Optimizing over FP/EDF Execution Times: Known Results and Open Problems

Enrico Bini, University of Turin, Italy

***Abstract*—In many use cases the execution time of tasks is unknown and can be chosen by the designer to increase or decrease the application features depending on the availability of processing capacity. If the application has real-time constraints, such as deadlines, then the necessary and sufficient schedulability test must allow the execution times to be left unspecified. By doing so, the designer can then perform optimization of the execution times by picking the schedulable values that minimize any given cost.**

In this paper, we review existing results on the formulation of both the Fixed Priority and Earliest Deadline First exact schedulability constraints. The reviewed formulations are expressed by a combination of linear constraints, which enables then optimization routines.

I. Introduction

The necessity to trade the accuracy of applications with the available processing capacity is common in many application domains. Some notable examples are the MPEG decoding which may be made at different degrees of detail, or the solution of optimization problems which may be solved with different proximity to the optimal solution. Notably, the workload of inference in neural network also belongs to this class as one may choose the desired accuracy of the answer given by a model.

If the application has real-time constrains, then schedulability must be taken into account. Schedulability tests, however, may not lend themselves to this type of problem as they normally need to know the execution times and they provide a "yes/no" answer to the schedulability questions, which is unfit for optimization. Instead, schedulability conditions which are formulated as combination of algebraic constraints between the parameters are better suited to contexts in which some of the parameters are unknown and are free to choose by the designer.

In this paper, we review some existing results on this form of constraints when the execution times are the free variables. We address both Fixed Priority (FP) and Earliest Deadline First (EDF) schedulability conditions and we present some open problems in this context.

II. System Model

We consider a set $\mathcal{T} = \{\tau_1, \ldots, \tau_n\}$ of n periodic *tasks*. Each task τ_i is characterized by

- a worst-case execution time C_i (which may be called *execution time* for simplicity),
- a *period* T_i, and
- a relative *deadline* D_i not greater than T_i (*constrained deadline* model).

All task parameters are assumed real-valued.

In the single processor context, which is addressed in this paper, this same model also captures tasks with sporadic releases. In such a case, T_i denotes the minimum interarrival time between two consecutive jobs.

Each task releases an infinite sequence of *jobs*. Jobs are indexed in $\mathbb{N}$ by the order of release. We assume $0 \in \mathbb{N}$. The *release time* of the j-th job of τ_i is denoted by $r_{i,j}$ and releases of consecutive jobs are constrained by

$$\forall j \in \mathbb{N}, \quad r_{i,j+1} \geq r_{i,j} + T_i. \tag{1}$$

Each job must also complete not later that its *absolute deadline* $d_{i,j}$, which is set by

$$d_{i,j} = r_{i,j} + D_i. \tag{2}$$

Finally, we use $U_i = \frac{C_i}{T_i}$ to denote τ_i's *utilization* as indeed, it is the fraction of CPU time utilized by τ_i.

For a more compact notation, we may be using

- $\mathbf{C} = [C_1, \ldots, C_n]$ to denote the vector of all execution times, and
- $\mathbf{U} = [U_1, \ldots, U_n]$ to denote the vector of utilizations.

We address single processor preemptive scheduling. This means that the scheduler may decide to preempt a running job to schedule another higher priority job. The interrupted job will then be continued later.

This paper considers only Fixed Priority (FP) and Earliest Deadline First (EDF) schedulers, in Sections III and IV, respectively. As proved by Liu and Layland [1], single processor preemptive FP and EDF have the following worst-case scenario for the job releases

$$\forall j \in \mathbb{N}, \quad r_{i,j} = j\, T_i, \tag{3}$$

which corresponds to all tasks starting to release jobs simultaneously and at the fastest rate. This means that if the jobs generated by the tasks are schedulable when released according to (3), then they are always schedulable for any releases fulfilling (1). From now on, we are then assuming the worst-case scenario of (3).

III. Fixed Priority

When tasks are scheduled by Fixed Priority (FP), we assume they are indexed by decreasing priority, that is τ_ℓ has priority higher than τ_i if and only if $\ell < i$.

For the purpose of optimizing over task execution time, it is convenient to borrow the exact schedulability condition, as formulated by Lehoczky et al. [2].

Theorem 1 (from [2]): A periodic task set $\mathcal{T}$ is schedulable under Fixed Priority **if and only if**

$$\forall i = 1, \ldots, n \quad \exists t \in [0, D_i] \quad C_i + \sum_{\ell=1}^{i-1} \left\lceil \frac{t}{T_\ell} \right\rceil C_\ell \leq t. \tag{4}$$

By using the more compact vector notation, the Eq. (4) can be rewritten as

$$\forall i = 1, \ldots, n \quad \exists t \in [0, D_i] \quad \mathbf{k}_i(t) \cdot \mathbf{C} \leq t \tag{5}$$

where

$$\mathbf{k}_i(t) = \Bigg(\overbrace{\left\lceil \frac{t}{T_1} \right\rceil, \left\lceil \frac{t}{T_2} \right\rceil, \ldots, \left\lceil \frac{t}{T_{i-1}} \right\rceil}^{\text{from } 0 \text{ to } i-1}, 1, \overbrace{0, \ldots, 0}^{\text{from } i+1 \text{ to } n} \Bigg).$$

Testing if Eq. (5) is true for any t in the dense interval $[0, D_i]$ is not practically feasible. In fact with elementary considerations Lehoczky suggested the equivalent

$$\bigwedge_{i=1}^{n} \bigvee_{t \in \mathcal{S}_i} \mathbf{k}_i(t) \cdot \mathbf{C} \leq t \tag{6}$$

with $\mathcal{S}_i$ being the discrete and finite set defined by

$$\mathcal{S}_i = \{ j\, T_\ell : 1 \leq \ell < i,\ j\, T_\ell \leq D_i \} \cup \{ D_i \}. \tag{7}$$

The set $\mathcal{S}_i$ contains all the release instants $r_{j,\ell}$ of any task τ_ℓ with priority higher than τ_i, with $r_{j,\ell} \leq D_i$, plus the deadline D_i of the task τ_i itself.

In Equation (6) we expressed the same condition of (5) through the logical AND/OR operator instead of the propositional operators $\forall/\exists$. This makes more clear that the space of FP-schedulable execution times is the intersection of unions of the half-spaces $\{\mathbf{k}_i(t) \cdot \mathbf{C} \leq t\}$ in space of execution times $\mathbf{C} \in \mathbb{R}^n$.

The challenge in a direct exploitation of (6) for the optimization over the execution times is that the cardinality of the set $\mathcal{S}_i$ of (7) may grow significantly as the periods of high priority tasks gets smaller and smaller w.r.t. the deadline D_i. This issue was then addressed [3], [4]. It was demonstrated that **if the priorities are Rate Monotonic**, then a task set is schedulable by RM, if and only if

$$\bigwedge_{i=1}^{n} \bigvee_{t \in \mathcal{P}_{i-1}(D_i)} \mathbf{k}_i(t) \cdot \mathbf{C} \leq t \tag{8}$$

with $\mathcal{P}_i(t)$ generically defined by

$$\begin{cases} \mathcal{P}_0(t) = \{t\} \\ \mathcal{P}_i(t) = \mathcal{P}_{i-1}\left(\left\lfloor \frac{t}{T_i} \right\rfloor T_i \right) \cup \mathcal{P}_{i-1}(t). \end{cases} \tag{9}$$

Figure 1 shows an example of computation of the two sets. In the example with $i = 3$, it can be observed that the points in $\mathcal{P}_2(D_3)$ are 4. Instead, the number of points in Lehoczky's $\mathcal{S}_3$ are 10. In general, the number of necessary and sufficient points that need to be tested through (9) is constant with a given number of tasks, whereas the number of points from (7) may grow arbitrarily large with the task set parameters. Figure 2 illustrates the space of RM schedulable execution times for an example with 2 tasks.

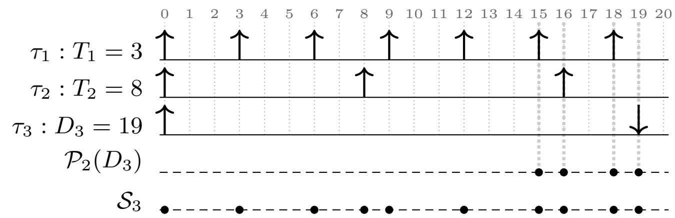

Fig. 1. An example of the schedulability points $\mathcal{S}_3$ and $\mathcal{P}_2(D_3)$ for a set of 3 tasks with $T_1 = 3$, $T_2 = 8$, and $D_3 = 19$. Notice that the set of points does not depend on the execution times.

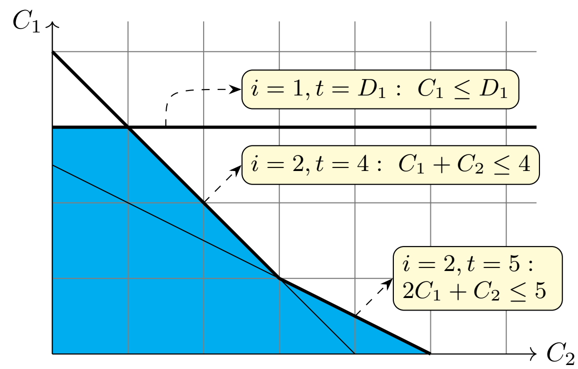

Fig. 2. Region of RM schedulable execution times. We draw C_2 along the horizontal axis and C_1 along the vertical axis. In this example, we assume $n = 2$ tasks and parameters: $T_1 = 4$, $D_1 = 3$, and $D_2 = 5$ and any $T_2 \geq D_2$. From (8), when $i = 1$ we have $\mathcal{P}_0(D_1) = \{D_1\} = \{3\}$ which gives $C_1 \leq 3$ as the only (and trivial) necessary and sufficient constraint to guarantee the schedulability of τ_1, the highest priority task. When $i = 2$ the schedulability points are $\mathcal{P}_2(D_2) = \{5, 4\}$, which yield the constraints $2C_1 + C_2 \leq 5$ and $C_1 + C_2 \leq 4$, respectively, both represented by thin lines. We need to make union among these constraints, as required by the logical OR of (8), thus getting the two oblique thick segments. Since the overall schedulability region is given by the intersection of the single-task regions, we find that the RM-schedulable execution times are the ones represented in the cyan area.

A. *Open problems*

In this section we sketch some problems which, to best of our knowledge, are open.

Non-DM priorities: The reduction of schedulability points of Eq. (9) can be made only when priorities are DM/RM [3], [4]. The proof does exploit the fact that higher priority tasks have a smaller period than the one under analysis. Is a construction similar to the one of (9) applicable to generic non-DM/RM priorities? In some preliminary experiments [5], it was shown a counter-example of a non-DM tasks' set with a schedulability point **not in** the set of (9). However, this investigation was not continued any further and the counter-example is lost.

Tightness of the points: The set of points determined by (9) is certainly smaller than the original Lehoczky's set of (7). It remains an open question if the reduced set of points can be further reduced:

- without exploiting information on the execution times, and
- keeping the set as necessary and sufficient condition.

Arbitrary deadlines: The direct extension of (8) to the arbitrary deadline case would be

$$\bigwedge_{i=1}^{n} \bigwedge_{j=0}^{\mathsf{lastbusy}(i)} \bigvee_{t\in\mathcal{P}_{i-1}(j\,T_i+D_i)} \mathbf{k}_i(t,j)\cdot\mathbf{C}\le t \tag{10}$$

with $\mathbf{k}_i(t,j)$ defined by

$$\mathbf{k}_i(t,j) = \Bigg(\overbrace{\left\lceil\frac{t}{T_1}\right\rceil, \left\lceil\frac{t}{T_2}\right\rceil, \ldots, \left\lceil\frac{t}{T_{i-1}}\right\rceil}^{\text{from } 0 \text{ to } i-1}, j{+}1, \overbrace{0,\ldots,0}^{\text{from } i+1 \text{ to } n}\Bigg).$$

properly extended to account the job j of τ_i. Also, in Eq. (10), $\mathsf{lastbusy}(i)$ denotes the index of the last τ_i job in the level-i busy interval [6]. The formulation of (10), however, poses a few challenges with no answer:

- is there any redundancy among the many schedulability points in $\mathcal{P}_{i-1}(j\,T_i + D_i)$ as j spans from 0 to $\mathsf{lastbusy}(i)$?
- since the execution times $\mathbf{C}$ are unknown, how long is the level-i busy interval? In the special case with $\sum_i U_i = 1$ and no hyperperiod H (irrational periods), which implies that every instant is level-n busy and then $\mathsf{lastbusy}(n) \to \infty$, how can we test the schedulability of τ_n?

IV. Earliest Deadline First

In this section, we illustrate some results following a similar investigation for the EDF scheduling policy. Also, we remark that in this section, we relax the constrained deadline case and allow deadlines to be arbitrary (possibly larger than the period of the corresponding task). Also we denote by H the *hyperperiod*, which is the least common multiple among all task periods $\{T_1,\ldots,T_n\}$. Observe that in our initial hypothesis we assumed all parameters to be **real-valued**. Hence, we assume that the hyperperiod H exists (as it normally happens in reality) and postpone the curious case of a non-existent hyperperiod H to Section IV-A for a related open problem.

If the n tasks in $\mathcal{T}$ are scheduled by preemptive EDF, the following condition is necessary and sufficient to ensure that no job deadline is missed.

Theorem 2 (Corollary 1 in [7]): The task set $\mathcal{T}$ is scheduled by preemptive EDF **if and only if**

$$\sum_{i=1}^{n} U_i \le 1, \tag{11}$$

and

$$\forall t\in\mathbb{N}, 0\le t\le H+\max_i\{D_i\},$$
$$\underbrace{\sum_{i=1}^{n}\max\left\{0, \left\lfloor\frac{t-D_i}{T_i}\right\rfloor+1\right\}C_i}_{\mathsf{dbf}(t)} \le t \tag{12}$$

The LHS of Eq. (12) is a very frequently used function in real-time EDF scheduling, and it is called *demand bound function* $\mathsf{dbf}(t)$. Since $\mathsf{dbf}(t)$ is piecewise constant, the inequality of Eq. (12) needs to be checked only at the instants t when the $\mathsf{dbf}(t)$ changes value, which are all the absolute deadlines of any job. Hence, the exact condition of Theorem 2 can be simplified as stated in next corollary.

Corollary 1: The task set $\mathcal{T}$ is scheduled by preemptive EDF **if and only if**

$$\forall t\in\mathcal{D},\quad \mathbf{h}(t)\cdot\mathbf{C}\le 1 \tag{13}$$

with

$$\mathcal{D} = \left\{d_{i,j} = j\,T_i + D_i : d_{i,j}\le H+\max_i\{D_i\}\right\}\cup\{0\} \tag{14}$$

and $\mathbf{h}(t) = [h_1(t),\ldots,h_n(t)]$,

$$h_i(t) = \begin{cases} \frac{1}{T_i} & \text{if } t=0 \\ \frac{1}{t}\max\left\{0, \left\lfloor\frac{t-D_i}{T_i}\right\rfloor+1\right\} & \text{otherwise}\end{cases} \tag{15}$$

In the definition of $\mathcal{D}$ of Eq. (14), we use the fictitious "deadline at 0" to encode the utilization constraint of (11) through the special definition of $h_i(0) = \frac{1}{T_i}$.

The necessary and sufficient schedulability condition of (13) already gives some information on the geometry of the EDF-schedulable computation times. In fact, the space of EDF-schedulable execution times is **convex** as it is the intersection between linear halfspaces yielded by (13). Figure 4 illustrates the EDF-schedulable execution times of the same simple example of Figure 2.

As Figure 4 seems to suggest, there may be some deadlines in $\mathcal{D}$ which are not necessary to be checked, as implied by other constraints. Most of the methods to reduce the number of deadlines to be checked [7], [8], [9], and then the complexity of the test exploit the values of the execution times with the general trend that "the smaller the utilization $\sum_i U_i$ of the whole set of tasks, the fewer deadlines are necessary". However, the question of whether some deadlines can be eliminated from $\mathcal{D}$ without compromising the schedulability and without exploiting the execution times received little attention.

In a recent work [10], it was proposed a method that exploits the convex hull of vectors to determine the smallest subset of deadlines $\mathcal{D}_{\mathsf{min}} \subseteq \mathcal{D}$ such that

$$\forall t\in\mathcal{D}_{\mathsf{min}},\quad \mathbf{h}(t)\cdot\mathbf{C}\le 1. \tag{16}$$

On the one hand this method shows that the number of necessary and sufficient linear constraints for EDF schedulability is orders of magnitude less than the full set of (13). See Figure 3 for an example. On the other hand it works as a "black box" algorithm producing the minimal set of constraint, providing no insight on why these few deadlines only matters. Also, the complexity of computing the convex hull of a large set of vectors is very high.

In Figures 5, 6 and 7 we experimentally investigate the dependency of the number of necessary and sufficient constraints on the hyperperiod H. In red, we draw the upper envelope, which then represents the hardest cases to be analyzed. Despite the linear growth of $|\mathcal{D}|$ with H as implied by (14), the number of necessary and sufficient constraints of the hardest cases grows with $\log H$. If this experimental evidence becomes a

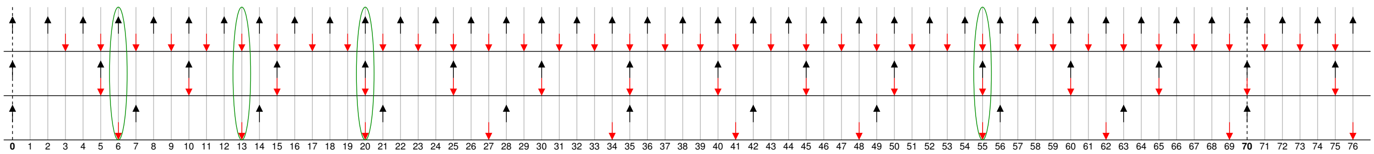

Fig. 3. The tight set of necessary and sufficient constraints for EDF. In this example, the parameters are $T_1 = 2, D_2 = 3$, $T_2 = 5, D_2 = 5$, and $T_3 = 7, D_3 = 6$. Job releases are represented by upward black arrows. Job deadlines are represented by downward red arrows. We represent the deadlines until $H + \max\{D_i\} = 70 + 6 = 76$, as required by (14). The number of total constraints to be checked is 49 corresponding to 48 deadlines plus the utilization constraint of (11). The reduced number of constraints, however, is only 5: 4 deadlines (circled in green) plus the utilization constraint.

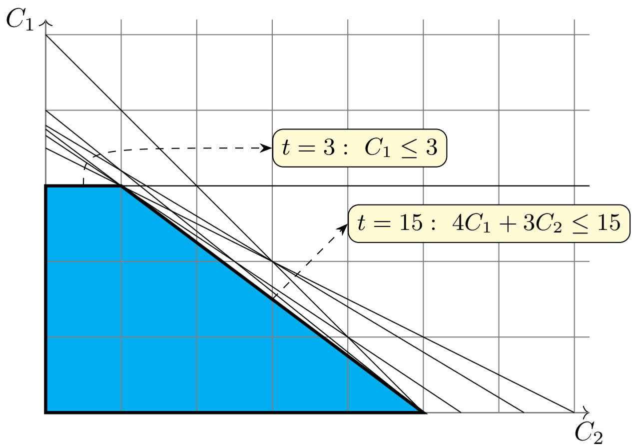

Fig. 4. Region of EDF schedulable execution times. As in Figure 2, we draw C_2 along the x axis and C_1 along the y axis. Using the same example of Figure 2, we assume 2 tasks with parameters: $T_1 = 4$, $D_1 = 3$, and $T_2 = D_2 = 5$. In this case, the set $\mathcal{D}$ from (14) of all deadlines to be considered is $\mathcal{D} = \{3, 5, 7, 10, 11, 15, 19, 0\}$. We remind that the "deadline 0" represents the utilization constraint of (11). All constraints are represented by a thin line, whereas the boundary of their intersection is represented by a thicker line. It can be observed that a large majority of the constraints does not contribute to determine the boundary of EDF-schedulable execution times. In this example, only the 2 deadlines at 3 and at 15 are needed to characterize the exact region.

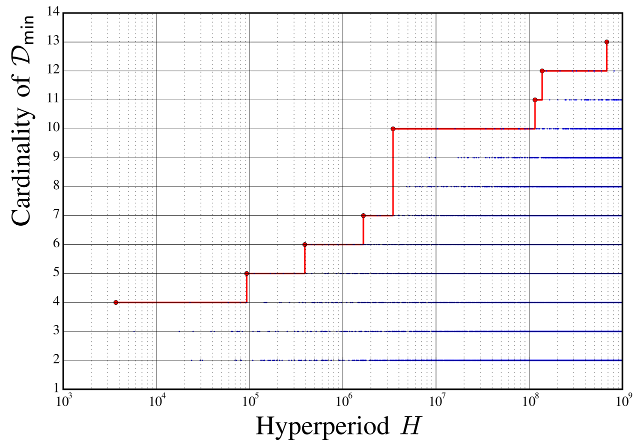

Fig. 5. In this experiment, randomly generated integer task period and deadlines for $n = 2$ tasks. For each experiment, we are plotting a dot at the corresponding hyperperiod H along the horizontal axis in log scale, and the number of necessary and sufficient constraints along the vertical axis. At the right, also the density of the minimal number of constraints over the sample space. We also plot the upper envelope of the points as such an envelope represents the hardest instances to be tested.

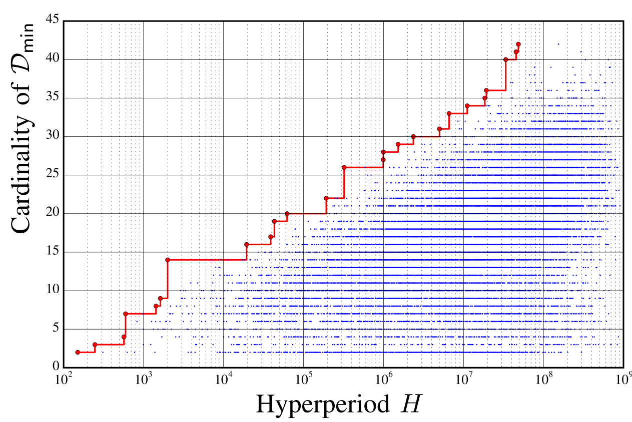

Fig. 6. Same experiment of Figure 5 with $n = 3$ tasks.

confirmed fact, it may allow the existence of an exact EDF test which is polynomial in the task periods [11], deferring then the harder complexity to the number n of tasks only. This seems not to contradict any existing result.

A. Open problems

Non-existent hyperperiod H: If the task periods are real-valued, indeed their least common multiple, the hyperperiod H, may not exist. In this case, the number of deadlines to be tested is infinite. Also, the experiments of Figures 5, 6, and 7 indicates that as H grows, the number of necessary and sufficient constraints grows (logarithmically) with H. Some existing EDF sufficient tests [12], [13], [14] do not require the existence of an hyperperiod. Perhaps, some of them become exact as H tends to infinity? Anyhow, how to test EDF schedulability with no hyperperiod H is unknown.

Other approaches to minimal set of constraints: The employment of the convex hull to determine the tight set of points [10] is indeed very complex. Is there any logic behind the points selected by the convex hull? Why are deadlines at 6, 13, 20, and 55 in the example of Figure 3 so special? If such a logic is found, then we could go straight to these constraints, with no complex machinery as the convex hull. Also, it may be possible that such an algorithm is polynomial in the periods as it grows with $\log(H)$.

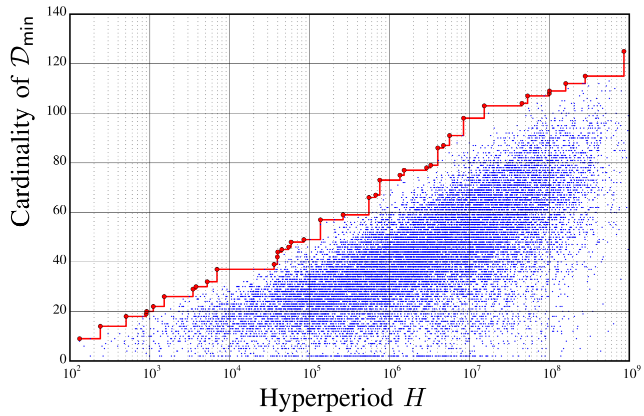

Fig. 7. Same experiment of Figure 5 with $n = 4$ tasks.

V. Related works

The research touched by this paper covers a very broad spectrum. As such it is surely incomplete. Next we report our best attempt to cover the related literature.

In Fixed Priority, the response time analysis (RTA) [15], [16], [17] is indeed very widely used. Many works have addressed the efficiency of the RTA by proposing a later initial instant for the iterations. In same cases, such initial instant depends on the execution times, hence it is unsuitable for optimization [18], [19]. Other works have proposed an initial start instant for RTA, which is independent of the tasks' execution times [20], [21]. Either way, the iterative nature of RTA makes it unfit for optimization unless costly binary search is employed [22].

If the periods have some good harmonic properties [23], it is possible to discard some of the points in (9) as shown by Zeng and Di Natale [24]. However, in the general case of periods not dividing each other, it is unknown if the same simplification is possible.

In the context of optimization of task parameters, the task model with imprecise computation [25], [26] was perhaps among the first ones to allow tasks to have a variable execution time. Reward-based scheduling was also a very good method to decide the duration of each individual job in EDF, assuming that the longer a job executed the more "reward" is accumulated [27].

Exploiting the set of reduced schedulability points [3], [4], Bini et al. [28] proposed to perform the sensitivity analysis on the task parameters, providing a closed-form expression for the acceptable margins for the execution times.

The elastic task model [29], [13] was also introduced as a way to adjust the parameters (the task periods) while preserving schedulability.

The reduction of constraints for EDF schedulability was also addressed by George and Hermant [30]. They proposed to solve an instance of a LP problem for each absolute deadline. As shown in their paper, however, their method is not capable to automatically cut all unnecessary deadlines.

The investigation of the case of irrational periods received, with no surprise, little attention from the research community. To best of our knowledge, the only known partial result is the computation of the task response time as supremum of the response time among all jobs [31].

Finally, it is worth mentioning the work by Singh [32], who proposed an interesting unification between FP and EDF schedulability tests.

Thanksgiving: The author would like to warmly thank the reviewers for the time spent in making a careful and detailed review of this paper.

Acknowledgment: This work is partially supported by the project "Trustworthy Cyber-Physical Pipelines", funded by the MAECI Italy-Sweden co-operation id. PGR02086, and the spoke "FutureHPC and BigData" of the ICSC — Centro Nazionale di Ricerca in High-Performance Computing, Big Data and Quantum Computing funded by European Union — NextGenerationEU.

References

[1] C. L. Liu and J. W. Layland, "Scheduling algorithms for multiprogramming in a hard real-time environment," *Journal of the Association for Computing Machinery*, vol. 20, no. 1, pp. 46–61, Jan. 1973.

[2] J. P. Lehoczky, L. Sha, and Y. Ding, "The rate-monotonic scheduling algorithm: Exact characterization and average case behavior," in *Proceedings of the* 10^{th} *IEEE Real-Time Systems Symposium*, Santa Monica (CA), U.S.A., Dec. 1989, pp. 166–171.

[3] Y. Manabe and S. Aoyagi, "A feasibility decision algorithm for rate monotonic and deadline monotonic scheduling," *Real-Time Systems*, vol. 14, no. 2, pp. 171–181, Mar. 1998.

[4] E. Bini and G. C. Buttazzo, "Schedulability analysis of periodic fixed priority systems," *IEEE Transactions on Computers*, vol. 53, no. 11, pp. 1462–1473, Nov. 2004.

[5] G. Yao, "A counter-example for the schedulability points of [3], [4], with non-DM priorities," Personal communication in Pisa, 2009.

[6] J. P. Lehoczky, "Fixed priority scheduling of periodic task sets with arbitrary deadline," in *Proceedings of the 11th IEEE Real-Time Systems Symposium*, Lake Buena Vista (FL), U.S.A., Dec. 1990, pp. 201–209.

[7] S. K. Baruah, A. K. Mok, and L. E. Rosier, "Preemptively scheduling hard-real-time sporadic tasks on one processor," in *Proceedings of the 11th IEEE Real-Time Systems Symposium*, Lake Buena Vista (FL), USA, Dec. 1990, pp. 182–190.

[8] I. Ripoll, A. Crespo, and A. K. Mok, "Improvement in feasibility testing for real-time tasks," *Real-Time Systems*, vol. 11, no. 1, pp. 19–39, 1996.

[9] F. Zhang and A. Burns, "Schedulability analysis for real-time systems with edf scheduling," *IEEE Transactions on Computers*, vol. 58, no. 9, pp. 1250–1258, Sep. 2009.

[10] E. Bini, "Cutting the unnecessary deadlines in EDF," in *2019 IEEE 25th International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*. IEEE, 2019, pp. 1–8.

[11] P. Ekberg, "If the number of necessary and sufficient constraints grows with $\log H$, then they are polynomial in the task periods," Personal communication in Uppsala, 2019.

[12] U. C. Devi, "An improved schedulability test for uniprocessor periodic task systems," in *Proceedings* 15^{th} *Euromicro Conference on Real-Time Systems*, Jul. 2003, pp. 23–30.

[13] T. Chantem, X. S. Hu, and M. D. Lemmon, "Generalized elastic scheduling," in *Proceedings of the* 27^{th} *IEEE Real-Time Systems Symposium*, Rio de Janeiro, Brazil, Dec. 2006, pp. 236–245.

[14] E. Bini and G. C. Buttazzo, "The space of EDF deadlines; the exact region and a convex approximation," *Real-Time Systems*, vol. 41, pp. 27–51, 2009.

[15] M. Joseph and P. K. Pandya, "Finding response times in a real-time system," *The Computer Journal*, vol. 29, no. 5, pp. 390–395, Oct. 1986.

[16] P. K. Harter, "Response times in level-structured systems," *ACM Transactions on Computer Systems*, vol. 5, no. 3, pp. 232–248, Aug. 1987.

[17] N. C. Audsley, A. Burns, M. Richardson, K. W. Tindell, and A. J. Wellings, "Applying new scheduling theory to static priority pre-emptive scheduling," *Software Engineering Journal*, vol. 8, no. 5, pp. 284–292, Sep. 1993.

[18] M. Sjödin and H. Hansson, "Improved response-time analysis calculations," in *Proceedings of the* 19^{th} *IEEE Real-Time Systems Symposium*, Madrid, Spain, Dec. 1998, pp. 399–408.

[19] R. J. Bril, W. F. J. Verhaegh, and E.-J. D. Pol, "Initial values for online response time calculations," in *Proceedings of the* 15^{th} *Euromicro Conference on Real-Time Systems*, Porto, Portugal, Jul. 2003, pp. 13–22.

[20] W.-C. Lu, K.-J. Lin, H.-W. Wei, and W.-K. Shih, "Period-dependent initial values for exact schedulability test of rate monotonic systems," in *2007 IEEE International Parallel and Distributed Processing Symposium*. IEEE, 2007, pp. 1–8.

[21] R. I. Davis, A. Zabos, and A. Burns, "Efficient exact schedulability tests for fixed priority real-time systems," *IEEE Transactions on Computers*, vol. 57, no. 9, pp. 1261–1276, 2008.

[22] S. Punnekkat, R. Davis, and A. Burns, "Sensitivity analysis of real-time task sets," in *Proceedings of the 3rd Asian Computing Science Conference on Advances in Computing Science*, Kathmandu, Nepal, Dec. 1997, pp. 72–82.

[23] T.-W. Kuo and A. K. Mok, "Load adjustment in adaptive real-time systems," in *Proceedings of the* 12^{th} *IEEE Real-Time Systems Symposium*, San Antonio (TX), U.S.A., Dec. 1991, pp. 160–170.

[24] H. Zeng and M. Di Natale, "An efficient formulation of the real-time feasibility region for design optimization," *IEEE Transactions on Computers*, vol. 62, no. 4, pp. 644–661, 2013.

[25] W.-K. Shih, J. W. S. Liu, and J.-Y. Chung, "Fast algorithms for scheduling imprecise calculations," in *Proceedings of the 10th IEEE Real-Time System Symposium*, Santa Monica (CA), USA, Dec. 1989, pp. 12–19.

[26] J.-Y. Chung, J. W. Liu, and K.-J. Lin, "Scheduling periodic jobs that allow imprecise results," *IEEE Transactions on Computers*, vol. 39, no. 9, pp. 1156–1174, Sep. 1990.

[27] H. Aydin, R. Melhem, D. Mossé, and P. Mejía-Alarez, "Optimal reward-based scheduling for periodic real-time tasks," *IEEE Transactions on Computers*, vol. 50, no. 2, pp. 111–130, Feb. 2001.

[28] E. Bini, M. D. Natale, and G. Buttazzo, "Sensitivity analysis for fixed-priority real-time systems," *Real-Time Systems*, vol. 39, no. 1–3, pp. 5–30, 2008.

[29] G. C. Buttazzo, G. Lipari, M. Caccamo, and L. Abeni, "Elastic scheduling for flexible workload management," *IEEE Transactions on Computers*, vol. 51, no. 3, pp. 289–302, Mar. 2002.

[30] L. George and J.-F. Hermant, "Characterization of the space of feasible worst-case execution times for earliest-deadline-first scheduling," *Journal of Aerospace Computing, Information, and Communication*, vol. 6, no. 11, pp. 604–623, 2009.

[31] E. Bini, A. Parri, and G. Dossena, "A quadratic-time response time upper bound with a tightness property," in *Proceedings of the 36th Real-Time Systems Symposium*, San Antonio, TX, USA, Dec. 2015.

[32] A. Singh, "Cutting-plane algorithms for preemptive uniprocessor scheduling problems," *Real-Time Systems*, vol. 60, no. 1, pp. 24–73, 2024.