

# EDF-schedulability of synchronous periodic task systems is coNP-hard

Friedrich Eisenbrand    Thomas Rothvoß

*Institute of Mathematics  
EPFL, Lausanne, Switzerland  
{friedrich.eisenbrand,thomas.rothvoss}@epfl.ch*

## Abstract

In the *synchronous periodic task model*, a set  $\tau_1, \dots, \tau_n$  of tasks is given, each releasing jobs of running time  $c_i$  with relative deadline  $d_i$ , at each integer multiple of the period  $p_i$ . It is a classical result that *Earliest Deadline First (EDF)* is an optimal preemptive uni-processor scheduling policy. For constrained deadlines, i.e.  $d_i \leq p_i$ , the EDF-schedule is feasible if and only if

$$\forall Q \geq 0 : \sum_{i=1}^n \left( \left\lfloor \frac{Q - d_i}{p_i} \right\rfloor + 1 \right) \cdot c_i \leq Q.$$

Though an enormous amount of literature deals with this topic, the complexity status of this test has remained unknown. We prove that testing EDF-schedulability of such a task system is (weakly) **coNP**-hard. This solves Problem 2 from the survey “Open Problems in Real-time Scheduling” by Baruah & Pruhs. The hardness result is achieved by applying recent results on inapproximability of Diophantine approximation.

## 1 Introduction

Nowadays more and more devices are controlled by embedded microprocessors, for example in power plants, car electronics, flight control systems, robotics and telecommunication systems, see Buttazzo [8] for an extensive introduction. Since many applications are safety critical, each task running on such a processor must produce the output not only correctly but also on time. Several tasks may run on the same processor and a *Real-time scheduling policy* decides which task should be active in which intervals, to guarantee that all deadlines are kept.

In the simple, but important *periodic task model* a set  $\tau_1, \dots, \tau_n$  of tasks is given, where each  $\tau_i$  is an infinite sequence of jobs, defined by an *execution time*  $c_i \in \mathbb{Q}_+$ , a (relative) *deadline*  $d_i \in \mathbb{Q}_+$  and a *period*  $p_i \in \mathbb{Q}_+$ . We assume that the tasks are *synchronous*, i.e. there is a time, say 0, at which all tasks release a job simultaneously. In other words for each  $i \in \{1, \dots, n\}$

and  $z \in \mathbb{Z}_{\geq 0}$ , a job of running time  $c_i$  and absolute deadline  $z \cdot p_i + d_i$  is released at  $z \cdot p_i$ . Furthermore we assume *constrained-deadlines*, hence  $d_i \leq p_i$  for each  $i \in \{1, \dots, n\}$ .

We consider *preemptive* uni-processor schedules, i.e. at any time a running job may be preempted and resumed later. As the name suggests, in the *Earliest Deadline First (EDF)* policy, at any time that job from the queue of released and not yet accomplished jobs is active, whose (absolute) deadline comes next. The EDF-schedule is provably optimal in this setting, meaning that if there is a schedule in which all jobs meet their deadlines, then the EDF-schedule is feasible as well (see Dertouzos [11]).

The main question of feasibility analysis however remains: Will each of the infinitely many jobs be finished in time? First observe, that

$$\left\lfloor \frac{Q - d_i}{p_i} \right\rfloor + 1$$

is the number of jobs of  $\tau_i$  that have both, their release time and deadline in the interval  $[0, Q]$ . Consequently the quantity

$$\text{DBF}(\tau_i, Q) = \left( \left\lfloor \frac{Q - d_i}{p_i} \right\rfloor + 1 \right) \cdot c_i$$

is the amount of running time that, regardless of the used scheduling policy, has to be spent on  $\tau_i$  in this interval. More general, the *demand bound function*

$$\text{DBF}(\mathcal{S}, Q) = \sum_{i=1}^n \left( \left\lfloor \frac{Q - d_i}{p_i} \right\rfloor + 1 \right) \cdot c_i$$

gives the running time of *all* jobs, which have their release time and deadline in the interval  $[0, Q]$ . As a consequence, for feasibility it is necessary, that  $\text{DBF}(\mathcal{S}, Q) \leq Q$  for all  $Q \geq 0$ . Baruah et al. [5] showed that this condition is in fact sufficient, hence an *EDF-schedulability test* is a test which checks validity of the following for-

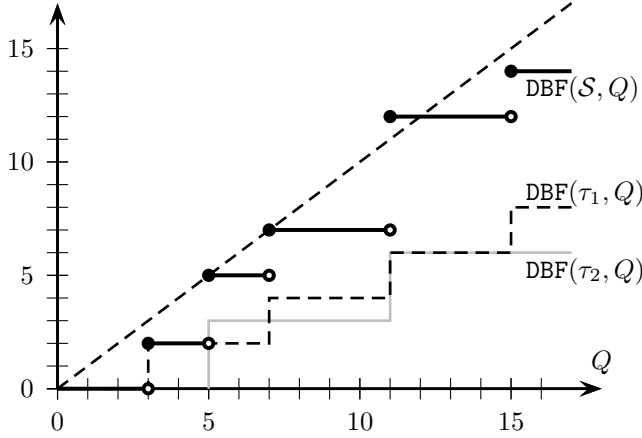


Figure 1: Constrained deadline task system  $\mathcal{S} = \{\tau_1, \tau_2\}$  with  $\tau_1 = (2, 3, 4)$ ,  $\tau_2 = (3, 5, 6)$ , using notation  $\tau_i = (c_i, d_i, p_i)$ . One has  $\text{DBF}(\mathcal{S}, Q) > Q$  for  $Q = 11$ , thus  $\mathcal{S}$  is not EDF-schedulable.

mula

$$\forall Q \geq 0 : \sum_{i=1}^n \left( \left\lfloor \frac{Q - d_i}{p_i} \right\rfloor + 1 \right) \cdot c_i \leq Q,$$

see Figure 1 for an illustration.

Much effort has been spent on developing sufficient polynomial or exact pseudo-polynomial time tests for EDF-schedulability of periodic tasks, see [2, 3, 5, 9, 12]. But none of the algorithms suggested in these papers was able to decide EDF-schedulability on a unit speed processor correctly and in polynomial time for all instances. The question whether EDF-schedulability can be decided in polynomial time is stated as a major open problem in the survey of Baruah & Pruhs [6] on open problems in Real-time scheduling. We settle the complexity status of testing EDF-schedulability by proving the following theorem.

**THEOREM 1.1.** *Given a set  $\mathcal{S} = \{\tau_1, \dots, \tau_n\}$  of synchronous, periodic, constrained-deadline tasks defined by rational numbers  $0 \leq c_i \leq d_i \leq p_i$ , it is (weakly) **coNP**-hard to decide, whether  $\mathcal{S}$  is EDF-schedulable, i.e. testing the condition*

$$\forall Q \geq 0 : \sum_{i=1}^n \left( \left\lfloor \frac{Q - d_i}{p_i} \right\rfloor + 1 \right) \cdot c_i \leq Q,$$

*is (weakly) **coNP**-hard. This holds even if  $d_i = p_i$  for  $i = 1, \dots, n - 1$ .*

This, together with the result in [5] implies the following corollary.

**COROLLARY 1.1.** *Given a set  $\mathcal{S} = \{\tau_1, \dots, \tau_n\}$  of sporadic tasks with worst-case execution time  $c_i$ , relative*

*deadline  $d_i$  and minimum inter-arrival time  $p_i$  it is (weakly) **coNP**-hard to determine, whether the EDF-schedule of  $\mathcal{S}$  is feasible.*

**Related work.** One approach to obtain algorithms to test EDF-feasibility lies in bounding the interval, in which the demand bound function has to be evaluated. Let  $u = \sum_{i=1}^n \frac{c_i}{p_i}$  be the *utilization* of a task system. Given that  $\mathcal{S}$  is not EDF-schedulable, the smallest  $Q > 0$ , certifying the infeasibility must have

$$Q < \frac{u}{1 - u} \max_{i=1, \dots, n} \{p_i - d_i\},$$

see e.g. [4, 17]. This admits a pseudo-polynomial time algorithm for the feasibility test, if the utilization of  $\mathcal{S}$  is bounded by  $1 - \varepsilon$  for some constant  $\varepsilon > 0$ .

Albers & Slomka [1] gave an FPTAS for approximating the speed of a processor, needed to make the EDF-schedule of  $\mathcal{S}$  feasible. Their algorithm is also interpreted as follows. It either asserts that the tasks are feasible, or it asserts that the tasks are infeasible on a processor of speed  $1 - \varepsilon$ . A similar result was also provided in the setting of fixed priority scheduling [15]. See [8] for more details on fixed priority scheduling policies and [9, 2, 12, 21] for further approaches to feasibility analyzes of EDF-schedules. Recently, Bonifaci et al. [7] extended the result of Albers & Slomka to the case of multiprocessor scheduling with migration. The algorithm asserts that a set of tasks is feasible on  $m$  speed- $(2 - 1/m + \varepsilon)$  machines or infeasible on  $m$  speed-1 machines.

In a popular special case, the tasks have *implicit-deadlines*, i.e.  $d_i = p_i$  for all  $i$ . In that case the condition  $\text{DBF}(\mathcal{S}, Q) \leq Q$  has only to be evaluated at  $Q = \text{scm}(p_1, \dots, p_n)$  and the set is EDF-schedulable if and only if the utilization is bounded by 1, see Liu & Layland [19]. In other words, the EDF-schedulability in this special case is decidable in polynomial time. If the tasks may be asynchronous, i.e. each task has an offset  $a_i$ , such that jobs are released at  $z \cdot p_i + a_i$ , then testing the feasibility is strongly **coNP**-hard [18]. This even holds if the utilization of the system is bounded from above by an arbitrarily small constant.

In the *sporadic* task model neither release times nor running times are predetermined. There,  $c_i$  denotes the *worst-case execution time* and  $p_i$  denotes the *minimum inter-arrival time*. But the worst-case is attained in a *synchronous arrival sequence*, that is when all tasks release jobs at time 0, all jobs fully use the worst-case execution time  $c_i$  and jobs arrive as early as permissible, see Baruah, Mok & Rosier [5]. In other words, the sporadic task system is EDF-schedulable if and only if this is true for the corresponding synchronous periodic task system.

## 2 Diophantine approximation

The EDF-schedulability test contains only one single unknown variable  $Q$ . This is unusual for **NP/coNP**-hard problems and helps us to narrow down the search for **NP/coNP**-hard remote relatives. The relative that we found helpful for problems in Real-time scheduling is *Diophantine approximation*, a problem in the field of algorithmic number theory (see e.g. [20]). Roughly speaking, there the objective is to replace a number or a vector, by another number or vector which is very close to the original, but less complex in terms of fractionality.

More precisely, a sequence  $\alpha_1, \dots, \alpha_n$  of rational numbers together with a bound  $N \in \mathbb{N}$  and an error bound  $\varepsilon \in \mathbb{Q}_+$  is given. One has to decide whether

$$(2.1) \quad \exists Q \in \{1, \dots, N\} : \max_{i=1, \dots, n} | \lceil Q\alpha_i \rceil - Q\alpha_i | \leq \varepsilon,$$

where  $\lceil x \rceil$  is the integer closest to  $x \in \mathbb{R}$ . In a seminal work, Lagarias [16] has shown, that testing (2.1) is **NP**-hard. This was later extended by Rössner & Seifert [22] and Chen & Meng [10] to inapproximability results. In [13], the authors of this paper applied these results to show that response-time computation of tasks in a *Rate-monotonic schedule* is **NP**-hard (under Turing reductions), where tasks with smaller period always preempt that of larger period.

The EDF-schedulability test uses a rounding operation, where one replaces a rational by the closest integer which is equal or smaller, i.e. one *rounds down*. In Diophantine approximation, one rounds up or down to the nearest integer. The variant of Diophantine approximation, where one has to round up is called *directed Diophantine approximation (DDA)*. Recently the authors of this paper provided the following hardness result for directed Diophantine approximation.

**THEOREM 2.1. (HARDNESS OF  $DDA_\rho$  [14])** *There is a constant  $c > 0$ , such that the following Directed Diophantine Approximation problem ( $DDA_\rho$ ) with gap parameter  $\rho = \lfloor n^{c/\log \log n} \rfloor$  is **NP**-hard: Given numbers  $\alpha_1, \dots, \alpha_n \in \mathbb{Q}$ , a bound  $N \in \mathbb{N}$  and an error bound  $\varepsilon \in \mathbb{Q}_+$  as input, distinguish the following cases*

- YES :  $\exists Q \in \{ \lceil N/2 \rceil, \dots, N \} :$   
 $\max_{i=1, \dots, n} (\lceil Q\alpha_i \rceil - Q\alpha_i) \leq \varepsilon$
- NO :  $\nexists Q \in \{1, \dots, \rho \cdot N\} :$   
 $\max_{i=1, \dots, n} (\lceil Q\alpha_i \rceil - Q\alpha_i) \leq 2^n \cdot \varepsilon$

Note that the union of the YES and NO cases does not represent all possible inputs. But there is a polynomial time reduction, taking the input of an **NP**-complete problem, say a SAT clause  $C$ , and yielding a  $DDA_\rho$  instance respecting the YES-case if  $C$  is satisfiable and

the NO-case otherwise. See, e.g., [23, 24] for more details on gap reductions.

Despite of some similarities between  $DDA_\rho$  and EDF-schedulability, we still observe crucial differences:

1.  $DDA_\rho$  contains a ceiling instead of a floor operation.
2. The number  $Q$  is restricted to be integer.
3. The approximation error is measured with  $\| \cdot \|_\infty$ -norm instead of  $\| \cdot \|_1$ -norm.
4. For  $DDA_\rho$ , one has a bound  $N$  on the number  $Q$ .

We can easily eliminate the first difference by observing that  $\lceil Q\alpha_i \rceil - Q\alpha_i = Q \cdot (-\alpha_i) - \lfloor Q(-\alpha_i) \rfloor$ . Consequently replacing the numbers by their negatives, we obtain a  $DDA_\rho$  problem with a floor operation. By adding a sufficiently large integer  $z$  and using  $Q(\alpha_i + z) - \lfloor Q(\alpha_i + z) \rfloor = Q\alpha_i - \lfloor Q\alpha_i \rfloor$  for  $Q \in \mathbb{N}$  we may then make the  $\alpha_i$ 's positive. We conclude that given  $\alpha_1, \dots, \alpha_n \in \mathbb{Q}_+$ ,  $N \in \mathbb{N}$  and  $\varepsilon \in \mathbb{Q}_+$ , it is **NP**-hard to distinguish

- YES :  $\exists Q \in \{ \lceil N/2 \rceil, \dots, N \} :$   
 $\max_{i=1, \dots, n} (Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq \varepsilon$
- NO :  $\nexists Q \in \{1, \dots, \rho \cdot N\} :$   
 $\max_{i=1, \dots, n} (Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq 2^n \cdot \varepsilon$

for  $\rho = \lfloor n^{c/\log \log n} \rfloor$ . In a next step, we introduce a variant of directed Diophantine approximation which incorporates differences (2) & (3). We use the notation  $[\alpha, \beta]$  to denote the set of real numbers  $[\alpha, \beta] = \{x \in \mathbb{R} : \alpha \leq x \leq \beta\}$ .

**THEOREM 2.2. (HARDNESS OF  $DDA_\rho^*$ )** *There exists a constant  $c > 0$ , such that the following  $DDA_\rho^*$  problem with gap parameter  $\rho = \lfloor n^{c/\log \log n} \rfloor$  is **NP**-hard: Given numbers  $\alpha_1, \dots, \alpha_n \in \mathbb{Q}_+$ , weights  $w_1, \dots, w_n \in \mathbb{Q}_+$ , a bound  $N \in \mathbb{N}$  and an error bound  $\varepsilon \in \mathbb{Q}_+$ , distinguish*

- YES :  $\exists Q \in [\lceil N/2 \rceil, N] : \sum_{i=0}^n w_i (Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq \varepsilon$
- NO :  $\nexists Q \in [1, \rho \cdot N] : \sum_{i=0}^n w_i (Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq \rho \cdot \varepsilon$

*Proof.* We reduce  $DDA_\rho$  to  $DDA_\rho^*$ . For this purpose let  $(\alpha_1, \dots, \alpha_n; N; \varepsilon)$  be the given  $DDA_\rho$  instance (with rounding down and  $\alpha_i > 0$  for all  $i$ ). Since the  $\alpha_i$ 's are rational numbers, we can write them as  $\alpha_i = \frac{a_i}{b_i}$  with pairwise co-prime integers  $a_i, b_i \in \mathbb{N}$ . Our  $DDA_\rho^*$  instance consists of the same numbers  $\alpha_1, \dots, \alpha_n$ , equipped with unit weights  $w_1 = \dots = w_n = 1$ . Furthermore we choose the same bound  $N$ , but a different error bound  $\varepsilon' = n \cdot \varepsilon$  and we add one more number  $\alpha_0 = 1$  with a very high weight of  $w_0 = 2 \cdot \max\{a_i : i = 1, \dots, n\} \cdot \varepsilon \cdot \rho \cdot n$ . Intuitively

the weight  $w_0$  is large enough, such that any reasonable  $\text{DDA}_\rho^*$  solution  $Q$  of this instance must be an integer. It suffices to show the following implications:

- YES :  
 $\exists Q \in \{[N/2], \dots, N\} : \max_{i=1, \dots, n} (Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq \varepsilon$   
 $\Rightarrow \exists Q \in [[N/2], N] : \sum_{i=0}^n w_i(Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq \varepsilon'$
- NO :  
 $\nexists Q \in \{1, \dots, \rho \cdot N\} : \max_{i=1, \dots, n} (Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq 2^n \cdot \varepsilon$   
 $\Rightarrow \nexists Q \in [1, \rho \cdot N] : \sum_{i=0}^n w_i(Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq \rho \cdot \varepsilon'$

YES-case: Clearly YES instances for  $\text{DDA}_\rho$  are mapped to YES instances of  $\text{DDA}_\rho^*$  by simply using the same solution  $Q$ . This is the case since given a  $Q \in \{[N/2], \dots, N\}$  that matches the conditions of the YES case for  $\text{DDA}_\rho$ , one has

$$\begin{aligned} & \sum_{i=0}^n w_i(Q\alpha_i - \lfloor Q\alpha_i \rfloor) \\ = & w_0 \cdot \underbrace{(Q - \lfloor Q \rfloor)}_{=0} + \sum_{i=1}^n 1 \cdot \underbrace{(Q\alpha_i - \lfloor Q\alpha_i \rfloor)}_{\leq \varepsilon} \\ \leq & n \cdot \varepsilon \\ = & \varepsilon'. \end{aligned}$$

NO-case: Now suppose that we have a  $Q \in [1, \rho \cdot N]$  with  $\sum_{i=0}^n w_i(Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq \rho \cdot \varepsilon' = \rho \cdot n \cdot \varepsilon$ . Decrease  $Q$  continuously until  $Q\alpha_j \in \mathbb{Z}$  for at least one  $j \in \{0, \dots, n\}$ . This can only decrease the approximation error since  $\lfloor Q\alpha_i \rfloor$  remains invariant for all  $i \in \{0, \dots, n\}$ . Furthermore  $Q$  will never be decreased below 1 since  $\alpha_0 = 1$ . If  $Q$  is then an integer, we are done since

$$\begin{aligned} \max_{i=1, \dots, n} (Q\alpha_i - \lfloor Q\alpha_i \rfloor) & \leq \sum_{i=0}^n w_i(Q\alpha_i - \lfloor Q\alpha_i \rfloor) \\ & \leq \rho \cdot n \cdot \varepsilon \\ & \leq 2^n \varepsilon \end{aligned}$$

for  $n$  large enough. Now suppose that  $Q$  is not integer. Then we may write  $Q\alpha_j = Q \frac{a_j}{b_j} =: z \in \mathbb{Z}$  for some  $j \in \{1, \dots, n\}$ , thus  $Q = \frac{zb_j}{a_j} \in \mathbb{Z} \frac{1}{a_j}$ . We write  $Q = \frac{y}{a_j}$  where  $y$  is integer but not a multiple of  $a_j$  (since  $Q \notin \mathbb{Z}$ ). Hence

$$Q - \lfloor Q \rfloor = \frac{y}{a_j} - \frac{\lfloor Q \rfloor a_j}{a_j} = \underbrace{(y - \lfloor Q \rfloor a_j)}_{\geq 1} \cdot \frac{1}{a_j} \geq \frac{1}{a_j}$$

where we use that  $y - \lfloor Q \rfloor a_j$  is a non-negative integer but  $y - \lfloor Q \rfloor a_j \neq 0$ . We obtain

$$\sum_{i=0}^n w_i(Q\alpha_i - \lfloor Q\alpha_i \rfloor) \geq w_0 \cdot (Q - \lfloor Q \rfloor) \geq w_0 \cdot \frac{1}{a_j} > \rho \cdot n \cdot \varepsilon$$

by the choice of  $w_0$ . This contradiction yields that  $Q \in \mathbb{N}$  and the claim follows.

### 3 Hardness of EDF-schedulability

In this section we will see that the **NP**-hard problem  $\text{DDA}_\rho^*$  is close enough to the EDF-schedulability condition to admit a direct reduction. To achieve this, YES (NO, resp.) instances for  $\text{DDA}_\rho^*$  are mapped to NO (YES, resp.) instances of EDF-schedulability. Intuitively this is done as follows: Suppose we are given a  $\text{DDA}_\rho^*$  instance  $(\alpha_1, \dots, \alpha_n; w_1, \dots, w_n; N; \varepsilon)$ . The first idea is to create implicit-deadline tasks  $\tau_1, \dots, \tau_n$  with  $p_i = d_i = \frac{1}{\alpha_i}$ . Then we have

$$\left\lfloor \frac{Q - d_i}{p_i} \right\rfloor + 1 = \lfloor Q\alpha_i \rfloor$$

hence a  $Q$  that maximizes  $\text{DBF}(\mathcal{S}, Q)/Q$ , minimizes the approximation error. On the other hand we need to forbid  $Q$  with  $Q \gg N$ . For this purpose we add a special task  $\tau_0$  which has a deadline of  $N/2$  and a sufficiently large period (we may imagine  $p_0 = \infty$ ). Then the quantity  $\text{DBF}(\tau_0, Q)/Q$  contributes significantly to  $\text{DBF}(\mathcal{S}, Q)/Q$  only if  $Q$  is of order  $N$ .

**THEOREM 3.1.** *Given an instance of  $\text{DDA}_\rho^*$  consisting of rational numbers  $\alpha_1, \dots, \alpha_n \in \mathbb{Q}_+$ , weights  $w_1, \dots, w_n \in \mathbb{Q}_+$ , a bound  $N \in \mathbb{N}_{\geq 2}$  and an error bound  $\varepsilon > 0$ , we can find in polynomial time a constrained-deadline task system  $\mathcal{S}$  consisting of  $n + 1$  tasks such that*

- YES:  $\exists Q \in [[N/2], N] : \sum_{i=1}^n w_i(Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq \varepsilon \Rightarrow \mathcal{S}$  not EDF-schedulable
- NO:  $\nexists Q \in [[N/2], 3N] : \sum_{i=1}^n w_i(Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq 3\varepsilon \Rightarrow \mathcal{S}$  EDF-schedulable

Furthermore  $n$  tasks in  $\mathcal{S}$  have implicit-deadlines.

*Proof.* A set of tasks is EDF-schedulable on a processor of speed  $\beta > 0$  if and only if the tasks with running times scaled by  $\frac{1}{\beta}$  are feasible on a unit speed processor. Thus we may assume to have an oracle for the test

$$\forall Q \geq 0 : \sum_{i=1}^n \left( \left\lfloor \frac{Q - d_i}{p_i} \right\rfloor + 1 \right) \cdot c_i \leq \beta \cdot Q$$

Let  $N \in \mathbb{N}, \alpha_1, \dots, \alpha_n, w_1, \dots, w_n \in \mathbb{Q}_+, \varepsilon > 0$  be the  $\text{DDA}_\rho^*$  instance. We choose a constrained-deadline task system  $\mathcal{S}$  consisting of  $n + 1$  tasks

$$\begin{aligned} \tau_i &= (c_i, d_i, p_i) = \left( w_i, \frac{1}{\alpha_i}, \frac{1}{\alpha_i} \right) \quad \forall i = 1, \dots, n \\ \tau_0 &= (c_0, d_0, p_0) = (3\varepsilon, [N/2], 12N) \end{aligned}$$

and processor speed

$$\beta = \frac{\varepsilon}{N} + \sum_{i=1}^n w_i \alpha_i$$

which just slightly exceeds the utilization.

YES-case: Suppose that we have a  $Q \in [\lceil N/2 \rceil, N]$  with  $\sum_{i=1}^n w_i(Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq \varepsilon$ . Then

$$\begin{aligned} & \text{DBF}(\{\tau_0, \dots, \tau_n\}, Q) \\ = & \text{DBF}(\tau_0, Q) + \sum_{i=1}^n \left( \left\lfloor \frac{Q - d_i}{p_i} \right\rfloor + 1 \right) c_i \\ = & 3\varepsilon + \sum_{i=1}^n \lfloor Q\alpha_i \rfloor w_i \\ \stackrel{(*)}{\geq} & 3\varepsilon + \left( \left( \sum_{i=1}^n Q\alpha_i w_i \right) - \varepsilon \right) \\ = & 2\varepsilon + Q \sum_{i=1}^n \alpha_i w_i \\ \stackrel{(**)}{>} & Q \cdot \underbrace{\left( \frac{\varepsilon}{N} + \sum_{i=1}^n \alpha_i w_i \right)}_{=\beta} \\ = & \beta Q \end{aligned}$$

Here we use  $\sum_{i=1}^n w_i(Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq \varepsilon$  in (\*) and  $Q \leq N < 2N$  in (\*\*). Thus the task system  $\mathcal{S}$  is not EDF-schedulable (on a processor of speed  $\beta$ ).

NO-case: Next we assume that  $\mathcal{S}$  is not EDF-schedulable. Then there exists a  $Q > 0$  such that  $\text{DBF}(\{\tau_0, \dots, \tau_n\}, Q) > \beta Q$ . We need to show that  $Q \in [\lceil N/2 \rceil, 3N]$  and  $\sum_{i=1}^n w_i(Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq 3\varepsilon$ .

Observe that using the definition of  $\beta$  and  $\lfloor Q\alpha_i \rfloor \leq Q\alpha_i$ , one has

$$\begin{aligned} \text{DBF}(\tau_0, Q) &= \text{DBF}(\mathcal{S}, Q) - \text{DBF}(\{\tau_1, \dots, \tau_n\}, Q) \\ &> \beta Q - \sum_{i=1}^n \lfloor Q\alpha_i \rfloor w_i \\ &\geq \beta Q - Q \sum_{i=1}^n \alpha_i w_i \\ &= \beta Q - Q \underbrace{\left( \frac{\varepsilon}{N} + \sum_{i=1}^n \alpha_i w_i \right)}_{=\beta} + Q \frac{\varepsilon}{N} \\ &= Q \frac{\varepsilon}{N} \end{aligned}$$

Since  $\tau_0$  has its first deadline at  $d_0 = \lceil N/2 \rceil$  and  $\text{DBF}(\tau_0, Q) > 0$  we must have  $Q \geq \lceil N/2 \rceil$ . Suppose

for contradiction that already the second deadline of  $\tau_0$  occurred before  $Q$ , i.e.  $Q \geq p_0 = 12N$ . Then

$$\text{DBF}(\tau_0, Q) \leq c_0 \cdot \left\lceil \frac{Q}{p_0} \right\rceil \leq 2 \cdot 3\varepsilon \cdot \frac{Q}{12N} < Q \frac{\varepsilon}{N},$$

leading to a contradiction. Hence, till time  $Q$  exactly one deadline of  $\tau_0$  has passed, thus  $\text{DBF}(\tau_0, Q) = 3\varepsilon$ . But we already inferred the bound  $\text{DBF}(\tau_0, Q) > Q \frac{\varepsilon}{N}$ , thus even  $Q < 3N$ . Finally

$$\begin{aligned} & \sum_{i=1}^n w_i(Q\alpha_i - \lfloor Q\alpha_i \rfloor) \\ = & Q \underbrace{\sum_{i=1}^n \alpha_i w_i}_{<\beta} - (\text{DBF}(\mathcal{S}, Q) - \text{DBF}(\tau_0, Q)) \\ \leq & \underbrace{Q\beta - \text{DBF}(\mathcal{S}, Q)}_{<0} + 3\varepsilon \\ \leq & 3\varepsilon \end{aligned}$$

and the claim follows.

Theorem 1.1 follows by combining Theorem 2.2 and 3.1, with  $\rho = 4$ .

## 4 Open problems

We obtained that testing EDF-schedulability of synchronous periodic tasks is **coNP**-hard. Nevertheless the starting point of the reduction is a problem that admits pseudo-polynomial time algorithms. Furthermore the utilization of the task system constructed in the reduction might be extremely close to 1. Hence we believe that the following statements are true:

**CONJECTURE 1.** *There is a pseudo-polynomial time algorithm for testing EDF-schedulability of synchronous constrained-deadline systems.*

Note that till now such an algorithm is only known if  $1/(1-u)$  is bounded by a polynomial in the input length (again  $u := \sum_{i=1}^n \frac{c_i}{p_i}$ ).

**CONJECTURE 2.** *For every fixed  $\varepsilon > 0$ , EDF-schedulability of a synchronous constrained-deadline task system  $\tau_1, \dots, \tau_n$  can be decided in polynomial time if  $u \leq 1 - \varepsilon$ .*

## References

- [1] K. ALBERS AND F. SLOMKA, *An event stream driven approximation for the analysis of real-time systems*, in ECRTS, IEEE Computer Society, 2004, pp. 187–195.

- [2] ———, *Efficient feasibility analysis for real-time systems with EDF scheduling*, in DATE, IEEE Computer Society, 2005, pp. 492–497.
- [3] S. K. BARUAH, D. CHEN, S. GORINSKY, AND A. K. MOK, *Generalized multiframe tasks*, Real-Time Systems, 17 (1999), pp. 5–22.
- [4] S. K. BARUAH AND J. GOOSSENS, *Scheduling real-time tasks: Algorithms and complexity*, in Handbook of Scheduling — Algorithms, Models, and Performance Analysis, J. Y.-T. Leung, ed., Chapman & Hall/CRC, 2004, ch. 28.
- [5] S. K. BARUAH, A. K. MOK, AND L. E. ROSIER, *Pre-emptively scheduling hard-real-time sporadic tasks on one processor*, in IEEE Real-Time Systems Symposium, 1990, pp. 182–190.
- [6] S. K. BARUAH AND K. PRUHS, *Open problems in real-time scheduling*, To appear in: Journal of Scheduling, (2009).
- [7] V. BONIFACI, A. MARCHETTI-SPACCAMELA, AND S. STILLER, *A constant-approximate feasibility test for multiprocessor real-time scheduling*, in ESA, D. Halperin and K. Mehlhorn, eds., vol. 5193 of Lecture Notes in Computer Science, Springer, 2008, pp. 210–221.
- [8] G. BUTTAZZO, *Hard Real-Time Computing Systems: Predictable Scheduling Algorithms and Applications*, Kluwer Academic Publishers, Boston, USA, 2000.
- [9] S. CHAKRABORTY, S. KÜNZLI, AND L. THIELE, *Approximate schedulability analysis*, in IEEE Real-Time Systems Symposium, 2002, pp. 159–168.
- [10] W. CHEN AND J. MENG, *An improved lower bound for approximating shortest integer relation in  $l_\infty$  norm ( $SIR_\infty$ )*, Information Processing Letters, 101 (2007), pp. 174–179.
- [11] M. L. DERTOUZOS, *Control robotics: The procedural control of physical processes*, in IFIP Congress, 1974, pp. 807–813.
- [12] U. C. DEVI, *An improved schedulability test for uniprocessor periodic task systems*, in ECRTS, IEEE Computer Society, 2003, p. 23.
- [13] F. EISENBRAND AND T. ROTHVOSS, *Static-priority Real-time Scheduling: Response Time Computation is NP-hard*, in IEEE Real-Time Systems Symposium (RTSS), 2008.
- [14] F. EISENBRAND AND T. ROTHVOSS, *New hardness results for diophantine approximation*, in 12-th International Workshop on Approximation Algorithms for Combinatorial Optimization Problems (APPROX 2009), LNCS, Springer, 2009.
- [15] N. FISHER AND S. K. BARUAH, *A fully polynomial-time approximation scheme for feasibility analysis in static-priority systems with arbitrary relative deadlines*, in ECRTS '05: Proceedings of the 17th Euromicro Conference on Real-Time Systems (ECRTS'05), Washington, DC, USA, 2005, IEEE Computer Society, pp. 117–126.
- [16] J. C. LAGARIAS, *The computational complexity of simultaneous Diophantine approximation problems*, SIAM Journal on Computing, 14 (1985), pp. 196–209.
- [17] J. LEUNG, *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, CRC Press, Inc., Boca Raton, FL, USA, 2004.
- [18] J. Y.-T. LEUNG AND M. L. MERRILL, *A note on pre-emptive scheduling of periodic, real-time tasks*, Information Processing Letters, 11 (1980), pp. 115–118.
- [19] C. L. LIU AND J. W. LAYLAND, *Scheduling algorithms for multiprogramming in a hard-real-time environment*, Journal of the Association for Computing Machinery, 20 (1973), pp. 46–61.
- [20] I. NIVEN, H. S. ZUCKERMAN, AND H. L. MONTGOMERY, *An introduction to the theory of numbers*, John Wiley & Sons Inc., New York, fifth ed., 1991.
- [21] R. PELLIZZONI AND G. LIPARI, *A new sufficient feasibility test for asynchronous real-time periodic task sets*, in ECRTS, IEEE Computer Society, 2004, pp. 204–211.
- [22] C. RÖSSNER AND J. P. SEIFERT, *Approximating good simultaneous Diophantine approximations is almost NP-hard*, in Mathematical foundations of computer science 1996 (Cracow), vol. 1113 of Lecture Notes in Comput. Sci., Springer, Berlin, 1996, pp. 494–505.
- [23] V. V. VAZIRANI, *Approximation algorithms*, Springer-Verlag, Berlin, 2001.
- [24] I. WEGENER, *Complexity theory*, Springer-Verlag, Berlin, 2005. Exploring the limits of efficient algorithms, Translated from the German by Randall Pruim.