Schedulability analysis of dependent probabilistic real-time tasks

Slim Ben-Amor
INRIA
2, rue Simone Iff
Paris, France
slim.ben-amor@inria.fr

Dorin Maxim
INRIA
2, rue Simone Iff
Paris, France
dorin.maxim@inria.fr

Liliana Cucu-Grosjean
INRIA
2, rue Simone Iff
Paris, France
liliana.cucu@inria.fr

ABSTRACT

The complexity of modern architectures has increased the timing variability of programs (or tasks). In this context, new approaches based on probabilistic methods are proposed to decrease the pessimism by associating probabilities to the worst case values of the programs (tasks) time execution. In this paper, we extend the original work of Chetto et al. [7] on precedence constrained tasks to the case of tasks with worst case execution times described by probability distributions. The precedence constraints between tasks are defined by acyclic directed graphs and these constraints are transformed in appropriate release times and deadlines. The new release times and deadlines are built using new maximum and minimum relations between pairs of probability distributions. We provide a probabilistic schedulability condition based on these new relations.

CCS Concepts

•**Computer systems organization** → **Embedded systems;** *Redundancy;* Robotics; •**Networks** → Network reliability;

Keywords

probabilistic, precedence constraints, schedulability

1. INTRODUCTION

Time critical systems are currently facing an increased complexity of the hardware architectures with a direct impact on the timing variability of the programs (and tasks). This increased complexity of the hardware architectures is motivated by numerous new functionalities that industries other than time critical systems require. Unfortunately the time critical systems do not have any impact on the design of such hardware architectures and need to adapt their timing analysis. Mainly worst case reasoning in presence of larger timing variability is becoming importantly more pessimistic and this conclusion has motivated the appearance of mixed-criticality models [45] with different possible values for the execution time of a program (or task). To our best knowledge, the only existing methods providing such estimation for the execution time of a program belong to the realm of probabilistic and statistical methods [3, 11, 13, 16]. In this context, the concept of probabilistic worst case execution time (pWCET) of a program (or task) has been proposed [10] as well as associated new schedulability analyses for task systems with worst case execution times described by probability distributions [34]. This paper proposes a schedulability analysis of task systems with execution times described by probability distributions and this in presence of precedence constraints. To our best knowledge, there is currently no solution to this problem.

Organization of our paper The existing results relevant with respect to our contribution are detailed in Section 2 following three main classes: tasks with precedence constraints, statistical and probabilistic estimation of tasks temporal parameters and schedulability and scheduling results for probabilistic task sets. Our contribution concerns a probabilistic extension of the Chetto et al. [7] and we provide the description of this result in Section 4. The main definitions and notations on probabilistic approaches are detailed in Section 4. Our model is presented in Section 5 and the associated schedulability analysis in Section 6. Some numerical considerations are provided in Section 7 and we conclude and present hints for future work in Section 8.

RTNS '16, October 19-21, 2016, Brest, France

DOI: http://dx.doi.org/10.1145/2997465.2997499

2. EXISTING WORK

2.1 Schedulability analysis of task systems with precedence constraints

Precedence constraints are together with deadlines probably the most used real-time constraints. This is motivated by the reactivity that real-time systems should ensure. This reactivity to some sensors input is obtained by regularly checking the sensors for new inputs and then producing a result that is sent to the actuators. The order of the execution is usually imposed by precedence constraints.

The precedence constraints are defined by (directed) graphs and the numerous results [4, 5, 7–9, 15, 17, 21, 25, 37, 39–42, 46] indicate the importance as well as the maturity of this topic. The existing results cover schedulability analysis, scheduling policies in both uniprocessor and multiprocessor case as well as shared resources.

2.2 Schedulability analysis of tasks systems with parameters described by probability distributions

The first papers in the real-time community with bearing on our work used the terms *stochastic analysis* [18], *probabilistic analysis* [33,43], *statistical analysis* [1] and *real-time queuing theory* [26] interchangeably. Since the publication of [14], the notion of *stochastic analysis* of real-time systems has been used regularly by the community, regardless of the approach. We use the terms *probabilistic analysis* or *statistical analysis* depending on the approach on which our solution is based. While the former provides the probability or chance of occurrence of future event, the latter searches for a model or some properties when studying some (often large) mass of data of observed past events. The term of *stochastic analysis* is appropriate in our context for models that are time-evolving which is not the case of our paper.

The results on probabilistic real-time systems may be classified following two main types:

- **Schedulability analysis** The seminal paper of Lehoczky [27] proposes the first schedulability analysis of task systems with probabilistic execution times. This result and several improvements [23, 48] consider a specific case of probability distributions for the probabilistic execution times (the difference between probabilistic execution times and probabilistic worst case execution time is detailed in Section 5. Tia et al. [44] and Gardner [19] propose probabilistic analyses for specific schedulers. Abeni et al. [2] proposes probabilistic analyses for tasks executed in isolation and a recent work consider time-evolving models [36]. The most general analysis for probabilistic systems with (worst-case) execution times of tasks described by random variables is proposed in [14]. The most general analysis for probabilistic systems with (worst-case) execution times and inter-arrival times of tasks described by random variables is proposed in [34]. A time-evolving model of probabilistic execution times is introduced in [30] and an associated schedulability analysis on multiprocessors is presented.

  This class of problems is the most studied among probabilistic methods. The next step to complete the existing results is to provide statistical methods for such problems as it has been done in [29]. The statistical methods have the advantage to be able to study more complex models or architectures. Comparisons between the two methods should close this class of problems.

- **Optimal scheduling algorithms** To our best knowledge, there is one paper presenting optimal fixed-priority uniprocessor scheduling algorithms for tasks with probabilistic worst case execution times [33]. For time-evolving probabilistic execution times of tasks an optimal fixed-priority algorithm is proposed for several processors [31].

2.3 Statistical and probabilistic estimation of temporal parameters

Re-sampling with respect to probabilistic worst-case parameters Schedulability analysis may have an important complexity directly related to the number of possible values of the random variables describing the parameters. This complexity may be decreased by using re-sampling techniques that ensure the safeness (the new response time PF upper bounds the result obtained without re-sampling) [35, 38]. For instance the response time analysis of [34] becomes interesting for large systems of tasks by using the re-sampling at the level of probabilistic worst case execution times and probabilistic minimal inter-arrival times. As indicated in [35] there is no optimal re-sampling technique and, thus, any new schedulability analysis should be provided with its own efficient re-sampling technique.

The estimation of the probabilistic parameters Since the seminal paper of Edgar and Burns [16], different papers [11,20,22,24,47] have proposed solutions for the problem of estimating probabilistic worst case execution times. To our best knowledge, one paper proposes a probabilistic execution time estimation [12]. Estimating probabilistic minimal inter-arrival time is an open problem and one short paper provides hints for this estimation [32].

3. EXISTING MODEL OF TASKS WITH PRECEDENCE CONSTRAINTS

In Chetto et al. [7] the authors study the schedulability analysis of task systems with precedence constraints on one processor. Let τ a set of tasks τ_i defined (r_i, C_i, d_i), $\forall i \in \{1, \cdots, n\}$, where r_i is the release time, C_i the execution time and d_i the deadline of τ. Let G be a directed acyclic graph (τ, E) defining a partial order between the tasks of τ. Two tasks τ_i and τ_j $(i \neq j)$ are related by a precedence if and only if (τ_i, τ_j) corresponds to an edge in E. All tasks have the same period and they are released only once.

For instance let τ be a task system τ of 4 tasks $\{\tau_1, \cdots, \tau_4\}$ with an associated defined with precedence graph G (see Figure 1).

In order to decide the schedulability of these task systems the authors of [7] propose the modification of the releases and the deadlines of each task as follows:

$$r_i^* = \max(r_i, r_j^* + C_j : j \to i) \tag{1}$$

$$d_i^* = \min(d_i, d_j^* - C_j : i \to j) \tag{2}$$

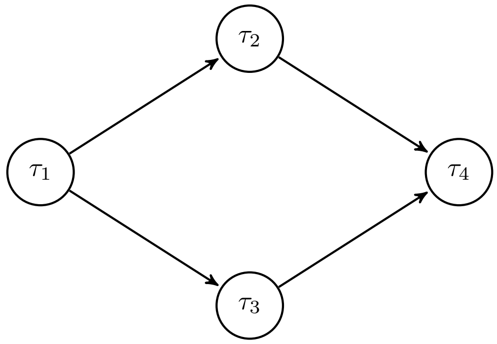

Figure 1: A graph describing the precedence constraints of a task system

Equation (1) imposes that the release of a task cannot be done before the release of its all succesors (with respect to the partial order defined by G). Equation (2) imposes that a task cannot be scheduled after its successors.

Equation (1) is applied from the sources (the tasks without predecessors) of G to the sinks (the tasks without successors) of G. Equation (2) is applied from the sinks of G to the sources of G.

Earliest Deadline First (EDF) is the scheduling policy applied preemptively to the new task systems τ^* containing only independent tasks and Theorem 1 describes a schedulability condition.

THEOREM 1. *[7] Let τ be a task set. τ is schedulable under a preemptive uniprocessor EDF scheduling if and only if $\forall\, j \in \{1, \cdots, n\}$ and $\forall\, i \in \{1, \cdots, n\}$ such that $r_i \leq r_j$, $d_i \leq d_j$,*

$$\sum_{k=1,\ r_i \leq r_k,\ d_k \leq d_j}^{k=n} C_k \leq d_j - r_i \tag{3}$$

The verification of Equation(3) could be expressed in an algorithmic form as follows.

for $j \leftarrow 1$ **to** n **do**
  for $i \leftarrow 1$ **to** n **do**
    $S = 0$
    if $(r_i \leq r_j) and (d_i \leq d_j)$ **then**
      for $k \leftarrow 1$ **to** n **do**
        if $(r_i \leq r_k) and (d_k \leq d_j)$ **then**
          $S = S + Ck$
        end
      end
      if $S > d_j - r_i$ **then**
        return *not schedulable*
      end
    end
  end
end
return *schedulable*

Algorithm 1: Algorithmic description of Theorem 1

4. DEFINITIONS AND NOTATIONS

A random variable $\mathcal{X}$ has a probability function (PF) $f_{\mathcal{X}}(\cdot)$ with $f_{\mathcal{X}}(x) = P(\mathcal{X} = x)$. The possible values of $\mathcal{X}_i$ belong to the interval $[X^{\min}, X^{\max}]$. In this paper, we associate the probabilities to the possible values of a variable using the following notation:

$$\mathcal{X} = \begin{pmatrix} X^0 = X^{min} & X^1 & \cdots & X^k = X^{max} \\ f_{\mathcal{X}}(X^{min}) & f_{\mathcal{X}}(X^1) & \cdots & f_{\mathcal{X}}(X^{max}) \end{pmatrix} \tag{4}$$

where $\sum_{j=0}^{k_i} f_{\mathcal{X}}(X^j) = 1$. A random variable may also be specified using its cumulative distribution function (CDF) $F_{\mathcal{X}}(x) = \sum_{z=X^{min}}^{x} f_{\mathcal{X}}(z)$.

DEFINITION 1. *Two random variables $\mathcal{X}_1$ and $\mathcal{X}_2$ are (probabilistically)* **independent** *if they describe two events such that the outcome of one event does not have any impact on the outcome of the other.*

DEFINITION 2. *The sum $\mathcal{Z}$ of two (probabilistically)* **independent** *random variables $\mathcal{X}_1$ and $\mathcal{X}_2$ is the* **convolution** *$\mathcal{X}_1 \otimes \mathcal{X}_2$ where $P\{\mathcal{Z} = z\} = \sum_{k=-\infty}^{k=+\infty} P\{\mathcal{X}_1 = k\} P\{\mathcal{X}_2 = z - k\}$.*

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

A complementary operator to the convolution is the operator $\ominus$, defined by $\mathcal{X} \ominus \mathcal{Y} = \mathcal{X} \otimes (-\mathcal{Y})$.

DEFINITION 3. *The* **coalescion** *of two partial random variables, denoted by the operator $\oplus$ represents the combination of the two partial random variables into a single (partial) random variable so that values that appear multiple times are kept only once gathering the summed probability mass of the respective values.*

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

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

DEFINITION 4. *[28] Let $\mathcal{X}_1$ and $\mathcal{X}_2$ be two probability distributions. We say that $\mathcal{X}_1$ is* **greater than** *$\mathcal{X}_2$ if $F_{\mathcal{X}_1}(x) \leq F_{\mathcal{X}_2}(x)$, $\forall x$, and denote it by $\mathcal{X}_1 \succeq \mathcal{X}_2$.*

For example, in Figure 2 $F_{\mathcal{X}_1}(x)$ never goes below $F_{\mathcal{X}_2}(x)$, meaning that $\mathcal{X}_2 \succeq \mathcal{X}_1$. Note that $\mathcal{X}_2$ and $\mathcal{X}_3$ are not comparable.

DEFINITION 5. *Let $\mathcal{X}_1$ and $\mathcal{X}_2$ two independent probability distributions and $\mathcal{Z} = \mathsf{Max}(\mathcal{X}_1, \mathcal{X}_2)$*

$$\begin{aligned} p(\mathcal{Z} \leqslant t) &= p(\mathsf{Max}(\mathcal{X}_1, \mathcal{X}_2) \leqslant t) \\ &= p(\mathcal{X}_1 \leqslant t, \mathcal{X}_2 \leqslant t) \\ &= p(\mathcal{X}_1 \leqslant t) p(\mathcal{X}_2 \leqslant t) \\ &= \sum_{i=min(\mathcal{X}_1)}^{t} p(\mathcal{X}_1 = i) \sum_{j=min(\mathcal{X}_2)}^{t} p(\mathcal{X}_2 = j) \end{aligned}$$

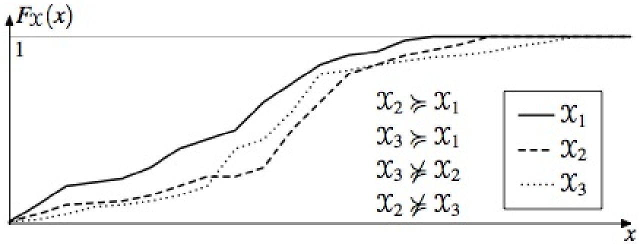

Figure 2: Possible relations between the CDFs of various random variables

If $\mathcal{X}_1$ and $\mathcal{X}_2$ are finite discrete distributions we may write that $p(\mathcal{Z} = t) = \sum_{max(i,j)=t} p(\mathcal{X}_1 = i)p(\mathcal{X}_2 = j)$.

We note that $\mathsf{Max}(\mathcal{X}_1, \mathcal{X}_2) \succeq \mathcal{X}_i, \forall i \in \{1, 2\}$.

Similarly, we define the minimum distribution

$$\mathcal{Z} = \mathsf{Min}(\mathcal{X}_1, \mathcal{X}_2)$$

with

$$p(\mathcal{Z} = t) = \sum_{min(i,j)=t} p(\mathcal{X}_1 = i)p(\mathcal{X}_2 = j)$$

.

We note that $\mathcal{X}_i \succeq \mathsf{Min}(\mathcal{X}_1, \mathcal{X}_2), \forall i \in \{1, 2\}$.

5. PROBABILISTIC MODEL

In this section, we propose a probabilistic description of the model of a task τ_i defined by $\mathcal{R}_i, \mathcal{C}_i, \mathcal{D}_i$ [7]. Our probabilistic description considers the release times, the worst case execution times and the deadlines described by discrete probability distributions and thus a probabilistic task τ.

Let $C_1^i, C_2^i, \cdots, C_m^i$ be m execution times of a task starting from a given scenario of execution S_i.

For a scenario S_i we may define a probabilistic execution time $\mathcal{C}_i$ as an empirical probability distribution of the execution time of that program for the given processor. For instance we have $\mathcal{C}_i$ defined by follows

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

The probabilistic worst-case execution time (pWCET) $\mathcal{C}$ of that program is then an upper bound on all possible probabilistic execution times $\mathcal{C}_i$ for all possible execution scenarios $S_i, \forall\, i \geq 1$. We obtain that $\mathcal{C} \succeq \mathcal{C}_i$.

Similarly if $\mathcal{R}_1^i, \mathcal{R}_2^i, \cdots, \mathcal{R}_m^i$ describes the release times of the task τ_i for m different scenarios, then we may define $\mathcal{R}_i$ as an upper bound on all possible releases. We have $\mathcal{R}_i \succeq \mathcal{R}_j^i, \forall i \in \{1, \cdots, n\}; j \in \{1, \cdots, m\}$.

5.1 Probabilistic transformation of release times and deadlines

In this section, we present the transformation of Equations (1) and (2) to a probabilistic formulation.

Let τ^* be the task system obtained as follows:

$$\mathcal{R}_i^* = \mathsf{Max}(\mathcal{R}_i, \mathcal{R}_j^* \otimes \mathcal{C}_j : j \to i) \tag{6}$$

$$\mathcal{D}_i^* = \mathsf{Min}(\mathcal{D}_i, \mathcal{D}_j^* \otimes (-\mathcal{C}_j) : i \to j) \tag{7}$$

Let τ^* be the task set obtained by applying Equations (6) and (7) to all tasks of a task set τ.

From Definition 5 we obtain that $\mathcal{R}_i^* \succeq \mathcal{R}_i$, and $\mathcal{D}_i \succeq \mathcal{D}_i^*$, $\forall i \in \{1, \cdots, n\}$.

For instance let us consider the task system with the precedence constraints described in Figure 1 and the parameters defined in Table 1.

Task	r_i	$\mathcal{C}_i$	d_i
τ_1	0	$\begin{pmatrix} 1 & 2 \\ 0.9 & 0.1 \end{pmatrix}$	3
τ_2	1	2	5
τ_3	0	2	4
τ_4	4	3	8

Table 1: Timing parameters

After transforming this task set with the defined Min and Max operations we get an independent probabilistic task set. The proposed transformation is applied as follows:

$$\begin{cases} \mathcal{R}_1^* = \mathcal{R}_1 = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ \mathcal{R}_2^* = \mathsf{Max}\,(\mathcal{R}_2, \mathcal{R}_1^* \otimes \mathcal{C}_1) \\ \mathcal{R}_3^* = \mathsf{Max}\,(\mathcal{R}_3, \mathcal{R}_1^* \otimes \mathcal{C}_1) \\ \mathcal{R}_4^* = \mathsf{Max}\,(\mathcal{R}_4, \mathcal{R}_2^* \otimes \mathcal{C}_2, \mathcal{R}_3^* \otimes \mathcal{C}_3) \end{cases}$$

$$\begin{cases} \mathcal{R}_1^* = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ \mathcal{R}_2^* = \mathsf{Max}\left(\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \bigotimes \begin{pmatrix} 1 & 2 \\ 0.9 & 0.1 \end{pmatrix}\right) \\ \mathcal{R}_3^* = \mathsf{Max}\left(\begin{pmatrix} 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \end{pmatrix} \bigotimes \begin{pmatrix} 1 & 2 \\ 0.9 & 0.1 \end{pmatrix}\right) \\ \mathcal{R}_4^* = \mathsf{Max}\left(\begin{pmatrix} 4 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 & 4 \\ 0.9 & 0.1 \end{pmatrix}, \begin{pmatrix} 3 & 4 \\ 0.9 & 0.1 \end{pmatrix}\right) \end{cases}$$

$$\begin{cases} \mathcal{R}_1^* = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \\ \mathcal{R}_2^* = \begin{pmatrix} 1 & 2 \\ 0.9 & 0.1 \end{pmatrix} \\ \mathcal{R}_3^* = \begin{pmatrix} 1 & 2 \\ 0.9 & 0.1 \end{pmatrix} \\ \mathcal{R}_4^* = \begin{pmatrix} 4 \\ 1 \end{pmatrix} \end{cases}$$

Moreover Equation 7 is applied to deadlines. Then, we obtain a transformed task set described in table 2.

Task	$\mathcal{R}_i^*$	$\mathcal{C}_i$	$\mathcal{D}_i^*$
τ_1	0	$\begin{pmatrix} 1 & 2 \\ 0.9 & 0.1 \end{pmatrix}$	2
τ_2	$\begin{pmatrix} 1 & 2 \\ 0.9 & 0.1 \end{pmatrix}$	2	5
τ_3	$\begin{pmatrix} 1 & 2 \\ 0.9 & 0.1 \end{pmatrix}$	2	4
τ_4	4	3	8

Table 2: The parameters of the transformed task set with precedence constraints described in Figure 1

6. SCHEDULABILITY ANALYSIS

In this section, we propose the modification of the Chetto et al. algorithm in order to decide the schedulability of the equivalent probabilistic task set τ^*.

6.1 Non-applicability of the Diaz relation

A natural candidate for replacing the relation $\leq$ in Equation(3) is the Diaz relation [28] defined by Definition 4. Unfortunately we may provide a counter-example of the utilization of the Diaz relation to decide the schedulability of task set.

Let $\tau = \{\tau_1\}$ be a task set with τ_1 defined by

$$(\mathcal{C}_1 = \begin{pmatrix} 1 & 3 \\ 0.9 & 0.1 \end{pmatrix}, \mathcal{D}_1 = \begin{pmatrix} 2 & 4 \\ 0.8 & 0.2 \end{pmatrix})$$

.

In the case when $C_1 = 3$ and $D_1 = 2$, the task set is not schedulable because $C_1 > D_1$. Nevertheless when comparing the CDFs of $F_{\mathcal{C}_1}(t) \geq F_{\mathcal{D}_1}(t), \forall\, t \in \mathbb{R}$ and according to Definition 4 $\mathcal{D}_1 \succeq \mathcal{C}_1$ the task set would be schedulable.

In conclusion, the Diaz relation can not replace the relation $\leq$ in Equation(3) or within the definition of the minimum and the maximum and we propose a new relation in Section 6.2 that justifies Definition 5 .

6.2 A new relation of comparison between two probability distributions

Based on the conclusion of the previous section we propose a new relation to compare the two quantities in Equation(3) as follows.

$$p(\mathcal{X}_1 \leqslant \mathcal{X}_2) = \int_{-\infty}^{\infty} p(\mathcal{X}_1 \leqslant t)p(\mathcal{X}_2 \geqslant t)dt$$

$$= \sum_{t=min(\mathcal{X}_1)}^{max(\mathcal{X}_2)} p(\mathcal{X}_1 \leqslant t)p(\mathcal{X}_2 \geqslant t)$$

$$= \sum_{t=min(\mathcal{X}_1)}^{max(\mathcal{X}_2)} \sum_{i=min(\mathcal{X}_1)}^{t} p(\mathcal{X}_1 = i) \sum_{j=t}^{max(\mathcal{X}_2)} p(\mathcal{X}_2 = j)$$

$$= \sum_{i \leqslant j} p(\mathcal{X}_1 = i)p(\mathcal{X}_2 = j)$$

Thus, the probabilistic comparison operation is given by:

$$\mathcal{X}_1 \leqslant \mathcal{X}_2 = \tag{8}$$

$$\begin{pmatrix} \mathcal{X}_1 \leqslant \mathcal{X}_2 & \mathcal{X}_1 > \mathcal{X}_2 \\ \sum_{i \leqslant j} p(\mathcal{X}_1 = i)p(\mathcal{X}_2 = j) & 1 - \sum_{i \leqslant j} p(\mathcal{X}_1 = i)p(\mathcal{X}_2 = j) \end{pmatrix}$$

If the comparaison $r_i \leq r_k$ (see Algorithm 1, line 6) is verified, then the execution time C_k of current task τ_k is added to the sum S. For the probabilistic extension, if $\mathcal{R}_i^*$ is less than $\mathcal{R}_j^*$ with probability p, we convolve the sum $\mathcal{S}$ with the worst case execution time $\mathcal{C}_k$, p proportion of the time (see p variable in Algorithm 2). Thus, the sum $\mathcal{S}$ is defined as follows.

$$\mathcal{S} = p \times (\mathcal{S} \otimes \mathcal{C}_k) \oplus (1-p) \times \mathcal{S} \tag{9}$$

Data: Transformed Task Set τ^* and the required Confidence level
Result: whether the system is schedulable or not at this confidence level
for $j \leftarrow 1$ **to** n **do**
  for $i \leftarrow 1$ **to** n **do**
    $\mathcal{S} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
    for $k \leftarrow 1$ **to** n **do**
      $p =$ *infEgalProba*$(\mathcal{R}_i^*, \mathcal{R}_j^*) \times$ *infEgalProba*$(\mathcal{D}_k^*, \mathcal{D}_j^*)$
      $\mathcal{S} =$ *coalescion*$(p \times$ *conv*$(\mathcal{S}, \mathcal{C}_k), (1-p) \times \mathcal{S})$;
    end
    if infEgalProba$(\mathcal{S}$, conv$(\mathcal{D}_j^*, -\mathcal{R}_i^*) <$ ConfidanceLevel **then**
      return *not schedulable*
    end
  end
end
return *schedulable*

Algorithm 2: Algorithmic description of the Equation(9)

By applying the modified algorithm to the previous task set $\tau = \{\tau_1\}$ be a task set with τ_1 (see Section 6.1), we obtain that the task set is schedulable with an associated probability 92%. Indeed by aplying Equation(8) we obtain

$$\mathcal{C}_1 \leqslant \mathcal{D}_1 =$$

$$\begin{pmatrix} \mathcal{C}_1 \leqslant \mathcal{D}_1 & \mathcal{C}_1 > \mathcal{D}_1 \\ \sum_{i \leqslant j} p(C = i)p(D = j) & 1 - \sum_{i \leqslant j} p(C = i)p(D = j) \end{pmatrix}$$

$$= \begin{pmatrix} \mathcal{C}_1 \leqslant \mathcal{D}_1 & \mathcal{C}_1 > \mathcal{D}_1 \\ 0.9 * 0.8 + 0.9 * 0.2 + 0.1 * 0.2 & 0.08 \end{pmatrix}$$

This task set is not schedulable if the probability of having the deadline met is larger than 92%. For such system a non-probabilistic schedulability analysis declares the system non-schedulable while its probability of meeting the deadline is high.

In order to evaluate the modified Algorithm 2 we run it for 10 different sets of 6 tasks where each parameter has 8 possible values randomly generated. These tasks have precedence constraints between each other according to the graph provided in Figure 3. Our random generation of the tasks takes as input an average utilization $\frac{1}{max_i\{\mathcal{D}_i\}}\Sigma_{i=1}^{6}\Sigma_{j=1}^{8}p_j^i\mathcal{C}_j^i$ equal to 75%. It uses also UUnifast algorithm [6] to share utilization between tasks. Then, the execution times and deadlines are generated with respect to the individual utilization.

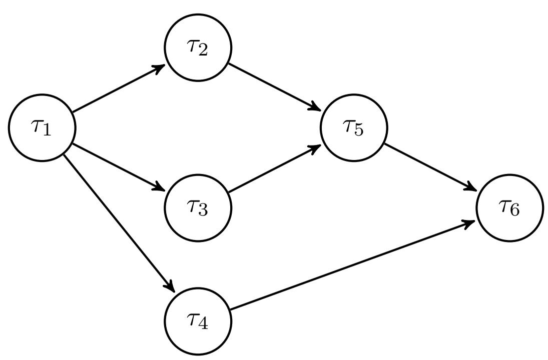

Figure 3: The graph describing the precedence constraints of the generated task system

We obtain all task sets non-schedulable with Equation(3) and we may notice an important schedulability ratio of our test.

Task set	Equation(3)	Our test
1	not schedulable	93,03%
2	not schedulable	87,37%
3	not schedulable	90,15%
4	not schedulable	95,6%
5	not schedulable	92,68%
6	not schedulable	90,82%
7	not schedulable	95,55%
8	not schedulable	98,07%
9	not schedulable	91,52%
10	not schedulable	90,47%

Table 3: The schedulability probability associated to different task sets

Random generation of the precedence constraints We have also evaluated an average schedulability ratio of our test in presence of randomly generated precedence constraints. The graph of precedences is generated by randomly positioning 1 and 0 in the precedence matrix of the graph. This generation may induce the presence of non-schedulable task sets (the probability of meeting the deadline is equal to 0). For instance for 100 task sets with 6 tasks each we obtain 22% of non-schedulable sets and 4% with probabilities of meeting the deadline smaller than 10%. In Figure 4 these task sets are grouped together with the non-schedulable and appear in the first column from the left of the probability distribution illustrated in the figure. We may notice that for the remaining 78% task sets with a probability larger than 10% our test indicate all schedulable (with smaller or larger probability than 10%), while the Chetto test declares 2% schedulable task sets (the first bin from the right of the probability distribution illustrated in the figure).

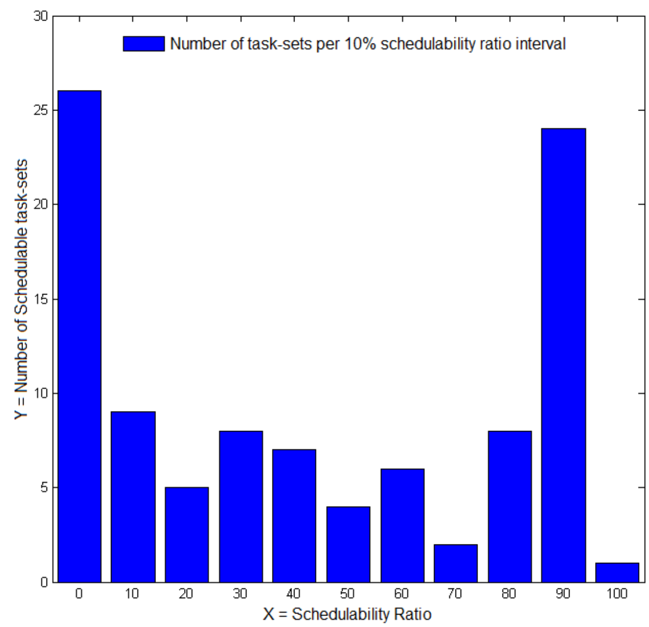

Figure 4: The schedulability ratios for 100 task sets with randomly generated precedence constraints

6.3 Theoretical proof of our probabilistic schedulability condition

In this section, we provide the proof of the schedulability test proposed in our probabilistic extension of Equation (3). The schedulability test is formulated as a corollary of Theorem 2.

THEOREM 2. *Let $\tau = \{\tau_1, \cdots, \tau_n\}$ be a task set with each task τ_i defined by $(r_i = max\{\mathcal{R}_i\}, C_i = max\{\mathcal{C}_i\}, D_i = min\{\mathcal{D}_i\})$ and $G = (\tau, E)$ a directed acyclic graph defining the precedence constraints between the tasks of τ. Let $\tau^* = \{\tau_1^*, \cdots, \tau_n^*\}$ the task set obtained by transforming for each task τ_i its release times and deadlines by using Equations (6) and (7). The task set τ is schedulable if and only if τ^* is schedulable with an associated probability of* 100%.

Proof (Only if part) We consider that τ is schedulable. This implies that it exists at least one schedule meeting both precedence constraints and deadlines. This indicates that all τ_i starts before its predecessors which indicates that the starting time of those tasks satisfies Equation (6) for all possible scenarios. Indeed by definition the execution time of a task τ_i $C_i = max\{\mathcal{C}_i\}$ is larger than any possible value of $\mathcal{C}_i$. Thus if τ_i^* has been assigned any of the values of the new release time $\mathcal{R}_i^*$, then the schedule would be valid because the schedule obtained with the initial r_i is valid by the *if only part* hypothesis.

If it exists at least one schedule meeting the deadlines, then all τ_i ends before its successors which indicates that the finishing time of those tasks satisfies Equation (7) for all possible scenarios. Indeed by definition the execution time of a task τ_i $C_i = max\{\mathcal{C}_i\}$ is larger than any possible value of $\mathcal{C}_i$ and the deadline of a τ_i $D_i = min\{\mathcal{D}_i\}$ is smaller than any possible value of $\mathcal{D}_i$. Thus if τ_i^* has been assigned any of the values of the new deadlines $\mathcal{D}_i^*$, then the schedule would be valid because the schedule obtained with the initial d_i is valid by the *if only part* hypothesis.

(If part) We consider that τ^* is schedulable. This implies that it exists at least one schedule with a probability of 100% that all executions are satisfying the deadline constraints for all possible scenarios (combinations between all values of $\mathcal{C}_i^*$ and $\mathcal{D}_i^*$). As $C_i = max\{\mathcal{C}_i\}$ and $D_i = min\{\mathcal{D}_i\}$, then it stays true also for the largest possible values. This statement is true because we consider the scheduling on one processor.

COROLLARY 1. *Let $\tau = \{\tau_1, \cdots, \tau_n\}$ be a task set with each task τ_i defined by $(r_i, C_i = max\{\mathcal{C}_i\}, D_i = min\{\mathcal{D}_i\})$ and $G = (\tau, E)$ a directed acyclic graph defining the precedence constraints between the tasks of τ. Let $\tau^* = \{\tau_1^*, \cdots, \tau_n^*\}$ the task set obtained by transforming for each task τ_i its release times and deadlines by using Equations (6) and (7). The task set τ is schedulable if and only if Equation (9) is true.*

Proof The corollary is a direct consequence of Theorem 2.

7. ANOTHER PROBABILISTIC RELATION AND NUMERICAL CONSIDERATIONS

We have also considered the maximum and the minimum relations with respect to the Diaz relation in order the define the release times and deadlines in the probabilistic extension of Equation(3). Let $\mathcal{X}_1, \mathcal{X}_2$ be two probability distributions, $F_{\mathcal{X}_1}, F_{\mathcal{X}_2}$ their corresponding cumulative distribution. In this case, we may define $\mathcal{Z}$ the maximum distribution of $\mathcal{X}_1$ and $\mathcal{X}_2$ as follows.

$$\forall\, t \in \mathbb{R}\;\; F_{\mathcal{Z}}(t) = \mathsf{min}(F_{\mathcal{X}_1}(t), F_{\mathcal{X}_2}(t))$$

and we note : $\mathcal{Z} = \mathsf{Max}_{Diaz}(\mathcal{X}_1, \mathcal{X}_2)$.

Similarly, we define the min operation

$$\mathcal{Z} = \mathsf{Min}_{Diaz}(\mathcal{X}_1, \mathcal{X}_2)$$

with

$$\forall\, t \in \mathbb{R}\;\; F_{\mathcal{Z}}(t) = \mathsf{max}(F_{\mathcal{X}_1}(t), F_{\mathcal{X}_2}(t))$$

For these new definitions we have considered the associated algorithmic description of the probabilistic extension of Equation (3) with respect to Min_{Diaz} and Max_{Diaz}. We may notice that with these definitions we do not obtain a probability associated to the deadline failures but an *yes or no* answer. This is a first identified limitation of such definitions.

for $j \leftarrow 1$ **to** n **do**
  for $i \leftarrow 1$ **to** n **do**
    $\mathcal{S} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$
    if $(\mathcal{R}_i^* \preceq \mathcal{R}_j^*) and (\mathcal{D}_i^* \preceq \mathcal{D}_j^*)$ **then**
      for $k \leftarrow 1$ **to** n **do**
        if $(\mathcal{R}_i^* \preceq \mathcal{R}_k^*) and (\mathcal{D}_k^* \preceq \mathcal{D}_j^*)$ **then**
          $\mathcal{S} = \mathcal{S} \otimes \mathcal{C}_k$
        end
      end
      if $\mathcal{S} \succ \mathcal{D}_j^* - \mathcal{R}_i^*$ **then**
        return *not schedulable*
      end
    end
  end
end
return *schedulable*

Algorithm 3: Algorithmic description of the probabilistic extension of Equation (3) with respect to Min_{Diaz} and Max_{Diaz}

We have generated randomly 10 examples of task sets with probabilistic worst case execution times. Our random generation of the tasks takes as input random generation of the worst case execution time probability distributions such that the average utilization $\frac{1}{max_i\{\mathcal{D}_i\}} \Sigma_{i=1}^{6} \Sigma_{j=1}^{8} p_j^i \mathcal{C}_j^i$ is equal to 75%.

While using the Diaz relation and the associated minimum and maximum we observe that only 40% of the task sets are comparable indicating that this relation is not useful within its current form. Indeed for instance $\mathcal{X}_2$ and $\mathcal{X}_3$ in Figure2 are not comparable and our numerical experiments have indicated that this is probably often the case. This is a second identified limitation of these definitions. Nevertheless it may be interesting in the future to analyze the crossing points between the CDFs of the two compared probability distributions and how they may refine our schedulability test.

8. CONCLUSION

In this paper, we have presented a probabilistic model of task sets with precedence constraints in the specific case of EDF scheduling for tasks described by directed acyclic graphs. Our extension considers worst case execution times described by probability distributions. This requires to adapt existing schedulability analysis by defining new probabilistic relations. We consider several possible relations and we show the best relation with respect to this precise schedulability problem.

As future work we would like to introduce multiple periods for the tasks related by precedence constraints while the periods may be described by probability distributions.

9. REFERENCES

[1] L. Abeni and G. Buttazzo. Integrating multimedia applications in hard real-time systems. In *the 19th IEEE Real-Time Systems Symposium (RTSS98)*, pages 4–13, 1998.

[2] L. Abeni and Buttazzo G. QoS guarantee using probabilistic deadlines. In *the 11th Euromicro Conference on Real-Time Systems (ECRTS99)*, 1999.

[3] S. Altmeyer, L Cucu-Grosjean, and R. Davis. Static probabilistic timing analysis for real-time systems using random replacement caches. *Real-Time Systems*, 51(1):77–123, 2015.

[4] Sanjoy Baruah. Federated scheduling of sporadic DAG task systems. In *2015 IEEE International Parallel and Distributed Processing Symposium, IPDPS 2015, Hyderabad, India, May 25-29, 2015*, pages 179–186, 2015.

[5] Sanjoy K. Baruah, Vincenzo Bonifaci, Alberto Marchetti-Spaccamela, Leen Stougie, and Andreas Wiese. A generalized parallel task model for recurrent real-time processes. In *the 33rd IEEE Real-Time Systems Symposium*, pages 63–72, 2012.

[6] E. Bini and G. C. Buttazzo. Measuring the performance of schedulability tests. *Real-Time Systems*, 30((1–2)):129 – 154, 2005.

[7] Houssine Chetto, Maryline Silly, and T. Bouchentouf. Dynamic scheduling of real-time tasks under precedence constraints. *Real-Time Systems*, 2(3):181–194, 1990.

[8] L. Cucu, R. Kocik, and Y. Sorel. Real-time scheduling for systems with precedence, periodicity and latency constraints. In *Real-time and Embedded Systems*, 2002.

[9] L. Cucu and Y. Sorel. Schedulability condition for systems with precedence and periodicity constraints without preemption. In *Real-time and Embedded Systems*, 2003.

[10] 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", the 6Oth birthday of Professor Alan Burns*, 2013.

[11] L. Cucu-Grosjean, L. Santinelli, M. Houston, C. Lo, T. Vardanega, L. Kosmidis, J. Abella, E. Mezzeti, E. Quinones, and F.J. Cazorla. Measurement-based probabilistic timing analysis for multi-path programs. In *the 24th Euromicro Conference on Real-time Systems*, 2012.

[12] L. David and I. Puaut. Static determination of probabilistic execution times. In *the Euromicro Conference on Real-Time Systems(ECRTS)*, 2004.

[13] R. Davis, L. Santinelli, S. Altmeyer, C. Maiza, and L. Cucu-Grosjean. Analysis of probabilistic cache related pre-emption delays. In *25th Euromicro Conference on Real-Time Systems*, 2013.

[14] J.L Díaz, D.F. Garcia, K. Kim, C.G. Lee, L.L. Bello, López J.M., and O. Mirabella. Stochastic analysis of periodic real-time systems. In *the 23rd IEEE Real-Time Systems Symposium (RTSS02)*, 2002.

[15] Unai Díaz-de-Cerio, Juan P. Uribe, Michael González Harbour, and J. Carlos Palencia. Adding precedence relations to the response-time analysis of EDF distributed real-time systems. In *22nd International Conference on Real-Time Networks and Systems, RTNS*, page 129, 2014.

[16] S. Edgar and A. Burns. Statistical analysis of WCET for scheduling. In *the 22nd IEEE Real-Time Systems Symposium*, 2001.

[17] Nathan Fisher and Sanjoy K. Baruah. The feasibility of general task systems with precedence constraints on multiprocessor platforms. *Real-Time Systems*, 41(1):1–26, 2009.

[18] M.K. Gardner and J.W. Lui. Analyzing stochastic fixed-priority real-time systems. In *the 5th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS99)*, pages 44–58, 1999.

[19] M.K. Gardner and J.W. Lui. Analyzing stochastic fixed-priority real-time systems. In *the 5th International Conference on Tools and Algorithms for the Construction and Analusis of Systems*, 1999.

[20] D. Griffin and A. Burns. Realism in Statistical Analysis of Worst Case Execution Times. In *the 10th International Workshop on Worst-Case Execution Time Analysis (WCET 2010)*, 2010.

[21] José C. Palencia Gutiérrez and Michael González Harbour. Exploiting precedence relations in the schedulability analysis of distributed real-time systems. In *the 20th IEEE Real-Time Systems Symposium*, pages 328–339, 1999.

[22] J. Hansen, S Hissam, and G. A. Moreno. Statistical-based wcet estimation and validation. In *the 9th International Workshop on Worst-Case Execution Time (WCET) Analysis*, 2009.

[23] Jeffery P. Hansen, John P. Lehoczky, Haifeng Zhu, and Ragunathan Rajkumar. Quantized edf scheduling in a stochastic environment. In *IPDPS*, 2002.

[24] D. Hardy and I. Puaut. Static probabilistic worst case execution time estimation for architectures with faulty instruction caches. In *the 21st International Conference on Real-Time and Network Systems (RTNS) RTNS*, 2013.

[25] Huang-Ming Huang, Christopher D. Gill, and Chenyang Lu. Mcflow: A real-time multi-core aware middleware for dependent task graphs. In *2012 IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, RTCSA 2012,*

Seoul, Korea (South), August 19-22, 2012, pages 104–113, 2012.

[26] J.P. Lehoczky. Fixed priority scheduling of periodic task sets with arbitrary deadlines. In *IEEE Real-Time Systems Symposium (RTSS)*, 1990.

[27] J.P. Lehoczky. Real-time queueing theory. In *the 10th IEEE Real-Time Systems Symposium (RTSS96)*, pages 186–195, 1996.

[28] J. López, J. Díaz, J. Entrialgo, and D. García. Stochastic analysis of real-time systems under preemptive priority-driven scheduling. *Real-Time Systems*, pages 180–207, 2008.

[29] Y. Lu, T. Nolte, I. Bate, and L. Cucu-Grosjean. A statistical response-time analysis of real-time embedded systems. In *IEEE Real-Time Systems Symposium (RTSS)*, 2012.

[30] S. Manolache, P. Eles, and Z. Peng. Schedulability analysis of applications with stochastic task execution times. *ACM Trans. Embedded Comput. Syst.*, 3(4):706–735, 2004.

[31] S. Manolache, P. Eles, and Z. Peng. Task mapping and priority assignment for soft real-time applications under deadline miss ratio constraints. *ACM Trans. Embedded Comput. Syst.*, 7(2), 2008.

[32] Cristian Maxim, Adriana Gogonel, Dorin Maxim, and Liliana Cucu. Estimation of Probabilistic Minimum Inter-arrival Times Using Extreme Value Theory. In *the 6th Junior Researcher Workshop on Real-Time Computing (JRWRTC)*, 2012.

[33] D. Maxim, O. Buffet, L. Santinelli, L. Cucu-Grosjean, and R. Davis. Optimal priority assignments for probabilistic real-time systems. In *the 19th International Conference on Real-Time and Network Systems*, 2011.

[34] D. Maxim and L. Cucu-Grosjean. Response time analysis for fixed-priority tasks with multiple probabilistic parameters. In *the IEEE Real-Time Systems Symposium*, 2013.

[35] D. Maxim, M. Houston, L. Santinelli, G. Bernat, Robert I. Davis, and L. Cucu-Grosjean. Re-sampling for statistical timing analysis of real-time systems. In *the 20th International Conference on Real-Time and Network Systems*, 2012.

[36] L. Palopoli, L. Abeni, D. Fontanelli, and N. Manica. An analytical bound for probabilistic deadlines. In *the 24th Eumicro Conference on Real-time Systems (ECRTS2012)*, 2012.

[37] K. Ramamritham and J. Stankovic. Scheduling algorithms and operating systems support for real-time systems. In *the IEEE Real-Time Systems Symposium*, 1994.

[38] K. Refaat and P.-E. Hladik. Efficient stochastic analysis of real-time systems via random sampling. In *the 22nd Euromicro Conference on Real-Time Systems*, 2010.

[39] Abusayeed Saifullah, David Ferry, Jing Li, Kunal Agrawal, Chenyang Lu, and Christopher D. Gill. Parallel real-time scheduling of dags. *IEEE Trans. Parallel Distrib. Syst.*, 25(12):3242–3252, 2014.

[40] M. Spuri and J.A. Stankovic. How to integrate precedence constraints and shared resources in real-time scheduling. *IEEE Transactions on Computers*, 43, 1994.

[41] Martin Stigge, Pontus Ekberg, Nan Guan, and Wang Yi. The digraph real-time task model. In *17th IEEE Real-Time and Embedded Technology and Applications Symposium*, pages 71–80, 2011.

[42] Martin Stigge and Wang Yi. Graph-based models for real-time workload: a survey. *Real-Time Systems*, 51(5):602–636, 2015.

[43] T.S. Tia, Z. Deng, M. Shankar, M. Storch, J. Sun, L.C. Wu, and J.S Liu. Probabilistic performance guarantee for real-time tasks with varying computation times. In *the 2nd IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS95)*, pages 164–174, 1995.

[44] T.S. Tia, Z. Deng, M. Shankar, M. Storch, J. Sun, L.C. Wu, and J.S Liu. Probabilistic performance guarantee for real-time tasks with varying computation times. In *IEEE Real-Time and Embedded Technology and Applications Symposium*, 1995.

[45] S. Vestal. Preemptive scheduling of multi-criticality systems with varying degrees of execution time assurance. In *the IEEE Real-Time Systems Symposium*, 2007.

[46] Jia Xu and David Lorge Parnas. Scheduling processes with release times, deadlines, precedence, and exclusion relations. *IEEE Trans. Software Eng.*, 16(3):360–369, 1990.

[47] L. Yue, I. Bate, T. Nolte, and L. Cucu-Grosjean. A new way about using statistical analysis of worst-case execution times. *ACM SIGBED Review*, September 2011.

[48] H. Zhu, J. Hansen, J. Lehoczky, and R. Rajkumar. Optimal partitioning for quantized EDF scheduling. In *IEEE Real-time Systems Symposium*, 2002.