set $\Psi \triangleq \{(\omega, 2, c_2) \,|\, \omega \in \Omega \wedge c_2 \in \mathbb{N}\} \subseteq \Omega^\star$. By distributivity, the sum $\sum_{\omega^\star \in \Psi} \mu^\star(\omega^\star) = \sum_{\omega \in \Omega, c_2 \in \mathbb{N}} \mu(\omega) \cdot f(2) \cdot f(c_2)$ simplifies to $f(2) \cdot (\sum_{\omega \in \Omega, c_2 \in \mathbb{N}} \mu(\omega) \cdot f(c_2))$, which in turn simplifies to $1/2 \cdot 1$, which matches the pWCET $\left(\begin{smallmatrix} 2 & 4 \\ 0.5 & 0.5 \end{smallmatrix}\right)$. Additionally, one can easily show that $\mathcal{C}^\star_{J_a}$ and $\mathcal{C}^\star_{J_b}$ are indeed independent; we omit the calculation due to space constraints.

What remains to be shown is that pET substitution (Def. 11) is pRT-monotonic (Def. 9), assuming that each F_i is an axiomatic pWCET (Def. 7), thereby establishing the adequacy of Def. 7. As it turns out, this is far from obvious.

VII. Axiomatic pWCET is pRT-monotonic

In fact, Def. 7 cannot be proven adequate *by itself*—recall from Sec. III-B that any guarantees inherently must depend on the scheduling policy. Therefore, we must assume that a *scheduling algorithm* $\mathbb{S}$ is employed consistently across $\omega \in \Omega$.

Scheduler. In analogy to $\vec{c}_\bullet$, let $\vec{a}_\bullet$ denote a vector of fixed arrival times. We model the scheduler $\mathbb{S}$ as a function that maps two vectors $\vec{a}_\bullet$ and $\vec{c}_\bullet$ of *fixed* values corresponding to the foundational job properties (*i.e.*, arrival times and costs, respectively) to a schedule of the system. Formally, the schedule σ and the scheduler $\mathbb{S}$ are related as follows: $\mathbb{S}[\mathcal{A}_\bullet(\omega), \mathcal{C}_\bullet(\omega)] = \sigma(\omega)$ for any $\omega \in \Omega$ (🏃). Recall that, specialized on a given $\omega \in \Omega$, $\mathcal{A}_\bullet(\omega)$ and $\mathcal{C}_\bullet(\omega)$ are fixed parameter vectors.

Clearly, $\mathbb{S}$ is not intended to resemble a real scheduler implementation. Rather, it is a minimal abstraction that serves two important purposes in the proof. First, it allows reasoning about related evolutions: since $\mathbb{S}$ is a function, identical input vectors result in an identical output schedule. Second, it allows restricting the scope to anomaly-free policies, as explained next.

We use $\mathbb{S}$ to define an algorithm $\mathbb{R}_J[\vec{a}_\bullet, \vec{c}_\bullet]$ that, given the parameter vectors $\vec{a}_\bullet$ and $\vec{c}_\bullet$, computes the response time of any given job J. It can be shown (🏃) that $\mathbb{R}_J[\mathcal{A}_\bullet(\omega), \mathcal{C}_\bullet(\omega)] = \mathcal{R}_J(\omega)$, that is, we prove that $\mathbb{R}_J[\vec{a}_\bullet, \vec{c}_\bullet]$ computes correctly. Let $\vec{c}_\bullet$ denote any cost vector, and $\vec{c}^+_\bullet$ a derived cost vector obtained from $\vec{c}_\bullet$ by *increasing* an arbitrary element of $\vec{c}_\bullet$. We say that $\mathbb{S}$ is *response-time monotone* (*RT-monotone*) iff, for any given job J and fixed vector of arrivals $\vec{a}_\bullet$, J's response time does not decrease: $\mathbb{R}_J[\vec{a}_\bullet, \vec{c}_\bullet] \leq \mathbb{R}_J[\vec{a}_\bullet, \vec{c}^+_\bullet]$.

Overview. We restrict our attention to RT-monotone schedulers. At a high level, the proof mirrors Sec. VI-C. First, we show

that one step $\mathfrak{T}^s$ (Def. 10) is a pRT-monotone transformation (Def. 9), which takes the Lion's share of the proof effort. In this part, we rely heavily on Fact 1 (LTP). Second, we apply $\mathfrak{T}^s$ repeatedly (Def. 11) to obtain a chain on inequalities $\mathcal{R}_J = \mathcal{R}_J^0 \preceq \mathcal{R}_J^1 \preceq \ldots \preceq \mathcal{R}_J^{|\mathbb{J}|} = \mathcal{R}_J^\star$. The final statement then simply follows from the transitivity of $\preceq$.

Our formal proof [9], if fully elaborated, would require many pages to describe in detail. Due to space constraints, we focus here on the essential steps, with an emphasis on the overall strategy. For further details, we refer the interested reader to the Coq development linked to throughout this section.

pRT-monotonic step $\mathfrak{T}^s$. Consider a system $(\Omega, \mu, \mathcal{A}_\bullet, \mathcal{C}_\bullet)$ transformed by $\mathfrak{T}^s$ into $(\widehat{\Omega}, \widehat{\mu}, \widehat{\mathcal{A}}_\bullet, \widehat{\mathcal{C}}_\bullet)$ to replace the pET of a *single* job $J_o \triangleq \kappa(s)$. By Def. 9, we need to establish:

Theorem 1 (🐾). $\forall J : \mathcal{R}_J \preceq \widehat{\mathcal{R}}_J$

Proof. To prove the dominance $\preceq$, we have to show the inequality $\mathbb{P}[\mathcal{R}_J > r] \leq \widehat{\mathbb{P}}[\widehat{\mathcal{R}}_J > r]$ for arbitrary J and r.

Introduce ξ via LTP. To perform a case analysis on all possible arrival sequences (🐾), we apply LTP to both sides of the inequality. Since $\mathfrak{T}^s$ does not change arrivals, arrival sequences in both $(\Omega, \mathbb{P})$ and $(\widehat{\Omega}, \widehat{\mathbb{P}})$ coincide (🐾). Hence, it suffices to establish the claim for an arbitrary, but *fixed* arrival sequence ξ appearing on both sides of the inequality.

Introduce S_l and $\widehat{S}_l$ via LTP. Now, by Def. 7, there is a partition $\mathfrak{S}$ satisfying Defs. 5 and 6 for job J_o. We show that $\mathfrak{T}^s$ respects the structure of $\mathfrak{S}$, so that there is a simple extension $\widehat{\mathfrak{S}}$ to the transformed system (🐾). Similarly to ξ, since $\mathfrak{S}$ and $\widehat{\mathfrak{S}}$ have identical structure, it suffices to establish the claim for an arbitrary, but fixed pair of "twin" events $S_l \in \mathfrak{S}$ and $\widehat{S}_l \in \widehat{\mathfrak{S}}$. After all this "zooming in," the following inequality remains to be shown: $\mathbb{P}[\mathcal{R}_J > r \wedge \xi \wedge S_l] \leq \widehat{\mathbb{P}}[\widehat{\mathcal{R}}_J > r \wedge \xi \wedge \widehat{S}_l]$.

Condition on $\xi \wedge S_l$ and $\xi \wedge \widehat{S}_l$. Next, we perform a case analysis on the probability of the event $\xi \wedge S_l$. If $\mathbb{P}[\xi \wedge S_l] = 0$, the claim is trivial, so assume otherwise. For the "twin" event in $(\widehat{\Omega}, \widehat{\mathbb{P}})$, we show $\mathbb{P}[\xi \wedge S_l] = \widehat{\mathbb{P}}[\xi \wedge \widehat{S}_l]$ (🐾). Recall the definition of conditional probability. Dividing by $\mathbb{P}[\xi \wedge S_l]$ on the LHS, and by $\widehat{\mathbb{P}}[\xi \wedge \widehat{S}_l]$ on the RHS, reduces the proof obligation to: $\mathbb{P}[\mathcal{R}_J > r | \xi \wedge S_l] \leq \widehat{\mathbb{P}}[\widehat{\mathcal{R}}_J > r | \xi \wedge \widehat{S}_l]$ (🐾).

Switch to $\mathbb{R}_J[\cdot, \cdot]$. We exploit the assumption that a scheduler $\mathbb{S}$ is used in all ω to replace $\mathcal{R}_J(\omega)$ with $\mathbb{R}_J[\mathcal{A}_\bullet(\omega), \mathcal{C}_\bullet(\omega)]$ (analogously for $\widehat{\mathcal{R}}_J(\omega)$) (🐾). For brevity, the expression $\mathbb{R}_J[\mathcal{A}_\bullet(\omega), \mathcal{C}_\bullet(\omega)] > r$ should be understood as an event $\{\omega \in \Omega \,|\, \mathbb{R}_J[\mathcal{A}_\bullet(\omega), \mathcal{C}_\bullet(\omega)] > r\}$ (analogously for the RHS):

$$\begin{aligned}&\mathbb{P}[\mathbb{R}_J[\mathcal{A}_\bullet(\omega), \mathcal{C}_\bullet(\omega)] > r | \xi \wedge S_l] \\ &\quad \leq \widehat{\mathbb{P}}[\mathbb{R}_J[\widehat{\mathcal{A}}_\bullet(\omega), \widehat{\mathcal{C}}_\bullet(\omega)] > r | \xi \wedge \widehat{S}_l].\end{aligned}$$

Pin $\mathcal{A}_\bullet$ and $\widehat{\mathcal{A}}_\bullet$ to $\vec{a}_\bullet$. Note that $\mathcal{A}_\bullet$ is completely determined by ξ. Since we consider one specific ξ, $\mathcal{A}_\bullet$ does not vary within the event $\xi \wedge S_l$. Hence, there is one $\vec{a}_\bullet$ equal to $\mathcal{A}_\bullet(\omega)$ for any $\omega \in \xi \cap S_l$ (analogous reasoning applies to $\widehat{\mathcal{A}}_\bullet$) (🐾):

$$\mathbb{P}[\mathbb{R}_J[\vec{a}_\bullet, \mathcal{C}_\bullet(\omega)] > r | \xi \wedge S_l] \leq \widehat{\mathbb{P}}[\mathbb{R}_J[\vec{a}_\bullet, \widehat{\mathcal{C}}_\bullet(\omega)] > r | \xi \wedge \widehat{S}_l].$$

Pin $\mathcal{C}_\bullet$ and $\widehat{\mathcal{C}}_\bullet$ to $\vec{c}_\bullet$. Next, let us apply LTP to perform a case analysis on *all job costs except for the job cost that we want to replace*. Recall $J_o = \kappa(s)$. We use the notation $\mathcal{C}_{\bullet \setminus J_o}$ and $\vec{c}_{\bullet \setminus J_o}$ to denote that the respective domains do *not* include job J_o. By applying LTP w.r.t. the set of all such reduced-domain cost vectors, we obtain as the remaining proof obligation (🐾):

$$\begin{aligned}&\sum_{\vec{c}_{\bullet \setminus J_o}} \mathbb{P}[\mathbb{R}_J[\vec{a}_\bullet, \mathcal{C}_\bullet(\omega)] > r \wedge \mathcal{C}_{\bullet \setminus J_o} = \vec{c}_{\bullet \setminus J_o} | \xi \wedge S_l] \\ &\leq \sum_{\vec{c}_{\bullet \setminus J_o}} \widehat{\mathbb{P}}[\mathbb{R}_J[\vec{a}_\bullet, \widehat{\mathcal{C}}_\bullet(\omega)] > r \wedge \widehat{\mathcal{C}}_{\bullet \setminus J_o} = \vec{c}_{\bullet \setminus J_o} | \xi \wedge \widehat{S}_l].\end{aligned}$$

Once again, both sums iterate over the same set of values (*i.e.*, they are structurally identical). Hence, it suffices to establish the claim for an arbitrary, but fixed $\vec{c}_{\bullet \setminus J_o}$ and prove that the LHS does not exceed the RHS for this specific cost vector.

Let us apply LTP once more to analyze the cost of J_o, which we will refer to simply as c (as in Def. 10) (🐾). Note that $\vec{c}_{\bullet \setminus J_o}$ and c together define all job costs, so let us "reassemble" the two, once-more obtaining a vector of all costs $\vec{c}_\bullet$. We exchange $\mathcal{C}_\bullet(\omega)$ with $\vec{c}_\bullet$, since any ω that satisfies both $\widehat{\mathcal{C}}_{\bullet \setminus J_o} = \vec{c}_{\bullet \setminus J_o}$ and $\widehat{\mathcal{C}}_{J_o} = c$ also satisfies $\mathcal{C}_\bullet(\omega) = \vec{c}_\bullet$ (similarly for the RHS) (🐾):

$$\begin{aligned}&\sum_{c} \mathbb{P}[\mathbb{R}_J[\vec{a}_\bullet, \vec{c}_\bullet] > r \wedge \mathcal{C}_{\bullet \setminus J_o} = \vec{c}_{\bullet \setminus J_o} \wedge \mathcal{C}_{J_o} = c | \xi \wedge S_l] \\ &\leq \sum_{c} \widehat{\mathbb{P}}[\mathbb{R}_J[\vec{a}_\bullet, \vec{c}_\bullet] > r \wedge \widehat{\mathcal{C}}_{\bullet \setminus J_o} = \vec{c}_{\bullet \setminus J_o} \wedge \widehat{\mathcal{C}}_{J_o} = c | \xi \wedge \widehat{S}_l].\end{aligned}$$

Why did we apply LTP twice, once on $\vec{c}_{\bullet \setminus J_o}$ and once on c, just to recombine the two afterwards? Ultimately, the reason is a technicality in the Coq proof that we gloss over here.

Exploit partition-independence. Both arguments of $\mathbb{R}_J[\vec{a}_\bullet, \vec{c}_\bullet]$ are now fixed values that do not depend on ω. Therefore, $\mathbb{R}_J[\vec{a}_\bullet, \vec{c}_\bullet] > r$ is a boolean *value* (and not a random variable). Hence, we can separate it from the probability expressions on both sides using the indicator function $\mathbb{1}[\cdot]$.

Next, consider the LHS: we exploit Def. 5 for $G = \mathbb{J} \setminus \{J_o\}$ to conclude (🐾) that the pET $\mathcal{C}_{J_o}$ is *conditionally independent* from other pETs $\mathcal{C}_{\bullet \setminus J_o}$ (conditioned on $\xi \wedge S_l$). On the RHS, the probability term can, by construction (Def. 10), be factored into $\mathbb{P}[\mathcal{C}_{\bullet \setminus J_o} = \vec{c}_{\setminus J_o} | \xi \wedge S_l] \cdot \mathbb{P}_f[\widehat{\mathcal{C}}_{J_o} = c]$ (🐾), where $\mathbb{P}_f$ denotes the probability measure induced by pWCET's PMF f_i.

The first factor of the RHS ($\mathbb{P}[\mathcal{C}_{\bullet \setminus J_o} = \vec{c}_{\setminus J_o} | \xi \wedge S_l]$) coincides with the corresponding term on the LHS; hence both cancel out. The second factor of the RHS $\mathbb{P}_f[\widehat{\mathcal{C}}_{J_o} = c]$ does not depend on ξ and $\widehat{S}_l$ since, by construction (Def. 10), $\widehat{\mathcal{C}}_{J_o}$ depends only on the added dimension. All in all, we arrive at (🐾):

$$\begin{aligned}&\sum_{c} \mathbb{1}[\mathbb{R}_J[\vec{a}_\bullet, \vec{c}_\bullet] > r] \cdot \mathbb{P}[\mathcal{C}_{J_o} = c | \xi \wedge S_l] \\ &\leq \sum_{c} \mathbb{1}[\mathbb{R}_J[\vec{a}_\bullet, \vec{c}_\bullet] > r] \cdot \mathbb{P}_f\Big[\widehat{\mathcal{C}}_{J_o} = c\Big].\end{aligned}$$

Use RT-monotonicity. Since the scheduler $\mathbb{S}$ is RT-monotone, if some c_0 causes J to have a response time exceeding r, then this is the case also for any $c > c_0$. Therefore, we consider three cases: **(i)** independently of c, $\mathbb{R}_J[\vec{a}_\bullet, \vec{c}_\bullet] > r$ is always false, **(ii)** independently of c, $\mathbb{R}_J[\vec{a}_\bullet, \vec{c}_\bullet] > r$ is always true, and

(iii) there exists a c_0 such that $c > c_0 \iff \mathbb{R}_J[\vec{a}_\bullet, \vec{c}_\bullet] > r$ (🐓). The first two cases are trivial: if (i), then the LHS and RHS both equal 0 (🐓), and if (ii), then both sides equal 1 (🐓).

Consider the last case. We replace $\mathbb{R}_J[\vec{a}_\bullet, \vec{c}_\bullet] > r$ with $c > c_0$ since, in case (iii), they are equivalent (🐓):

$$\sum_c \mathbb{1}[c > c_0]\mathbb{P}[\mathcal{C}_{J_o} = c | \xi \wedge S_l] \leq \sum_c \mathbb{1}[c > c_0]\mathbb{P}_f[\widehat{\mathcal{C}}_{J_o} = c].$$

The LHS and RHS of the inequality can be simplified to $\mathbb{P}[\mathcal{C}_{J_o} > c_0 | \xi \wedge S_l]$ and $\mathbb{P}_f[\widehat{\mathcal{C}}_{J_o} > c_0]$, respectively. Using the fact that $\mathbb{P}[a > b] \leq \mathbb{P}[c > d] \iff \mathbb{P}[a \leq b] \geq \mathbb{P}[c \leq d]$, we transform the inequality to obtain (🐓):

$$\mathbb{P}[\mathcal{C}_{J_o} \leq c_0 | \xi \wedge S_l] \geq \mathbb{P}_f[\widehat{\mathcal{C}}_{J_o} \leq c_0].$$

Finally, by construction (Def. 10), $\mathbb{P}_f[\widehat{\mathcal{C}}_{J_o} \leq c_0] = F_i(c_0)$. Hence, we end up with $\mathbb{P}[\mathcal{C}_{J_o} \leq c_0 | \xi \wedge S_l] \geq F_i(c_0)$, which follows (🐓) from partition-dominance (Def. 6). □

Transformation $\mathfrak{T}$ is pRT-Monotone. We are now ready to prove the main result, which follows easily from Theorem 1.

Theorem 2 (🐓). *Consider a job $J \in \mathbb{J}$. Let $\mathcal{R}_J$ be the pRT of J in $(\Omega, \mathbb{P}, \mathcal{A}_\bullet, \mathcal{C}_\bullet)$ and $\mathcal{R}^\star_J$ be the pRT of J in $(\Omega^\star, \mu^\star, \mathcal{A}^\star_\bullet, \mathcal{C}^\star_\bullet)$, where the latter is defined as in Def. 11. Then $\mathcal{R}_J \preceq \mathcal{R}^\star_J$.*

Proof. We construct a chain of dominance relations:

$$\mathcal{R}_J = \mathcal{R}^0_J \preceq \mathcal{R}^1_J \preceq \ldots \preceq \mathcal{R}^{|\mathbb{J}|}_J = \mathcal{R}^\star_J, \tag{1}$$

where $\mathcal{R}^s_J$ denotes pRT of J in system $(\Omega^s, \mu^s, \mathcal{A}^s_\bullet, \mathcal{C}^s_\bullet)$ after the s-th step of the transformation. The claim then follows from the transitivity of $\preceq$, Theorem 1, and the fact that, for each task τ_i, the function F_i is an axiomatic pWCET (Def. 7) w.r.t. system $(\Omega^s, \mu^s, \mathcal{A}^s_\bullet, \mathcal{C}^s_\bullet)$ for $0 \leq s \leq |\mathbb{J}|$ (🐓). □

Theorem 2 establishes axiomatic pWCET's *adequacy* in the sense of Def. 9, under any RT-monotone scheduling policy. This means that pWCET distributions satisfying Def. 7 may be used to "replace" pETs (as in Def. 11) to enable *safe* IID reasoning. Axiomatic pWCET is the first notion of pWCET for which the IID guarantee has been formally established.

VIII. Practical Applicability

Axiomatic pWCET (Def. 7) is defined w.r.t. a space of all evolutions Ω, which is rarely (if ever) known. While this might raise questions about the practicality of the proposed approach, it is actually not an issue since Ω is just a modeling construct. In fact, even though Ω usually cannot be *enumerated*, it can still be possible to *partition* it (Defs. 4–6), which suffices.

To illustrate this idea, let us focus on a result by Frias et al. [23], who consider the problem of DFP derivation for randomized robotics applications as found in an autonomous vehicle. While the execution costs observed in the real system are statistically dependent and thus do not satisfy the IID assumption, Frias et al. show that a *hidden Markov model* (HMM) can faithfully describe the application's behavior.

One can interpret the HMM learning task (*i.e.*, parameter inference from a corpus of traces) as the derivation of a partition satisfying Def. 5. Consider an HMM H with k states $H_1, \ldots, H_k$, and let $H^J(\omega)$ denote the state of H in which a job J is released in evolution $\omega \in \Omega$. Each state of the HMM corresponds to a disjoint subset of Ω. More precisely, for each job J, we can define a suitable partition $\mathfrak{S}^J \triangleq \left\{ \left\{ \omega \,\middle|\, H^J(\omega) = H_i \right\} \middle|\, i \in \{1, \ldots, k\} \right\}$ (recall from Def. 7 that a different partition may be chosen for each job). Such a partition satisfies Def. 5 since, by the definition of an HMM, the cost of J is independent of any other cost *conditioned on the current state of the HMM—$H^J(\omega)$*. Thus, we have derived a finite partition satisfying Def. 5 for an opaque, unenumerated set of evolutions Ω.

To satisfy Def. 6, it suffices to simply upper-bound the distribution of job costs in each HMM state (in the sense of Def. 2) by taking a point-wise minimum of the execution-cost CDFs associated with $H_1, \ldots, H_k$. In conclusion, if a task's behavior can be described with an HMM (*e.g.*, as demonstrated in practice by Frias et al. [23]), then Def. 7 applies nicely.

Finally, by design, Def. 7 is *in spirit* very close to Def. 1, just minimally strengthened to guarantee IID reasoning. Indeed, Def. 4 refines the notion of a "scenario of operation" in precise mathematical language, and the essence of Def. 6 coincides in both definitions. It stands to reason that, for any application for which it is possible to satisfy Def. 1 *and* the additional requirements needed for IID reasoning (as given by Davis and Cucu-Grosjean [18, 19]), it will also be possible to apply Def. 7.

IX. Conclusion

Motivated by the observation that previous pWCET definitions are prone to misinterpretation (Sec. III-A), we have developed axiomatic pWCET (Sec. VI-A), the first truly formal notion of pWCET backed by rigorous proof. Axiomatic pWCET follows in the tradition of the prevailing intuition about pWCET (Sec. II-C), but improves upon prior proposals by providing the first mathematically precise and complete justification for the desirable guarantees commonly ascribed to pWCET.

The name "axiomatic pWCET" derives from the fact that Def. 7 is the weakest precondition for which we could find a mechanized, Coq-verified proof of pRT-monotonicity (Sec. VII), which we consider to be the minimum adequacy requirement that any reasonable pWCET definition should satisfy (Sec. VI-C). Overall, axiomatic pWCET significantly advances the state of the art by illuminating exactly *how*, *when*, and *why* the pWCET abstraction enables IID reasoning.

We leave all practical considerations to future work. In particular, it will be interesting to understand, ideally formally, which of the existing pWCET derivation methods (Sec. II-B) are compatible with axiomatic pWCET. Another interesting empirical direction is *hypothesis testing*: given a pWCET distribution claimed to satisfy Def. 7 for a real system, what observations are necessary to accept or reject the claim with high confidence? Finally, on the theoretical side, there is the opportunity to formally verify probabilistic schedulability analyses based on the proposed precise, Coq-backed semantics.

Acknowledgements

We thank Robert I. Davis for his insightful and constructive feedback, which helped to substantially improve the paper.

This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation program (grant agreement No 803111).

References

[1] J. Abella, M. Padilla, J. del Castillo, and F. J. Cazorla, "Measurement-based worst-case execution time estimation using the coefficient of variation," *ACM Transactions on Design Automation of Electronic Systems*, vol. 22, no. 4, pp. 72:1–72:29, 2017.

[2] S. Altmeyer, L. Cucu-Grosjean, and R. I. Davis, "Static probabilistic timing analysis for real-time systems using random replacement caches," *Real Time Systems*, vol. 51, no. 1, pp. 77–123, 2015.

[3] S. K. Baruah and A. Burns, "Sustainable scheduling analysis," in *27th IEEE Real-Time Systems Symposium (RTSS'06), December 5-8, Rio de Janeiro, Brazil*. IEEE, 2006, pp. 159–168.

[4] K. Bedarkar, M. Vardishvili, S. Bozhko, M. Maida, and B. B. Brandenburg, "From intuition to Coq: A case study in verified response-time analysis of FIFO scheduling," in *43rd IEEE Real-Time Systems Symposium (RTSS'22), December 5-8, Houston, TX, USA*. IEEE, 2022, pp. 197–210.

[5] K. Berezovskyi, L. Santinelli, K. Bletsas, and E. Tovar, "WCET measurement-based and extreme value theory characterisation of CUDA kernels," in *22nd International Conference on Real-Time Networks and Systems (RTNS'14), October 8-10, Versaille, France*. ACM, 2014, p. 279.

[6] G. Bernat, A. Colin, and S. Petters, "pWCET: A tool for probabilistic worst-case execution time analysis of real-time systems," Tech. Rep., 2003.

[7] G. Bernat, A. Burns, and M. Newby, "Probabilistic timing analysis: An approach using copulas," *Journal of Embedded Computing*, vol. 1, no. 2, pp. 179–194, 2005.

[8] S. Bozhko, G. von der Brüggen, and B. B. Brandenburg, "Monte Carlo response-time analysis," in *42nd IEEE Real-Time Systems Symposium (RTSS'21), December 7-10, Dortmund, Germany*. IEEE, 2021, pp. 342–355.

[9] S. Bozhko, F. Marković, G. von der Brüggen, and B. B. Brandenburg, "What really is pWCET? A rigorous axiomatic proposal (Artifact)," 2023. [Online]. Available: https://doi.org/10.5281/zenodo.8414267

[10] A. Burns and S. Edgar, "Predicting computation time for advanced processor architectures," in *12th Euromicro Conference on Real-Time Systems (ECRTS'00), June 19-21, Stockholm, Sweden*. IEEE, 2000, pp. 89–96.

[11] F. Cerqueira, F. Stutz, and B. B. Brandenburg, "PROSA: A case for readable mechanized schedulability analysis," in *28th Euromicro Conference on Real-Time Systems (ECRTS'16), July 5-8, Toulouse, France*. IEEE, 2016, pp. 273–284.

[12] F. Cerqueira, G. Nelissen, and B. B. Brandenburg, "On strong and weak sustainability, with an application to self-suspending real-time tasks," in *30th Euromicro Conference on Real-Time Systems (ECRTS'18), July 3-6, Barcelona, Spain*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, pp. 26:1–26:21.

[13] K.-H. Chen and J.-J. Chen, "Probabilistic schedulability tests for uniprocessor fixed-priority scheduling under soft errors," in *12th IEEE International Symposium on Industrial Embedded Systems (SIES'17), June 14-16, Toulouse, France*. IEEE, 2017, pp. 1–8.

[14] K.-H. Chen, M. Günzel, G. von der Brüggen, and J.-J. Chen, "Critical instant for probabilistic timing guarantees: Refuted and revisited," in *43rd IEEE Real-Time Systems Symposium (RTSS'22), December 5-8, Houston, TX, USA*. IEEE, 2022, pp. 145–157.

[15] "The Coq proof assistant, project web site." [Online]. Available: https://coq.inria.fr

[16] 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*, pp. 29–37, 2013.

[17] L. David and I. Puaut, "Static determination of probabilistic execution times," in *16th Euromicro Conference on Real-Time Systems (ECRTS'04), June 30-July 2, Catania, Italy*. IEEE, 2004, pp. 223–230.

[18] R. I. Davis and L. Cucu-Grosjean, "A survey of probabilistic schedulability analysis techniques for real-time systems," *Leibniz Transactions on Embedded Systems*, vol. 6, no. 1, pp. 04:1–04:53, 2019.

[19] ——, "A survey of probabilistic timing analysis techniques for real-time systems," *Leibniz Transactions on Embedded Systems*, vol. 6, no. 1, pp. 03:1–03:60, 2019.

[20] R. I. Davis, A. Burns, and D. Griffin, "On the meaning of pWCET distributions and their use in schedulability analysis," in *In Proceedings Real-Time Scheduling Open Problems Seminar at (ECRTS'17)*, 2017.

[21] J. L. Díaz, D. F. García, K. Kim, C. Lee, L. L. Bello, J. M. López, S. L. Min, and O. Mirabella, "Stochastic analysis of periodic real-time systems," in *23rd IEEE Real-Time Systems Symposium (RTSS'02), December 3-5, Austin, TX, USA*. IEEE, 2002, pp. 289–300.

[22] J. L. Díaz, J. M. López, M. G. Vazquez, A. M. Campos, K. Kim, and L. L. Bello, "Pessimism in the stochastic analysis of real-time systems: Concept and applications," in *25th IEEE Real-Time Systems Symposium (RTSS'04), December 5-8, Lisbon, Portugal*. IEEE, 2004, pp. 197–207.

[23] B. V. Frias, L. Palopoli, L. Abeni, and D. Fontanelli, "Probabilistic real-time guarantees: There is life beyond the i.i.d. assumption," in *23rd IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'17), April 18-21, Pittsburg, PA, USA*. IEEE, 2017, pp. 175–186.

[24] P. R. Halmos, *Naive Set Theory*. Springer, 1974.

[25] J. P. Hansen, S. A. Hissam, and G. A. Moreno, "Statistical-based WCET estimation and validation," in *9th International Workshop on Worst-Case Execution Time Analysis, (WCET'09), July 1-3, Dublin, Ireland*. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2009.

[26] P. S. Heidmann, "A statistical model for designers of rate monotonic systems," in *Proceedings of the Second Annual Rate Monotonic User's Forum (hosted by the SEI)*, 1993, available at https://heidmann.com/rma/paper2.htm.

[27] Y. Liang and T. Mitra, "Cache modeling in probabilistic execution time analysis," in *45th Design Automation Conference (DAC'08), June 8-13, Anaheim, CA, USA*. ACM, 2008, pp. 319–324.

[28] G. Lima and I. Bate, "Valid application of EVT in timing analysis by randomising execution time measurements," in *23rd IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS'17), April 18-21, Pittsburg, PA, USA*. IEEE, 2017, pp. 187–198.

[29] D. Maxim and L. Cucu-Grosjean, "Response time analysis for fixed-priority tasks with multiple probabilistic parameters," in *34th IEEE Real-Time Systems Symposium (RTSS'13), December 3-6, Vancouver, BC, Canada*. IEEE, 2013, pp. 224–235.

[30] D. Maxim, R. I. Davis, L. Cucu-Grosjean, and A. Easwaran, "Probabilistic analysis for mixed criticality systems using fixed priority preemptive scheduling," in *25th International Conference on Real-Time Networks and Systems (RTNS'17), October 4-6, Grenoble, France*. ACM, 2017, pp. 237–246.

[31] "PROSA — a foundation for formally proven schedulability analysis." [Online]. Available: http://prosa.mpi-sws.org

[32] F. Reghenzani, G. Massari, and W. Fornaciari, "Probabilistic-WCET reliability: Statistical testing of EVT hypotheses," *Microprocess. Microsystems*, vol. 77, pp. 103–135, 2020.

[33] K. P. Silva, L. F. Arcaro, and R. S. de Oliveira, "On using GEV or Gumbel models when applying EVT for probabilistic WCET estimation," in *38rd IEEE Real-Time Systems Symposium (RTSS'17), December 5-8, Paris, France*. IEEE, 2017, pp. 220–230.

[34] G. Smolka, *Modeling and Proving in Computational Type Theory Using the Coq Proof Assistant*, 2021. [Online]. Available: https://www.ps.uni-saarland.de/~smolka/drafts/icl2021.pdf

[35] J. Tassarotti, "Proba — a probability theory library for the Coq theorem prover." [Online]. Available: https://github.com/jtassarotti/coq-proba

[36] J. Tassarotti and R. Harper, "A separation logic for concurrent randomized programs," *Proceedings of the ACM on Programming Languages*, vol. 3, no. POPL, pp. 1–30, 2019.

[37] T. Tia, Z. Deng, M. Shankar, M. F. Storch, J. Sun, L. Wu, and J. W. Liu, "Probabilistic performance guarantee for real-time tasks with varying computation times," in *1st IEEE Real-Time Technology and Applications Symposium (RTAS'95), May 15-17, Chicago, Illinois, USA*. IEEE, 1995, pp. 164–173.

[38] G. von der Brüggen, N. Piatkowski, K.-H. Chen, J.-J. Chen, and K. Morik, "Efficiently approximating the probability of deadline misses in real-time systems," in *30th Euromicro Conference on Real-Time Systems (ECRTS'18), July 3-6, Barcelona, Spain*. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2018, pp. 6:1–6:22.