# Uniprocessor EDF Feasibility is an Integer Problem

Enrico Bini

Scuola Superiore Sant'Anna, Real-Time Systems Lab
Piazza Martiri della Libertà 33, 56127, Pisa, Italy

**Abstract.** The research on real-time scheduling has mostly focused on the development of algorithms that allows to test whether the constraints imposed on the task execution (often expressed by deadlines) are verified or not. However, in many design scenarios the task set is only partially known and these algorithms cannot be applied because they require the complete knowledge of all the parameters of the task set. Moreover, very often the designer has the freedom to select some of the task set parameters in order to maximize the system performance, and an arbitrary selection of the free parameters can lead either to poor performance or to a constraint violation. It is then useful to describe the *feasibility region* of the task set parameters by equations instead of by algorithms, so that optimization algorithms can be applied to find the best assignment to the free variables.
In this paper we formulate the EDF schedulability on a single processor through a combination of linear constraints. We study the geometry of the feasibility region of task deadlines when computation times and periods are known.

**Keywords.** EDF schedulability condition, optimal deadline assignment.

## 1 Introduction

The software of control systems is typically implemented through a set of periodic activities performing data sampling, sensory processing, control, action planning, and actuation. When several of such activities execute concurrently in the same processor, however, each control task may experience a variable delay and jitter, mainly due to the interference created by the other tasks. If not properly taken into account, delays and jitter may degrade the performance of the system and even jeopardize its stability [1–3].

A common practice to reduce jitter and delay is to limit the execution interval of each task by setting a suitable relative deadline. Working on this line, Baruah at al. [4] proposed two methods (with different complexity and performance) for assigning shorter relative deadlines to tasks and guaranteeing the schedulability of the task set.

Several authors [5–8] independently proposed different algorithms for computing the minimum deadline of a newly arrived task, assuming the existing task set is feasibly schedulable by Earliest Deadline First (EDF). The problem of these methods is that they can hardly be extended to reduce a set of arbitrary deadlines, but can only be applied to a single task at a time, following a given order, as suggested by [8]. In this way, however, the only task which experiences a significant deadline reduction is the first task in the sequence, since it can use all the slack available in the task set to minimize its deadline, leaving little margin for the remaining tasks. To apply a more uniform deadline reduction in the task set, Balbastre et al. [7] proposed an algorithm able to scale all deadlines by the same factor. The problem of this approach, however, is that a uniform deadline reduction may not achieve a significant improvement in terms of jitter and delays, because all deadlines are reduced and, in some cases, the schedule could even remain unchanged.

In this paper, we formally demonstrate an alternative condition for testing the schedulability under the EDF scheduling algorithm. This new formulation allows the description of the *feasibility region of the task deadlines*. The knowledge of such a region is extremely useful in the design process, since it allows the designer to select the set of relative deadlines that maximizes a given performance index defined over the task set. This paper greatly simplifies the approach proposed by Bini and Buttazzo [9] and it provides deeper insights. Nonetheless also a convex restriction of the feasibility region was described [9].

## 2 Terminology and Notation

We now present the adopted terminology and notation.

The execution platform is a single processor and the preemption is allowed.

We consider a set $\mathcal{T}$ of $n$ *periodic tasks*. Each task, denoted by $\tau_i$, is activated periodically over time. The amount of work that must be executed by a task at each activation is called *job*. All the jobs of the task $\tau_i$ have the same *execution time* $C_i$ and are activated every *period* $T_i$. The jobs of $\tau_i$ must complete not later than $D_i$ time units after the activation ($D_i$ is called relative deadline or, sometimes, simply *deadline*). We say that the task set $\mathcal{T}$ has *implicit deadlines* if $D_i = T_i$, *constrained deadlines* if $D_i \leq T_i$, or *arbitrary deadlines* if the deadlines are not constrained. In this paper we assume arbitrary deadlines. Sometime we denote the task $\tau_i$ also by the triplet $(C_i, T_i, D_i)$ and the task set $\mathcal{T}$ by $(\mathbf{C}, \mathbf{T}, \mathbf{D})$, which are the vectors of computation times, periods, and deadlines.

The *critical scenario* of activations (i.e. if no deadline is missed in the critical scenario then no deadline is ever missed) occurs when all the tasks start activating jobs at the same time. It is then convenient to label by 0 the instant when all the tasks start activating their first job. Given these hypothesis, it follows that the activation of the $j^{\text{th}}$ job of $\tau_i$ occurs at $a_{ij} = (j-1)T_i$ and its *absolute deadline* $d_{ij}$ is at $d_{ij} = (j-1)T_i + D_i$.

Another significant feature related to the task $\tau_i$ is its *utilization* $U_i = \frac{C_i}{T_i}$. It represents the fraction of time that is required by the task. The sum of all the task utilizations $U = \sum_{i=1}^{n} U_i$ is the *total utilization* of the task set. It is straightforward to see that if $U > 1$, then some deadline is going to be missed because the processor is overloaded.

All these parameters ($C_i$, $T_i$, $D_i$) are real-valued. We extend the notion of least common multiple also to real numbers meaning that, given $a, b \in \mathbb{R}$ then

$$\mathsf{lcm}(a, b) = \inf\{x \in \mathbb{R} : \exists p, q \in \mathbb{N}_+ \; x = p\,a = q\,b\}$$

We schedule the jobs by EDF [10]. EDF assigns the highest priority to the job which has the earliest absolute deadline $d_{ij}$. If two jobs have the same absolute deadline, tie is broken arbitrarily.

## 3 Standard Schedulability Analysis in EDF

The schedulability analysis is developed to check whether some deadlines can be missed or not. The classic way of performing schedulability analysis is based on the "demand-supply" approach. In this approach, it is required that the work demanded by the task set does not exceed the time supplied by the execution platform.

In the EDF scheduling algorithm, the demand is given by amount of computation required by all the jobs whose activation $a_{ij}$ and absolute deadline $d_{ij}$ can fit in an interval long $t$. Since the activation $a_{i1}$ of the first job of $\tau_i$ occurs at 0, the maximum number of $\tau_i$ jobs that can fit the interval $[0, t]$ is

$$\max\left\{0, \left\lfloor \frac{t + T_i - D_i}{T_i} \right\rfloor\right\}$$

Figure 1 shows the number of jobs given by the previous expression as $t$ varies.

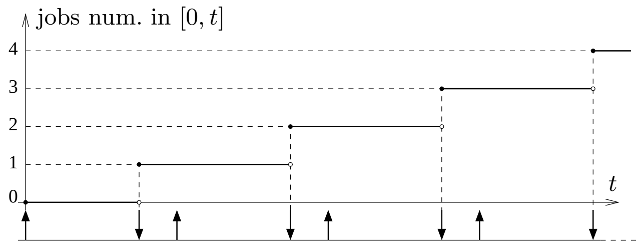

**Fig. 1.** Number of jobs of a single task.

Since the critical scenario occurs when all the tasks are activated simultaneously the demand of the task set is given by the sum of the demand of each task

$$\sum_{i=1}^{n} \max\left\{0, \left\lfloor \frac{t + T_i - D_i}{T_i} \right\rfloor\right\} C_i$$

Hence the next theorem provides a necessary and sufficient condition for the EDF schedulability.

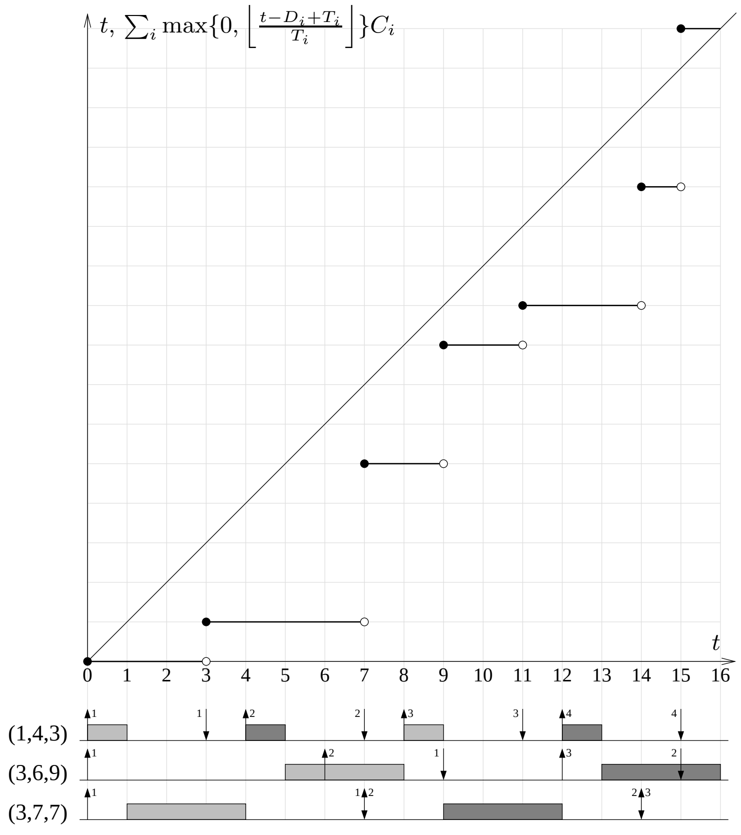

**Fig. 2.** The EDF schedulability test.

**Theorem 1 (Lemma 3 in [11]).** *The task set* $\mathcal{T} = \{(C_i, T_i, D_i) : i = 1, \ldots, n\}$ *is schedulable by EDF* ***if and only if****:*

$$\forall t \geq 0 \quad \sum_{i=1}^{n} \max\left\{0, \left\lfloor \frac{t + T_i - D_i}{T_i} \right\rfloor\right\} C_i \leq t \tag{1}$$

In Figure 2 we provide also a graphical representation of Theorem 1. In the figure, the task parameters are reported in the $(C_i, T_i, D_i)$ format on the left of the task schedule. The activations are labelled by the index of the task job on the right of the upward arrow. Similarly the deadlines are labelled by the job index on the left of the corresponding downward arrow. Adjacent jobs are colored by alternative colors.

In the scenario depicted in Figure 2 it can be noticed that at time 15 the second job of task $\tau_2$ misses its deadline. This deadline miss is revealed by Theorem 1. In fact, it can be seen in the figure that the Equation (1) is not verified at time 15.

The necessary and sufficient condition of Eq. (1) is clearly impractical, since it requires to check an infinite number of inequalities. Several works addressed the problem of reducing the number of instants where the inequality of Eq. (1) can be checked while preserving the necessity of the condition. First it can be observed that for any pair of adjacent absolute deadlines $d_a$ and $d_b$, if Eq. (1) is true at $d_a$ then it is also true $\forall t \in [d_a, d_b[$, because the left hand side of Eq. (1) remains constant whereas the right hand side increases. This allows to restrict the test to the set of all the absolute deadlines. Using upper bounds of the demand the set of deadlines can be further reduced to a finite set [11].

## 4 Integer Problem Formulation

Very often, some of the task parameters can be freely selected by the designer. For example, in control systems the algorithm that needs to be executed is known, whereas the period and the deadline of the control task can be programmed. In this circumstance, the designer desires a characterization of the feasible assignment of $T_i$ and $D_i$ given the value of the computation times $C_i$. Sometime it is also possible that the performance of the system can be expressed as a function of periods and deadlines. In this case the design problem becomes an optimization problem, where the variables are periods and deadlines, the goal function is the system performance, and the constraint is given by the condition of EDF schedulability.

Unfortunately, the condition expressed by Theorem 1 is not well suited to be used in optimizations, because the presence of the floor $\lfloor \cdot \rfloor$ operator breaks any property of the constraints that is desirable for optimization (such as linearity). For this reason, some efforts have been devoted to the derivation of alternative way to formulate the necessary and sufficient condition for EDF schedulability.

The following Theorem provides a convenient way to formulate an equivalent condition of Theorem 1 that is expressed by a combination of linear constraints.

**Theorem 2.** *The task set $\mathcal{T} = \{(C_i, T_i, D_i) : i = 1, \ldots, n\}$ is schedulable by EDF **if and only if**:*

$$\forall \mathbf{k} \in \mathbb{N}^n \setminus \{\mathbf{0}\} \quad \exists i \in I_{\mathbf{k}} \quad (T_i - C_i)k_i - \sum_{j \neq i} C_j k_j \geq T_i - D_i \tag{2}$$

*where*

$$I_{\mathbf{k}} = \{j : k_j \neq 0\}$$

*is the set of non-zero indexes in* $\mathbf{k}$.

*Proof.* We will prove that Equations (1) and (2) are equivalent.

Eq. (1) $\Rightarrow$ Eq. (2): we are given a vector $\mathbf{k} \in \mathbb{N}^n$ different that $\mathbf{0} \in \mathbb{N}^n$ and we must find an index $i$ in $I_{\mathbf{k}} = \{j : k_j \neq 0\}$ using the hypothesis of Eq. (1).

Since $\mathbf{k} \neq \mathbf{0}$ then $I_{\mathbf{k}} \neq \emptyset$. Let us define

$$\forall j \in I_{\mathbf{k}} \qquad d_j = D_j + (k_j - 1)T_j$$

which is the absolute deadline of the $k_j$ job of $\tau_j$. Notice that from $k_j \geq 1$ it follows that $d_j \geq 0$.

We claim that the index $i$ satisfying Eq. (2) is such that

$$d_i = \max_{j \in I_{\mathbf{k}}} \{d_j\} \tag{3}$$

This value is well defined because $I_{\mathbf{k}} \neq \emptyset$.

Now we exploit the Equation (1) for $t = d_i$. We have

$$\begin{aligned}
&\sum_{j=1}^{n} \max\left\{0, \left\lfloor \frac{d_i + T_j - D_j}{T_j} \right\rfloor\right\} C_j \leq d_i \\
&\max\left\{0, \left\lfloor \frac{d_i + T_i - D_i}{T_i} \right\rfloor\right\} C_i + \sum_{j \neq i} \max\left\{0, \left\lfloor \frac{d_i + T_j - D_j}{T_j} \right\rfloor\right\} C_j \leq d_i \\
&\max\{0, k_i\} C_i + \sum_{j \neq i} \max\left\{0, \left\lfloor \frac{d_i + T_j - D_j}{T_j} \right\rfloor\right\} C_j \leq D_i + (k_i - 1)T_i \\
&(T_i - C_i)k_i - \sum_{j \neq i} \max\left\{0, \left\lfloor \frac{d_i + T_j - D_j}{T_j} \right\rfloor\right\} C_j \geq T_i - D_i \qquad (4)
\end{aligned}$$

Equation (4) is quite similar to Equation (2) that we want to prove. To conclude the demonstration we need to find a lower bound of $\max\left\{0, \left\lfloor \frac{d_i + T_j - D_j}{T_j} \right\rfloor\right\}$. We have

$$\begin{aligned}
\forall j \in I_{\mathbf{k}} \quad \max\left\{0, \left\lfloor \frac{d_i + T_j - D_j}{T_j} \right\rfloor\right\} &\geq \left\lfloor \frac{d_i + T_j - D_j}{T_j} \right\rfloor \geq \\
&\geq \left\lfloor \frac{d_j + T_j - D_j}{T_j} \right\rfloor = k_j \qquad (5)
\end{aligned}$$

Similarly

$$\forall j \notin I_{\mathbf{k}} \quad \max\left\{0, \left\lfloor \frac{d_i + T_j - D_j}{T_j} \right\rfloor\right\} \geq 0 = k_j \tag{6}$$

Finally, from Equation (4) by using the lower bounds of Equations (5) and (6), it follows that for the index $i$ selected such that Eq. (3) holds, we have

$$(T_i - C_i)k_i - \sum_{j \neq i} C_j k_j \geq (T_i - C_i)k_i - \sum_{j \neq i} \max\left\{0, \left\lfloor \frac{d_i + T_j - D_j}{T_j} \right\rfloor\right\} C_j \geq T_i - D_i$$

as required.

Eq. (2) $\Rightarrow$ Eq. (1). Given $t \geq 0$, we build the mapping $t \mapsto \mathbf{k} = (k_1, \ldots, k_n) \in \mathbb{N}^n$ defined by

$$k_j = \max\left\{0, \left\lfloor \frac{t + T_j - D_j}{T_j} \right\rfloor\right\}$$

Given this mapping, the Equation (1) that needs to be proved can be rewritten as

$$\forall t \geq 0 \quad \sum_{j=1}^{n} k_j C_j \leq t \tag{7}$$

For all the $t \geq 0$ such that $\mathbf{k} = \mathbf{0}$ we have

$$\sum_{j=1}^{n} k_j C_j = 0 \leq t$$

On the other hand, for all the $t \geq 0$ such that $\mathbf{k} \neq \mathbf{0}$ we can invoke Equation (2) from which it follows that we have an index $i \in I_{\mathbf{k}}$ such that

$$(T_i - C_i)k_i - \sum_{j \neq i} C_j k_j \geq T_i - D_i$$

$$\sum_{j=1}^{n} C_j k_j \leq (k_i - 1)T_i + D_i = \left( \left\lfloor \frac{t + T_i - D_i}{T_i} \right\rfloor - 1 \right) T_i + D_i \leq \frac{t - D_i}{T_i} T_i + D_i$$

$$\sum_{j=1}^{n} C_j k_j \leq t$$

Hence Equation (7) is demonstrated and the Theorem is proved.

Formulating the EDF schedulability condition through Theorem 2 allows an easier characterization of the region of the feasible parameters of the task set. For example, from Theorem 2 it follows directly that a task set $\mathcal{T}$ is schedulable by EDF if and only if

$$\mathcal{T} \in \bigcap_{\mathbf{k} \in \mathbb{N}^n \setminus \{\mathbf{0}\}} \bigcup_{i \in I_{\mathbf{k}}} \left\{ (\mathbf{C}, \mathbf{T}, \mathbf{D}) : (T_i - C_i)k_i - \sum_{j \neq i} C_j k_j \geq T_i - D_i \right\} \tag{8}$$

Below we examine in details some special cases when the information on the task set are partially known.

## 5 The region of feasible deadlines

We consider the problem of determining the region of the feasible task deadlines, given the computation times $\mathbf{C}$ and the periods $\mathbf{T}$. This problem can arise when the designer has to assign the deadlines to the tasks in order to minimize the response times or the jitter in the task execution. In fact response times and jitter can heavily affect the performance, for example, in control systems [12].

Since the computation times and the periods are known, it is convenient to rewrite Equation (8) by isolating the free deadline variables. We can then say that the task set $\mathcal{T}$ is schedulable by EDF if and only if

$$\mathbf{D} \in \mathsf{domD} = \bigcap_{\mathbf{k} \in \mathbb{N}^n \setminus \{\mathbf{0}\}} \bigcup_{i \in I_{\mathbf{k}}} \{ D_i \geq \mathbf{k} \cdot \mathbf{C} - (k_i - 1)T_i \} \tag{9}$$

where $\mathsf{domD}$ is then the region of all the deadlines that make the task set feasible. Since the expression of right hand side of Eq. (9) is a lower bound on the deadline, then we can compactly call it $D_i^{\text{lb}}$. Hence the region of feasible deadlines $\mathsf{domD}$ becomes

$$\mathsf{domD} = \bigcap_{\mathbf{k}\in\mathbb{N}^n\setminus\{\mathbf{0}\}} \bigcup_{i=1,\ldots,n} \{\mathbf{D} \in \mathbb{R}^n : D_i \geq D_i^{\text{lb}}\} \tag{10}$$

where $D_i^{\text{lb}}$ is defined as

$$D_i^{\text{lb}}(\mathbf{k}) = \begin{cases} \mathbf{k}\cdot\mathbf{C} - (k_i - 1)T_i & \text{if } k_i \neq 0 \\ +\infty & \text{if } k_i = 0 \end{cases} \tag{11}$$

To gain some knowledge on the geometry of $\mathsf{domD}$ we propose an example. Let us consider a set of two tasks with computation times $\mathbf{C} = (2, 3)$ and periods $\mathbf{T} = (4, 7)$. By applying Eq. (11), we can compute the deadline lower bounds $D_i^{\text{lb}}$ associated to some integer vectors purposely selected for the example. For each selected vector $\mathbf{k}$, Table 1 shows the two resulting coordinates of the corresponding bounds $\mathbf{D}^{\text{lb}}(\mathbf{k})$ derived by Eq. (11). In Figure 3 we draw the six regions

| $\mathbf{k} = (k_1, k_2)$ | $D_1^{\text{lb}}(\mathbf{k})$ | $D_2^{\text{lb}}(\mathbf{k})$ |
|---|---|---|
| $(1, 0)$ | $C_1 = 2$ | $+\infty$ |
| $(0, 1)$ | $+\infty$ | $C_2 = 3$ |
| $(1, 1)$ | $C_1 + C_2 = 5$ | $C_1 + C_2 = 5$ |
| $(2, 1)$ | $2C_1 + C_2 - T_1 = 3$ | $2C_1 + C_2 = 7$ |
| $(0, 2)$ | $+\infty$ | $2C_2 - T_2 = -1$ |
| $(1, 2)$ | $C_1 + 2C_2 = 8$ | $C_1 + 2C_2 - T_2 = 1$ |

**Table 1.** Vertexes for the sample task set.

corresponding to the $\mathbf{k}$ values selected in Table 1. It can be noticed that when both $k_1$ and $k_2$ are different than zero then the resulting region is the union of two half-planes. When there is only one non-zero coordinate in $\mathbf{k}$ then the constraint becomes an half-plane. Finally notice that the constraints corresponding to $\mathbf{k} = (1, 0)$ and $\mathbf{k} = (0, 1)$ are the (necessary) conditions $D_1 \geq C_1$ and $D_2 \geq C_2$ respectively.

Finally, when intersecting all the regions of Figure 3 as required by the definition of $\mathsf{domD}$ (see Eq. (10) then we obtain the region shown in Fig. 4. Notice that the exact $\mathsf{domD}$ resulting from Eq. (10) can be smaller than the region shown in Figure 4 because Eq. (10) requires to intersect regions such as in Fig. 3 for all $\mathbf{k} \in \mathbb{N}^n$, whereas Figure 4 is derived for a subset of integer vectors.

Moreover it can be noticed that the first four vertexes *dominate* the others, meaning that the region described by them is not restricted by the other constraints.

This example suggests that the exact feasibility region $\mathsf{domD}$ may be simply computed from a finite set of integer vectors. Section 5.1 provides a method for determining such a set, by removing redundant points.

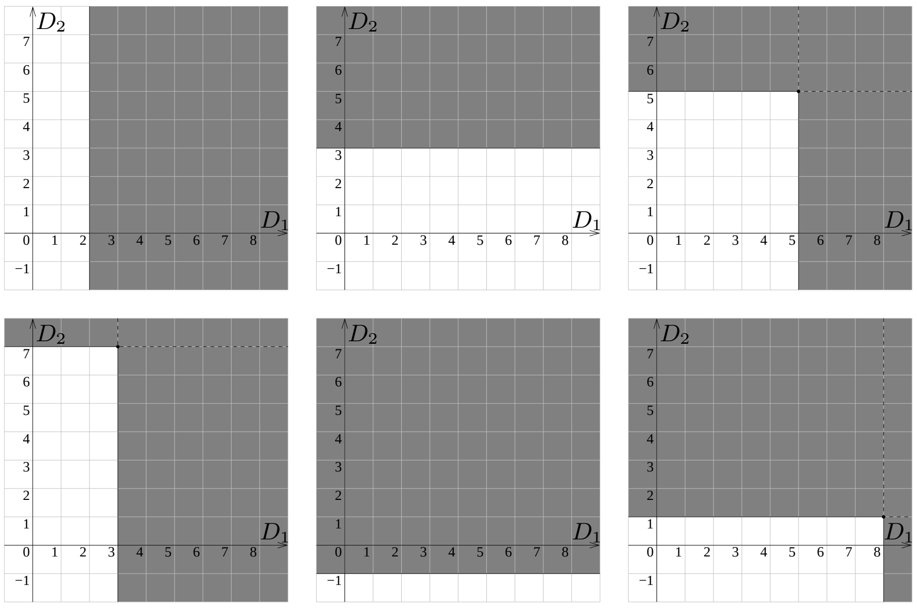

**Fig. 3.** Deadlines satisfying the condition for the values of $\mathbf{k}$ in Table 1.

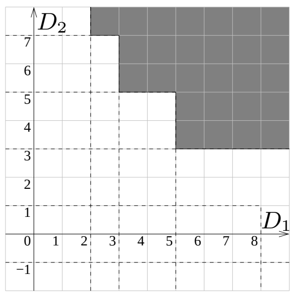

**Fig. 4.** $\mathsf{domD}$ when $\mathbf{k} \in \{(1,0),(0,1),(1,1),(0,2),(1,2),(2,1)\}$.

### 5.1 Reducing the set of k's

The derivation of the feasible deadline space based on Eq. (10) is still infeasible, because it requires the intersection of infinite regions, derived from all possible $\mathbf{k}$'s ranging in $\mathbb{N}^n \setminus \{\mathbf{0}\}$. However, as we observed in the previous section, a region associated to a vector $\hat{\mathbf{k}}$ can be ignored if $\hat{\mathbf{k}}$ is *dominated* by some other integer vector $\mathbf{k}$. For instance, in Figure 4, $(1,2)$ is dominated by $(0,1)$. We start by defining formally the idea of domination.

**Definition 1.** *We say that an integer vector* $\hat{\mathbf{k}}$ *is* dominated *by the integer vector* $\mathbf{k}$ *if*

$$\forall i = 1,\ldots,n \qquad D_i^{\mathrm{lb}}(\hat{\mathbf{k}}) \le D_i^{\mathrm{lb}}(\mathbf{k}). \tag{12}$$

In fact, if all the coordinates of $\mathbf{D}^{\mathrm{lb}}(\hat{\mathbf{k}})$ are smaller than or equal to the corresponding coordinates of $\mathbf{D}^{\mathrm{lb}}(\mathbf{k})$, then the intersection with the region associated with $\hat{\mathbf{k}}$ cannot eliminate any portion of the space of deadlines that has not already been eliminated by the region associated with $\mathbf{k}$. Below we will exploit this property to make an effective computation of the deadline space.

Now we translate Eq. (12) as a constraint on the integers $\mathbf{k}$ and $\hat{\mathbf{k}}$.

$$\begin{aligned}
&\begin{cases} \forall i \in I_{\mathbf{k}} & D_i^{\mathrm{lb}}(\hat{\mathbf{k}}) \le D_i^{\mathrm{lb}}(\mathbf{k}) \\ \forall i \notin I_{\mathbf{k}} & D_i^{\mathrm{lb}}(\hat{\mathbf{k}}) \le D_i^{\mathrm{lb}}(\mathbf{k}) \end{cases} \\
&\begin{cases} \forall i \in I_{\mathbf{k}} & D_i^{\mathrm{lb}}(\hat{\mathbf{k}}) \le \mathbf{k}\cdot\mathbf{C} - (k_i - 1)T_i \\ \forall i \notin I_{\mathbf{k}} & D_i^{\mathrm{lb}}(\hat{\mathbf{k}}) \le +\infty \end{cases} \\
&\forall i \in I_{\mathbf{k}} \quad \hat{\mathbf{k}}\cdot\mathbf{C} - (\hat{k}_i - 1)T_i \le \mathbf{k}\cdot\mathbf{C} - (k_i - 1)T_i \\
&\forall i \in I_{\mathbf{k}} \quad (\hat{k}_i - k_i)(T_i - C_i) - \sum_{j\neq i}(\hat{k}_j - k_j)C_j \ge 0
\end{aligned} \tag{13}$$

For a given $\mathbf{k}$, Eq. (13) describes the region of the integer vectors $\hat{\mathbf{k}}$ that are dominated by $\mathbf{k}$. Such a region is a cone whose vertex is at $\mathbf{k}$, which can be formally defined by

$$\mathsf{cone}(\mathbf{k}, I_{\mathbf{k}}) = \{\hat{\mathbf{k}} \in \mathbb{N}^n : \forall i \in I_{\mathbf{k}} \ (\hat{k}_i - k_i)(T_i - C_i) - \sum_{j\neq i}(\hat{k}_j - k_j)C_j \ge 0\} \tag{14}$$

These cones are important because they allow a significant reduction of the number of integer vectors to be tested. In fact, suppose there is a finite subset of integer vectors $\mathsf{domK} \subseteq \mathbb{N}^n \setminus \{\mathbf{0}\}$ such that $\mathbb{N}^n \setminus \{\mathbf{0}\}$ can be covered by cones with vertex in $\mathsf{domK}$ as follows

$$\mathbb{N}^n \setminus \{\mathbf{0}\} = \bigcup_{\mathbf{k}\in\mathsf{domK}} \mathsf{cone}(\mathbf{k}, I_{\mathbf{k}}) \tag{15}$$

then $\mathsf{domD}$ can be **equivalently** defined as

$$\mathsf{domD} = \bigcap_{\mathbf{k}\in\mathsf{domK}} \ \bigcup_{i=1,\ldots,n} \{\mathbf{D} \in \mathbb{R}^n : D_i \ge D_i^{\mathrm{lb}}\} \tag{16}$$

In fact, the vectors in $\mathsf{domK}$ are explicitly checked by the test of Eq. (16), whereas the vectors in

$$\bigcup_{\mathbf{k}\in\mathsf{domK}} (\mathsf{cone}(\mathbf{k}, I_{\mathbf{k}}) \setminus \{\mathbf{k}\})$$

which are the points in the cones except the vertexes, are not required to be tested because dominated by some vector in $\mathsf{domK}$.

This remarkable property deserves a definition.

**Definition 2.** *We say that a set* $\mathsf{domK} \subseteq \mathbb{N}^n \setminus \{\mathbf{0}\}$ *is* dominant *when all the integers* $\hat{\mathbf{k}} \in \mathbb{N}^n \setminus \{\mathbf{0}\}$ *are dominated by some integer* $\mathbf{k} \in \mathsf{domK}$.

Figure 5 shows three examples of $\mathsf{cone}(\mathbf{k}, I_{\mathbf{k}})$, when $n = 2$. The black dots represent three possible vertexes of the cones (resp. $(0, 2)$, $(1, 1)$, and $(4, 0)$), whereas the white dots represent the sets of integers which are dominated by $\mathsf{cone}((0, 2), \{2\})$, $\mathsf{cone}((1, 1), \{1, 2\})$, and $\mathsf{cone}((4, 0), \{1\})$ respectively.

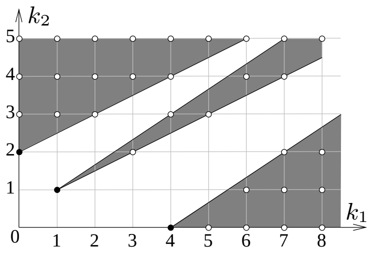

**Fig. 5.** Examples on $\mathsf{cone}(\mathbf{k}, I_{\mathbf{k}})$.

The key problem is then to find a small dominant $\mathsf{domK}$, since a small $\mathsf{domK}$ results in a faster computation of $\mathsf{domD}$ from Eq. (16). It is quite clear that for large cones, a dominant set $\mathsf{domK}$ can be smaller, because a large cone allows dominating more integer vectors, and consequently the number of integer vectors that need to be explicitly checked is smaller. Hence we proceed by studying in depth the cones.

To derive the properties of the cones, it can be noticed from Eq. (14) that the coefficients of the hyperplanes delimiting the cone do not change with $\mathbf{k}$, meaning that the "shape" of the cone remains unaltered for the same set of non-zero indexes $I_{\mathbf{k}}$. More formally, we can state the property of "invariance for translation" as follows

$$\mathsf{cone}(\mathbf{k}, I_{\mathbf{k}}) = \mathsf{cone}(\mathbf{0}, I_{\mathbf{k}}) + \mathbf{k} \tag{17}$$

with the implicit meaning that the plus sign at the right-hand side of Eq. (17) means a translation of the set. Due to this translation property, we can restrict the study to the cone placed at the origin $\mathsf{cone}(\mathbf{0}, I)$ and then extend the results to the other cones by translation.

The following theorem allows the definition of a finite dominant subset $\mathsf{domK}$, starting from some vector $\mathbf{k}^{\max} \in \mathsf{cone}(\mathbf{0}, \{1, \ldots, n\})$.

**Theorem 3.** *Let* $\mathbf{k}^{\max} \in \mathsf{cone}(\mathbf{0}, \{1, \ldots, n\})$, $\mathbf{k}^{\max} \neq \mathbf{0}$. *Then*

$$\mathsf{domK} = \{\mathbf{k} \in \mathbb{N}^n \setminus \{\mathbf{0}\} : 0 \le k_i \le k_i^{\max}\} \tag{18}$$

*satisfies the covering property of Eq. (15).*

*Proof.* We have to prove that $\mathbb{N}^n \setminus \{\mathbf{0}\}$ can be covered by cones whose vertexes are in $\mathsf{domK}$. Given $\hat{\mathbf{k}} \in \mathbb{N}^n \setminus \{\mathbf{0}\}$ we will build the proper $\mathbf{k} \in \mathsf{domK}$ such that $\hat{\mathbf{k}} \in \mathsf{cone}(\mathbf{k}, I_{\mathbf{k}})$.

Through the euclidean division, let's define $q_i$ and $r_i$ as follows

$$\forall i = 1, \ldots, n \quad \hat{k}_i = q_i k_i^{\max} - r_i \qquad q_i \in \mathbb{N},\ 0 \le r_i \le k_i^{\max} - 1$$

and let us set $q = \max_i\{q_i\}$. Since $\hat{\mathbf{k}} \neq \mathbf{0}$ we have $q \ge 1$. Let us also set $I = \{i = 1, \ldots, n : q_i = q\}$. We claim that the cone vertex $\mathbf{k}$ such that $\hat{\mathbf{k}} \in \mathsf{cone}(\mathbf{k}, I_{\mathbf{k}})$ is defined as follows

$$\begin{cases} k_i = k_i^{\max} - r_i & \text{if } i \in I \\ k_i = 0 & \text{if } i \notin I \end{cases}.$$

First of all, we verify that $\mathbf{k} \in \mathsf{domK}$. Since $0 \le r_i \le k_i^{\max}$, it follows that $1 \le k_i \le k_i^{\max}$. Moreover $\mathbf{k} \neq \mathbf{0}$ because $I$ is not empty. A posteriori we verify that such a definition of $\mathbf{k}$ allows asserting that $I = \{i : k_i \neq 0\} = I_{\mathbf{k}}$. In fact

$$\begin{aligned} &i \in I \ \Rightarrow\ k_i = k_i^{\max} - r_i \ \Rightarrow\ k_i \neq 0 \ \Rightarrow\ i \in I_{\mathbf{k}} \\ &i \notin I \ \Rightarrow\ k_i = 0 \ \Rightarrow\ i \notin I_{\mathbf{k}} \end{aligned}$$

Finally, we verify that the constructed $\mathbf{k}$ dominates $\hat{\mathbf{k}}$, as required.

$$\hat{\mathbf{k}} \in \mathsf{cone}(\mathbf{k}, I_{\mathbf{k}}) = \mathsf{cone}(\mathbf{k}, I) \ \Leftrightarrow\ \forall i \in I\ (\hat{k}_i - k_i)(T_i - C_i) - \sum_{j \neq i} (\hat{k}_j - k_j) C_j \ge 0. \tag{19}$$

We will proceed by finding lower estimates of the previous inequality, for all $i \in I$.

$$\begin{aligned}
&(\hat{k}_i - k_i)(T_i - C_i) - \sum_{\substack{j \neq i \\ j \in I}} (\hat{k}_j - k_j) C_j - \sum_{\substack{j \neq i \\ j \notin I}} (\hat{k}_j - k_j) C_j = \\
&(q_i k_i^{\max} - r_i - (k_i^{\max} - r_i))(T_i - C_i) - \sum_{\substack{j \neq i \\ j \in I}} (q_j k_j^{\max} - r_j - k_j) C_j - \sum_{\substack{j \neq i \\ j \notin I}} \hat{k}_j C_j = \\
&(q-1) k_i^{\max}(T_i - C_i) - (q-1) \sum_{\substack{j \neq i \\ j \in I}} k_j^{\max} C_j - \sum_{\substack{j \neq i \\ j \notin I}} \hat{k}_j C_j \ge \\
&(q-1) k_i^{\max}(T_i - C_i) - (q-1) \sum_{\substack{j \neq i \\ j \in I}} k_j^{\max} C_j - \sum_{\substack{j \neq i \\ j \notin I}} (q-1) k_j^{\max} C_j = \\
&(q-1) \left( k_i^{\max}(T_i - C_i) - \sum_{j \neq i} k_j^{\max} C_j \right) \ge 0
\end{aligned}$$

because $q \ge 1$ and $\mathbf{k}^{\max} \in \mathsf{cone}(\mathbf{0}, \{1, \ldots, n\})$. Hence the inequality of Eq. (19) is proved and the theorem follows.

Theorem 3 allows reducing the problem of computing the EDF feasible deadlines $\mathsf{domD}$ to the problem of finding an integer vector $\mathbf{k}^{\max}$ in $\mathsf{cone}(\mathbf{0}, \{1, \ldots, n\})$ other than the vertex $\mathbf{0}$. If this point is found, then the Equation (16) allows computing the space of EDF feasible deadlines. Unfortunately, the complexity is still proportional to the cardinality of $\mathsf{domK}$, which is $\prod_{i=1}^{n}(k_i^{\max}+1)-1$. How do we search for a suitable point $\mathbf{k}^{\max}$ that minimizes the complexity? Does $\mathbf{k}^{\max}$ always exist?

We propose to span onto all the integer vectors in $\mathsf{cone}(\mathbf{0},\{1,\dots,n\})$, moving to the direction of the vertex at $\mathbf{0}$. This strategy can be implemented by an Integer Linear Programming (ILP) problem.

The first constraint of the ILP problem must translate the property that $\mathbf{k} \in \mathsf{cone}(\mathbf{0},\{1,\dots,n\})$. From Eq. (14), it follows that the constraint is expressed by the inequality

$$\forall i = 1,\dots,n \quad k_i(T_i - C_i) - \sum_{j \neq i} k_j C_j \geq 0. \tag{20}$$

Then a second constraint must erase the vertex from the cone because we require (see Theorem 3) that $\mathbf{k} \neq \mathbf{0}$. Hence we add

$$\sum_{i=1}^{n} k_i \geq 1 \tag{21}$$

which erases only $\mathbf{0}$ from $\mathbb{N}^n$.

Finally, we set the minimization direction for reducing the cardinality of $\mathsf{domK}$. The goal of the problem is to minimize $\prod_i (k_i+1)$. However this function is not well suited for the ILP problem because it is not linear. Hence we choose the following convenient linear cost function

$$\sum_{i=1}^{n} k_i \tag{22}$$

which approximates $|\mathsf{domK}|$ at the first order in the point $\mathbf{0}$.

*An example* Now, it is worth showing how the previous result can be applied to compute a dominant set $\mathsf{domK}$ for the sample task set used in this paper. When $\mathbf{C} = (2,3)$ and $\mathbf{T} = (4,7)$, the solution of the ILP problem is $\mathbf{k}^{\max} = (2,1)$. In Figure 6, black dots represent the integers that belong to $\mathsf{domK}$ as defined by Eq. (18). For each $\mathbf{k} \in \mathsf{domK}$ we also draw the corresponding $\mathsf{cone}(\mathbf{k}, I_{\mathbf{k}})$. It can be seen that all the integers are dominated by some $\mathbf{k} \in \mathsf{domK}$, as proved in Theorem 3. It can also be noticed that $\mathsf{cone}((2,0),\{1\}) \subseteq \mathsf{cone}((1,0),\{1\})$ meaning that $\mathsf{domK} \setminus \{(2,0)\}$ is dominant as well. This allows us to assert that the region of feasible deadline $\mathsf{domD}$ that was drawn in Figure 4 is exact, since $\mathsf{domK}$ is contained in the set of integer vectors $\mathbf{k}$ that was selected for drawing the figure.

In the next subsection we investigate what are the parameters affecting the complexity of a dominant set $\mathsf{domK}$, computed through the ILP formulation of Equations (20), (21), and (22).

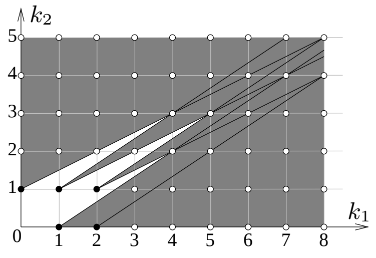

**Fig. 6.** An example of domK.

## 5.2 Complexity of domK

From the Figures 5 and 6 it is quite clear that the solution of the ILP problem described in Equations (20), (21), and (22) depends on the width of the cone. The wider the cone, the smaller the solution, and consequently the fewer points in domK.

For measuring the cone width we translated the $n$ hyperplanes of the boundary in such way the the $i^{\text{th}}$ hyperplane passes through the vector $\mathbf{e}_i$, which has all zeros except in the $i^{\text{th}}$, position where it has 1 (see Figure 7 for a graphical representation). Then, we computed the intersection point $\mathbf{k}^{\text{int}}$: if $\mathbf{k}^{\text{int}}$ has large

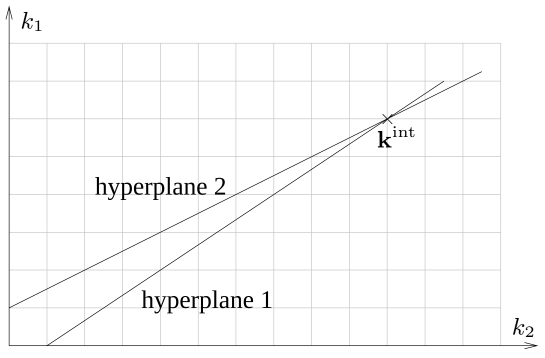

**Fig. 7.** Finding the width of the cone.

coordinates then the cone is narrow, and vice versa. The coordinates of $\mathbf{k}^{\text{int}}$ are the solution of the following linear system:

$$\begin{pmatrix} T_1 - C_1 & -C_2 & \cdots & -C_n \\ -C_1 & T_2 - C_2 & \cdots & -C_n \\ \vdots & \vdots & \ddots & \vdots \\ -C_1 & -C_2 & \ldots & T_n - C_n \end{pmatrix} \cdot \mathbf{k}^{\text{int}} = \begin{pmatrix} T_1 - C_1 \\ T_2 - C_2 \\ \vdots \\ T_n - C_n \end{pmatrix}$$

and, by the Cramer's rule, we find

$$k_i^{\text{int}} = \frac{(T_i - C_i)(1 - \sum_{j\neq i} U_j) + \sum_{j\neq i} C_j(1 - U_j)}{T_i(1-U)}.$$

The coordinates of $\mathbf{k}^{\text{int}}$ are inversely proportional to $1-U$, meaning that the set $\mathsf{domK}$ grows as $U$ approaches 1. This means that, for small values of $U$, the size of $\mathsf{domK}$ is small. On the other hand, as the total utilization $U$ approaches 1, the size of a dominant set $\mathsf{domK}$ increases and, consequently, the complexity of Eq. (16) grows exponentially with $n$.

Note that the dependency of the complexity on the total processor utilization $U$ is also typical of the processor demand test proposed by Baruah et al. [11], which requires to test all the absolute deadline not exceeding $\frac{\sum_i (T_i - D_i)U_i}{1-U}$. This confirms that, as the total utilization $U$ approaches 1, the EDF schedulability guarantee tests become intrinsically more complex.

*$U$ approaching* 1 We already observed that as $U$ approaches 1, the cone $\mathsf{cone}(\mathbf{0}, \{1,\ldots,n\})$ becomes narrower. When $U$ is exactly 1, the cone becomes a line. Let us consider more closely the case when $n = 2$ and $U = U_1 + U_2 = 1$. When $I_{\mathbf{k}} = \{1\}$, the constraints of the cone that describes the $\hat{\mathbf{k}} \in \mathbb{N}^2$ dominated by some $\mathbf{k}$ (Eq. (14)) becomes

$$\begin{aligned}\hat{\mathbf{k}} \in \mathsf{cone}(\mathbf{k}, \{1\}) &\Leftrightarrow (\hat{k}_1 - k_1)(T_1 - C_1) - (\hat{k}_2 - k_2)C_2 \geq 0\\ &\Leftrightarrow T_1(\hat{k}_1 - k_1)(1 - U_1) - T_2(\hat{k}_2 - k_2)U_2 \geq 0\\ &\Leftrightarrow T_1(\hat{k}_1 - k_1) - T_2(\hat{k}_2 - k_2) \geq 0 \qquad (23)\end{aligned}$$

Similarly, when $I_{\mathbf{k}} = \{2\}$ we have

$$\begin{aligned}\hat{\mathbf{k}} \in \mathsf{cone}(\mathbf{k}, \{2\}) &\Leftrightarrow (\hat{k}_2 - k_2)(T_2 - C_2) - (\hat{k}_1 - k_1)C_1 \geq 0\\ &\Leftrightarrow T_1(\hat{k}_1 - k_1) - T_2(\hat{k}_2 - k_2) \leq 0 \qquad (24)\end{aligned}$$

Finally, when $I_{\mathbf{k}} = \{1,2\}$ both Equation (23) and (24) must hold, then we have

$$\hat{\mathbf{k}} \in \mathsf{cone}(\mathbf{k}, \{1,2\}) \Leftrightarrow T_1(\hat{k}_1 - k_1) - T_2(\hat{k}_2 - k_2) = 0 \qquad (25)$$

To clarify the two tasks case we propose a numerical example to show a tight set of integers $\mathsf{domK}$ and the resulting region of feasible deadlines $\mathsf{domD}$. To achieve $U = 1$ we choose $\mathbf{C} = (2, 3.5)$ and $\mathbf{T} = (4, 7)$. For these values the set

$$\begin{aligned}\mathsf{domK} = \{&(0,1),(1,0),(1,1),(2,1),(2,2),(3,2),(4,2),(4,3),\\ &(5,3),(6,3),(6,4),(7,4)\} \qquad (26)\end{aligned}$$

is dominant, as it can be noticed in Figure 8. In the figure we can see that $\mathsf{cone}((1,0),\{1\})$, the lower gray triangle, and $\mathsf{cone}((0,1),\{2\})$, the upper gray triangle, dominate most of the integers (the dominated integer vectors are represented by white dots). The other degenerate cones, represented by lines whose

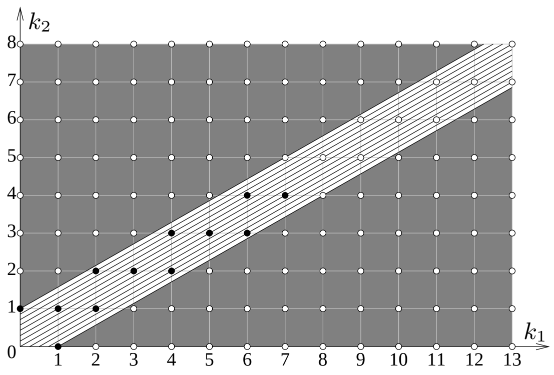

**Fig. 8.** domK when $U = 1$.

equation is Eq. (25), can dominate all the remaining points in $\mathbb{N}^2 \setminus \{(0,0)\}$ when they are placed at the integers in domK (represented by black dots).

We conclude this example by drawing the resulting region of feasible deadlines. Since previously we showed that the set domK as in Eq. (26) is dominant, then it is sufficient to evaluate the deadline values $\mathbf{D}^{\text{lb}}(\mathbf{k})$ for all $\mathbf{k} \in$ domK to have an exact description of the feasible region of deadlines. In Table 2 we

| $\mathbf{k} = (k_1, k_2)$ | $\mathbf{D}^{\text{lb}}(\mathbf{k})$ |
|---|---|
| $(0, 1)$ | $(+\infty, 3.5)$ |
| $(1, 0)$ | $(2, +\infty)$ |
| $(1, 1)$ | $(5.5, 5.5)$ |
| $(2, 1)$ | $(3.5, 7.5)$ |
| $(2, 2)$ | $(7, 4)$ |
| $(3, 2)$ | $(5, 6)$ |
| $(4, 2)$ | $(3, 8)$ |
| $(4, 3)$ | $(6.5, 4.5)$ |
| $(5, 3)$ | $(4.5, 6.5)$ |
| $(6, 3)$ | $(2.5, 8.5)$ |
| $(6, 4)$ | $(6, 5)$ |
| $(7, 4)$ | $(4, 7)$ |

**Table 2.** $\mathbf{D}^{\text{lb}}$ for the example.

computed these deadline and in Figure 9 we report the corresponding region of feasible deadlines.

## References

1. Kalman, R., Bertram, J.: A unified approach to the theory of sampling systems. J. Franklin Inst. **267** (1959) 405–436

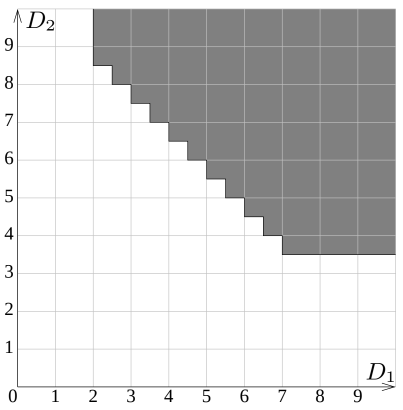

**Fig. 9.** The region of feasible deadline of the example.

2. Kushner, H.J., Tobias, L.: On the stability of randomly sampled systems. IEEE Transactions on Automatic Control **14** (1969) 319–324
3. Davidson, C.: Random sampling and random delays in optimal control. PhD thesis, Department of Optimization and Systems Theory, Royal Institute of Technology, Sweden (1973)
4. Baruah, S.K., Buttazzo, G., Gorinsky, S., Lipari, G.: Scheduling periodic task systems to minimize output jitter. In: Proceedings of the $6^{\text{th}}$ International Conference on Real-Time Computing Systems and Applications, Hong Kong (1999) 62–69
5. Zheng, Q., Shin, K.G.: On the ability of establishing real-time channels in point-to-point packet-switched networks. IEEE Transactions on Communications **42** (1994) 1096–1105
6. Buttazzo, G., Sensini, F.: Optimal deadline assignment for scheduling soft aperiodic task in hard real-time environments. IEEE Transactions on Computers **48** (1999) 1035–1052
7. Balbastre, P., Ripoll, I., Crespo, A.: Optimal deadline assignment for periodic real-time tasks in dynamic priority systems. In: Proceedings of the $18^{\text{th}}$ Euromicro Conference on Real-Time Systems, Dresden, Germany (2006) 65–74
8. Hoang, H., Buttazzo, G., Jonsson, M., Karlsson, S.: Computing the minimum EDF feasible deadline in periodic systems. In: Proceedings of the 12th IEEE International Conference on Embedded and Real-Time Computing Systems and Applications, Sydney, Australia (2006) 125–134
9. Bini, E., Buttazzo, G.: The space of EDF feasible deadlines. In: Proceedings of the 19th Euromicro Conference on Real-Time Systems, Pisa, Italy (2007) 19–28
10. Liu, C.L., Layland, J.W.: Scheduling algorithms for multiprogramming in a hard real-time environment. Journal of the Association for Computing Machinery **20** (1973) 46–61
11. Baruah, S.K., Mok, A.K., Rosier, L.E.: Preemptively scheduling hard-real-time sporadic tasks on one processor. In: Proceedings of the $11^{\text{th}}$ IEEE Real-Time Systems Symposium, Lake Buena Vista (FL), U.S.A. (1990) 182–190
12. Cervin, A., Lincoln, B., Eker, J., Årzén, K.E., Buttazzo, G.: The jitter margin and its application in the design of real-time control systems. In: Proceedings of the $10^{\text{th}}$ International Conference on Real-Time and Embedded Computing Systems and Applications, Göteborg, Sweden (2004)