technische universität dortmund

Efficient Computation of Deadline-Miss Probability and Potential Pitfalls

Kuan-Hsun Chen, Niklas Ueter, Georg von der Brüggen, and Jian-Jia Chen

Department of Informatics, TU Dortmund University, Germany

https://ls12-www.cs.tu-dortmund.de/

Citation: 10.23919/DATE.2019.8714908

BibTeX:

```
@inproceedings{DBLP:conf/date/ChenUBC19,
  author    = {Kuan{-}Hsun Chen and
               Niklas Ueter and
               Georg von der Br{\"{u}}ggen and
               Jian{-}Jia Chen},
  editor    = {J{\"{u}}rgen Teich and
               Franco Fummi},
  title     = {Efficient Computation of Deadline-Miss Probability and Potential Pitfalls},
  booktitle = {Design, Automation {\&} Test in Europe Conference {\&} Exhibition,
               {DATE} },
  pages     = {896--901},
  publisher = {{IEEE}},
  year      = {2019},
  url       = {https://doi.org/10.23919/DATE.2019.8714908},
  doi       = {10.23919/DATE.2019.8714908},
}
```


cs 12 computer science 12

Efficient Computation of Deadline-Miss Probability and Potential Pitfalls

Kuan-Hsun Chen, Niklas Ueter, Georg von der Brüggen, and Jian-Jia Chen
Department of Informatics, TU Dortmund University, Germany
E-mail: {kuan-hsun.chen, niklas.ueter, georg.von-der-brueggen, jian-jia.chen} @tu-dortmund.de

***Abstract*—In soft real-time systems, applications can tolerate *rare* deadline misses. Therefore, probabilistic arguments and analyses are applicable in the timing analyses for this class of systems, as demonstrated in many existing researches. Convolution-based analyses allow to derive tight deadline-miss probabilities, but suffer from a high time complexity. Among the analytical approaches, which result in a significantly faster runtime than the convolution-based approaches, the Chernoff bounds provide the tightest results. In this paper, we show that calculating the deadline-miss probability using Chernoff bounds can be solved by considering an equivalent convex optimization problem. This allows us to, on the one hand, decrease the runtime of the Chernoff bounds while, on the other hand, ensure a tighter approximation since a larger variable space can be searched more efficiently, i.e., by using binary search techniques over a larger area instead of a sequential search over a smaller area. We evaluate this approach considering synthesized task sets. Our approach is shown to be computationally efficient for large task systems, whilst experimentally suggesting reasonable approximation quality compared to an exact analysis.**

***Index Terms*—Deadline miss probability, Soft real-time systems**

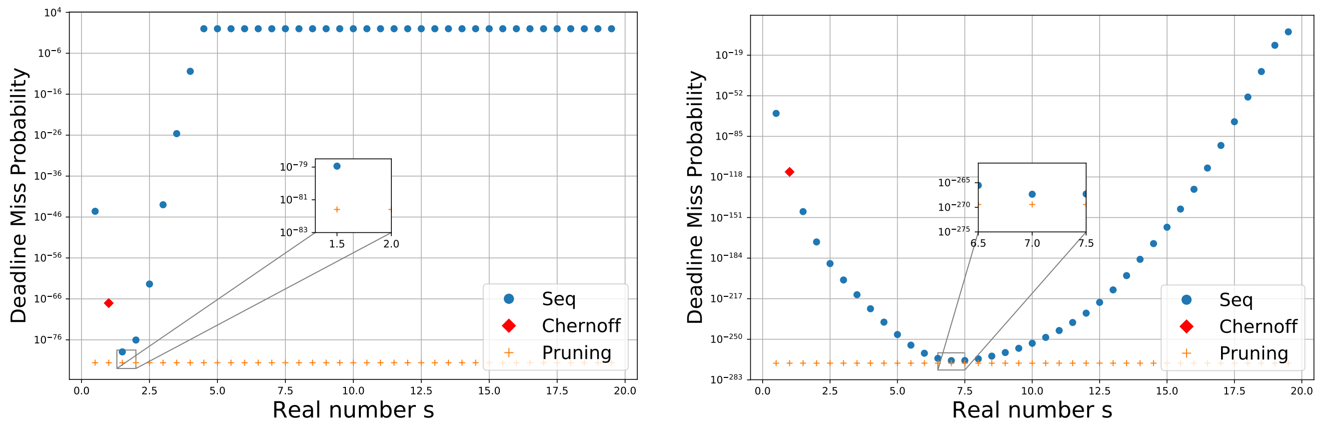
I. Introduction

In embedded and cyber-physical systems, timeliness is an essential feature. The strongest timeliness requirement is to provide hard real-time guarantees. That is, all computing entities must not only be correct functionally, but also compute the respective outputs within given timing constraints. Hard real-time constraints are necessary if any deadline miss may lead to catastrophic consequences. However, many embedded systems can still be functionally correct for occasional, i.e., quantified and bounded, deadline misses. The relevance of these classes of systems to industry is evident from safety standards such as IEC-61508 [11] and ISO-26262 [12] that require (very) low failure probability but not necessarily a failure probability of zero. Further examples of relevance are soft-error recovering systems, where soft errors that occur during a task's execution trigger error-recovery routines. Based on the probabilistic characteristic of the soft error occurrences, the task system exhibits probabilistic behavior. In these cases, probabilistic task models and analyses can help the system designer to achieve expectedly high system utilization, whilst quantifying the probabilistic system behavior using a deadline-miss metric.

We consider sporadic real-time task systems, in which a sporadic task τ_i releases an infinite number of task instances, called jobs, that are separated by a minimum inter-arrival time T_i. All tasks are scheduled under a preemptive fixed-priority algorithm on a uniprocessor system. Suppose that S_t is the maximal cumulative amount of workload resulting from τ_i and its higher-priority tasks over any interval of length t. Now, the probability that a task τ_i is not finished at time t is given by the probability that S_t is strictly larger than the interval, i.e., $\Pr(S_t > t)$. Under the assumption that all jobs are independent, i.e., the execution times are uncorrelated, the deadline miss probability can be inferred using convolution to compose the probability-density function of S_t. Previous job-level convolution-based approaches [1], [9], [14], [16] in probabilistic response-time analyses suffered from their high time complexity, i.e., only a small number of tasks could be considered. Recently, von der Brüggen et al. [18] proposed a task-level convolution-based approach that efficiently decreases the number of states that need to be considered. The authors demonstrate the capability to analyze task systems with up to 100 tasks. Despite the fact that the task-level convolution-based approach is more efficient than the job-level approaches, the time complexity remains exponential with respect to the number of tasks. Furthermore, the experimental results presented by von der Brüggen et al. [18] show that analyzing a task system with 100 tasks can take several hours. Chen and Chen in [7] proposed to use Chernoff bounds to safely over-approximate the deadline-miss probabilities of tasks. Their approach trades a loss of accuracy for improved runtime efficiency as reported in [18].

The Chernoff bound is a parametric bound that holds true for any real number $s > 0$. While the Chernoff bound is an over-approximation with no non-trivial analytical guarantees for the approximation quality, the quality varies with the choice of s. Hence, in order to optimize the approximation quality, it is beneficial to find the smallest Chernoff bound efficiently, based on all possible s values. The following example motivates the studied problem in this paper, and demonstrates the performance of the aforementioned approaches.

Motivational Example: Consider a real-time embedded system with a set of sporadic tasks, i.e., $\Gamma = \{\tau_1, \tau_2, \cdots, \tau_{25}\}$, on a uniprocessor. We follow a similar setup as described in [7], [17], [18]. That is, a task has two modes with associated probabilities, where a mode is characterized by its worst-case execution time (WCET). Such a setting is common when software based fault-tolerance techniques are considered, i.e., a task τ_i has a normal execution mode with a related WCET C_i^N and probability $\mathbb{P}_i^N$ as well as an abnormal execution mode with WCET C_i^A and associated probability $\mathbb{P}_i^A$. Further, $C_i^N \leq C_i^A$, $\mathbb{P}_i^A < \mathbb{P}_i^N$, and $\mathbb{P}_i^N = 1 - \mathbb{P}_i^A$. We use the



(a) Set 0 (b) Set 1

Fig. 1: Derived results comparison: The red diamonds represent the obtained upper bound on the probability of deadline misses by setting $s = 1$ with the Chernoff bound approach [7]. The blue curves are drawn by iterating over different s with step 0.5 by using the Chernoff bound approach. The bottom orange lines are the derived results from the task-level convolution-based approach in [18].

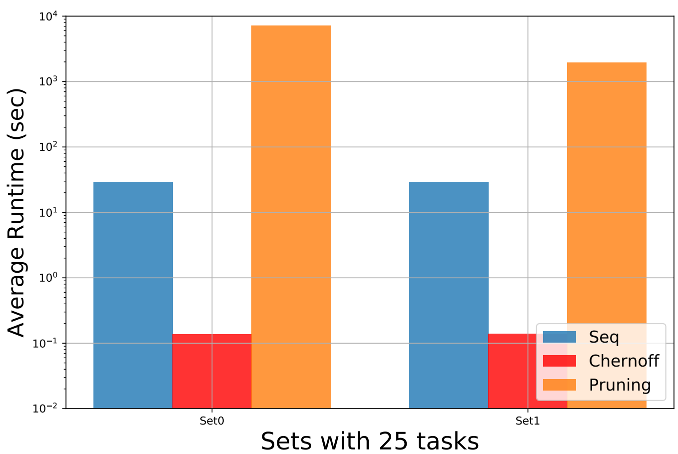


Fig. 2: Runtime comparison: The average required runtime with different state-of-the-arts for the selected task sets in the motivational example.

UUniFast method [4] to synthesize sporadic implicit-deadline task sets with a given normal mode utilization of 60%, setting $\mathbb{P}_i^A = 10^{-4}$ and $C_i^A = 1.83 \cdot C_i^N$ for all tasks. The tasks are scheduled according to rate-monotonic preemptive scheduling, i.e., tasks with shorter periods have higher priority.

For these task sets, we used the state-of-the-art approaches to calculate the deadline miss probability (DMP), i.e., the Chernoff bound approach[1] considering a fixed value $s = 1$ in [7], denoted by **Chernoff**, and the task-level convolution-based approach in [18] with the proposed runtime optimization that prunes out unnecessary states, denoted by **Pruning**. Furthermore, we iteratively calculated the Chernoff bound for different values of s with a step size 0.5, denoted as **Seq**.

[1]The scripts were downloaded from https://github.com/kuanhsunchen/EPST on 23 July 2018. While the script considers multiple s in $(0, 1]$, we modified it to calculate the Chernoff bound for one given real number s.

Figure 1 displays these results and shows that using **Chernoff** with a fixed value can lead to a large gap compared to the DMP for **Pruning**. This also holds true if only a small interval of possible values for s can be considered, i.e., in [7] $s \in (0, 1]$ and in [18] $s \in (0, 3]$ was considered to achieve a good runtime. When **Chernoff** is used iteratively for multiple real number s to find out a suitable s, the gap to **Pruning** may be reasonably small as **Seq** shows. Regarding the required runtime for analyses, shown in Figure 2, **Pruning** is orders of magnitude slower than **Chernoff**. However, the required runtime for finding a suitable s iteratively as in **Seq** highly depends upon the step size and the considered interval. Unfortunately, the results in Figure 1 suggest 1) that the actual value of s differs largely based on the considered task set, and 2) that being slightly off from the best value may lead to a large difference from the value obtained by **Pruning**. However, Figure 1 also suggests that for a given task set the Chernoff bound is a convex function with respect to s.

In this paper, we show that finding the smallest Chernoff bound based on all possible s values in fact poses a convex optimization problem. This allows to increase the precision of the Chernoff bound compared to the results in [7], since a wider range of possible s values can be covered. The reason is that the convex property allows to search the possible interval of s values more efficiently. Hence, the reduced runtime directly leads to a more precise DMP estimation. Furthermore, the task-level convolution-based approach in [18] has a runtime complexity that is exponential in the number of considered tasks for each point in time, while the Chernoff bounds have a runtime complexity that is linear with respect to both the number of tasks and the number of values that are considered for s. Therefore, Chernoff bounds are the only method to determine the deadline miss probability for really large task sets, e.g., 1000 tasks, while still resulting in a reasonable approximation quality.

Our Contributions: Despite the fact that the Chernoff bound offers no specific approximation quality guarantees, the loss of approximation quality compared to [18] can be reasonably small, whilst the analysis is significantly faster. In summary, our contributions are as follows:

- To efficiently compute the upper bound on the deadline-miss probability, we first show that the Chernoff-bound approach is a convex optimization problem, and provide an equivalent logarithm equation to mitigate the computational difficulty as detailed in Section V.
- Throughout several numerical simulations, we demonstrate that the upper bound of the deadline-miss probability derived by the proposed optimization is reasonably small compared to the state-of-the-art while reducing the necessary runtime in Section VI.

Our result enables the possibility to analyze the probability of deadline misses with sufficient accuracy whilst keeping the time complexity in pseudo-polynomial time[2].

II. Related Work

Several approaches that calculate the deadline-miss probability based on probabilistic response-time analyses are known from the literature [2], [9], [14], [16], [18]. For instance, Diaz et al. [9] provided a framework to determine the deadline-miss probability. Another approach was presented by Tanasa et al. [16] that applies a customized decomposition procedure based on the Weierstrass Approximation in order to derive the deadline-miss probability. Unfortunately, both approaches are only suitable for small problem instances, i.e., task systems with at most 7 to 25 jobs in the hyper-period. That constraint holds true, even for the simplest task model, i.e., periodic real-time task systems.

In the research context of sporadic real-time task systems, Axer et al. in [2] derived the response-time distribution for non-preemptive fixed-priority scheduling iteratively with respect to the activations of released jobs.

Maxim and Cucu-Grosjean [14] proposed a probabilistic response time analysis exploiting a (job-level) convolution-based approach, which can handle task sets with up to 10 tasks [7], [18]. By changing from a job-level to a task-level perspective, von der Brüggen et al. [18] reduced the time complexity of the convolution-based approach and introduced several techniques to efficiently decrease the runtime. While it was shown in [18] that their approach can handle sets with 100 tasks, it still has a very high runtime. The approaches in [14] and [18] are tight for computing the deadline miss probability. Hence, to the best of our knowledge, the approach by von der Brüggen et al. in [18] presents the state-of-the-art with respect to *tightness* and time complexity amongst the class of convolution-based response time analyses.

For improving the time complexity of probabilistic response time analyses, Chen and Chen [7] proposed to derive the deadline-miss probability based on Chernoff bounds. Other analytical bounds, i.e., the the Hoeffding inequality and the Bernstein inequality, were applied by von der Brüggen et al. in [18] to derive the deadline-miss probability. As shown in [18], the Chernoff bound approach leads to the tightest results among the analytical bounds.

Based on the Chernoff bound approach [7] or the task-level convolution approach [18], Chen and Chen [8] presented the first analytical upper bound for the deadline miss rate of sporadic constrained-deadline task systems. Since in their approach the deadline miss probability has to be determined not only for the first but also for consecutive jobs, this introduces an additional need for a fast and tight over-approximation of the deadline miss rate. We believe that improving the efficiency of the Chernoff bound approach may bring the value of such probabilistic timing analyses into focus in the future.

III. System Model

In this paper, we consider a real-time system with n independent sporadic tasks $\Gamma = \{\tau_1, \tau_2, \cdots, \tau_n\}$ on a uniprocessor. Each task τ_i is modeled by a tuple $(C_i, \mathbb{P}_i, D_i, T_i)$ where C_i is a random variable that is distributed according to a sampled (discrete) probability-density function $\mathbb{P}_i(u) = \sum_{k=1}^{K_i} \mathbb{P}_i^k \cdot \delta(u - C_i^k)$, where the C_i^k denotes the possible execution times, K_i is the total number of possible values, and δ function here is a Dirac delta function.

Each task releases task instances (or jobs) to the system according to the minimal inter-arrival time constraint T_i such that if a job is released at time t_i then the next job may be released no earlier than $t_i + T_i$, and the job should finish its execution before $t_i + D_i$ in order to meet the deadline, i.e., D_i is the relative deadline of τ_i. Furthermore, the execution time for each job is drawn according to the probability-density function at each job release. In this paper, we consider *implicit-deadline* task sets, i.e., $D_i = T_i \ \forall \tau_i \in \Gamma$, and *constrained-deadline* task sets, i.e., $D_i \leq T_i \ \forall \tau_i \in \Gamma$.

We assume that all realizations of the jobs execution times are independent and identically distributed (i.i.d.) in order to make it analytically tractable, which aligns with the usual assumption in the literature when modeling multiple possible execution times, e.g., [3], [7]–[9], [14], [18]. Despite the known limitations with this assumption, we hope our results to be useful in foresight of possible techniques that may approximate the task sets statistical characteristics by a family of i.i.d. random processes or for problem instances where the i.i.d. assumption is reasonable enough.

We further assume the system is scheduled by a preemptive fixed-priority scheduling policy. Namely each task set Γ is ordered according to the task priorities such that for any two instances of two distinct tasks τ_i and τ_j that are eligible to execute at a given time t, the execution of τ_i precedes the execution of τ_j if $i < j$.

Moreover, we comply with the common assumption that the system is reset once a deadline-miss of any job occurred. By this property, it is ensured that no task suffers backlog, i.e., multiple jobs of the same task are not pending at any time. Nonetheless, if this assumption does not hold, the here presented approach can be used in conjunction with the analysis proposed by Chen et al. in [7] or the task-level convolution-based approach by von der Brüggen et al. [18].

[2]The Chernoff bound approach in [7] deriving the probability of deadline misses still requires the time complexity in pseudo-polynomial time.

IV. Safe Bound of Deadline-Miss Probability

We consider the computation of a safe upper bound of the deadline-miss probability for a specific task τ_j. More precisely, we want to derive an upper bound Φ_j such that

$$\sup_{\sigma \in S} \{Pr(\tau_j \text{ misses deadline in schedule } \sigma)\} \leq \Phi_j \quad (1)$$

where S denotes the set of all possible schedules.

In the literature, it is demonstrated that this upper bound can be derived by considering the maximal amount of workload that every task with higher priority task than τ_j in the task set under analysis can contribute in any interval of length t.

As shown in [14], [18], the maximal amount of workload that each task can contribute to any interval of length t is a random variable whose probability-density function can be computed using the convolution-theorem of independent and equally-distributed random variables. The problem with such convolution-based approaches is that the time complexity of these methods is very high and hence the computation takes a lot of time for the task-level convolution and is infeasible for the job-level convolution if larger task systems are considered [18]. To reduce the time complexity, Chen and Chen [7] proposed to adopt the moment-generating function (MGF) to represent the probabilistic characteristics of such workloads. The MGF is defined as follows:

Definition 1 (Moment-Generating Function)**.** *The moment-generating function of a random variable X_i is defined by the expected value of $e^{X_i \cdot s}$ for any non-negative real number s, i.e., $\text{MGF}(X_i) = E\left[e^{X_i \cdot s}\right]$.*

We use MGF_i to denote the MGF of the execution time of task τ_i. Based on the theory of Chernoff bound [6], [15], it is possible to compute an upper bound of a task's deadline-miss probability without computing the joint-probability density function using convolution. Note, that the Chernoff bound holds for any non-negative real-value s, and thus poses an optimization problem to find the smallest upper bound.

V. Optimization Problem

The objective of this work is to show how to efficiently compute the smallest upper bound of each task's deadline-miss probability, using the Chernoff bound approach in [7] and to evaluate the precision loss compared to the convolution-based approach. That is, the following optimization problem transformed from Eq. 7 in [7] must be solved

$$\Pr(S_t \geq t) \leq \inf_{s_j > 0} \left\{ \left(\prod_{i=1}^{j} \text{MGF}_i(s_j)^{\lceil t/T_i \rceil} \right) \cdot e^{-s_j \cdot t} \right\} \quad (2)$$

where

$$\text{MGF}_i(s_j) = \sum_{k=1}^{K_i} e^{C_i^k \cdot s_j} \cdot \mathbb{P}_i^k. \quad (3)$$

This means, for each task τ_j in the task set, a non-negative real value s_j must be identified that minimizes Eq. (2) for some given time t and a given set of all higher-priority tasks $\tau_1, \tau_2, \ldots, \tau_{j-1}$. In the following, we show that this optimization problem shown in Eq. (2) is log-convex. It thus exhibits a unique minimum and is efficiently solvable by various numerical algorithms.

Theorem 1 (Boyd [5])**.** *Let $y \in Y$ then if a function $f(x, y)$ is log-convex in x for each $y \in Y$ and $f(x, y) \geq 0$, the function $g(x) = \int_Y f(x, y)\ dy$ is log-convex.*

Lemma 1. *The moment-generating function of a task τ_i*

$$\text{MGF}_i = \int_{-\infty}^{\infty} e^{u \cdot s} \cdot \mathbb{P}_i(u)\ du, \quad (4)$$

for a given probability density function $\mathbb{P}_i(u)$ and any $s \in \mathbb{R}^+$, is log-convex.

Proof. By definition of a probability density function, $P_i(u) \geq 0$ for any $u \in \mathbb{R}$ and thus satisfies the conditions stated in Theorem 1. Since the logarithm of the integrand, i.e., $s \cdot u + \ln(\mathbb{P}_i(u))$, is linear in s for any $u \in \mathbb{R}$, it is convex. Therefore, by the arguments of Theorem 1, MGF_i is log-convex. □

Theorem 2. *The moment-generating function of the cumulative execution time of a given number of job-releases is log-convex.*

Proof. By the i.i.d assumption, we know that the moment-generating function of the cumulative execution time of a given number of job-releases can be given as the multiplication of the individual moment-generating functions of each job instance, i.e., $\prod_{i=1}^{j} \text{MGF}_i$ for j job releases. By the property of the logarithm function, the logarithm of the cumulative moment-generating function, i.e., $ln(\prod_{i=1}^{j} \text{MGF}_i)$ is equivalent to $\sum_{i=1}^{j} ln(\text{MGF}_i)$. Since convexity is closed with respect to addition and Lemma 1, we know that the moment-generating function of the cumulative execution time of a given number of job-releases is log-convex. □

In order to minimize Eq. (2), we use the common approach to minimize the logarithm of the equation instead. Since the logarithm is strictly monotonically increasing, the minimum will be the same for both equations. In conclusion, for each task τ_j, we solve the following convex optimization problem

$$\inf_{s_j > 0} \left(\sum_{i=1}^{j} \lceil t/T_i \rceil \cdot ln(\text{MGF}_i(s_j)) \right) - s_j \cdot t \quad (5)$$

where t is given from a finite set of values L_j [7]. The probability that task τ_j misses its deadline is thus upper bounded by

$$\Phi_j = \min_{t \in L_j} \left\{ \inf_{s_j > 0} \left\{ \left(\sum_{i=1}^{j} \lceil t/T_i \rceil \cdot ln(\text{MGF}_i(s_j)) \right) - s_j \cdot t \right\} \right\}. \quad (6)$$

Since this optimization problem is a set of finitely many convex optimization problems, it can be efficiently and unequivocally solved by $|L_j|$ binary search techniques.

Another problem in the numerical computation of the minimal s_j is the sum of exponential functions in MGF_i that may lead to over or underflow if not handled properly.

Numerical Issues: With respect to an implementation of the above optimization problem, the floating-point arithmetic is of special concern due to overflow and underflow problems. There are two types of floating-point arithmetics that can be used, namely either finite-precision with hardware support (by default), or arbitrary-precision with software supports, e.g., the *mpmath* library in Python [13]. In the former case, the computation may suffer from over- and underflow problems since $C_i^k \cdot s$ varies with the parameter range of s. In the latter case, the number of digits that can be used for numeric presentation is only limited by the available memory of the computing system. Note that truncation and approximation errors are unavoidable in both arithmetics due to the limitations of binary representation. Although arbitrary-precision arithmetic is considerably slower than finite-precision arithmetic due to the incurred software overhead, we still adopt it to evaluate the proposed approach to avoid any over- and underflow problems.

VI. Evaluations

In this section, we experimentally evaluate the approximation quality and the runtime of the deadline-miss probability computation of the proposed approach compared to the state-of-the-art, i.e., [18], for different experimental settings.

A. Experimental Setup

We generated sporadic constrained-deadline task sets where the utilization values of the individual tasks are generated using the UUniFast method [4]. Following the suggestion proposed in [10], we generated each tasks period according to a log-uniform distribution within two orders of magnitude, i.e., $T_i \in \{10ms, 1000ms\}$.

Similar to the evaluation in [18] where the task-level convolution based approach was introduced, we consider tasks with two distinctive execution modes and corresponding probabilities, i.e., a normal execution $\left\{C_i^N, P_i^N\right\}$ and an abnormal execution $\left\{C_i^A, P_i^A\right\}$, where $C_i^A = 1.83 \cdot C_i^N \forall\ \tau_i \in \Gamma$, $\mathbb{P}_i^A = \{2.5\%\}$, and $\mathbb{P}_i^N = 1 - \mathbb{P}_i^A$. This system setup is inspired by software based fault-tolerant techniques to handle soft-errors. In such systems, in order to account for the resulting overhead of error recovery, i.e., re-execution, we assume the error detection costs 20% of the task execution time and set C_i^A by $\frac{2.2}{1.2} \approx 1.83 \cdot C_i^N$ for all tasks. Details regarding this setup can be found in [17].

Further, we only calculated the resulting deadline-miss probability of the lowest-priority task under the rate-monotonic scheduling policy in each setting in order to speed up the experimental evaluations and for simplified presentation.

We considered the following approaches: **Chernoff** referring to the Chernoff bound based approach that uses the golden-section search to find an optimal s value, and **Pruning** referring to the task-level convolution-based approach with the pruning technique described in [18]. The evaluations are deployed on an Intel Core i7-4770 and 16GB DDR3 RAM.

B. Results

In the following evaluations, we compare the computed deadline-miss probabilities of the lowest-priority tasks and the

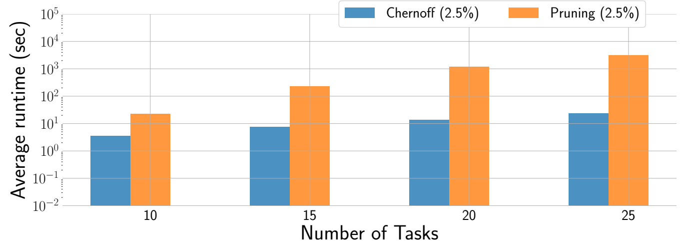


Fig. 3: Average runtime time for the **Chernoff** and **Pruning** method for soft-error implicit-deadline task sets with 50% utilization. The soft-error probability is set to 2.5%.

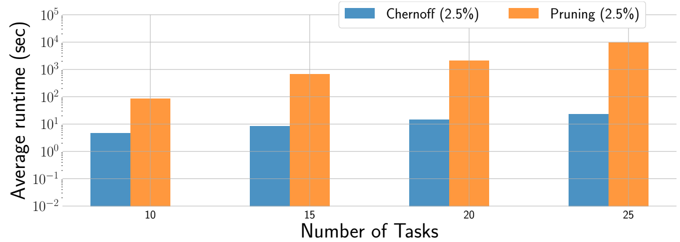


Fig. 4: Average runtime time for the **Chernoff** and **Pruning** method for soft-error implicit-deadline task sets with 70% utilization. The soft-error probability is set to 2.5%.

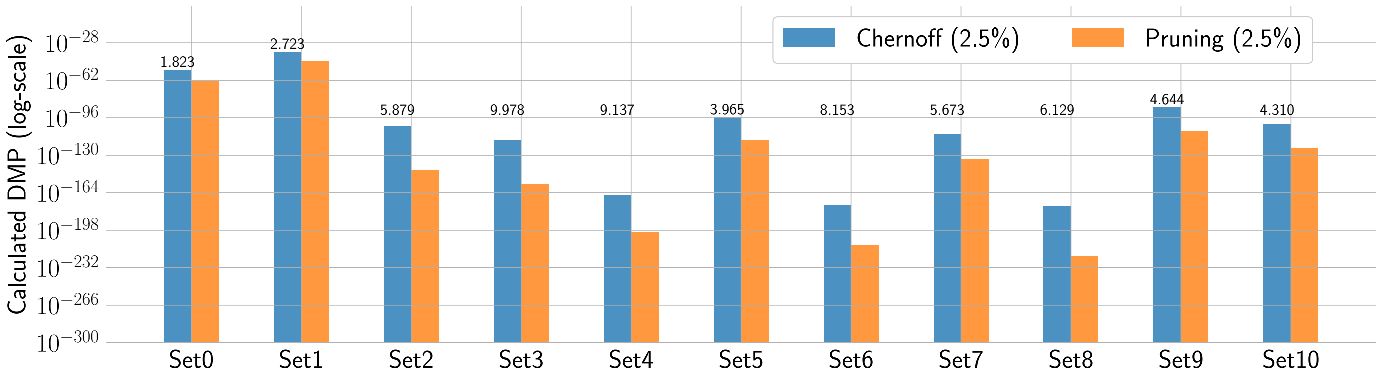
required runtime by **Chernoff** and **Pruning**. Further, we used task sets with a varying number of tasks, i.e., $\{10, 15, 20, 25\}$. Due to the high runtime required by **Pruning**, we created a varying number of task sets depending on the number of tasks, i.e., 100 sets with 10 tasks, 50 sets with 15 tasks, 25 sets with 20 tasks, and 5 sets with 25 tasks. We considered two normal-mode utilizations, namely 50% and 70%.

Fig. 3 and Fig. 4 show that the average computation time used for the Chernoff bounds is $1-3$ magnitudes faster than **Pruning**. Additionally, unlike **Pruning**, the runtime of **Chernoff** is insensitive to the task set utilization. With a larger number of tasks per task set, i.e., 100 tasks, **Pruning** cannot derive any results even over 24 hours, whereas **Chernoff** can finish the computations in average 507.5 sec over 5 sets.

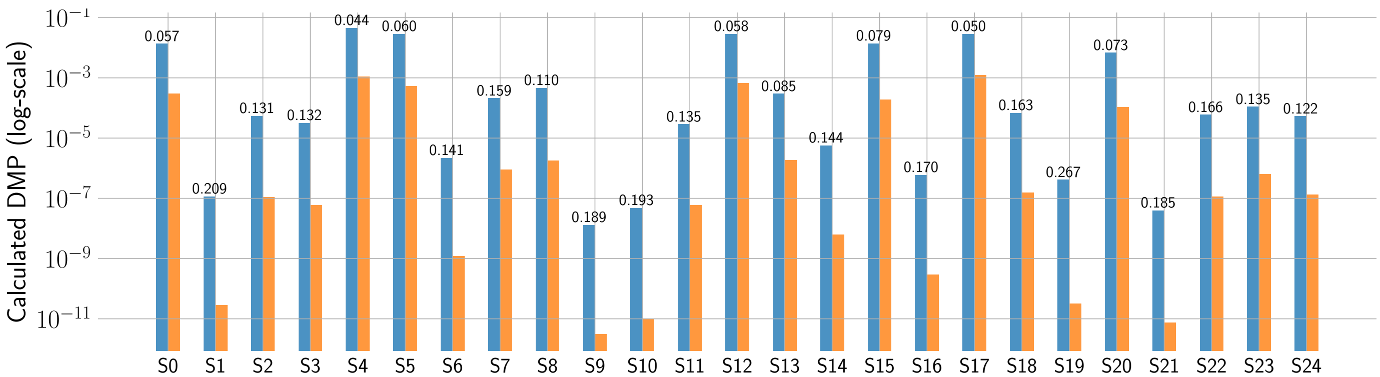
With respect to the approximation quality, Fig. 5 displays the calculated deadline-miss probabilities of the lowest-priority tasks in each analyzed task set consisting of 20 tasks each with cumulative utilization of 50%. As expected, due to the lack of an analytical bound on approximation performance of **Chernoff**, the difference between the two methods can be arbitrarily large. Moreover, the differences are larger for task sets where the deadline-miss probability is already very low. By contrast, in the cases where the deadline-miss probability is higher, e.g., 10^{-3} to 10^{-1}, the differences are relatively smaller. Fig. 5 also shows that the optimal value of s depends on the specific settings of the task set, e.g., the utilization, which empirically shows that testing only a specific range as in [7] is not enough.

VII. Conclusion

In this paper, we demonstrated how to compute an upper bound on the deadline-miss probability by convex-



(a)



(b)

Fig. 5: Approximation quality for the Chernoff and Pruning method for soft-error recovering implicit-deadline task sets with 50% Fig. (a) and 70% Fig. (b) utilization for different task sets. The bars are annotated with the computed optimal value of s.

optimizing the parametric Chernoff bound using the golden-section search. It could be shown that the computations are substantially faster than the state-of-the-art convolution-based approaches. Further, it could be shown that in the cases where the cut-off deadline miss probability of interest is in the region of 10^{-3} to 10^{-1}, the differences in approximation quality are reasonable in light of the runtime improvement.

Acknowledgments

Part of this work has been supported by Deutsche Forschungsgemeinschaft (DFG), as part of the priority program "Dependable Embedded Systems" - SPP1500 GetSURE, and the Collaborative Research Center SFB 876 (http://sfb876.tu-dortmund.de/), subproject A1. Special thanks go to Prof. Rahnenführer for his valuable initial comments.

References

[1] Y. Abdeddaïm and D. Maxim. Probabilistic schedulability analysis for fixed priority mixed criticality real-time systems. In *Design Automation and Test in Europe (DATE)*, 2017.

[2] P. Axer and R. Ernst. Stochastic response-time guarantee for non-preemptive, fixed-priority scheduling under errors. In *Design Automation Conference (DAC)*, 2013.

[3] S. Ben-Amor, D. Maxim, and L. Cucu-Grosjean. Schedulability analysis of dependent probabilistic real-time tasks. In *24th International Conference on Real-Time Networks and Systems (RTNS)*, 2016.

[4] E. Bini and G. Buttazzo. Measuring the performance of schedulability tests. *Real-Time Systems*, 2005.

[5] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004.

[6] J. Bucklew. *Introduction to Rare Event Simulation*. Springer-Verlag, New York, NY, USA, 1st edition, 2004.

[7] 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*, pages 1–8, June 2017.

[8] K.-H. Chen, G. v. d. Brüggen, and J.-J. Chen. Analysis of deadline miss rates for uniprocessor fixed-priority scheduling. In *IEEE International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, August 2018.

[9] J. L. Diaz, D. F. Garcia, K. Kim, C.-G. Lee, L. L. Bello, J. M. Lopez, S. L. Min, and O. Mirabella. Stochastic analysis of periodic real-time systems. In *23rd IEEE Real-Time Systems Symposium (RTSS)*, 2002.

[10] P. Emberson, R. Stafford, and R. I. Davis. Techniques for the synthesis of multiprocessor tasksets. In *International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems*, 2010.

[11] International Electrotechnical Commission (IEC). Functional safety of electrical / electronic / programmable electronic safety-related systems ed2.0. 2010.

[12] International Organization for Standardization (ISO). Iso/fdis26262: Road vehicles - functional safety. 2000.

[13] F. Johansson et al. *mpmath: a Python library for arbitrary-precision floating-point arithmetic (version 0.18)*, December 2013. `http://mpmath.org/`.

[14] D. Maxim and L. Cucu-Grosjean. Response time analysis for fixed-priority tasks with multiple probabilistic parameters. In *Real-Time Systems Symposium (RTSS), IEEE 34th*, pages 224–235, 2013.

[15] M. Mitzenmacher and E. Upfal. *Probability and computing - randomized algorithms and probabilistic analysis*. Cambridge University Press, 2005.

[16] B. Tanasa, U. D. Bordoloi, P. Eles, and Z. Peng. Probabilistic response time and joint analysis of periodic tasks. In *27th Euromicro Conference on Real-Time Systems*, pages 235–246, July 2015.

[17] G. von der Bruggen, K.-H. Chen, W.-H. Huang, and J.-J. Chen. Systems with dynamic real-time guarantees in uncertain and faulty execution

environments. In *IEEE Real-Time Systems Symposium, RTSS*, pages 303–314, 2016.
[18] 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 *EUROMICRO Conference on Real-Time Systems (ECRTS)*, 2018.