Certificates of Real-Time Schedulability

Sanjoy Baruah
Washington University in Saint Louis
baruah@wustl.edu

Pontus Ekberg
Uppsala University
pontus.ekberg@it.uu.se

***Abstract*—One method of showing that a hard-real-time system is schedulable is to present a "certificate" of its schedulability — e.g., a (static) schedule which can be verified to always meet all deadlines. We identify some widely-studied real-time systems for which short (i.e., polynomial-sized) certificates of schedulability exist that can be verified in polynomial time, and apply ideas and results from computational complexity theory to identify other systems for which such certificates are unlikely to exist.**

***Index Terms*—Periodic Task Systems; Schedulability; Polynomial-time Verification**

I. Introduction

Cyclic-executive (CE) [1], [2] based approaches have proved to be a successful means of demonstrating that a safety-critical real-time system will meet all its timing constraints. In such approaches, the system developer provides the certification authority (CA) with a lookup table, called the CE, that explicitly enumerates which task will execute at each instant; the CA checks that repeated execution of this table assigns adequate computing to each task to allow all its timing constraints to be met (provided, of course, that no task executes for a duration exceeding its *worst-case execution time* or WCET).

When used in this manner, we can think of the CE as a **certificate** of the schedulability of the system. For periodic task systems [3], [4] the size of the CE is proportional to the hyper-period and may therefore in general be of size exponential in the representation of the task system under consideration. Furthermore, verifying the correctness of such a certificate takes the CA time at least linear in the size of the certificate; i.e., exponential in the representation of the task system.

In this work we adopt a more expansive notion of a certificate than merely an explicit enumeration of a schedule lookup table. As an illustrative problem we consider the preemptive uniprocessor scheduling of synchronous periodic constrained-deadline task systems (see Section II). We discuss alternative certificates of schedulability for such systems, and provide informal assurance arguments as to why these certificates may be considered adequate for the purposes of verifying schedulability. We apply standard results from computational complexity theory in order to identify, in Section III, a particular schedulability analysis problem for which a system developer is able to provide *certificates that can be verified in time polynomial in the size of the representation of the task system*. By defining explainability in terms of the guaranteed existence of such polynomial-time verifiable certificates, we also identify, in Sections IV and V, a pair of schedulability analysis problems that are unlikely to be explainable according to this notion. In Section VI we propose some directions for research upon such a notion of explainability: seeking explainable *sub*problems of problems that are unlikely to be explainable in general.

II. System Model

We consider a synchronous periodic constrained-deadline task system Γ that is to be scheduled upon a single preemptive processor. Each periodic task $\tau_i \in \Gamma$ is characterized by three integer parameters: its worst-case execution time C_i, its relative deadline D_i, and its period T_i; we restrict attention to constrained-deadline systems in which $D_i \leq T_i$. To recap the synchronous periodic task model: each task τ_i releases a job at each time instant $k \times T_i$ for all $k \in \mathbb{N}$; the job released at time-instant $k \times T_i$ has a WCET C_i and a deadline at time-instant $(k \times T_i + D_i)$.

Notation: we will use $H(\Gamma)$ to denote the hyper-period (least common multiple of all the periods) of task system Γ, and $U(\Gamma)$ to denote its utilization, $U(\Gamma) = \sum_{\tau_i \in \Gamma} C_i/T_i$.

Explainability. A system developer that chooses to model their system as a periodic task system for the purposes of obtaining certification would presumably need to justify this choice to the CA. They would, for instance, need to provide some justification for the values they have assigned to the WCET parameters characterizing their tasks[1], and explain why one may model the processor as being preemptive. We will not address this issue of model-justification any further in this note, other than pointing out that safety-critical system design methodologies (e.g., the Rate-Monotonic Analysis methodology [6]) exist that explain how this may be done.

In Section V, we will consider an extended version of this problem in which each task τ_i is additionally characterized by a *best case execution time (BCET)* B_i. This model, if used, must also be justified, and a convincing argument provided as to why it is safe to assume that each job of a task will execute for a duration no smaller than the value that is assigned to the BCET parameter value of the task.

III. Fixed-Priority Scheduling

Fixed-priority (FP) scheduling is a priority-based scheduling scheme in which each task in the system is assigned a unique priority, and at each instant in time the highest-priority

[1]Such justification may, for instance, take the form of stating that the values were obtained using tools [5] that have been certified for this purpose.

task needing execution is executed on the shared processor. *Response-time analysis (RTA)* [7], [8] is the standard technique for determining whether a constrained-deadline synchronous periodic task system is schedulable or not under FP scheduling with given priorities. RTA is based on the observation [7] that if a constrained-deadline task system is schedulable under FP, then the maximum possible duration between the release of a job of τ_i and the instant this job completes execution (called the *worst-case response time* of task τ_i) is equal to the smallest positive value of R_i that satisfies the following recurrence:

$$R_i = C_i + \sum_{\tau_j \in \mathrm{hp}(\tau_i)} \left\lceil \frac{R_i}{T_j} \right\rceil \times C_j \qquad (1)$$

(Here $\mathrm{hp}(\tau_i)$ denotes all jobs in the task system that have scheduling priority greater than τ_i's scheduling priority.)

Explainability. A system developer that chooses to use FP scheduling must first have the CA accept the validity of RTA.[2] Assuming the CA accepts this, the certificate for schedulability for a given task system Γ is a value for R_i for each $\tau_i \in \Gamma$ that satisfies Expression 1 and is $\leqslant D_i$'s. The certificate comprises $|\Gamma|$ numbers, and so is polynomial (in fact linear) in the representation of the task system Γ. It is straightforward to observe that each claimed R_i can be verified to be a solution to Equation 1 in linear time.

IV. EDF Scheduling

Earliest-deadline-first (EDF) scheduling is another priority-based scheduling algorithm. In an EDF-scheduled system, at each instance the currently *active* (i.e., needing execution) job with the earliest deadline is executed — ties may be broken arbitrarily. *Processor-demand analysis (PDA)* [11] is the standard technique for determining whether a synchronous periodic task system is schedulable or not under EDF scheduling. PDA asserts that constrained-deadline synchronous periodic task system Γ is EDF-schedulable if (and only if) the following constraint is satisfied for all $t \in [0, H)$:

$$\left(\sum_{\tau_i \in \Gamma} \max \left(\left\lfloor \frac{t - D_i}{T_i} \right\rfloor + 1, 0 \right) \times C_i \right) \leqslant t \qquad (2)$$

Explainability. Getting the CA to accept the validity of PDA does not yield a means to generate a certificate for a polynomial-time verification algorithm. Indeed, the existence of such a certificate seems highly unlikely since it would then follow (from the definition of the complexity class NP) that EDF schedulability of constrained-deadline synchronous periodic task systems is in NP. But this schedulability problem has previously been shown [12]–[14] to be coNP-hard, hence its membership in NP would immediately imply that coNP = NP, which runs counter to the expectations of most researchers in computational complexity theory.

[2]This has essentially been achieved: there appears to be wide-spread acceptance by most certification authorities that Equation 1 is indeed correct. Additionally in recent years, this RTA (and much more) has been formally proven correct [9] with machine-verified proofs in the Prosa [10] framework.

V. Best-case Response Times

We have seen that there are polynomial-time verifiable certificates for FP-schedulability but it is unlikely that such certificates exist for EDF-schedulability. In showing [non]-existence of polynomial-time verifiable certificates, the devil is very much in the details of the exact question being asked — we demonstrate this below by showing that a problem, which is very closely related to the FP-schedulability problem, is coNP-hard and therefore does *not* allow for polynomial-time verifiable certificates if NP $\neq$ coNP.

The problem we consider is to establish a lower bound on the best-case response-time (BCRT) of a task under FP-scheduling. This is a practically relevant problem since bounding both the BCRT (from below) and the WCRT (from above) allows us to bound the jitter in task responses. For the BCRT problem to be meaningful we need the task model to also include a best-case execution-time B_i of each task. The hardness of the BCRT problem that we will establish here does not stem from any complicated relationship between the B_i and C_i parameters; in the following we show that the BCRT problem is coNP-hard even if $B_i = C_i$ for all tasks.

The Best-Case Response-Time (BCRT) Problem

INSTANCE: An FP-scheduled synchronous periodic task system Γ with each task $\tau_i \in \Gamma$ additionally characterized by a best-case execution time (BCET) B_i, and a positive integer a.

QUESTION: Does each job of the lowest-priority task in Γ have a response time $\geqslant a$?

We will establish the coNP-hardness of the BCRT problem by relating it to the worst-case response-time (WCRT) problem. Determining FP-schedulability is equivalent to determining whether the WCRT R_i (as in Eq. 1) of each task is no larger than its relative deadline D_i. We find it convenient in our derivation below to use the following utilization-restricted variant of the WCRT problem, which has itself been shown [15] to be NP-complete.

The Worst-Case Response-Time (WCRT) Problem

INSTANCE: An FP-scheduled synchronous periodic task system Γ with $U(\Gamma) \leqslant \ln 2$, and a positive integer a.

QUESTION: Does each job of the lowest-priority task in Γ have a response time $\leqslant a$?

We note that the key difference between the above two problem formulations is that we are asked if the given number a is an upper bound to the possible response times in the WCRT case,

and a lower bound in the BCRT case. We will use a simple trick to reduce from the WCRT problem to the complement of the BCRT problem, thereby showing coNP-hardness for the BCRT problem.

We reduce from the WCRT problem to the BCRT problem by copying the task set Γ of the former problem to a task set Γ' for the new problem, but changing the period of the lowest-priority task τ_{low} in Γ' to equal the hyper-period,

$$T_{\text{low}} = H(\Gamma),$$

and assigning best-case execution times

$$B_i = C_i$$

to all tasks $\tau_i \in \Gamma'$.

The change to τ_{low}'s period effectively means that it will only release the first job in every hyper-period in Γ' compared to Γ. It is well-known that if the first job in the hyper-period has a response-time $\leqslant T_{\text{low}}$, then that job has the maximum response time [16]. Since we have $U(\Gamma) \leqslant \ln 2$, the response-time of the first job must be $\leqslant T_{\text{low}}$ by Liu and Layland's utilization bound [3], and so τ_{low}'s WCRT must be the same in Γ and Γ'. But since τ_{low} only releases a single job per hyper-period in Γ', and since all tasks have $B_i = C_i$, it must also be the case that τ_{low}'s WCRT and BCRT are the same in Γ'. In order to answer the WCRT problem for Γ

> "Does each job of the lowest-priority task in Γ have a response time $\leqslant a$?"

we can simply answer the BCRT problem for Γ'

> "Does each job of the lowest-priority task in Γ' have a response time $\geqslant a + 1$?"

and negate the answer. It follows that the BCRT problem is coNP-hard and therefore does not allow polynomial-time verifiable certificates if NP $\neq$ coNP.

As an immediate corollary, we may conclude that the following problem for bounding the response time within an interval is both NP-hard and coNP-hard, and therefore is unlikely to have either polynomial-time verifiable certificates or counterexamples.

The Response-Time Jitter Problem

INSTANCE: An FP-scheduled synchronous periodic task system Γ with each task $\tau_i \in \Gamma$ additionally characterized by a best-case execution time (BCET) B_i, and positive integers a, b.

QUESTION: Does each job of the lowest-priority task in Γ have a response time in interval $[a, b]$?

Explainability. Despite being so closely related to the WCRT problem, and hence the FP-schedulability problem, we have seen that there are likely no polynomial-time verifiable certificates for the BCRT problem (if there are, then NP = coNP).

We consider the coNP-hardness of the BCRT problem an interesting result in itself; this section additionally demonstrates how computational complexity may change with only small variations of the questions asked, and how this can determine the existence of efficiently-verifiable certificates.

VI. Summary & Discussion

One effective means of "explaining" that a system is schedulable has been by presenting verifiable certificates of its schedulability, as is evidenced by the prevalence of cyclic-executive based scheduling approaches in important safety-critical application domains such as avionics. A more general notion of certificate than the explicit schedule as provided by cyclic executives is some more abstract *proof* of schedulability that can be independently verified by, say, a certification authority. A very formal (and very interesting) approach to this are the machine-checkable certificates generated by the foundational response-time analysis of Maida et al. [17].

The notion of explainability that we focus upon in this note is this: *is a certificate guaranteed to exist for all schedulable task systems, that can be verified in time polynomial in the representation of the task system whose schedulability is to be verified?* Under this interpretation, explainability implies membership in the computational complexity class NP; and as a contrapositive, if a schedulability analysis problem is $\notin$ NP, then the problem is not explainable in general – it is not the case that all instances have polynomial-sized certificates. Showing, as we have done in this note, that a problem is coNP-hard offers very strong evidence that it is $\notin$ NP, since we otherwise would have NP = coNP.

We have demonstrated this equivalence between computational complexity and this notion of explainability via examples upon some commonly-studied preemptive uniprocessor schedulability analysis problems. Under FP scheduling, we noted that upper bounds on worst-case response time are explainable, but showed that lower bounds on best-case response times are likely not. For EDF scheduling, we noted that determining schedulability is unlikely to be explainable.

We can of course apply this thinking to other complexity results in real-time scheduling theory. To exemplify with a few complexity results for some *multiprocessor* schedulability analysis problems:

- Partitioned FP scheduling of constrained-deadline synchronous periodic task systems is in NP, this follows for example from an ILP formulation by Zheng et al. [18]. (In fact it is NP-complete even for unrelated heterogeneous processors, see [19, Sec. VII] for an overview on the complexity of partitioned schedulability problems.) Hence partitioned FP scheduling of constrained-deadline synchronous periodic task systems is explainable.
- In contrast, it is currently *unknown* if the global FP-schedulability problem of constrained-deadline synchronous periodic task systems has polynomial-sized certificates. This problem is NP-hard (which follows directly from the hardness of the single-processor case [15]), but to the best of our

knowledge no better lower bounds on its complexity are known, and it is not known to be in NP.

- Multiprocessor schedulability for a single *conditional DAG (C-DAG)* [20]–[22] under restricted processor assignment is unlikely to be explainable, since this schedulability analysis problem is known [23] to be PSPACE-complete (and since it is generally believed that NP $\neq$ PSPACE).

Directions for future research. Although this note (and indeed, this workshop) deals with explainability, the traditional focus of real-time scheduling theory research has primarily been on devising efficient algorithms for determining, rather than explaining, schedulability. In this traditional context if a schedulability analysis problem is shown to be computationally hard, one approach has been to try to identify sub-problems that are solvable in polynomial time (for example, FP and EDF schedulability analysis of *harmonic* task systems may be looked upon as such sub-problems of the schedulability analysis problems considered in Sections III and IV, for which exact polynomial-time schedulability tests are known [24], [25]).

This flavor of prior research suggests a promising future research direction on the approach to explainability that we have investigated in this note: If some schedulability analysis problem that arises frequently in practice is $\notin$ NP and therefore unlikely to have polynomial-sized certificates, there may be sub-problems of it that are in NP and hence possess polynomially-verifiable certificates of schedulability. Since P $\subseteq$ NP, there is an obvious possibility that there are more or larger practically relevant sub-problems of this type than there are sub-problems that are efficiently solvable. In other words, if explainability is a main concern it may be meaningful to search for practically relevant sub-problems not only for their efficient solvability, but also for their explainability

References

[1] T. P. Baker and A. Shaw. The cyclic executive model and Ada. In *Proceedings of the 9th Real-Time Systems Symposium (RTSS)*, pages 120–129, 1988.

[2] T. P. Baker and A. Shaw. The cyclic executive model and Ada. *Real-Time Systems*, 1(1):7–25, 1989.

[3] C. Liu and J. Layland. Scheduling algorithms for multiprogramming in a hard real-time environment. *Journal of the ACM*, 20(1):46–61, 1973.

[4] Joseph Y.-T Leung and M. Merrill. A note on the preemptive scheduling of periodic, real-time tasks. *Information Processing Letters*, 11:115–118, 1980.

[5] Reinhard Wilhelm, Jakob Engblom, Andreas Ermedahl, Niklas Holsti, Stephan Thesing, David Whalley, Guillem Bernat, Christian Ferdinand, Reinhold Heckmann, Tulika Mitra, Frank Mueller, Isabelle Puaut, Peter Puschner, Jan Staschulat, and Per Stenström. The worst-case execution-time problem – overview of methods and survey of tools. *ACM Transactions on Embedded Computing Systems*, 7(3):36:1–36:53, May 2008.

[6] Mark H. Klein, Thomas Ralya, Bill Pollak, Ray Obenza, and Michael González Harbour. *A Practitioner's Handbook for Real-time Analysis – Guide to Rate Monotonic Analysis for Real-Time Systems*. Kluwer Academic Publishers, Norwell, MA, USA, 1993.

[7] M. Joseph and P. Pandya. Finding response times in a real-time system. *The Computer Journal*, 29(5):390–395, October 1986.

[8] J. Lehoczky, L. Sha, and Y. Ding. The rate monotonic scheduling algorithm: Exact characterization and average case behavior. In *Proceedings of the 10th Real-Time Systems Symposium (RTSS)*, pages 166–171, Santa Monica, California, USA, December 1989. IEEE Computer Society Press.

[9] Sergey Bozhko and Björn B. Brandenburg. Abstract Response-Time Analysis: A Formal Foundation for the Busy-Window Principle. In Marcus Völp, editor, *Proceedings of the 32nd Euromicro Conference on Real-Time Systems (ECRTS)*, volume 165 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 22:1–22:24, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[10] Felipe Cerqueira, Felix Stutz, and Björn B. Brandenburg. PROSA: A case for readable mechanized schedulability analysis. In *Proceedings of the 28th Euromicro Conference on Real-Time Systems (ECRTS)*, pages 273–284, 2016.

[11] S. Baruah, A. Mok, and L. Rosier. Preemptively scheduling hard-real-time sporadic tasks on one processor. In *Proceedings of the 11th Real-Time Systems Symposium (RTSS)*, pages 182–190, Orlando, Florida, 1990. IEEE Computer Society Press.

[12] Friedrich Eisenbrand and Thomas Rothvoß. EDF-schedulability of synchronous periodic task systems is coNP-hard. In *Proceedings of the Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, January 2010.

[13] P. Ekberg and W. Yi. Uniprocessor feasibility of sporadic tasks with constrained deadlines is strongly coNP-complete. In *Proceedings of the 27th Euromicro Conference on Real-Time Systems (ECRTS)*, pages 281–286, 2015.

[14] P. Ekberg and W. Yi. Uniprocessor feasibility of sporadic tasks remains coNP-complete under bounded utilization. In *Proceedings of the 36th Real-Time Systems Symposium (RTSS)*, pages 87–95, 2015.

[15] Pontus Ekberg and Wang Yi. Fixed-priority schedulability of sporadic tasks on uniprocessors is NP-hard. In *Proceedings of the 38th Real-Time Systems Symposium (RTSS)*, pages 139–146. IEEE Computer Society, 2017.

[16] J.P. Lehoczky. Fixed priority scheduling of periodic task sets with arbitrary deadlines. In *Proceedings of the 11th Real-Time Systems Symposium (RTSS)*, pages 201–209, 1990.

[17] Marco Maida, Sergey Bozhko, and Björn B. Brandenburg. Foundational Response-Time Analysis as Explainable Evidence of Timeliness. In Martina Maggio, editor, *Proceedings of the 34th Euromicro Conference on Real-Time Systems (ECRTS)*, volume 231 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 19:1–19:25, Dagstuhl, Germany, 2022. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[18] Wei Zheng, Qi Zhu, Marco Di Natale, and Alberto Sangiovanni Vincentelli. Definition of task allocation and priority assignment in hard real-time distributed systems. In *Proceedings of the 28th Real-Time Systems Symposium (RTSS)*, RTSS '07, page 161–170, USA, 2007. IEEE Computer Society.

[19] Pontus Ekberg and Sanjoy Baruah. Partitioned scheduling of recurrent real-time tasks. In *Proceedings of the 42nd Real-Time Systems Symposium (RTSS)*, pages 356–367, 2021.

[20] Jose Fonseca, Vincent Nelis, Gurulingesh Raravi, and Luis Miguel Pinho. A Multi-DAG model for real-time parallel applications with conditional execution. In *Proceedings of the ACM/ SIGAPP Symposium on Applied Computing (SAC)*, Salamanca, Spain, April 2015. ACM Press.

[21] Sanjoy Baruah, Vincenzo Bonifaci, and Alberto Marchetti-Spaccamela. The global EDF scheduling of systems of conditional sporadic DAG tasks. In *Proceedings of the 26th Euromicro Conference on Real-Time Systems (ECRTS)*, ECRTS '15, pages 222–231, Lund (Sweden), 2015. IEEE Computer Society Press.

[22] Alessandra Melani, Marko Bertogna, Vincenzo Bonifaci, Alberto Marchetti-Spaccamela, and Giorgio Buttazzo. Response-time analysis of conditional DAG tasks in multiprocessor systems. In *Proceedings of the 26th Euromicro Conference on Real-Time Systems (ECRTS)*, ECRTS '15, pages 222–231, Lund (Sweden), 2015. IEEE Computer Society Press.

[23] Sanjoy Baruah and Alberto Marchetti-Spaccamela. Feasibility Analysis of Conditional DAG Tasks. In Björn B. Brandenburg, editor, *Proceedings of the 33rd Euromicro Conference on Real-Time Systems (ECRTS)*, volume 196 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 12:1–12:17, Dagstuhl, Germany, 2021. Schloss Dagstuhl – Leibniz-Zentrum für Informatik.

[24] V. Bonifaci, A. Marchetti-Spaccamela, N. Megow, and A. Wiese. Polynomial-time exact schedulability tests for harmonic real-time tasks. In *Proceedings of the 34th Real-Time Systems Symposium (RTSS)*, pages 236–245, Dec 2013.

[25] Thi Huyen Chau Nguyen, Werner Grass, and Klaus Jansen. Exact polynomial time algorithm for the response time analysis of harmonic tasks. In Cristina Bazgan and Henning Fernau, editors, *Combinatorial Algorithms*, pages 451–465, Cham, 2022. Springer International Publishing.