Worst-Case Utilization Bound for EDF Scheduling on Real-Time Multiprocessor Systems

J.M. López, M. García, J.L. Díaz, D.F. García
University of Oviedo
Department of Computer Science
Campus de Viesques, 33204 Gijón, Spain
$\{$chechu,manuel,jdiaz,daniel$\}$@atc.uniovi.es

Abstract

In this paper we present the utilization bound for Earliest Deadline First (EDF) scheduling on homogeneous multiprocessor systems with partitioning strategies.

Assuming that tasks are pre-emptively scheduled on each processor according to the EDF algorithm, and allocated according to the First Fit (FF) heuristic, we prove that the worst-case achievable utilization is $0.5(n+1)$, where n is the number of processors. This bound is valid for arbitrary utilization factors. Moreover, if all the tasks have utilization factors under a value α, the previous bound is raised, and the new utilization bound considering α is calculated.

In addition, we prove that no pair of uniprocessor scheduling algorithm-allocation algorithm can provide a higher worst-case achievable utilization than that of EDF-FF. Finally, simulation provides the average-case achievable utilization for EDF-FF.

1. Introduction

Dertouzos [3] proved EDF scheduling to be optimal for any kind of tasks, periodic or not, in uniprocessors. Unfortunately, EDF scheduling is not optimal in multiprocessor systems [4].

A new issue arises in multiprocessor scheduling with regard to the uniprocessor case; that is which processor executes each task at a given time. There are two major strategies to deal with this issue: *partitioning strategies*, and *non-partitioning strategies* (also known as global) [7]. In a partitioning strategy, once a task is allocated to a processor, it is executed exclusively on that processor. In a non-partitioning strategy any instance of a task can be executed on a different processor, or even be pre-empted and moved to a different processor, before it is completed.

Partitioning strategies have several advantages over non-partitioning strategies. Firstly, the scheduling overhead associated with a partitioning strategy is lower than the overhead associated with a non-partitioning strategy. Secondly, partitioning strategies allow us to apply well-known uniprocessor scheduling algorithms to each processor.

In this paper, we follow the partitioning strategy, and we assume that all the tasks allocated to a processor are pre-emptively scheduled using EDF, as this is the optimal scheduling algorithm for uniprocessors. Once the scheduler has been chosen, the only degree of freedom is the allocation algorithm.

The problem of allocating a set of tasks to a set of processors is analogous to the bin-packing problem. In this case the bin is a processor with a capacity of one, according to Liu & Layland's bound for EDF scheduling [6]. The bin-packing problem is known to be NP-hard in the strong sense [5]. Thus, searching for optimal allocation algorithms is not practical. Several heuristic allocation algorithms have been proposed [7], [5], [1], [2], [8]. One of these, known as the First Fit (FF) algorithm, which is focused on in this paper.

A common metric to compare the performance of different allocation algorithms is the expression N/N_{opt}, where N is the number of processors required to allocate the tasks using a given allocation algorithm, and N_{opt} is the number of processors required by the optimal allocation algorithm. Using FF allocation, Garey et al. [5] showed that $N \leq \lceil 17N_{\text{opt}}/10\rceil$. For high values of N_{opt} this expression is approximately equivalent to $N/N_{\text{opt}} \leq 1.7$. The metric N/N_{opt} is useful to compare different allocation algorithms but not to perform a schedulability test. There are two reasons for this. Firstly, N_{opt} can not be calculated in pseudo-polynomial time. Secondly, even if we did know N_{opt}, we would not obtain a tight bound. For example, consider a set of $10n$ tasks of utilization factors $(0.5+\epsilon)$. In this case $N_{\text{opt}} = 10n$, and applying the metric $N/N_{\text{opt}} \leq 1.7$ we would need at least $17n$ processors to guarantee the schedulability of the task set using FF allocation. The total utilization of this task set is $5n$ when $\epsilon \to 0$, much less than the utilization $0.5(17n+1)$, which is presented for $17n$ processors.

Our paper proves that the worst-case achievable utilization for EDF scheduling and FF allocation (EDF-FF) takes the value

$$U_{\text{wc}}^{\text{EDF-FF}}(n) = 0.5(n+1) \tag{1}$$

independent of the number of tasks, where n is the number of processors. If all the tasks have an utilization factor under a value α, the worst-case achievable utilization is proved to be that given by equation (2).

$$U_{\text{wc}}^{\text{EDF-FF}}(n, \beta) = \frac{\beta n + 1}{\beta + 1} \tag{2}$$

where $\beta = \lfloor 1/\alpha \rfloor$.

Equation (1) is obtained from equation (2) making $\alpha = 1$, and therefore $\beta = 1$. As α decreases, both β and the bound given by equation (2) increase. In the limit, when $\alpha \to 0$, then $\beta \to \infty$, and the bound is n. That is, for low utilization factors EDF-FF is a nearly optimal multiprocessor scheduling algorithm.

Equation (2) allows us not only to carry out a fast schedulability test, but also gives the worst-case behaviour that could be expected from multiprocessor EDF-FF scheduling. For example, for a multiprocessor made up of two processors, any task set of total utilization 1.5 or less is schedulable by EDF-FF, which implies an average utilization of 0.75 per processor. If the utilization factors of the tasks are known to be less than or equal to $\alpha = 0.3$ then $\beta = 3$, and so the worst-case achievable utilization is 1.75, which implies an average utilization of 0.875 per processor.

The rest of the paper is organized as follows. Section 2 defines the system we deal with. The expression of the worst-case achievable utilization given by equation (2) is proved in Section 3. Section 4 provides the worst-case achievable utilization for other allocation heuristics under EDF scheduling. Section 5 analyzes the expression of the worst-case achievable utilization. Section 6 provides the achievable utilization for the average-case by means of simulation. Finally, our conclusions are presented in Section 7.

2. System definition

The task set model consists of m independent periodic tasks $\{\tau_1, \dots, \tau_m\}$, of computation times $\{C_1, \dots, C_m\}$, periods $\{T_1, \dots, T_m\}$, and hard deadlines equal to the task periods. The utilization factor u_i of any task τ_i, defined as $u_i = C_i/T_i$, is assumed to be $0 < u_i \leq \alpha \leq 1$, where α is the maximum value that can be taken by the utilization factor of any task. Thus, the total utilization of the task set defined as $U = \sum_{i=1}^{m} u_i$ is less than or equal to $m\alpha$. No particular ordering is assumed among the utilization factors.

Tasks are allocated to an array of n identical processors $\{P_1, \dots, P_n\}$, and are executed independently of each other. Once a task is allocated to a processor, it is executed only on that processor. Within each processor, tasks are scheduled pre-emptively using EDF. The allocation is carried out using the *First Fit* (FF) allocation algorithm.

Each periodic task τ_i, of utilization factor u_i, is assigned by the FF heuristic to the first processor, P_j, with enough capacity. Under EDF scheduling, the capacity of each processor is one, i.e, the total utilization of the tasks allocated to one processor, U_j, must be less than or equal to one [6]. Thus, the task is allocated to the first processor fulfilling $u_i + U_j \leq 1$, where U_j is the total utilization of the tasks previously allocated to processor P_j. Processors are visited in the order $P_1, P_2, \dots, P_n$. If no processor has enough capacity to hold τ_i, then the periodic task set is said to be non-schedulable (under EDF scheduling and FF allocation).

3. Worst-case achievable utilization

In this section we obtain the worst-case achievable utilization $U_{\text{wc}}^{\text{EDF-FF}}$ for EDF scheduling and FF allocation (EDF-FF) on multiprocessors, which is defined as follows.

Definition 1 *The worst-case achievable utilization for EDF-FF is defined as the real number $U_{wc}^{EDF\text{-}FF}$, fulfilling the following properties:*

- *Any periodic task set of total utilization $U \leq U_{wc}^{EDF\text{-}FF}$ fits into n processors, using EDF scheduling, and FF allocation. Therefore, it is schedulable using EDF scheduling, and FF allocation.*
- *For any total utilization $U > U_{wc}^{EDF\text{-}FF}$, it is always possible to find a periodic task set with that utilization, which does not fit into n processors, using EDF scheduling and FF allocation. Therefore, a task set with total utilization U may or may not be schedulable.*

Consequently, the worst-case achievable utilization can be utilized as a sufficient schedulability condition for multiprocessors.

Before calculating the worst-case achievable utilization, we introduce the parameter β, defined as *the maximum number of tasks of utilization factor α, which fit into one processor*. This parameter is a key concept in obtaining $U_{\text{wc}}^{\text{EDF-FF}}$. Parameter β is a function of α, as given by Lemma 1.

Lemma 1

$$\beta = \lfloor 1/\alpha \rfloor \tag{3}$$

Proof:
According to the definition of β, β tasks of utilization factor α fit into one processor. Thus, from the utilization bound for EDF scheduling we get $\beta\alpha \leq 1$. Finding β, we obtain $\beta \leq 1/\alpha$. Since β is an integer we get $\beta \leq \lfloor 1/\alpha \rfloor$. Since β is the maximum number of tasks of utilization factor α that fit into one processor, $(\beta + 1)$ tasks of utilization factor α do not fit into one processor. Thus, $(\beta + 1)\alpha > 1$. Finding β we obtain $\beta > 1/\alpha - 1$. Since β is an integer we get $\beta \geq \lfloor 1/\alpha \rfloor$. In conclusion $\beta = \lfloor 1/\alpha \rfloor$. ■

The value of β can be used to establish the schedulability of some task sets. From the definition of β, β tasks of utilization factor α fit into each processor. Since all the tasks have utilization factors less than or equal to α, at least β tasks of arbitrary utilization factors ($\leq \alpha$) fit into each processor. Therefore, a multiprocessor made up of n processors can allocate at least βn tasks. Subsequently, if a task set is made up of m tasks, with utilization factors $u_i \leq \alpha$, and $m \leq \beta n$, the task set is schedulable by EDF and any *reasonable* allocation algorithm on n processors. By a *reasonable* allocation algorithm we means one which fails to allocate a task only when there is no processor in the multiprocessor with sufficient capacity to hold the task. For example, FF and the optimal allocation algorithm are *reasonable*.

Figure 1 depicts β as a function of α, also showing the sufficient schedulability condition $m \leq \beta n$. For example, if α is in the interval $(1/3, 1/2]$ then $\beta = 2$. In this case, the task set is schedulable if it has $2n$ tasks or less.

Another consequence of the schedulability condition $m \leq \beta n$, is that it is worthwhile obtaining the value of the worst-case achievable utilization only for the case $m > \beta n$. Otherwise, the task set is directly schedulable. From now on we assume that $m > \beta n$, so the above schedulability condition can not be applied.

The worst-case achievable utilization depends (at most) on the restrictions of the task set and multiprocessor, that is, the number of processors, n, the number of tasks, m, and the maximum reachable utilization factor, α. Thus, hereafter we use $U_{\text{wc}}^{\text{EDF-FF}}(m, n, \alpha)$ to denote it.

Next, we present the strategy for the calculation of the worst-case achievable utilization under EDF-FF scheduling.

1. Theorem 1 gives an upper bound on the worst-case achievable utilization for any allocation algorithm. Therefore, this bound also applies to FF allocation.

2. Lemma 2 is proved. This lemma is necessary in order to prove Theorem 2.

3. Theorem 2 proves an expression which relates the worst-case achievable utilization of m tasks and n processors, with the worst-case achievable utilization of $(m - \beta)$ tasks and $(n - 1)$ processors for EDF-FF.

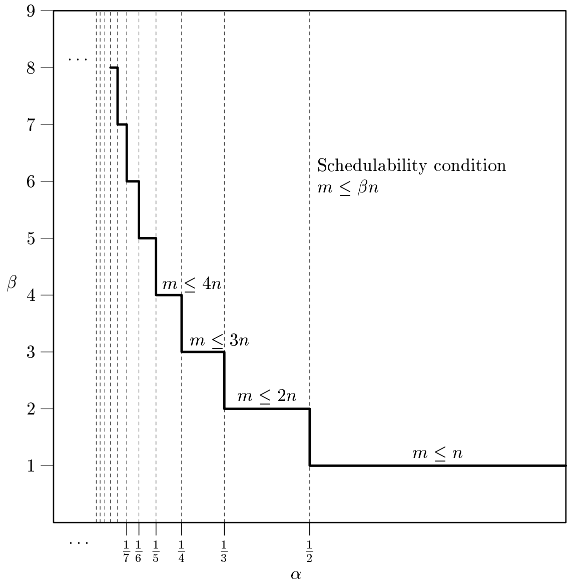

Figure 1. Representation of the function $\beta(\alpha)$, and the associated schedulability condition.

4. From the result given in step 3, and the bound for EDF scheduling on uniprocessors, Theorem 3 obtains a lower bound on the worst-case achievable utilization for EDF-FF.

5. The upper and lower bounds on the worst-case achievable utilization for EDF-FF given in steps 1 and 4 are the same. Thus, both coincide with the worst-case achievable utilization, as stated finally by Theorem 3.

Theorem 1 gives an upper bound on the worst-case achievable utilization for EDF scheduling, and an arbitrary allocation algorithm, represented by AA. The proof is based on finding a task set which does not fit into the processors.

Theorem 1 *If $m > \beta n$ then*

$$U_{wc}^{\textit{EDF-AA}}(m, n, \alpha) \leq \frac{\beta n + 1}{\beta + 1}$$

Proof:
We will prove that a set of m tasks $\{\tau_1, \ldots, \tau_m\}$ exists, with utilization factors $0 < u_i \leq \alpha$ for all $i = 1, \ldots, m$, and total utilization $U = (\beta n + 1)/(\beta + 1) + \epsilon$, being $\epsilon \to 0^+$, which does not fit into n processors using any allocation algorithm and EDF scheduling on each processor.

We will construct this set of m tasks as composed of two subsets: a first subset with $(m - \beta n - 1)$ tasks, and a second subset with $(\beta n + 1)$ tasks.

All the tasks of the first subset have the same utilization factor of value

$$u_i = \frac{\epsilon}{m}$$

where $i = 1, \ldots, (m - \beta n - 1)$.

All the tasks of the second subset have the same utilization factor of value

$$u_i = \frac{1}{(\beta + 1)} + \frac{\epsilon}{m}$$

where $i = (m - \beta n), \ldots, m$.

It can be seen that the total utilization of the whole task set is $(\beta n + 1)/(\beta + 1) + \epsilon$.

Firstly, it is necessary to prove that the utilization factors of both subsets are valid, i.e, $0 < u_i \leq \alpha$ for all $i = 1, \ldots, m$.

Check of the utilization factors of the first subset. Choosing a small enough value for ϵ, we obtain $0 < u_i = \frac{\epsilon}{m} \leq \alpha$.

Check of the utilization factors of the second subset. By definition of β, $(\beta + 1)$ tasks of utilization factor α do not fit into one processor, therefore $(\beta + 1)\alpha > 1$, and

$$\alpha > \frac{1}{(\beta + 1)}$$

It is always possible to find one real number between two real numbers. Hence, a positive value ϵ exists such that

$$\alpha > \frac{1}{(\beta + 1)} + \frac{\epsilon}{m} = u_i$$

which proves that the utilization factors of the second subset are less than α when $\epsilon \to 0^+$. In addition, the utilization factors of the second subset are obviously greater than zero.

From the above results, we conclude that the proposed task set is valid. Below we prove that it does not fit into n processors, using EDF scheduling and any allocation algorithm.

There are $(\beta n + 1)$ tasks in the second subset. Hence at least one processor of the n available should allocate $(\beta + 1)$ or more of these tasks. However, no processor can allocate $(\beta + 1)$ or more tasks of the second subset, since $(\beta + 1)$ of these tasks together have an utilization over one.

$$(\beta + 1)\left(\frac{1}{\beta + 1} + \frac{\epsilon}{m}\right) > 1$$

We conclude that the proposed task set of total utilization $(\beta n + 1)/(\beta + 1) + \epsilon$ does not fit into n processors when $\epsilon \to 0^+$, so the worst-case achievable utilization, $U_{\text{wc}}^{\text{EDF-AA}}(m, n, \alpha)$, must be less than or equal to $(\beta n + 1)/(\beta + 1)$.

NOTE: the tasks of the first subset are necessary in the proof only to fulfil the restriction of having m tasks. ■

The proof of Theorem 2 requires Lemma 2, which is proved below. It relates the worst-case achievable utilization for the same number of processors but a different number of tasks, using any *reasonable* allocation algorithm, represented by RA. The result is valid for FF allocation, as it is for any *reasonable* allocation algorithm.

Lemma 2

$$U_{wc}^{\textit{EDF-RA}}(q, n, \alpha) \geq U_{wc}^{\textit{EDF-RA}}(m, n, \alpha) \quad \textit{for all } q < m$$

Proof:

This lemma will be proved by contradiction.

Let us suppose that a pair of integers q and m exists, such that $q < m$, and $U_{\text{wc}}^{\text{EDF-RA}}(q, n, \alpha) < U_{\text{wc}}^{\text{EDF-RA}}(m, n, \alpha)$. Between two real numbers, it is always possible to find another real number, so we can find an $\epsilon > 0$ such that

$$\begin{aligned} U_{\text{wc}}^{\text{EDF-RA}}(q, n, \alpha) &< U_{\text{wc}}^{\text{EDF-RA}}(m, n, \alpha) - \epsilon \\ &< U_{\text{wc}}^{\text{EDF-RA}}(m, n, \alpha) \end{aligned}$$

According to the definition of worst-case achievable utilization, there exists at least one set of q tasks, $\{\tau_1, \ldots, \tau_q\}$, of total utilization

$$\sum_{i=1}^{q} u_i = U_{\text{wc}}^{\text{EDF-RA}}(m, n, \alpha) - \epsilon$$

(greater than the worst-case achievable utilization for q tasks), which does not fit into n processors. Next, we prove that this gives rise to a contradiction.

If we add to this task set $(m - q)$ new tasks, $\{\tau_{q+1}, \ldots, \tau_m\}$, each of utilization factor $\epsilon/(m - q)$, we obtain a task set made up of m tasks of total utilization $\sum_{i=1}^{m} u_i = U_{\text{wc}}^{\text{EDF-RA}}(m, n, \alpha)$, which fits into n processors. Hence, the first q tasks fit into n processors, which is a contradiction. ■

Next, we prove an expression which relates the worst-case achievable utilization of multiprocessors with n and $(n - 1)$ processors for EDF-FF. This will allow us to obtain a lower bound on the worst-case achievable utilization for EDF-FF, going from the case $n = 1$ (uniprocessor case) to a general multiprocessor case with an arbitrary n.

Theorem 2 *If $m > \beta n$ then*

$$U_{wc}^{\textit{EDF-FF}}(m, n, \alpha) \geq \frac{\beta}{\beta + 1} + U_{wc}^{\textit{EDF-FF}}(m - \beta, n - 1, \alpha)$$

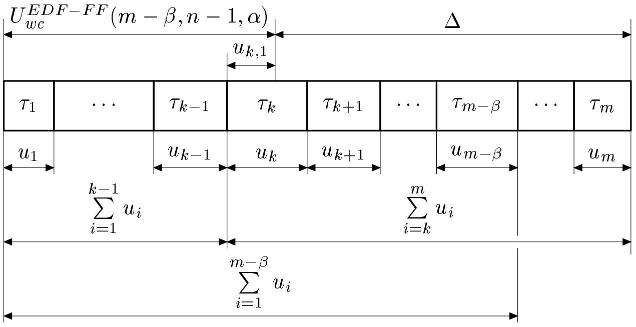

Figure 2. General situation in *Case 2* of Theorem 2.

Proof:
We will prove that any set of m tasks $\{\tau_1, \ldots, \tau_m\}$, with utilization factors $0 < u_i \leq \alpha$ for all $i = 1, \ldots, m$, and total utilization less than or equal to

$$\frac{\beta}{\beta+1} + U_{\text{wc}}^{\text{EDF-FF}}(m-\beta, n-1, \alpha)$$

fits into n processors using EDF scheduling on each processor and FF allocation.
There are two possible cases:

Case 1: The first $(m-\beta)$ tasks have total utilization less than or equal to $U_{\text{wc}}^{\text{EDF-FF}}(m-\beta, n-1, \alpha)$, that is, $\sum_{i=1}^{m-\beta} u_i \leq U_{\text{wc}}^{\text{EDF-FF}}(m-\beta, n-1, \alpha)$. In this case the whole set of m tasks always fits into n processors, because the first $(m-\beta)$ tasks fit into the first $(n-1)$ processors (since their total utilization is below the bound), and the remaining β tasks fit into the last processor, since the definition of β implies that a least β tasks always fit into one processor.

Case 2: The first $(m-\beta)$ tasks have a total utilization greater than $U_{\text{wc}}^{\text{EDF-FF}}(m-\beta, n-1, \alpha)$, that is, $\sum_{i=1}^{m-\beta} u_i > U_{\text{wc}}^{\text{EDF-FF}}(m-\beta, n-1, \alpha)$. In this case we will prove that the whole set of m tasks still fits into n processors if the total utilization is equal to $U_{\text{wc}}^{\text{EDF-FF}}(m-\beta, n-1, \alpha) + \Delta$, provided $\Delta \in \mathbb{R}$, and $\Delta \leq \frac{\beta}{\beta+1}$.

A task τ_k must exist, whose u_k added to the previous utilizations u_i, causes the bound $U_{\text{wc}}^{\text{EDF-FF}}(m-\beta, n-1, \alpha)$ to be exceeded. This situation is depicted in Figure 2, which is a graphical representation of the utilization factors of each task and the relationships between several quantities and summations used through this proof. The value of k is obtained as the integer which fulfills

$$\sum_{i=1}^{k-1} u_i \leq U_{\text{wc}}^{\text{EDF-FF}}(m-\beta, n-1, \alpha) < \sum_{i=1}^{k} u_i$$

Note that $k \leq m-\beta$ (if $k > m-\beta$ we would be in *Case 1*).

It can be seen that the first $(k-1)$ tasks fit into the first $(n-1)$ processors. The total utilization of the first $(k-1)$ tasks fulfills

$$\sum_{i=1}^{k-1} u_i \leq U_{\text{wc}}^{\text{EDF-FF}}(m-\beta, n-1, \alpha)$$

Applying Lemma 2 with $(m-\beta) > (k-1)$ and RA=FF, we get

$$U_{\text{wc}}^{\text{EDF-FF}}(m-\beta, n-1, \alpha) \leq U_{\text{wc}}^{\text{EDF-FF}}(k-1, n-1, \alpha)$$

and thus

$$\sum_{i=1}^{k-1} u_i \leq U_{\text{wc}}^{\text{EDF-FF}}(k-1, n-1, \alpha)$$

Therefore, the first $(k-1)$ tasks fit into the first $(n-1)$ processors. It is sufficient to prove that the remaining $(m-k+1)$ tasks fit into the last processor.

We will assume without loss of generality that all the tasks τ_i in $\{\tau_k, \ldots, \tau_m\}$ have $u_i > u_{k,1}$, where

$$u_{k,1} = U_{\text{wc}}^{\text{EDF-FF}}(m-\beta, n-1, \alpha) - \sum_{i=1}^{k-1} u_i$$

as shown in Figure 2. Note that if there were a task τ_i in $\{\tau_k, \ldots, \tau_m\}$ with $u_i \leq u_{k,1}$, we could always allocate this task to the first $(n-1)$ processors (since the addition of this new task does not cause the total utilization to exceed the bound), and we would be in a situation analogous to the current one, with k one unit greater. This reasoning can be repeated until no task τ_i with $u_i \leq u_{k,1}$ exists among the last $(m-k+1)$ tasks, or until we are in *Case 1*.

In order to prove that the last $(m-k+1)$ tasks fit into the last processor we have to prove that the total utilization of these tasks is not greater than one, that is,

$$\sum_{i=k}^{m} u_i \leq 1$$

Figure 2 shows that

$$\sum_{i=k}^{m} u_i = u_{k,1} + \Delta \tag{4}$$

As already stated, all the utilization factors u_i in this summation are greater than $u_{k,1}$, so

$$\begin{aligned}(m-k+1)u_{k,1} &< u_{k,1} + \Delta \\ &< u_{k,1} + \frac{\beta}{\beta+1} \quad \text{by the definition of } \Delta\end{aligned}$$

and we can find $u_{k,1}$.

$$u_{k,1} < \frac{\beta}{(\beta+1)(m-k)} \tag{5}$$

Substituting the value of $u_{k,1}$ from (5) into (4) we obtain

$$\begin{aligned}\sum_{i=k}^{m} u_i &< \frac{\beta}{(\beta+1)(m-k)} + \Delta \\ &< \frac{\beta}{(\beta+1)(m-k)} + \frac{\beta}{(\beta+1)} \quad \text{by def. of } \Delta \\ &= \frac{(m-k+1)\beta}{(\beta+1)(m-k)} \\ &= \frac{1+1/(m-k)}{1+1/\beta}\end{aligned}$$

We know that $k \le (m-\beta)$ in *Case 2*, so $(m-k) \ge \beta$. Therefore,

$$\sum_{i=k}^{m} u_i \le 1$$

This equation shows that the last $(m-k+1)$ tasks meet the EDF schedulability condition, so they fit into the last processor.

We have proved that any task set with m tasks and a total utilization

$$U_{\text{wc}}^{\text{EDF-FF}}(m-\beta, n-1, \alpha) + \Delta \le U_{\text{wc}}^{\text{EDF-FF}}(m-\beta, n-1, \alpha) + \frac{\beta}{\beta+1}$$

fits into n processors, so the worst-case utilization bound, $U_{\text{wc}}^{\text{EDF-FF}}(m,n,\alpha)$, must be greater than or equal to $U_{\text{wc}}^{\text{EDF-FF}}(m-\beta, n-1, \alpha) + \frac{\beta}{\beta+1}$, and the theorem is proved. ■

The worst-case achievable utilization for EDF-FF is obtained from Theorem 1 and Theorem 2.

Theorem 3 *If $m > \beta n$ then $U_{wc}^{EDF\text{-}FF}$ depends only on n, and β, and is given by*

$$U_{wc}^{EDF\text{-}FF}(n,\beta) = \frac{\beta n + 1}{\beta + 1} \tag{6}$$

Proof:
Firstly, we obtain a lower bound for the worst-case achievable utilization for a set of m tasks on a multiprocessor with n processors.

Theorem 2 relates the worst-case achievable utilization of sets of m tasks on multiprocessors of n processors, with the worst-case achievable utilization of sets of $(m-\beta)$ tasks on multiprocessors with $(n-1)$ processors.

$$U_{\text{wc}}^{\text{EDF-FF}}(m,n,\alpha) \ge \frac{\beta}{\beta+1} + U_{\text{wc}}^{\text{EDF-FF}}(m-\beta, n-1, \alpha) \tag{7}$$

Theorem 2 also relates the worst-case achievable utilization of sets of $(m-\beta)$ tasks on multiprocessors of $(n-1)$ processors, with the worst-case achievable utilization of sets of $(m-2\beta)$ tasks on multiprocessors of $(n-2)$ processors.

$$U_{\text{wc}}^{\text{EDF-FF}}(m-\beta, n-1, \alpha) \ge \frac{\beta}{\beta+1} + U_{\text{wc}}^{\text{EDF-FF}}(m-2\beta, n-2, \alpha) \tag{8}$$

Substituting (8) into (7) we get

$$U_{\text{wc}}^{\text{EDF-FF}}(m,n,\alpha) \ge \frac{2\beta}{\beta+1} + U_{\text{wc}}^{\text{EDF-FF}}(m-2\beta, n-2, \alpha)$$

This procedure can be repeated, until finally relating the worst-case achievable utilization of sets of m tasks on multiprocessors of n processors, with the worst-case achievable utilization of sets of $(m-(n-1)\beta)$ tasks on an uniprocessor.

$$U_{\text{wc}}^{\text{EDF-FF}}(m,n,\alpha) \ge \frac{(n-1)\beta}{\beta+1} + U_{\text{wc}}^{\text{EDF-FF}}(m-(n-1)\beta, 1, \alpha) \tag{9}$$

The worst-case achievable utilization for $(m-(n-1)\beta)$ tasks and one processor is one, independently of the values of m or α.

$$U_{\text{wc}}^{\text{EDF-FF}}(m-(n-1)\beta, 1, \alpha) = 1 \tag{10}$$

Substituting (10) into (9), we obtain a lower bound on the worst-case achievable utilization of m tasks on n processors.

$$U_{\text{wc}}^{\text{EDF-FF}}(m,n,\alpha) \ge \frac{\beta n + 1}{\beta + 1} \tag{11}$$

Theorem 1 proved that

$$U_{\text{wc}}^{\text{EDF-AA}}(m,n,\alpha) \le \frac{\beta n + 1}{\beta + 1} \tag{12}$$

where AA is an arbitrary allocation algorithm. Hence, it also applies to the FF heuristic. From equation (11) and equation (12) we get

$$U_{\text{wc}}^{\text{EDF-FF}}(m,n,\alpha) = \frac{\beta n + 1}{\beta + 1}$$

We observe that $U_{\text{wc}}^{\text{EDF-FF}}$ only depends on the number of tasks, m, and parameter β, so we finally conclude that

$$U_{\text{wc}}^{\text{EDF-FF}} = U_{\text{wc}}^{\text{EDF-FF}}(n, \beta) = \frac{\beta n + 1}{\beta + 1} \qquad \blacksquare$$

Therefore, any set of periodic tasks, with utilization factors less than or equal to α, and total utilization less than or equal to

$$U_{\text{wc}}^{\text{EDF-FF}}(n, \beta) = \frac{\beta n + 1}{\beta + 1}$$

is feasibly scheduled by EDF on n processors using FF allocation, where $\beta = \lfloor 1/\alpha \rfloor$. This is a tight sufficient schedulability condition. For any value of total utilization greater than $U_{\text{wc}}^{\text{EDF-FF}}(n, \beta)$ it is possible to find a task set with this utilization which does not fit into the processors using EDF-FF scheduling.

Having calculated the general expression of the worst-case achievable utilization, and making $\alpha = 1$ we remove the restriction $u_i \leq \alpha$, so the utilization factors of the tasks can now be in the interval $(0, 1]$. In this case $\beta = 1$, and the worst-case achievable utilization is given by

$$U_{\text{wc}}^{\text{EDF-FF}}(n, 1) = 0.5(n + 1) \tag{13}$$

4. Worst-case achievable utilization for other allocation heuristics

This paper focuses on the FF heuristic. However it would be interesting to know whether it is possible to find other heuristics providing higher worst-case achievable utilization than that provided by FF. Theorem 1 stated that

$$U_{\text{wc}}^{\text{EDF-AA}}(m, n, \alpha) \leq \frac{\beta n + 1}{\beta + 1}$$

where AA is an arbitrary allocation algorithm. Since the worst-case achievable utilization for EDF-FF (given by equation (6)) equals this bound, it follows that FF allocation provides the maximum worst-case achievable utilization under EDF scheduling with partitioning. Therefore, it coincides with the worst-case achievable utilization under EDF scheduling for the optimal allocation algorithm. In addition, EDF being the optimal uniprocessor scheduling algorithm, it follows that no pair of *uniprocessor scheduling algorithm-allocation algorithm* exists with higher worst-case achievable utilization than that of EDF-FF.

The fact that FF provides the maximum worst-case achievable utilization for EDF scheduling does not preclude the existence of task sets which can be allocated by other algorithms but not by FF. In fact, different allocation algorithms should also be compared in terms of average-case achievable utilization.

Under EDF scheduling, other *reasonable* heuristics such as *First Fit Decreasing* (FFD), *Best Fit* (BF), and *Best Fit Decreasing* (BFD) have the same worst-case achievable utilization as FF. However, they are computationally more expensive. In addition, neither FFD nor BFD can be applied on-line. The definition of these heuristics can be found in [5]. The proof of the worst-case achievable utilization for BF is analogous to that shown for FF, so it is not repeated. The proof for FFD and BFD is direct. Both heuristics require a process of sorting the utilization factors before applying the pure FF or BF heuristic. The process of sorting restricts the possible task sets to allocate with regard to the general task sets considered in the case of pure FF or BF allocation. Therefore, the worst-case achievable utilization for FFD and BFD can not be lower than that of FF and BF. In addition from Theorem 1, the worst-case achievable utilization can not be higher, so finally we conclude that the worst-case achievable utilization for FFD, BFD, FF, and BF is the same, given by equation (6).

After observing the coincidence among the worst-case achievable utilization of different *reasonable* allocation algorithms, one might think that the worst-case achievable utilization may be a constant for all the *reasonable* allocation algorithms. However, this is not the case. For example, consider the *reasonable* allocation algorithm called *Worst Fit* (WF), which assigns a task to the processor with the lowest capacity, among all the processors with sufficient capacity. Consider also a multiprocessor made up of $n = 2$ processors, and a set of $m = 3$ tasks of utilization factors 0.1, 0.1 and 0.91. The first task of utilization factor 0.1 is assigned to the first processor, and the second task of utilization 0.1 to the second processor. After that, the third task of utilization factor 0.91 can not be allocated to any of the two processors, so the worst-case achievable utilization for EDF-WF is less than $(0.1 + 0.1 + 0.91) = 1.11$. This value is much lower than 1.5, the worst-case achievable utilization for EDF-FF with $n = 2$.

5. Analysis of the theoretical results

In this section, we analyze the expression of the worst-case achievable utilization, $U_{\text{wc}}^{\text{EDF-FF}}(n, \beta)$, for EDF-FF given by equation (6), which is repeated below.

$$U_{\text{wc}}^{\text{EDF-FF}}(n, \beta) = \frac{\beta n + 1}{\beta + 1}$$

In spite of n being an integer, the function $U_{\text{wc}}^{\text{EDF-FF}}(n, \beta)$ is represented as a continuous function in Figure 3, with the aim of improving its visualization. Figure 3 depicts the worst-case achievable utilization for EDF-FF as a function of the numbers of processors, for different values of β. The representation has been normalized dividing $U_{\text{wc}}^{\text{EDF-FF}}(n, \beta)$ by the number of processors, in order to show the average

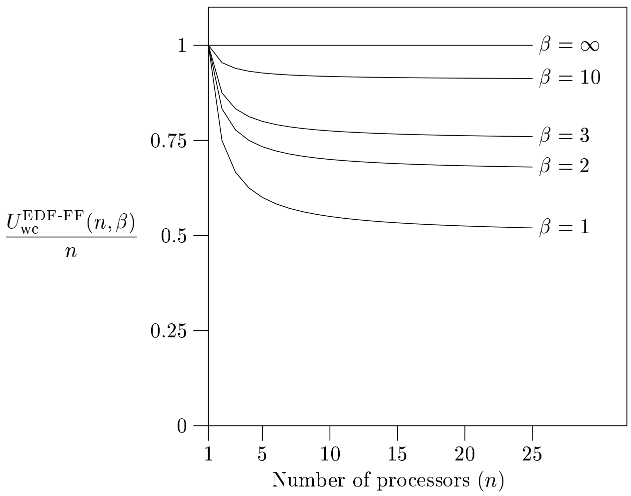

Figure 3. Worst-case achievable utilization for EDF-FF.

degree of utilization of the processors. The addition of each new processor increments the worst-case achievable utilization in the magnitude $\frac{\beta}{\beta+1}$. Furthermore, any decrement of α produces an increment of $\beta = \lfloor 1/\alpha \rfloor$ and an increment of the worst-case achievable utilization. In the limit, if $\alpha \to 0$ then $\beta \to \infty$, and it follows that

$$\lim_{\alpha\to 0} U_{\text{wc}}^{\text{EDF-FF}}(n,\beta) = n$$

Hence, if the utilization factor of all the tasks is low, the multiprocessor system using EDF-FF scheduling approximates an ideal uniprocessor n-times faster than each processor of the multiprocessor.

The worst-case achievable utilization provides the maximum total utilization guaranteeing the schedulability on a given number of processors. Nevertheless, there are practical cases in which we are interested in the worst number of processors, $n_{\text{wc}}^{\text{EDF-FF}}$, required to feasibly schedule a task set of a given total utilization.

In this paper we have presented two schedulability conditions, from which we obtain different numbers of processors. The first schedulability condition (see Figure 1) coming from the definition of β is $m \le \beta n_{\text{wc}}^{\text{EDF-FF}}$, so finding $n_{\text{wc}}^{\text{EDF-FF}}$ and choosing the minimum integer, we obtain

$$n_{\text{wc}}^{\text{EDF-FF}} = \left\lceil \frac{m}{\beta} \right\rceil \tag{14}$$

The second schedulability condition coming from equation (6) is

$$U \le \frac{\beta n_{\text{wc}}^{\text{EDF-FF}} + 1}{\beta + 1}$$

so finding $n_{\text{wc}}^{\text{EDF-FF}}(n,\beta)$, and choosing the minimum integer we obtain

$$n_{\text{wc}}^{\text{EDF-FF}} = \left\lceil \frac{(\beta+1)U - 1}{\beta} \right\rceil \tag{15}$$

Equation (15) is not valid for $U \le 1/(\beta+1)$. However for any total utilization $U \le 1$ a single processor is enough to feasibly schedule the task set. From this observation, equation (14), and equation (15), we get

$$n_{\text{wc}}^{\text{EDF-FF}}(m,U,\beta) = \begin{cases} 1 & \text{if } U \le 1 \\ \min\left\{ \left\lceil \frac{m}{\beta} \right\rceil, \left\lceil \frac{(\beta+1)U-1}{\beta} \right\rceil \right\} & \text{otherwise} \end{cases} \tag{16}$$

For example, to guarantee the schedulability of 100 tasks, of total utilization 15, and utilization factors under 0.25 (i.e. $\beta = 4$), 19 processors are necessary

$$n_{\text{wc}}^{\text{EDF-FF}}(100,15,4) = \min\left\{ \left\lceil \frac{100}{4} \right\rceil, \left\lceil \frac{(4+1)15-1}{4} \right\rceil \right\} = 19$$

6. Average-case behaviour

The worst-case achievable utilization is obtained for task sets which might be infrequent in practice. In order to perceive the pessimism of the worst-case achievable utilization, we will define the *average-case achievable utilization*, $U_{\text{ac}}^{\text{EDF-FF}}(m,n,p)$, for EDF-FF as follows.

Definition 2 *A task set made up of m tasks, of total utilization $U = U_{ac}^{EDF\text{-}FF}(m,n,p)$ is schedulable by EDF and FF on n processors with a probability equal to p.*

If task sets made up of m tasks with a total utilization $U = U_{\text{ac}}^{\text{EDF-FF}}(m,n,p)$ were randomly generated, $p\%$ of the task sets would be schedulable by EDF and FF on n processors. The other $(100-p)\%$ would correspond to task sets which do not fit into the processors.

The statistical distribution chosen to generate the task sets is the *Beta Distribution*. This distribution has two positive parameters, which allow us to select the mean, μ, and standard deviation, σ, of the distribution. In addition, it is suitable to generate the utilization factors of the tasks, as it only generates values in the interval $(0,1)$. The maximum σ for a given value of μ is $\sigma_{\text{max}} = \sqrt{\mu(1-\mu)}$, which is obtained by considering the restrictions of the Beta distribution. Varying the value of the standard deviation from zero to the maximum, the utilization factors of the tasks vary from being equal, to having unlike values. Thus, the standard deviation of the distribution affects the schedulability of the task sets.

With the aim of establishing the pessimism of the worst-case achievable utilization for EDF-FF, given by equation (6), we have carried out simulation experiments to obtain $U_{\text{ac}}^{\text{EDF-FF}}(m,n,p)$ for $p = 99\%$. At least 1000 task sets were randomly generated using the beta distribution of probability to obtain each tuple $\{m, n, U_{\text{ac}}^{\text{EDF-FF}}(m,n,p)\}$, for each value of standard deviation from $\sigma = 0.001\sigma_{\text{max}}$ to $\sigma = 0.9\sigma_{\text{max}}$. Figure 4 was obtained as the minimum, among all the standard deviations from $\sigma = 0.001\sigma_{\text{max}}$ to $\sigma = 0.9\sigma_{\text{max}}$, of the associated curves of average-case achievable utilization. Figure 4 also represents the worst-case utilization bound to be compared with the average-case utilization bound. The difference is significant for task sets with a high number of tasks, since in this case the utilization factors of the tasks are small in probability terms. Nevertheless, for the uniprocessor case, the average-case coincides with the worst-case achievable utilization, given by the bound $U \leq 1$.

From Figure 4, we observe an achievable utilization of around 80% per processor (with a probability of 99%) for a multiprocessor made up of four processors. Assuming that any task set of utilization n or less can be scheduled using non-partitioning, the improvement that may be obtained by means of non-partitioning is about 20% per processor. However in practice, the improvement is much less. Firstly, we must consider the high overload associated with non-partitioning schemes. Secondly, we must bear in mind that task sets with total utilization greater than that given by Figure 4 could be schedulable by EDF-FF. In addition, the implementation of non-partitioning schemes is more complex and less predictable than the implementation of EDF-FF. Therefore, from a practical point of view, EDF-FF is superior to non-partitioning schemes for scheduling independent hard-real time periodic tasks on multiprocessors.

7. Conclusions

A schedulability bound based on utilizations has been obtained for real-time scheduling of periodic tasks using EDF-FF. This bound is a function of the number of processors and the maximum value that can be reached by any utilization factor of the task set. For the case of tasks with low utilization factors, the bound is greatly raised, reaching asymptotically the value n when all the utilization factors are close to zero.

Among all the partitioning scheduling algorithms, EDF-FF has been proved to be optimal in terms of worst-case achievable utilization.

Simulation has provided an average-case achievable utilization bound much higher than the worst-case achievable utilization, showing the pessimism of equation (6).

Finally, simulation indicated that in general EDF-FF scheduling is more practical than non-partitioning scheduling schemes when scheduling independent hard-real time periodic tasks on multiprocessors.

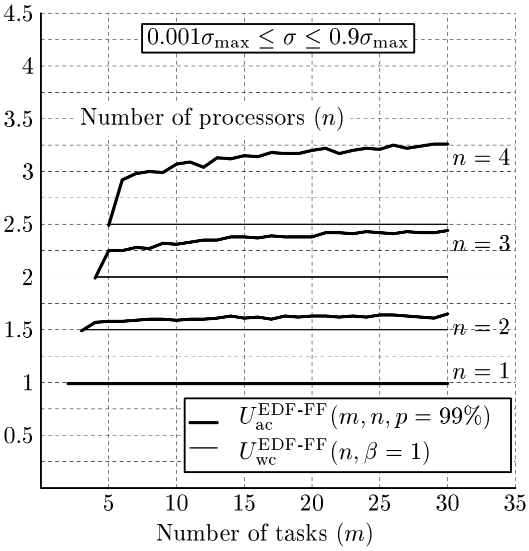

Figure 4. Average-case achievable utilization for arbitrary standard deviation.

References

[1] A. Burchard, J. Liebeherr, Y. Oh, and S. Son. New strategies for assigning real-time tasks to multiprocessor systems. *IEEE Transactions on Computers*, 44(12), 1995.

[2] S. Dall and C. Liu. On a real-time scheduling problem. *Operations Research*, 6(1):127–140, 1978.

[3] M. L. Dertouzos. Control robotics: The procedural control of physical processes. *Proceedings of IFIP Congress*, pages 807–813, 1974.

[4] M. L. Dertouzos and A. K. Mok. Multprocessor on-line scheduling of hard-real-time tasks. *Transactions on Software Engineering*, 15(12):1497–1506, 1989.

[5] M. Garey and D. Johnson. *Computers and Intractability*. W.H. Freman, New York, 1979.

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

[7] Y. Oh and S. Son. Allocating fixed-priority periodic tasks on multiprocessor systems. *Real-Time Systems*, 9(3):207–239, 1995.

[8] S. Sáez, J. Vila, and A. Crespo. Using exact feasibility tests for allocating real-time tasks in multiprocessor systems. *Proceedings of the 10th Euromicro Workshop on Real-Time Systems*, pages 53–60, 1998.