Recent Hardness Results for Periodic Uni-processor Scheduling

Friedrich Eisenbrand and Thomas Rothvoss

Institute of Mathematics
École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
\{friedrich.eisenbrand, thomas.rothvoss\}@epfl.ch

Abstract In the synchronous periodic task model, a set \(\tau_1, \ldots, \tau_n\) of tasks is given, each releasing \(n\) jobs of running time \(c_i\) and 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 Barnah & Pruls. 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 [1] 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 \(\tau_1, \ldots, \tau_n\) 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 \(i \in \{1, \ldots, n\}\) 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 \(i \in \{1, \ldots, n\}\).

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 [2]).

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

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

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

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

gives 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

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

Dagstuhl Seminar Proceedings 10071
Scheduling
http://drops.dagstuhl.de/opus/volltexte/2010/2545
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 $DBF(S, Q) \leq Q$ for all $Q > 0$. Baruah et al. [3] showed that this condition is in fact sufficient, hence an EDF-schedulability test is a test which checks validity of the following formula

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

see Figure 1 for an illustration.

![Figure 1](image-url)

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

Much effort has been spent on developing sufficient polynomial or exact pseudo-polynomial time tests for EDF-schedulability of periodic tasks, see [4,5,3,6,7]. 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 [8] on open problems in Real-time scheduling. We settle the complexity status of testing EDF-schedulability by proving the following theorem.

**Theorem 1.** Given a set $S = \{\tau_1, \ldots, \tau_n\}$ 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 $S$ 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 $i = 1, \ldots, n - 1$.

This, together with the result in [3] implies the following corollary.

**Corollary 1.** Given a set $S = \{\tau_1, \ldots, \tau_n\}$ 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 $S$ 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 $u = \sum_{i=1}^{n} \frac{p_i}{d_i}$ be the utilization of a task system. Given that $S$ is not EDF-schedulable, the smallest $Q > 0$, certifying the infeasibility must have

$$Q < \frac{u}{1 - u} \max_{i=1, \ldots, n} \{p_i - d_i\},$$
see e.g. [9,10]. This admits a pseudo-polynomial time algorithm for the feasibility test, if the utilization of \( S \) is bounded by \( 1 - \varepsilon \) for some constant \( \varepsilon > 0 \).

Albers & Slomka [11] gave an FPTAS for approximating the speed of a processor, needed to make the EDF-schedule of \( S \) 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 [12]. See [1] for more details on fixed priority scheduling policies and [6,4,7,13] for further approaches to feasibility analysis of EDF-schedules. Recently, Bonifaci et al. [14] 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 \( \text{DF}(S, Q) \leq Q \) has only to be evaluated at \( Q = \text{scm}(p_1, \ldots, p_n) \) and the set is EDF-scheduleable if and only if the utilization is bounded by \( 1 \), see Liu & Layland [13]. In other words, the EDF-scheduleability in this special case is decidable in polynomial time. If the set may be asynchronous, i.e. each task has an offset \( a_i \), such that jobs are released at \( z : p_i + a_i \), then testing the feasibility is strongly \( \text{coNP} \)-hard [16]. This even holds if the utilization of the system is bounded from above by an arbitrarily small constant.

In the \textit{sporadic} task model neither release times nor running times are predetermined. There, \( c_i \) denotes the \textit{worst-case execution time} and \( p_i \) denotes the \textit{minimum inter-arrival time}. But the worst-case is attained in a \textit{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 [3]. In other words, the sporadic task system is EDF-scheduleable if and only if this is true for the corresponding synchronous periodic task system.

## 2 Diophantine approximation

The EDF-scheduleability test contains only one single unknown variable \( Q \). This is unusual for \text{NP}/\text{coNP}-hard problems and helps us to narrow down the search for \text{NP}/\text{coNP}-hard remote relatives. The relative that we found helpful for problems in Real-time scheduling is \textit{Diophantine approximation}, a problem in the field of algorithmic number theory (see e.g. [17]). Roughly speaking, 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 fractalinity.

More precisely, a sequence \( \alpha_1, \ldots, \alpha_n \) 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

\[
\exists Q \in \{1, \ldots, N\} : \max_{i=1, \ldots, n} \{ |Q \alpha_i - Q \alpha_i| \} \leq \varepsilon,
\]

(1)

where \([x]\) is the integer closest to \( x \in \mathbb{R} \). In a seminal work, Lagarias [18] has shown, that testing (1) is \text{NP}-hard. This was later extended by Rössner & Seifert [19] and Chen & Meng [20] to inapproximability results. In [21], the authors of this paper applied these results to show that response-time computation of tasks in a \textit{Rate-monotonic schedule} is \text{NP}-hard, where tasks with smaller period always preempt that of larger period.

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

\textbf{Theorem 2 (Hardness of DDA \( \rho \) [22])}. There exists 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 \text{NP}-hard: Given numbers \( \alpha_1, \ldots, \alpha_n \in \mathbb{Q} \), a bound \( N \in \mathbb{N} \) and an error bound \( \varepsilon \in \mathbb{Q}_+ \) as input, distinguish the following cases

- \textbf{Yes} : \( \exists Q \in \{ \lceil N/2 \rceil, \ldots, N \} : \max_{i=1, \ldots, n} \{ [Q \alpha_i] - Q \alpha_i \} \leq \varepsilon \)
- \textbf{No} : \( \not\exists Q \in \{ 1, \ldots, \rho \cdot N \} : \max_{i=1, \ldots, n} \{ [Q \alpha_i] - Q \alpha_i \} \leq 2n \cdot \varepsilon \)
Note that the union of the \(\text{Yes}\) and \(\text{No}\) cases does not represent all possible inputs. But there is a polynomial time reduction, taking the input of an \(\text{NP}\)-complete problem, say a \(\text{SAT}\) clause \(C\), and yielding a \(\text{DDA}_p\) instance respecting the \(\text{Yes}\)-case if \(C\) is satisfiable and the \(\text{No}\)-case otherwise. See, e.g., [23,24] for more details on \(\text{gap}\) reductions.

Despite of some similarities between \(\text{DDA}_p\) and EDF-schedulability, we still observe crucial differences:

1. \(\text{DDA}_p\) 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 \(\text{DDA}_p\), one has a bound \(N\) on the number \(Q\).

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

\[- \text{Yes} : \exists Q \in \{\lceil N/2 \rceil, \ldots, N\} : \max_{i=1, \ldots, n}(Q\alpha_i - |Q\alpha_i|) \leq \varepsilon\]
\[- \text{No} : \exists Q \in \{1, \ldots, \rho \cdot N\} : \max_{i=1, \ldots, n}(Q\alpha_i - |Q\alpha_i|) \leq 2^n \cdot \varepsilon\]

for \(\rho = \lceil n^{1/\log \log n} \rceil\). In a next step, we introduce a variant of directed Diophantine approximation which incorporates differences (2) & (3). We use the notation \([\alpha, \beta] = \{x \in \mathbb{R} : \alpha \leq x \leq \beta\}\).

**Theorem 3 (Hardness of \(\text{DDA}_p^*\)).** There exists a constant \(c > 0\), such that the following \(\text{DDA}_p\) problem with \(\text{gap}\) parameter \(\rho = \lceil n^{1/\log \log n} \rceil\) is \(\text{NP}\)-hard: Given numbers \(\alpha_1, \ldots, \alpha_n \in \mathbb{Q}_+, \) weights \(w_1, \ldots, w_n \in \mathbb{Q}_+, \) a bound \(N \in \mathbb{N}\) and an error bound \(\varepsilon \in \mathbb{Q}_+,\) distinguish

\[- \text{Yes} : \exists Q \in \{\lceil N/2 \rceil, \ldots, N\} : \sum_{i=0}^{n} w_i(Q\alpha_i - |Q\alpha_i|) \leq \varepsilon\]
\[- \text{No} : \exists Q \in \{1, \ldots, \rho \cdot N\} : \sum_{i=0}^{n} w_i(Q\alpha_i - |Q\alpha_i|) \leq \rho \cdot \varepsilon\]

Proof. We reduce \(\text{DDA}_p\) to \(\text{DDA}_p^*\). Let \((\alpha_1, \ldots, \alpha_n; \varepsilon)\) be the given \(\text{DDA}_p\) instance (with rounding down and \(\alpha_i > 0\) for all \(i\)). Since the \(\alpha_i\)'s are rational numbers, we can write them as \(\alpha_i = \frac{a_i}{n_i}\) with pairwise co-prime integers \(a_i, n_i \in \mathbb{N}\). Our \(\text{DDA}_p^*\) instance consists of the same numbers \(\alpha_1, \ldots, \alpha_n\), equipped with unit weights \(w_1 = \cdots = w_n = 1\). 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 \(w_0 = 2 \cdot \max \{\alpha_i; i = 1, \ldots, n\} \cdot \varepsilon \cdot \rho \cdot n\). Intuitively the weight \(w_0\) is large enough, such that any reasonable \(\text{DDA}_p^*\) solution \(Q\) of this instance must be an integer. It suffices to show the following implications:

\[- \text{Yes} : \exists Q \in \{\lceil N/2 \rceil, \ldots, N\} : \max_{i=1, \ldots, n}(Q\alpha_i - |Q\alpha_i|) \leq \varepsilon\]
\[- \text{No} : \exists Q \in \{1, \ldots, \rho \cdot N\} : \max_{i=1, \ldots, n}(Q\alpha_i - |Q\alpha_i|) \leq 2^n \cdot \varepsilon\]

\[- \text{Yes-case:} \text{ Clearly \text{Yes} instances for } \text{DDA}_p \text{ are mapped to } \text{Yes} \text{ instances of } \text{DDA}_p^* \text{ by simply using the same solution } Q. \text{ This is the case since given a } Q \in \{\lceil N/2 \rceil, \ldots, N\} \text{ that matches the conditions of the } \text{Yes} \text{ case for } \text{DDA}_p, \text{ one has}

\[\sum_{i=0}^{n} w_i(Q\alpha_i - |Q\alpha_i|) = w_0 \cdot |Q - |Q|| + \sum_{i=1}^{n} 1 \cdot |Q\alpha_i - |Q\alpha_i|| \leq n \cdot \varepsilon = \varepsilon'.\]

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

\[\max_{i=1, \ldots, n} (Q\alpha_i - |Q\alpha_i|) \leq \sum_{i=0}^{n} w_i(Q\alpha_i - |Q\alpha_i|) \leq \rho \cdot n \cdot \varepsilon \leq 2^n \varepsilon.\]
for \( n \) large enough. Now suppose that \( Q \) is not integer. Then we may write \( Q \alpha_j = Q_{\bar{a}_j} =: z \in \mathbb{Z} \), thus \( Q = \frac{z y}{\bar{a}_j} \in \mathbb{Z} \frac{1}{\bar{a}_j} \). We write \( Q = \frac{y}{a_j} \) where \( y \) is integer but not a multiple of \( a_j \) (since \( Q \notin \mathbb{Z} \)). Hence

\[
Q - [Q] = \frac{y}{a_j} - \left\lfloor \frac{Q}{a_j} \right\rfloor = \left( y - \left\lfloor Q \right\rfloor a_j \right) \cdot \frac{1}{a_j} \geq 1
\]

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

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

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 \textbf{NP}-hard problem \textsc{DDA}\textsuperscript{*} is close enough to the EDF-schedulability condition to admit a direct reduction. To achieve this, \textsc{YES} (\textit{No}, resp.) instances for \textsc{DDA}\textsuperscript{*} are mapped to \textsc{No} (\textsc{YES}, resp.) instances of EDF-schedulability. Intuitively this is done as follows: Suppose we are given a \textsc{DDA}\textsuperscript{*} instance \((\alpha_1, \ldots, \alpha_n; w_1, \ldots, w_n; N; \varepsilon)\). The first idea is to create implicit-deadline tasks \( \tau_1, \ldots, \tau_n \) with \( p_i = d_i = \frac{1}{\alpha_i} \). Then we have

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

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 \( Q \gg N \), a common multiple of all \( p_i \)'s. 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 \( \text{DBF}(\tau_0)/Q \) contributes significantly to \( \text{DBF}(\mathcal{S}, Q)/Q \) only if \( Q \) is of order \( N \).

**Theorem 4.** Given an instance of \textsc{DDA}\textsuperscript{*} consisting of rational numbers \( \alpha_1, \ldots, \alpha_n \in \mathbb{Q}_+ \), weights \( w_1, \ldots, w_n \in \mathbb{Q}_+ \), a bound \( N \in \mathbb{N}_{>2} \) and an error bound \( \varepsilon > 0 \), we can in polynomial time a constrained-deadline task system \( \mathcal{S} \) consisting of \( n + 1 \) tasks such that

\[
- \textsc{YES}: \exists Q \in \left( \frac{[N/2]}{N} \right): \sum_{i=1}^{n} w_i (Q \alpha_i - [Q \alpha_i]) \leq \varepsilon \Rightarrow \mathcal{S} \text{ not EDF-schedulable}
\]

\[
- \textsc{NO}: \forall Q \in \left( \frac{[N/2]}{N} \right): \exists Q = 3 \varepsilon \Rightarrow \mathcal{S} \text{ EDF-schedulable}
\]

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

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

Let \( N \in \mathbb{N}, \alpha_1, \ldots, \alpha_n, w_1, \ldots, w_n \in \mathbb{Q}_+, \varepsilon > 0 \) be the \textsc{DDA}\textsuperscript{*} instance. We choose a constrained-deadline task system \( \mathcal{S} \) consisting of \( n + 1 \) tasks

\[
\tau_i = (c_i, d_i, p_i) = \left( \frac{w_i}{\alpha_i}, \frac{1}{\alpha_i}, \frac{1}{\alpha_i} \right) \quad \forall i = 1, \ldots, n
\]

\[
\tau_0 = (c_0, d_0, p_0) = (3 \varepsilon, [N/2], 12N)
\]

and processor speed

\[
\beta = \frac{\varepsilon}{N} + \sum_{i=1}^{n} w_i \alpha_i = \frac{\varepsilon}{N} + u(\{\tau_1, \ldots, \tau_n\})
\]

which just slightly exceeds the utilization.
**Yes-case:** Suppose that we have a \( Q \in [[N/2], N] \) with \( \sum_{i=1}^{n} w_i (Q\alpha_i - \lfloor Q\alpha_i \rfloor) \leq \varepsilon \). Then

\[
\text{DBF}(\{\tau_0, \ldots, \tau_n\}, Q) = \text{DBF}(\tau_0, Q) + \sum_{i=1}^{n} \left( \left\lfloor \frac{Q - d_i}{p_i} \right\rfloor + 1 \right) c_i
\]

\[
= 3\varepsilon + \sum_{i=1}^{n} [Q\alpha_i] w_i
\]

\[
(*) \geq 3\varepsilon + \left( \sum_{i=1}^{n} Q\alpha_i w_i - \varepsilon \right)
\]

\[
= 2\varepsilon + Q \sum_{i=1}^{n} \alpha_i w_i
\]

\[
(**) > Q \cdot \left( \frac{\varepsilon}{N} + \sum_{i=1}^{n} \alpha_i w_i \right)
\]

\[
= \beta Q
\]

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 \( S \) is not EDF-schedulable (on a processor of speed \( \beta \)).

**No-case:** Next we assume that \( S \) is not EDF-schedulable. Then there is a \( Q > 0 \) such that \( \text{DBF}(\{\tau_0, \ldots, \tau_n\}, Q) > \beta Q \). We need to show that \( Q \in [[N/2], 3N] \) 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 \( [Q\alpha_i] \leq Q\alpha_i \), one has

\[
\text{DBF}(\tau_0, Q) = \text{DBF}(S, Q) - \text{DBF}(\{\tau_1, \ldots, \tau_n\}, Q)
\]

\[
> \beta Q - \sum_{i=1}^{n} [Q\alpha_i] w_i
\]

\[
\geq \beta Q - Q \sum_{i=1}^{n} \alpha_i w_i
\]

\[
= \beta Q - Q \left( \frac{\varepsilon}{N} + \sum_{i=1}^{n} \alpha_i w_i \right) + Q \frac{\varepsilon}{N}
\]

\[
= Q \frac{\varepsilon}{N}
\]

Since \( \tau_0 \) has its first deadline at \( d_0 = [N/2] \) and \( \text{DBF}(\tau_0, Q) > 0 \) we must have \( Q \geq [N/2] \). Suppose for contradiction that already the second deadline of \( \tau_0 \) occurred before \( Q \), i.e. \( Q \geq \rho_0 = 12N \). Then

\[
\text{DBF}(\tau_0, Q) \leq c_0 \cdot \left\lfloor \frac{Q}{\rho_0} \right\rfloor \leq 2 \cdot 3\varepsilon \cdot \frac{Q}{12N} < Q \frac{\varepsilon}{N},
\]

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

\[
\sum_{i=1}^{n} w_i (Q\alpha_i - \lfloor Q\alpha_i \rfloor) = Q \sum_{i=1}^{n} \alpha_i w_i - (\text{DBF}(S, Q) - \text{DBF}(\tau_0, Q)) \leq Q\beta - \text{DBF}(S, Q) + 3\varepsilon \leq 3\varepsilon
\]

and the claim follows.

Theorem 1 follows by combining Theorem 3 and 4, with \( \rho = 4 \).
References