Further improved schedulability analysis of EDF on multiprocessor platforms
TR-051001

Theodore P. Baker
Department of Computer Science
Florida State University
Tallahassee, FL 32306-4530
e-mail: baker@cs.fsu.edu

Abstract

This report shows how to tighten the analysis of global EDF scheduling on multiprocessor platforms, so as to verify the feasibility of a significantly larger range of task systems than has been possible using the previously known tests, including those recently by Bertogna, Cirinei, and Lipari. The improved EDF feasibility test for sporadic task systems with arbitrary deadlines is proven correct, and evaluated in comparison to prior schedulability tests by simulation.

1 Introduction

This report shows how to tighten the analysis of global EDF scheduling on multiprocessor platforms, so as to verify the feasibility of a significantly larger range of task systems than has been possible using the previously known tests, including those proposed recently by Bertogna, Cirinei, and Lipari[4] and those proposed earlier by Baker[2, 3] and Goossens, Funk, and Baruah [5].

2 Definitions

The analysis of EDF scheduling failures in [3] is based on obtaining upper and lower bounds on the computational load over a time interval preceding a first missed deadline of some task τ_k. The computational load is defined in terms of all the computation done in the interval by tasks that can preempt τ_k. As pointed out by [4], when doing this sort of analysis one does not need to consider all the work done by tasks that *can* preempt τ_k, but only the time that such tasks that actually *do* prempt τ_k. That distinction is captured by the following definitions.

Definition 1 *The* work $W_i(a, b)$ *done by a task* τ_k *over a time interval* $[a, b)$ *is the sum of the lengths of all the subintervals in which a job of* τ_i *executes. The* total work $W(a, b)$ *done in a time interval* $[a, b)$ *is the sum*

$$W(a, b) = \sum_{i=1}^{N} W_i(a, b)$$

The interference $I_k(t-\Delta,t)$ *of a task* τ_k *over a time interval* $[t-\Delta,t)$ *is the sum of the lengths of all the subintervals during which* τ_k *is backlogged but unable to execute due to preemption. The* interference contribution $I_{i,k}(t-\Delta,t)$ *of a task* τ_i *to* $I_k(t-\Delta,t)$ *is the amount of time during the interval that* τ_k *is backlogged while* τ_i *and* $m-1$ *other tasks are executing. (Adapted from [4].)*

The block busy time $B(a,b)$ *of a time interval* $[a,b)$ *is the sum of the lengths of all the subintervals during which all* m *processors are executing. The* block busy time $B_i(a,b)$ of a task τ_i *is the amount of time that* τ_i *executes in parallel with* $m-1$ *other tasks.*

A consequence the above definitions is that $I_k(a,b) \leq B(a,b) \leq W(a,b)$, and $I_{i,k}(a,b) \leq B_i(a,b) \leq W_i(a,b)$

The following lemma is adapted from [4].

Lemma 1 *If* $[a,b)$ *is an interval in which it is always the case that at least one processor is executing and* $B(a,b) > x$ *then one of the following is true*

$$\sum_{i=1}^{N} \min\{B_i(a,b),x)\} > mx \tag{1}$$

$$B_i(a,b) < x \iff B_i(a,b) = 0 \tag{2}$$

proof:

Suppose $B(a,b) > x$. Let $S = \{\ i\ | B_i(a,b) \geq x\}$ and $\xi = |S|$. Observe that $\xi \leq m$. If $\xi < m$, then

$$\begin{aligned}
\sum_{i=1}^{N} \min\{B_i(a,b),x\} &= \xi x + \sum_{i:B_i<x} B_i(a,b) \\
&= \xi x + mB(a,b) - \sum_{i:B_i \geq x} B_i(a,b) \\
&\geq \xi x + mB(a,b) - \xi B(a,b) \\
&> \xi x + (m-\xi)x = mx
\end{aligned}$$

Otherwise, if $\xi = m$ and $\sum_{i=1}^{N} \min\{B_i(a,b),x\} \leq mx$ then $mx \geq \xi x + \sum_{i:i\notin S} B_i(a,b)$.

It follows that $B_i(a,b) < x$ if-and-only-if $B_i(a,b) = 0$.

□

Observe that the lemma is also true if one replaces $B(a,b)$ by $I_k(a,b)$ and $B_i(a,b)$ by $B_{i,k}(a,b)$.

In the next two sections the core results of [3] are refined to make use of block busy time and the above lemma.

3 Lower Bound

Lemma 2 (lower bound) *If* t *is a first missed deadline of* τ_k *and* $[t-\Delta,t)$ *is the corresponding maximal* τ_k*-busy interval then*

$$I_k(t-\Delta,t) > \Delta - \frac{c_k}{T_k}(\Delta + T_k - d_k)$$

proof: $W_k(t-\Delta, t)$ is the amount of time τ_k executes in the interval. Since τ_k is continually backlogged over the interval, the only subintervals in which τ_k does not execute are the ones in which all m processors are busy executing jobs that interfere with τ_k, so

$$I_k(t-\Delta, t) = \Delta - W_k(t-\Delta, t) \tag{3}$$

Let $j \geq 0$ be the number of jobs of τ_k that execute in the interval. Consider the cases $j = 0$ and $j > 0$ separately.

If $j = 0$ then $x = 0$. Since $\Delta \geq d_k$ and $T_k > 0$, it follows that

$$I_k(t-\Delta, t) = \Delta > \Delta - \frac{c_k}{T_k}(\Delta + T_k - d_k)$$

If $j > 0$ then since τ_k is backlogged over the entire interval,

$$W_k(t-\Delta, t) < jc_k \tag{4}$$

Since τ_k is not backlogged at the start of the interval, all of the j jobs that execute in the interval are released on or after $t-\Delta$ and not later than $t-d_k$ (because t is a missed deadline), and since the release times are separated by at least T_k,

$$(j-1)T_k + d_k \leq \Delta \tag{5}$$

It follows from (3)-(5) that

$$I_k(t-\Delta, t) = \Delta - W_k(t-\Delta, t) > \Delta - \frac{\Delta - d_k}{T_k}c_k - c_k = \Delta - \frac{c_k}{T_k}(\Delta + T_k - d_k) \tag{6}$$

□

Note that since $I_k(t-\Delta, t) \leq B(t-\Delta, t) \leq W(t-\Delta, t)$, the lower bound on interference provided by the above lemma also applies to busy time $B(t-\Delta, t)$ and work $W(t-\Delta, t)$.

4 Upper Bound

The work $W_i(t-\Delta, t)$ done by τ_i in an interval $[t-\Delta, t)$ is bounded by the interval length Δ and may include:

1. a portion of the execution times of zero or more jobs that are released before $t-\Delta$ but are unable to complete by that time, which are called *carried-in* jobs;
2. the full execution times c_i of zero or more jobs that are released on or after time $t-\Delta$ and complete by time t;
3. a portion $\epsilon \leq c_i$ of the execution time of at most one job that is released at some time $t-\delta$, $0 < \delta \leq \Delta$, but is unable to complete by time t.

To bound the size of the carried-in contribution of τ_i the maximal τ_k-busy interval will be extended downward as far as possible while still maintaining a lower bound on block busy time as in Lemma 2.

Definition 2 *An interval $[t-\Delta, t)$ is* τ_k^λ-busy *for a given constant $\lambda \geq c_k/T_k$ if*

$$B(t-\Delta, t) > \Delta - \lambda(\Delta + T_k - d_k)$$

It is a maximal τ_k^λ*-busy interval if it is τ_k^λ-busy and there is no $\Delta' > \Delta$ such that $[t-\Delta', t)$ is also τ_k^λ-busy.*

Lemma 3 *If t is a first missed deadline of τ_k and $\lambda \geq c_k/T_k$ then there is a unique maximal τ_k^λ-busy interval $[t-\hat{\Delta}, t)$, and $\hat{\Delta} \geq d_k$.*

proof: Since τ_k misses a deadline at time t, by Lemma 2, the interval $[t-d_k, t)$ is τ_k^λ-busy. Therefore, the set of all starting points $t' \leq t-\Delta$ of τ_k^λ-busy intervals $[t', t)$ is non-empty. This set must have a minimal member, since there is a start time of the system, before which no jobs arrive. Let $\hat{\Delta} = t - t'$ for this minimum value t' and the lemma is satisfied.

□

Definition 3 *Given a task set S, a release-time assignment r, a task τ_k that has a first missed deadline at time t, and a value $\lambda \geq c_k/T_k$, the unique interval $[t-\hat{\Delta}, t)$ that is guaranteed by Lemma 3 is called* the τ_k^λ-busy interval of τ_k*. From this point on, let $[t-\hat{\Delta}, t)$ denote such an interval.*

The next step in the analysis is to find an upper bound on the work $W_i(t-\hat{\Delta}, t)$ done by each task τ_i in a τ_k^λ-busy interval $[t-\hat{\Delta}, t)$.

Lemma 4 (upper bound) *If t is a first missed deadline of task τ_k, $\lambda \geq c_k/T_k$ and $[t-\hat{\Delta}, t)$ is the corresponding τ_k^λ-busy interval, then for any task τ_i such that $i < k$*

$$\frac{W_i(t-\hat{\Delta}, t)}{\hat{\Delta}} \leq \beta_k^\lambda(i)$$

where

$$\beta_k^\lambda(i) = \begin{cases} \max\{\frac{c_i}{T_i}, \frac{c_i}{T_i}(1 - \frac{d_i}{d_k}) + \frac{c_i}{d_k}\} & \text{if } \frac{c_i}{T_i} \leq \lambda \\ \frac{c_k}{T_k} & \text{if } \frac{c_i}{T_i} > \lambda \text{ and } \lambda \geq \frac{c_i}{d_i} \\ \frac{c_i}{T_i} + \frac{c_i - \lambda d_i}{d_k} & \text{if } \frac{c_i}{T_i} > \lambda \text{ and } \frac{c_i}{d_i} > \lambda \end{cases}$$

proof: The only interesting cases are those where $W_i(t-\hat{\Delta}, t)$ is nonzero, which means that τ_i executes in the interval.

Let $t-\hat{\Delta}-\phi$ be the release time of the first job of τ_i that is released before $t-\hat{\Delta}$ and executes in the interval, if such exists; otherwise, let $\phi = 0$. Observe that the way ϕ is chosen guarantees $\phi < d_i$.

If $\phi > 0$ the interval $[t-\hat{\Delta}-\phi, t-\hat{\Delta})$ is non-empty and must be τ_i-busy. Call this the *preamble* with respect to t_i of $[t-\hat{\Delta}, t)$. Reasoning similar to that of Lemma 2 will be used to bound the amount of execution time that τ_i may carry from the preamble into $[t-\hat{\Delta}, t)$.

$W_i(t-\hat{\Delta}-\phi, t-\hat{\Delta})$ is the total amount of time spent executing jobs of τ_i in the preamble. Since τ_i is backlogged over $[t-\hat{\Delta}-\phi, t-\hat{\Delta})$ the only times in this interval that τ_i does not execute are when it is preempted by jobs of m other tasks. Therefore,

$$B(t-\hat{\Delta}-\phi, t-\hat{\Delta}) \geq \phi - W_i(t-\hat{\Delta}-\phi, t-\hat{\Delta}) \tag{7}$$

Since $[t-\hat{\Delta}, t)$ is τ_k^λ-busy,

$$B(t-\hat{\Delta}, t) > \hat{\Delta} - \lambda(\hat{\Delta} + T_k - d_k) \tag{8}$$

Since $[t-\hat{\Delta}, t)$ is maximal τ_k^λ-busy and $\phi > 0$,

$$B(t-\hat{\Delta}-\phi, t) \leq \hat{\Delta} - \lambda(\hat{\Delta} + T_k - d_k) \tag{9}$$

The block busy time of the concatenation of any two contiguous intervals is the sum of the block busy times of the intervals, so

$$B(t-\hat{\Delta}-\phi, t-\hat{\Delta}) + B(t-\hat{\Delta}, t) = B(t-\hat{\Delta}-\phi, t) \tag{10}$$

From (7)-(10) it follows that

$$\begin{array}{rcl} \phi - W_i(t-\hat{\Delta}-\phi, t-\hat{\Delta}) + \hat{\Delta} - \lambda(\hat{\Delta}+T_k-d_k) & < & B(t-\hat{\Delta}-\phi, t-\hat{\Delta}) + B(t-\hat{\Delta}, t) \\ & \leq & \hat{\Delta} + \phi - \lambda(\hat{\Delta}+\phi+T_k-d_k) \end{array}$$

and so,

$$W_i(t-\hat{\Delta}-\phi, t-\hat{\Delta}) > \phi\lambda \tag{11}$$

Let $j > 0$ be the number of jobs of τ_i that execute in the interval $[t-\hat{\Delta}-\phi, t)$. An upper bound on the total execution time of τ_i in $[t-\hat{\Delta}-\phi, t)$ is

$$W_i(t-\hat{\Delta}-\phi, t) \leq jc_i \tag{12}$$

From (11) and (12),

$$W_i(t-\hat{\Delta}, t) = W_i(t-\hat{\Delta}-\phi, t) - W_i(t-\hat{\Delta}-\phi, t-\hat{\Delta}) < jc_i - \phi\lambda \tag{13}$$

By the minimum separation constraint T_i,

$$j - 1 \leq \frac{\hat{\Delta} + \phi - d_i}{T_i} \tag{14}$$

Combining (13) and (14),

$$\frac{W_i(t-\hat{\Delta}, t)}{\hat{\Delta}} < \frac{c_i(\hat{\Delta}+\phi-d_i)}{T_i\hat{\Delta}} + \frac{c_i - \phi\lambda}{\hat{\Delta}} \tag{15}$$

Let $f(\phi, \hat{\Delta})$ be the function defined by the formula on the right side of inequality (15) above. That is

$$f(\phi, \hat{\Delta}) = \frac{c_i}{T_i} + \frac{c_i}{T_i}\frac{T_i - d_i}{\hat{\Delta}} + \frac{\phi}{\hat{\Delta}}(\frac{c_i}{T_i} - \lambda)$$

The value of $f(\phi, \hat{\Delta})$ is bounded by consideration of the following cases.

Case 1: Suppose $\frac{c_i}{T_i} > \lambda$. It follows that f is increasing with respect to ϕ, and since $\phi < d_i$,

$$\begin{array}{rcl} f(\phi, \hat{\Delta}) & \leq & \frac{c_i}{T_i} + \frac{c_i}{T_i}\frac{T_i - d_i}{\hat{\Delta}} + \frac{d_i}{\hat{\Delta}}(\frac{c_i}{T_i} - \lambda) \\ & = & \frac{c_i}{T_i} + \frac{c_i}{\hat{\Delta}} - \frac{d_i}{\hat{\Delta}}\lambda \end{array}$$

Case 1.1: Suppose $\frac{c_i}{d_i} > \lambda$. It follows that f is decreasing with respect to $\hat{\Delta}$, and since $\hat{\Delta} \geq d_k$,

$$f(\phi, \hat{\Delta}) \leq \frac{c_i}{T_i} + \frac{c_i - \lambda d_i}{d_k} = \beta_k^\lambda(i)$$

Case 1.2: Suppose $\frac{c_i}{d_i} \leq \lambda$. It follows that f is non-decreasing with respect to $\hat{\Delta}$, and taking the limit as $\hat{\Delta} \rightarrow \infty$,

$$f(\phi, \hat{\Delta}) \leq \frac{c_i}{T_i} = \beta_k^\lambda(i)$$

Case 2: Suppose $\frac{c_i}{T_i} \leq \lambda$. It follows that f is non-increasing with respect to ϕ, and since $\phi > 0$,

$$f(\phi, \hat{\Delta}) \leq \frac{c_i}{T_i} + \frac{c_i}{T_i} \frac{T_i - d_i}{\hat{\Delta}}$$

The expression on the right above is decreasing with respect to $\hat{\Delta}$ if-and-only-if $T_i > d_i$. Since $\hat{\Delta} \geq d_k$, it follows that

$$\begin{aligned} f(\phi, \hat{\Delta}) &\leq \frac{c_i}{T_i} + \max\{0, \frac{c_i}{T_i} \frac{T_i - d_i}{d_k}\} \\ &= \max\{\frac{c_i}{T_i}, \frac{c_i}{T_i}(1 - \frac{d_i}{d_k}) + \frac{c_i}{d_k}\} = \beta_k^\lambda(i) \end{aligned}$$

The above upper bounds on $W_i(t - \hat{\Delta}, t)$ and $\frac{W_i(t-\hat{\Delta},t)}{\hat{\Delta}}$ entirely cover the case $\phi > 0$. In the remaining case, where $\phi = 0$, it is clear that $W_i(t - \hat{\Delta}, t)$ cannot be any larger, since there is no carried-in execution of τ_i.

□

5 Schedulability Condition

Lemmas 1 and 4 can be combined to obtain the following schedulability test.

Theorem 1 (EDF Schedulability Condition) *Let $S = \{\tau_1, \ldots, \tau_N\}$ be a set of sporadic tasks, let $\beta_k^\lambda(i)$ be as defined as in Lemma 4 and let $\lambda_k = \lambda \max\{1, \frac{T_k}{d_k}\}$. The task set S is schedulable on m processors using global preemptive EDF scheduling if, for every task τ_k, $k = m+1, \ldots, N$ there exists $\lambda \geq \frac{c_k}{T_k}$ such that one or more of the the following numbered criteria is satisfied*

$$\sum_{i=1}^{N} \max\{\beta_k^\lambda(i), 1 - \lambda_k\} < m(1 - \lambda_k) \tag{16}$$

$$\sum_{i=1}^{N} \min\{\beta_k^\lambda(i), 1 - \lambda_k\} = m(1 - \lambda_k) \text{ and } \exists_i \; 0 < \beta_k^\lambda(i) < 1 - \lambda_k \tag{17}$$

$$\sum_{i=1}^{N} \min\{1, \beta_k^\lambda(i)\} \leq m(1 - \lambda_k) + \lambda_k \tag{18}$$

proof: The proof is by contradiction. Suppose there is a task set S with a release time assignment r for which some task τ_k has a first missed deadline at time t. Let $[t - \hat{\Delta}, t)$ be the τ_k^λ-busy interval guaranteed by Lemma 3.

By the definition of τ_k^λ-busy,

$$\frac{B(t-\hat{\Delta},t)}{\hat{\Delta}} \geq 1 - \lambda + \lambda \frac{T_k - d_k}{\hat{\Delta}} \tag{19}$$

Case A: If $T_k \leq d_k$ then the value of the expression on the right side of the inequality above is non-decreasing with respect to $\hat{\Delta}$, and since $\hat{\Delta} \geq d_k$,

$$\frac{B(t-\hat{\Delta},t)}{\hat{\Delta}} > 1 - \lambda + \lambda \frac{T_k - d_k}{d_k} = 1 - \lambda \frac{T_k}{d_k} \tag{20}$$

Case B: If $T_k > d_k$ then the value of the expression on the right side of inequality (19) is decreasing with respect to $\hat{\Delta}$, and so

$$\frac{B(t-\hat{\Delta},t)}{\hat{\Delta}} > 1 - \lambda \tag{21}$$

Since $\lambda_k = \lambda \max\{1, \frac{T_k}{d_k}\}$, (20) and (21) can be combined into

$$\frac{B(t-\hat{\Delta},t)}{\hat{\Delta}} > 1 - \lambda_k \tag{22}$$

Observe that $W_i(t-\hat{\Delta},t) \leq \hat{\Delta}$ and so by the definition of block busy time

$$\sum_{i=1}^{N} \min\{W_i(t-\hat{\Delta},t), \hat{\Delta}\} = \sum_{i=1}^{N} W_i(t-\hat{\Delta},t) = W(t-\hat{\Delta},t) > (m-1)B(t-\hat{\Delta},t) + \hat{\Delta} \tag{23}$$

It follows from Lemma 4 and (22) that

$$\sum_{i=1}^{N} \min\{\beta_k^\lambda(i), 1\} > \frac{(m-1)B(t-\hat{\Delta},t) + \hat{\Delta}}{\hat{\Delta}} > m(1-\lambda_k) + \lambda_k \tag{24}$$

Therefore condition (18) of the theorem must be false. It remains to show that conditions (16) and (18) must also be false.

By Lemma 1 with $x = 1 - \lambda_k$ and using (22), one of the following two cases must hold:

Case 1:

$$\sum_{i=1}^{N} \min\{\frac{B_i(t-\hat{\Delta},t)}{\hat{\Delta}}, 1 - \lambda_k\} > m(1-\lambda_k) \tag{25}$$

Combining Lemma 4 and (25), and using $B_i(t-\hat{\Delta},t) \leq W_i(t-\hat{\Delta},t)$,

$$\sum_{i=1}^{N} \min\{\beta_k^\lambda(i), 1 - \lambda_k\} > m(1-\lambda_k) \tag{26}$$

This contradicts conditions (16) and (17) of the theorem.

Case 2: $B_i(t-\hat{\Delta},t) < 1-\lambda_k$ if-and-only-if $B_i(t-\hat{\Delta},t) = 0$.

This case directly contradicts condition (17). To see that it also contradicts condition (16), observe that

$$\sum_{i=1}^{N} \min\{\frac{B_i(t-\hat{\Delta},t)}{\hat{\Delta}}, 1-\lambda_k\} = m(1-\lambda_k) \tag{27}$$

Since $W_i(t-\hat{\Delta},t) \geq B_i(t-\hat{\Delta},t)$, by Lemma 4 and (27),

$$\sum_{i=1}^{N} \min\{\beta_k^{\lambda}(i), 1-\lambda_k\} \geq m(1-\lambda_k) \tag{28}$$

□

The above theorem can be used as a schedulability test by testing the three conditions for each value of k. The test is of complexity $O(N^3)$ since the only values of λ that need be considered are the minimum and the points where $\beta_k^{\lambda}(i)$ is discontinuous, i.e.,

- $\lambda = c_i/Ti$, $i = 1, \ldots, N$
- $\lambda = c_i/d_i$, if $d_i > T_i$

6 Prior Work

The theoretical worst-case achievable processor utilizations of the global and partitioned scheduling approaches have been shown to be very similar, for sporadic or aperiodic task sets with deadline equal to period. Andersson, Baruah, and Jonsson[1] showed that the utilization guarantee for EDF or any other static-priority multiprocessor scheduling algorithm – partitioned or global – cannot be higher than $(m+1)/2$ for an m-processor platform.

Goossens, Funk, and Baruah [5] showed that a system of independent periodic tasks can be scheduled successfully on m processors by EDF scheduling if the total utilization is at most $m(1-u_{\max})+u_{\max}$, where $u_{\max}$ is the maximum utilization of any individual task. They also showed that this utilization bound is tight, in the sense that there is no utilization bound $\hat{U} > m(1-u_{\max}) + u_{\max} + \epsilon$, where $\epsilon > 0$, for which $U \leq \hat{U}$ guarantees EDF schedulability.

Srinivasan and Baruah[6] also examined the global EDF scheduling of periodic tasks on multiprocessors, and showed that any system of independent periodic tasks for which the utilization of every individual task is at most $m/(2m-1)$ can be scheduled successfully on m processors if the total utilization is at most $m^2/(2m-1)$.

In 2002, Srinivasan and Baruah[6] proposed a method for dealing with a few heavy tasks, using a *hybrid* scheduling policy. Their idea is to give highest (fixed) priority to to tasks of utilization greater than some constant ζ, and schedule the other tasks according to the basic EDF algorithm. This algorithm is called EDF-US[ζ]. Algorithm EDF-US[$m/(2m-2)$] was shown to correctly schedule on m processors any periodic task system with total utilization $U \leq m^2/(2m-2)$.

Baker[2, 3] derived several sufficient feasibility tests for m-processor preemptive EDF scheduling of sets of periodic and sporadic tasks with arbitrary deadlines, and showed that the optimal value of ζ in EDF-US[ζ] with respect to maximizing the worst-case guaranteed schedulable utilization is $\zeta = 1/2$, for which the utilization bound is $(m+1)/2$. It follows from the argument in [1] that this bound is tight, and it is identical to the worst-case utilization bound for EDF-based first-fit-decreasing (FFD) partitioned scheduling. Baker also proposed extending the EDF-US[ζ] hybrid scheduling model, to define “heaviness” in terms of $\lambda_i = c_i/\min\{D_i, T_i\}$ rather than u_i and to not use a fixed cut-off value.

Bertogna, Cirinei and Lipari[4] made further improvements in global EDF schedulability tests. First, they observed that the proof of the utilization bound test of [5] extends naturally to cover pre-period deadlines if the utilization u_i is replaced by c_i/D_i. As observed by Sanjoy Baruah[1], the same proof extends to the case of post-period deadlines if c_i/D_i is replaced by $\lambda_i = c_i/\min\{D_i, T_i\}$.

Theorem 2 (GFB) *A set of sporadic tasks $\tau_1,\ldots,\tau_N$ is EDF schedulable on m identical processors if*

$$\sum_{i=1}^{N} \lambda_i \leq m - \lambda_{\max}(m-1)$$

where $\lambda_{\max} = \max\{\lambda_i | i = 1, \ldots, N\}$.

Bertogna *et al.* also developed the following new schedulability test.

Theorem 3 (BCL) *A set of sporadic tasks $\tau_1,\ldots,\tau_N$ (with constraint $d_i \leq T_i$) is EDF schedulable on m identical processors if for each task τ_k one of the following is true:*

$$\sum_{i \neq k} \min\{\beta_i, 1 - \lambda_k\} < m(1 - \lambda_k) \tag{29}$$

$$\begin{array}{l} \sum_{i \neq k} \min\{\beta_i, 1 - \lambda_k\} = m(1 - \lambda_k) \quad \textit{and} \\ \exists i \neq k : 0 < \beta_i \leq 1 - \lambda_k \end{array} \tag{30}$$

where

$$\beta_i = \frac{N_i c_i + \min\{c_i, \max\{0, d_k - N_i T_i\}\}}{d_k}$$

and

$$N_i = \left\lfloor \frac{d_k - d_i}{T_i} \right\rfloor + 1$$

Bertogna *et al.* demonstrated that the BCL, GFB, and Baker[2, 3] tests are generally incomparable, but observed that the BCL test seemed to do better than the rest on task sets with a few "heavy" (high utilization) tasks. They reported simulations on collections of pseudo-randomly generated tasks sets with a few heavy tasks, for which the BCL was able to discover significantly more schedulable task sets than either of the other two tests. However, they did not compare these results against the EDF-US[ζ] hybrid method of handling heavy tasks proposed in 2002 by Srinivasan and Baruah[6] or the more general hybrid models proposed by Baker.

The main contribution of the current paper is to take the observation of BCL, that if a task τ_k misses a deadline the maximum fraction of the workload of any task that can contribute to the interference is $1 - \lambda_k$, and combine it with the busy-interval analysis of [3], to obtain a tighter bound than either method could achieve alone. A secondary contribution is to demonstrate that a hybrid scheme, in which up to m "heavy" tasks are chosen to receive top priority, is significantly more effective than pure EDF scheduling, for task systems with some heavy tasks.

7 Empirical Comparisons

To demonstrate that the new schedulability test does offer an improvement in accuracy over the previous tests, a series of experiments were conducted on pseudo-randomly generated sets of sporadic tasks. The tests compared are:

[1] personal communication

BAK Baker's test from [2, 3]
GFB Goosens, Funk and Baruah's test, extended to arbitrary deadlines by Bertogna, Cirinei and Lipari, as stated Theorem 2 above.
BCL Bertogna, Cirinei, and Lipari's test, as stated in Theorem 3 above.
BAK2 Baker's revised test, as stated in Theorem 1 above.
GFB+BCL Apply the GFB test first, and if it fails, apply the BCL test.
GFB+BAK2 Apply the GFB test first, and if it fails, apply the BAK2 test.
GFB+BCL+BAK2 Apply the GFB+BCL, and if it fails, apply BAK2.

7.1 Generation of Datasets

The performance of the above schedulability tests was evaluated on several datasets. Each dataset contained 1,000,000 sets of tasks. The task periods were generated pseudo-randomly with a uniform distribution between 1 and 1000. The processor utilizations (and, implicitly, the compute times) were chosen according to the following distributions, truncated to bound the utilization between 0.001 and 0.999.

1. uniform distribution, between 1/period and 1
2. bimodal distribution: heavy tasks uniform between 0.5 and 1; light tasks uniform between 1/period and 0.5; probability of being heavy = 1/3[2]
3. exponential distribution with mean 0.25
4. exponential distribution with mean 0.50

The deadlines were chosen in two different ways:

A constrained: deadlines uniformly distributed between the execution time and the period

B unconstrained: deadlines uniformly distributed between the execution time and 4 times the period (for all but the BCL test)

Separate datasets were generated for 2, 4, and 8 processor systems, as follows. An initial set of $m+1$ tasks was generated, and tested. Then another task was generated and added to the previous set, and all the schedulability tests were run on the new set. This process of adding tasks was repeated until the total processor utilization exceeded m. The whole procedure was then repeated, starting with a new initial set of $m + 1$ tasks.

All five schedulability tests, and were run on these 24 datasets. Only the results of a few of the experiments are reported here, due to space limitations and because the trends across the experiments were quite similar.

7.2 Results for Basic EDF

Figures 1-3 compare the success rates of the GFB, BCL, and BAK2 tests on three of the datasets. Like most of the other graphs in this paper, each is a histogram of 100 buckets, each bucket corresponding to a range of 1 percent of the full range of total utilizations possible for a given number of processors. So, with four processors the buckets each represent a total processor utilization range of four percent. The three lower lines in each graph show how many task sets were verifiably feasible according to each of the three tests. The upper line shows the total number of task sets in each bucket, including both feasible and infeasible task sets.

[2]Intended to bias toward cases with a few heavy tasks, similar to the experiments of Bertogna, Cirinei, and Lipari[4].

Infeasible task sets are included in the count because the only necessary and sufficient test for feasibility of global EDF scheduling of N tasks on m processors known to this author has worst-case execution time of the order $O(mN \cdot \Pi_{i=1}^{N} T_i c_i)$. The author implemented and tested that algorithm, but running it on datasets of the size considered here was not practical. Reporting the relative performance of the efficient sufficient tests of feasibility on large numbers of tasks sets seemed more important than comparing them against perfection on a much smaller number of task sets, with smaller periods.

Note that in [4] it is claimed that "simulation of the schedule up to the hyper-period checking for missed deadlines" is a necessary and sufficient test for schedulability. However, this author is not aware of any proof that such a simulation is a sufficient test for feasibility of sporadic task sets, or even of periodic tasks sets with arbitrary initial release time offsets. Even under the assumption of strictly periodic tasks and simultaneous start times, if periods can exceed deadlines simulation to the hyperperiod is not sufficient.

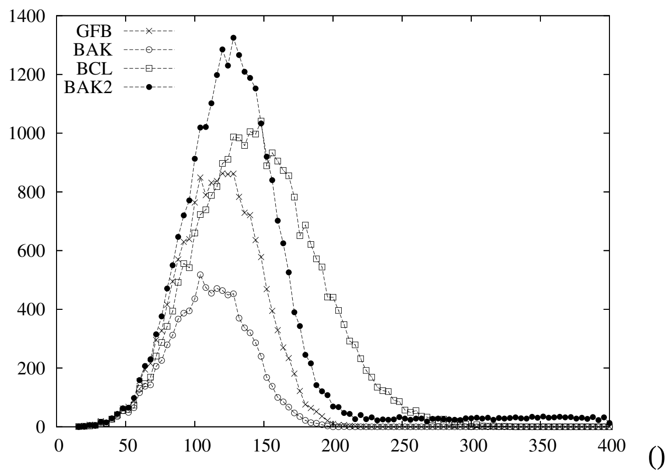

Figure 1. Constrained deadline, bimodal utilization distribution, 4 CPUs

Figure 1 is for one of the datasets were the BCL test was more effective than the other tests over some range of the utilization distribution. The dataset shown is for constrained deadlines and bimodal task utilizations on four processors.

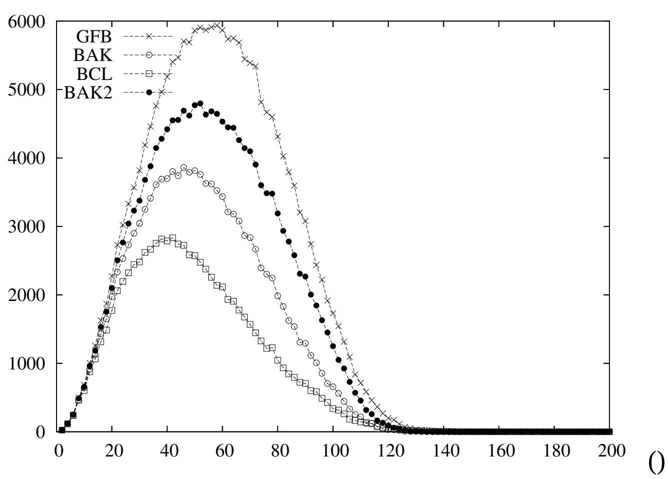

Figure 2. Constrained deadline, exponential utilization w/mean 0.25, 2 CPUs

Figure 2 is for one of the datasets where the GFB test was most accurate over some range of the utilization distribution.

The dataset shown is for unconstrained deadlines and exponentially distributed task utilizations with mean 0.25 on two processors.

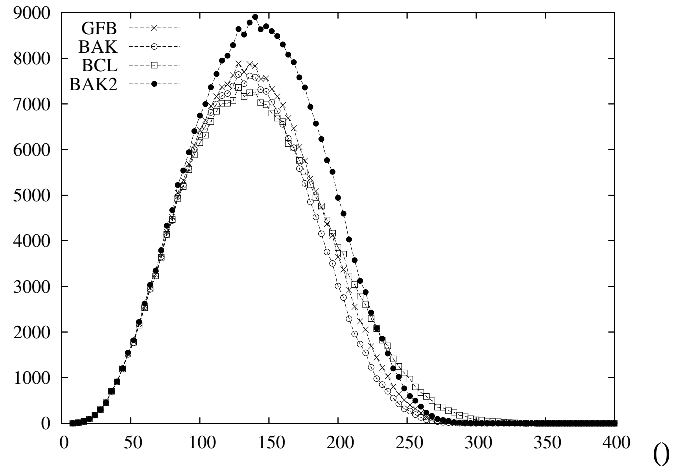

Figure 3. Unconstrained deadline, exponential utilization w/mean 0.25, 4 CPUs

Figure 3 is for a 4-cpu dataset with unconstrained deadlines and exponentially distributed task utilizations with mean 0.25.

Because the three tests each have some cases where they are more accurate in determining feasibility, it makes sense to apply them in combination, starting with the computationally least expensive, that is:

1. apply the GFB test
2. if the GFB test fails, apply the BCL test
3. if the BCL test fails, apply the BAK2 test

Figures 4-6 show the results of applying such a three-stage test on the same datasets reported in Figures 1-3. These are typical of the results on all of the 24 datasets.

It is clear that the combination of the three tests, GFB+BCL+BAK2, is the winner among the feasibility tests for pure EDF scheduling. This test is called “GBB” for short, from this point on.

7.3 Hybrid Scheduling Schemes

In addition to basic EDF scheduling, the performance of the following hybrids of EDF and highest-utilization-first scheduling was tested:

1. EDF-US[1/2]: give special priority to the tasks of utilization greater than 1/2, which is the cut-off value that guarantees the highest worst-case utilization when deadline=period;

2. EDF-UM: give special priority to the k tasks with highest utilization, where k is the smallest value between 0 and m for which the system can be verified as schedulable according to the GBB Test.

3. EDF-LM: give special priority to the k tasks with highest value of $c_i / \max\{T_i, d_i\}$, where k is the smallest value between 0 and m for which the system can be verified as schedulable according to the GBB Test.

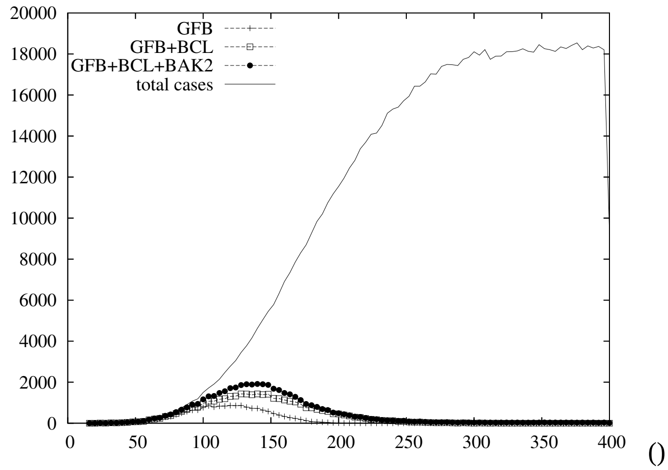

Figure 4. Constrained deadline, bimodal utilization distribution, 4 CPUs

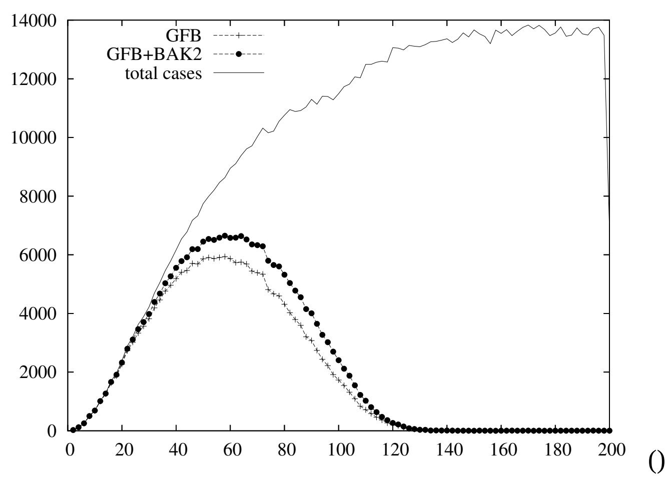

Figure 5. Constrained deadline, exponential utilization w/mean 0.25, 2 CPUs

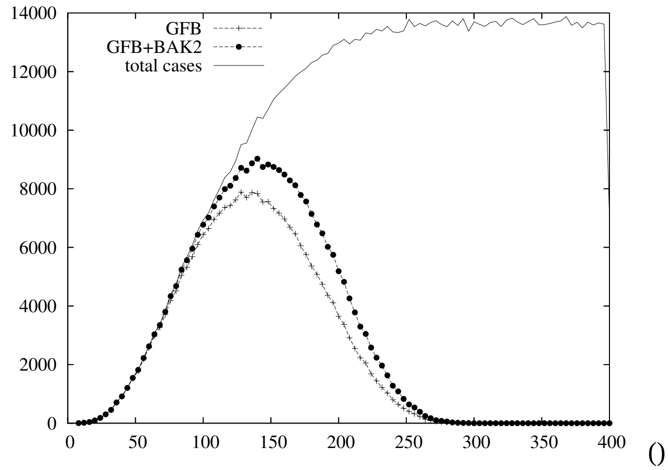

Figure 6. Unconstrained deadline, exponential utilization w/mean 0.25, 4 CPUs

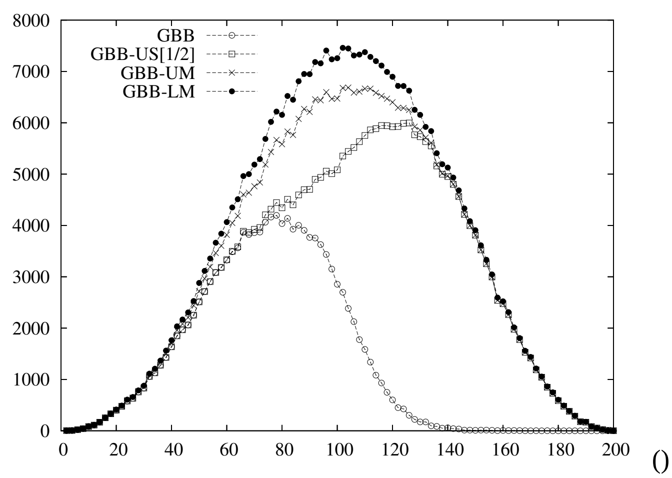

Figure 7. Constrained deadline, bimodal utilization, 2 CPUs

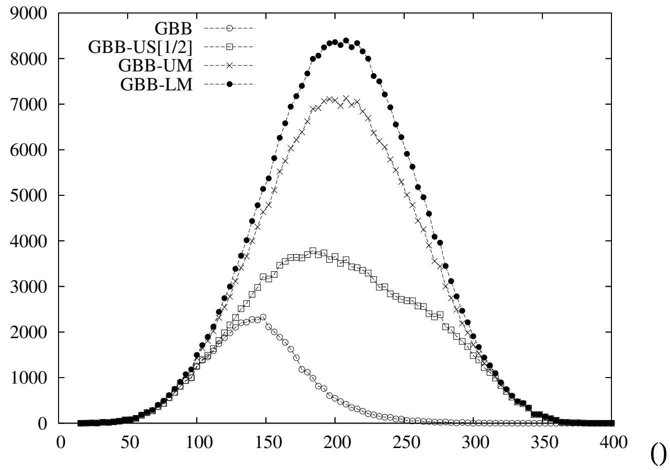

Figure 8. Constrained deadline, bimodal utilization, 4 CPUs

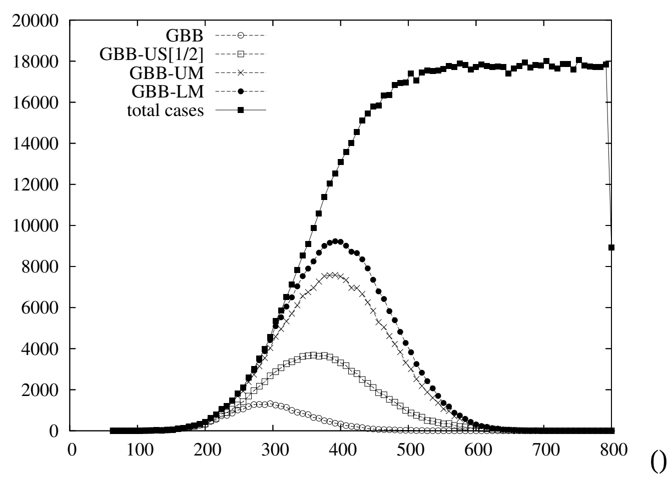

Figure 9. Constrained deadline, bimodal utilization, 8 CPUs

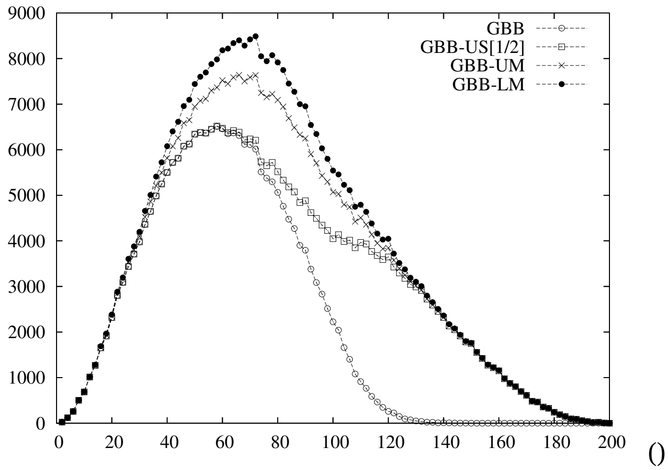

Figure 10. Constrained deadline, exponential utilization w/mean 0.25, 2 CPUs

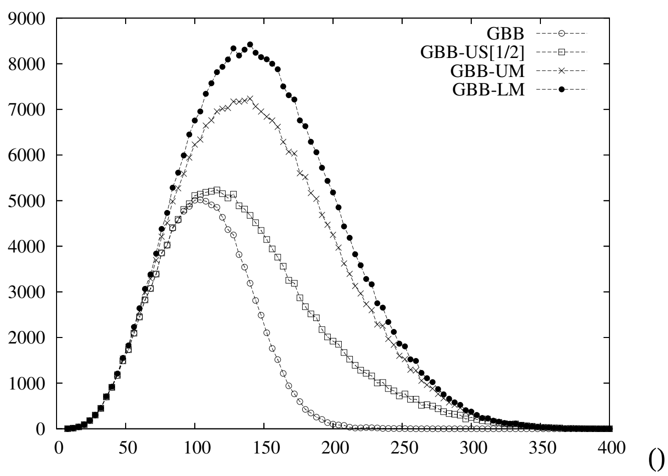

Figure 11. Constrained deadline, exponential utilization w/mean 0.25, 4 CPUs

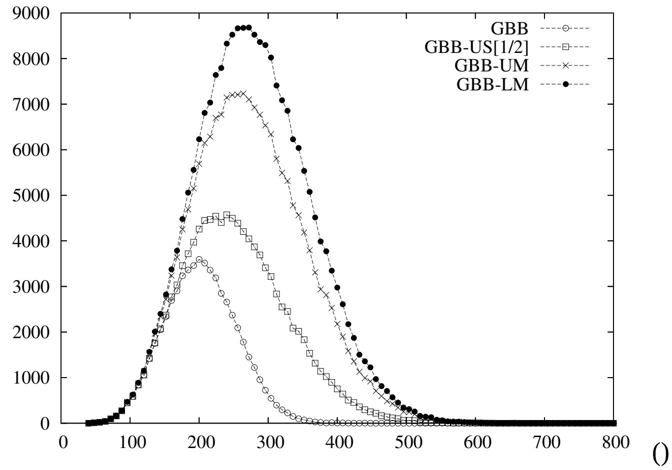

Figure 12. Constrained deadline, exponential utilization w/mean 0.25, 8 CPUs

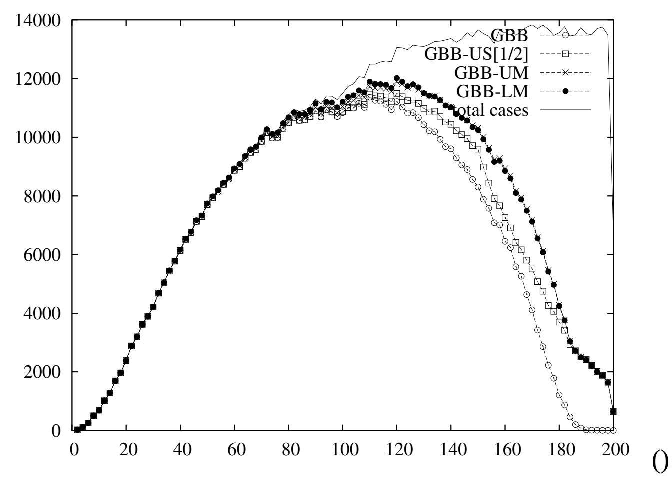

Figure 13. Unconstrained deadline, exponential utilization w/mean 0.25, 2 CPUs

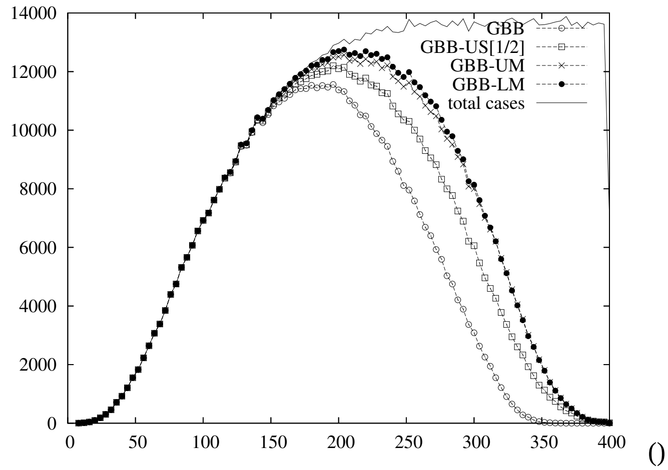

Figure 14. Unconstrained deadline, exponential utilization w/mean 0.25, 4 CPUs

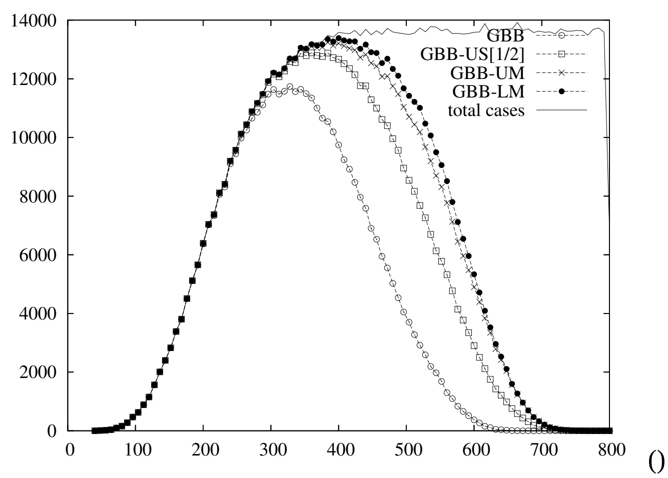

Figure 15. Unconstrained deadline, exponential utilization w/mean 0.25, 8 CPUs

Figures 7-15 show the results of applying these three hybrid EDF scheduling policies with the GBB test on the same datasets reported in Figures 1-3. The GBB test was applied to the remaining $N-k$ tasks on $m-k$ processors, after choosing k special tasks to receive top priority. These result are typical of what was observed on all of the 24 datasets. The EDF-LM hybrid scheme clearly finds the highest number of verifiably schedulable task sets at every total utilization level.

8 Conclusions

Theorem 1 provides a new schedulability test, that is able to verify the feasibility under EDF multiprocessor scheduling of a significant number of task systems that could not be verified using previously known tests, including those recently by Bertogna, Cirinei, and Lipari[4] and those proposed earlier by Baker[2, 3] and Goossens, Funk, and Baruah [5]. Combining these tests has been shown to be advantageous, especially in a hybrid scheduling scheme where which a few tasks, up to one minus the number of processors, are given special top priority and the other tasks are scheduled by EDF priority.

References

[1] B. Andersson, S. Baruah, and J. Jonsson. Static-priority scheduling on multiprocessors. In *Proc. 22nd IEEE Real-Time Systems Symposium*, pages 193–202, London, UK, Dec. 2001.

[2] T. P. Baker. Multiprocessor EDF and deadline monotonic schedulability analysis. In *Proc. 24th IEEE Real-Time Systems Symposium*, pages 120–129, 2003.

[3] T. P. Baker. An analysis of EDF scheduling on a multiprocessor. *IEEE Trans. on Parallel and Distributed Systems*, 15(8):760–768, Aug. 2005.

[4] M. Bertogna, M. Cirinei, and G. Lipari. Improved schedulability analysis of EDF on multiprocessor platforms. In *Proc. of the 17th Euromicro Conference on Real-Time Systems*, Palma de Mallorca, Spain, July 2005.

[5] J. Goossens and R. Devillers. Feasibility intervals for the deadline driven scheduler with arbitrary deadlines. In *Proc. 6th International Conf. Real-Time Computing Systems and Applications (RTCSA'99)*, 1999.

[6] A. Srinivasan and S. Baruah. Deadline-based scheduling of periodic task systems on multiprocessors. *Information Processing Letters*, 84:93–98, 2002.