

UNIVERSIDAD DE OVIEDO

Departamento de Ingeniería Eléctrica,
Electrónica, de Computadores y Sistemas

Tesis doctoral

ANÁLISIS DE PLANIFICABILIDAD BASADO EN UTILIZACIONES DE
SISTEMAS DE TIEMPO REAL IMPLEMENTADOS SOBRE
MULTIPROCESADORES CON TÉCNICAS DE PARTICIONADO

presentada por

José María López López

para la obtención del grado de

Doctor Ingeniero Industrial

Director: Daniel Fernando García Martínez

Gijón, Enero de 2001

A mi esposa Silvia, y a mi hija Andrea.

Agradecimientos

La consecución de esta tesis no hubiera sido posible sin el apoyo y trabajo de muchas personas.

En primer lugar, me gustaría agradecer a mi esposa Silvia y a mis padres todo el apoyo que me han prestado. A mi esposa Silvia y a mi chiquitina Andrea por todo el tiempo que les he robado. A mis compañeros del área de Arquitectura y Tecnología de Computadores, Antonio, Arias, Daniel, Francisco, Javi, Joaquín, José Luis y Manuel, pues la tesis es en buena parte fruto del trabajo de todos ellos.

Muchas gracias a todos.

Resumen

Esta tesis se enmarca dentro del contexto general de investigación del Área de Arquitectura y Tecnología de Computadores de la Universidad de Oviedo, relativa al estudio de sistemas de tiempo real. En particular se estudia la planificabilidad de sistemas de tiempo real multiprocesador mediante la obtención de límites de utilización. La planificación de las tareas en el multiprocesador se realiza siguiendo la estrategia de particionado.

Se obtienen los límites de utilización para planificación EDF y RM empleando diferentes algoritmos de asignación de tareas a procesadores. Esto permite analizar la planificabilidad del sistema de una forma rápida, teniendo en cuenta además la influencia de parámetros clave como el tamaño de las tareas.

Aparte del análisis teórico de planificabilidad se realizan numerosos experimentos de simulación que permiten obtener el comportamiento típico de diferentes opciones de planificación monoprocesador y asignación.

Inicialmente el modelo de tareas considerado contiene exclusivamente tareas periódicas independientes. Al final de la tesis se extiende este modelo, lo que permite analizar conjuntos de tareas más realistas.

Abstract

This thesis is included in the general research context of the Computer Engineering group in the University of Oviedo, related to the investigation in Real-Time systems. In particular, this work deals with schedulability issues by means of utilization bounds. Task scheduling is performed following the partitioning strategy.

The utilization bounds for EDF and RM scheduling, and for different allocation algorithms are obtained. This allows us to carry out not only fast schedulability tests, but also to consider key parameters, such as the task size.

Many simulation experiments are carried out in order to obtain the average-case behaviour for different uniprocessor schedulers and different allocation algorithms.

The initial model of tasks includes only periodic and independent tasks. This model is generalised at the end of the thesis to cope with more realistic task sets.

Abreviaturas

AA	Algoritmo arbitrario de asignación
BF	Best Fit
BFD	Best Fit Decreasing
BFI	Best Fit Increasing
EDF	Earliest Deadline First
FF	First Fit
FFD	First Fit Decreasing
FFI	First Fit Increasing
PCP	Priority Ceiling Protocol
POSIX	Portable Operating System Interface
RA	Algoritmo razonable de asignación
RAD	Algoritmo razonable de asignación de tareas con ordenación en sentido decreciente
RF	Random Fit
RFD	Random Fit Decreasing
RFI	Random Fit Increasing
RM	Rate Monotonic
SRP	Stack Resource Policy
WF	Worst Fit
WFD	Worst Fit Decreasing
WFI	Worst Fit Increasing

Índice General

1	Introducción	1
1.1	Sistemas de tiempo real	1
1.2	El problema de la planificación en los sistemas de tiempo real	2
1.2.1	Recursos	4
1.2.2	Consumidores de recursos	5
1.2.3	Algoritmos de planificación	7
1.2.4	Análisis de planificabilidad	10
1.3	Problemas de planificación en el contexto de la tesis	13
1.4	Planificación en sistemas monoprocesador	15
1.4.1	Planificación RM	16
	Técnicas de análisis de planificabilidad basadas en utilizaciones	16
	Técnicas de análisis de planificabilidad basadas en tiempos de respuesta	17
1.4.2	Planificación EDF	18
	Técnicas de análisis de planificabilidad basadas en utilizaciones	18
	Técnicas de análisis de planificabilidad basadas en tiempos de respuesta	19
1.5	Planificación en sistemas multiprocesador	20
2	Definición del problema	25
2.1	Definición de las instancias del problema	25
2.1.1	Algoritmos de asignación de tareas a procesadores	28
2.2	Definición del problema en términos de utilización	34
3	Límites de utilización empleando planificación EDF	37
3.1	Definición y existencia de los límites de utilización	38
3.2	Cálculo de los límites de utilización	40
3.2.1	Rango de los límites de utilización	42
3.2.2	Límite de utilización para EDF-FF	46
3.2.3	Límite de utilización para EDF-BF	53
3.2.4	Límite de utilización para EDF-WF	58
3.2.5	Límite de utilización para EDF-RF	61
3.2.6	Límite de utilización para EDF-OPT	62
3.2.7	Límites de utilización empleando ordenación	63
	Algoritmos de asignación FFD y FFI	64
	Algoritmos de asignación BFD y BFI	64

	Algoritmos de asignación WFD y WFI	65
	Algoritmos de asignación RFD y RFI	69
3.3	Análisis de los resultados teóricos	70
3.3.1	Análisis del límite $L_{EDF}(n, \alpha)$	71
3.3.2	Análisis del límite $H_{EDF}(n, \beta_{EDF})$	73
3.4	Obtención de los límites de utilización estadísticos	76
3.5	Comparación de los algoritmos de asignación de tareas	129
4	Límites de utilización empleando planificación RM	141
4.1	Definición y existencia de los límites de utilización	142
4.2	Cálculo de los límites de utilización	144
4.2.1	Rango de los límites de utilización	146
4.2.2	Límite de utilización para RM-FF	158
4.2.3	Límite de utilización para RM-BF	167
4.2.4	Límite de utilización para RM-WF	174
4.2.5	Límite de utilización para RM-RF	180
4.2.6	Límite de utilización para RM-OPT	181
4.2.7	Límites de utilización empleando ordenación	182
	Algoritmos de ordenación en sentido decreciente	183
	Algoritmos de ordenación en sentido creciente	186
4.3	Análisis de los resultados teóricos	195
4.3.1	Análisis del límite $L_{RM}(m, n, \alpha)$	197
4.3.2	Análisis del límite $I_{RM}(m, n, \alpha)$	199
4.3.3	Análisis del límite $K_{RM}(m, n, \alpha)$	200
4.3.4	Análisis del límite $H_{RM}(n > 1, \beta_{RM})$	204
4.4	Obtención de los límites de utilización estadísticos	206
5	Mejora de los límites de utilización en multiprocesadores	213
5.1	Consideración del máximo factor de utilización	215
5.1.1	Planificación EDF	215
5.1.2	Planificación RM	219
5.2	Consideración de los factores de utilización de las tareas	224
6	Extensiones del modelo de tareas	227
6.1	Acceso a recursos compartidos	227
6.1.1	Planificación EDF	229
6.1.2	Planificación RM	231
6.2	<i>Jitter</i> de activación	232
6.3	Plazos menores que los periodos	234
6.4	Tareas aperiódicas	235
6.5	Cambios de modo	236
7	Conclusiones	241
A	Relaciones matemáticas entre números enteros	245
	Referencias	246

Índice de Figuras

1.1	Los diferentes aspectos en la planificación de sistemas de tiempo real.	3
1.2	Problemas de planificación en el contexto de la tesis.	14
3.1	Representación de la función $\beta_{EDF}(\alpha)$, y la condición de planificabilidad asociada.	41
3.2	Situación general en el <i>caso 2</i> del teorema 3.	49
3.3	Situación general en el <i>caso 2</i> del teorema 5.	55
3.4	Representación del límite $L_{EDF}(n, \alpha)$.	71
3.5	Representación del límite $H_{EDF}(n, \beta_{EDF})$.	73
3.6	Representación de la diferencia $H_{EDF}(n, \beta_{EDF}) - L_{EDF}(n, \alpha)$ para el caso $n = 2$.	75
3.7	Relación entre los conceptos de límite de utilización absoluto y estadístico para unos valores de m y n fijos, y para una distribución estadística dada.	77
3.8	Representación del límite de utilización exacto para el caso de factores de utilización iguales y planificación EDF-RA.	79
3.9	Límites de utilización para planificación EDF-FF y $p = 50\%$.	81
3.10	Límites de utilización para planificación EDF-FF y $p = 75\%$.	82
3.11	Límites de utilización para planificación EDF-FF y $p = 90\%$.	83
3.12	Límites de utilización para planificación EDF-FF y $p = 99\%$.	84
3.13	Límites de utilización para planificación EDF-FFD y $p = 50\%$.	85
3.14	Límites de utilización para planificación EDF-FFD y $p = 75\%$.	86
3.15	Límites de utilización para planificación EDF-FFD y $p = 90\%$.	87
3.16	Límites de utilización para planificación EDF-FFD y $p = 99\%$.	88
3.17	Límites de utilización para planificación EDF-FFI y $p = 50\%$.	89
3.18	Límites de utilización para planificación EDF-FFI y $p = 75\%$.	90
3.19	Límites de utilización para planificación EDF-FFI y $p = 90\%$.	91
3.20	Límites de utilización para planificación EDF-FFI y $p = 99\%$.	92
3.21	Límites de utilización para planificación EDF-BF y $p = 50\%$.	93
3.22	Límites de utilización para planificación EDF-BF y $p = 75\%$.	94
3.23	Límites de utilización para planificación EDF-BF y $p = 90\%$.	95
3.24	Límites de utilización para planificación EDF-BF y $p = 99\%$.	96
3.25	Límites de utilización para planificación EDF-BFD y $p = 50\%$.	97
3.26	Límites de utilización para planificación EDF-BFD y $p = 75\%$.	98
3.27	Límites de utilización para planificación EDF-BFD y $p = 90\%$.	99
3.28	Límites de utilización para planificación EDF-BFD y $p = 99\%$.	100

3.29	Límites de utilización para planificación EDF-BFI y $p = 50\%$	101
3.30	Límites de utilización para planificación EDF-BFI y $p = 75\%$	102
3.31	Límites de utilización para planificación EDF-BFI y $p = 90\%$	103
3.32	Límites de utilización para planificación EDF-BFI y $p = 99\%$	104
3.33	Límites de utilización para planificación EDF-WF y $p = 50\%$	105
3.34	Límites de utilización para planificación EDF-WF y $p = 75\%$	106
3.35	Límites de utilización para planificación EDF-WF y $p = 90\%$	107
3.36	Límites de utilización para planificación EDF-WF y $p = 99\%$	108
3.37	Límites de utilización para planificación EDF-WFD y $p = 50\%$	109
3.38	Límites de utilización para planificación EDF-WFD y $p = 75\%$	110
3.39	Límites de utilización para planificación EDF-WFD y $p = 90\%$	111
3.40	Límites de utilización para planificación EDF-WFD y $p = 99\%$	112
3.41	Límites de utilización para planificación EDF-WFI y $p = 50\%$	113
3.42	Límites de utilización para planificación EDF-WFI y $p = 75\%$	114
3.43	Límites de utilización para planificación EDF-WFI y $p = 90\%$	115
3.44	Límites de utilización para planificación EDF-WFI y $p = 99\%$	116
3.45	Límites de utilización para planificación EDF-RF y $p = 50\%$	117
3.46	Límites de utilización para planificación EDF-RF y $p = 75\%$	118
3.47	Límites de utilización para planificación EDF-RF y $p = 90\%$	119
3.48	Límites de utilización para planificación EDF-RF y $p = 99\%$	120
3.49	Límites de utilización para planificación EDF-RFD y $p = 50\%$	121
3.50	Límites de utilización para planificación EDF-RFD y $p = 75\%$	122
3.51	Límites de utilización para planificación EDF-RFD y $p = 90\%$	123
3.52	Límites de utilización para planificación EDF-RFD y $p = 99\%$	124
3.53	Límites de utilización para planificación EDF-RFI y $p = 50\%$	125
3.54	Límites de utilización para planificación EDF-RFI y $p = 75\%$	126
3.55	Límites de utilización para planificación EDF-RFI y $p = 90\%$	127
3.56	Límites de utilización para planificación EDF-RFI y $p = 99\%$	128
3.57	Comparación de los límites de utilización estadísticos para planificación EDF-OPT y EDF-FFD para $p = 50\%$	132
3.58	Comparación de los límites de utilización estadísticos para planificación EDF-OPT y EDF-FFD para $p = 75\%$	133
3.59	Comparación de los límites de utilización estadísticos para planificación EDF-OPT y EDF-FFD para $p = 90\%$	134
3.60	Comparación de los límites de utilización estadísticos para planificación EDF-OPT y EDF-FFD para $p = 99\%$	135
3.61	Comparación de los límites de utilización estadísticos para planificación EDF-FFD y EDF-FF para $p = 50\%$	137
3.62	Comparación de los límites de utilización estadísticos para planificación EDF-FFD y EDF-FF para $p = 75\%$	138
3.63	Comparación de los límites de utilización estadísticos para planificación EDF-FFD y EDF-FF para $p = 90\%$	139
3.64	Comparación de los límites de utilización estadísticos para planificación EDF-FFD y EDF-FF para $p = 99\%$	140

4.1	Representación de la función $\beta_{RM}(\alpha)$, y la condición de planificabilidad asociada.	145
4.2	Representación de las cotas inferiores de los límites de utilización proporcionadas por los teoremas 12, 13, y 14.	155
4.3	Situación general en el <i>caso 2</i> del teorema 17.	163
4.4	Situación general en el <i>caso 2</i> del teorema 20.	169
4.5	Asignación de las $(m - 1)$ primeras tareas en el <i>caso 1</i> del teorema 22, para $m = 18$ y $n = 5$	176
4.6	Asignación de las $(m - 1)$ primeras tareas en el <i>caso 2</i> del teorema 22, para $m = 18$ y $n = 5$	178
4.7	Ejemplo de asignación de tareas empleando el algoritmo WFI, para $m = 18$ y $n = 5$	188
4.8	Asignación de las $(m - 1)$ primeras tareas en el <i>caso 1</i> del teorema 25, para $m = 18$ y $n = 5$	191
4.9	Representación del límite $L_{RM}(m, n, \alpha)$	198
4.10	Representación del límite $I_{RM}(m, n, \alpha)$	200
4.11	Representación del límite $K_{RM}(m, n, \beta_{RM})$	201
4.12	Representación del límite $K_{RM}(m, n, 1)$	202
4.13	Representación del límite $K_{RM}(m, n, 3)$	203
4.14	Representación del límite $H_{RM}(n > 1, \beta_{RM})$	205
4.15	Relación entre los conceptos de límite de utilización (absoluto) y estadístico para unos valores de m y n fijos, y para una distribución estadística dada.	207
4.16	Representación del límite de utilización para el caso de factores de utilización iguales y planificación RM-RA.	209
4.17	Comparación de los límites de utilización estadísticos para planificación EDF-FF y RM-FF.	210
4.18	Evolución del límite de utilización estadístico para planificación RM-WF cuando se incrementa la desviación típica de la distribución.	211
6.1	Concepto de independencia y definición de macrotareas.	229
6.2	Ejemplo de tarea periódica con <i>jitter</i>	232
6.3	Transformación de un conjunto de tareas con <i>jitter</i> a un conjunto de tareas sin <i>jitter</i>	233
6.4	Cambios de modo en multiprocesadores.	238
7.1	Representación del límite de utilización para planificación RM-FF, y tareas de factor de utilización menor o igual que 0'26. La línea discontinua muestra el límite de utilización propuesto por Oh y Baker.	244

Índice de Tablas

1.1	Opciones del problema de planificación consideradas en la tesis. . . .	13
1.2	Valores de la métrica (N_{AA}/N_{OPT}) cuando $N_{opt} \rightarrow \infty$, para diferentes parejas algoritmo de planificación monoprocesador-algoritmo de asignación de tareas.	23
5.1	Máximos incrementos del límite de utilización multiprocesador L_{RM} , para distintos valores de α	225
6.1	Ejemplo de análisis de planificabilidad de tareas dependientes en multiprocesadores.	231

Capítulo 1

Introducción

Este capítulo presenta en primer lugar una breve introducción a los sistemas de tiempo real en el apartado 1.1.

El apartado 1.2 presenta de forma general el problema de planificación de sistemas de tiempo real.

En el apartado 1.3 se restringe el problema general de planificación, considerando sólo aquellos subproblemas de planificación relevantes a la tesis.

1.1 Sistemas de tiempo real

Los sistemas de tiempo real se definen como aquellos sistemas computacionales que deben reaccionar a los eventos de su entorno cumpliendo ciertas restricciones temporales. Como consecuencia, el comportamiento correcto de dichos sistemas depende no sólo de los resultados de la computación, sino que depende también de los instantes en los cuales se producen dichos resultados (Stankovic, 1988; Buttazzo, 1997).

Lo que distingue a los sistemas de tiempo real de los sistemas computacionales tradicionales es la presencia de restricciones temporales sobre los instantes en los que se generan los resultados.

Estas restricciones temporales son habitualmente una consecuencia del impacto que tienen los sistemas que controlan actividades de su entorno. Por ejemplo, si el sistema de navegación de un avión no es capaz de generar las señales de control dentro de un margen de tiempo establecido, las consecuencias pueden ser muy graves. De nada sirve una respuesta de gran calidad pero tardía del sistema de navegación, pues esta respuesta puede llegar cuando el avión ha perdido el control.

Otras aplicaciones en las cuales se requieren sistemas de tiempo real son las siguientes:

- Control industrial
- Monitores industriales
- Robótica
- Sistemas militares

- Sistemas multimedia
- Automóviles
- Sistemas transaccionales

Puesto que la presencia de restricciones temporales es la característica diferenciadora de los sistemas de tiempo real con respecto a los sistemas computacionales tradicionales, resulta natural clasificar a los sistemas de tiempo real de acuerdo con el tipo de restricciones temporales. Desde un punto de vista simplista, los sistemas de tiempo real pueden clasificarse en críticos (*hard real-time*) y acríticos (*soft real-time*).

Un sistema de tiempo real crítico es aquel en el cual el incumplimiento de alguna de las restricciones temporales ocasiona o puede ocasionar consecuencias muy graves. Por el contrario, en un sistema de tiempo real acrítico el cumplimiento de las restricciones temporales es deseable pero su incumplimiento no trae consigo graves consecuencias. Por supuesto, el término “grave” es relativo y depende de cada caso particular.

Uno de los problemas que aparecen en los sistemas de tiempo real es el problema de la correcta administración de recursos que asegure el cumplimiento de las restricciones temporales. A esta administración de recursos se la conoce como planificación. El procesador, o en general los procesadores, son los recursos que más habitualmente se consideran en la planificación de sistemas de tiempo real. Sin embargo, cada vez más se considera la administración de recursos de entrada/salida, como son las redes de comunicación y los sistemas masivos de almacenamiento de información.

1.2 El problema de la planificación en los sistemas de tiempo real

La teoría de planificación de sistemas de tiempo real considera cuatro aspectos básicos del problema:

- Recursos (*hardware*¹ del sistema)
- Consumidores de recursos (tareas)
- Algoritmos de asignación de recursos a consumidores (algoritmos de planificación)
- Análisis de la planificabilidad (¿el algoritmo de planificación asigna recursos a tareas de tal manera que se cumplen las restricciones temporales?)

La figura 1.1 muestra de manera esquemática los cuatro aspectos implicados en la planificación de sistemas de tiempo real, así como las opciones más habituales en cada uno de ellos.

¹En ocasiones también se consideran recursos de más alto nivel.

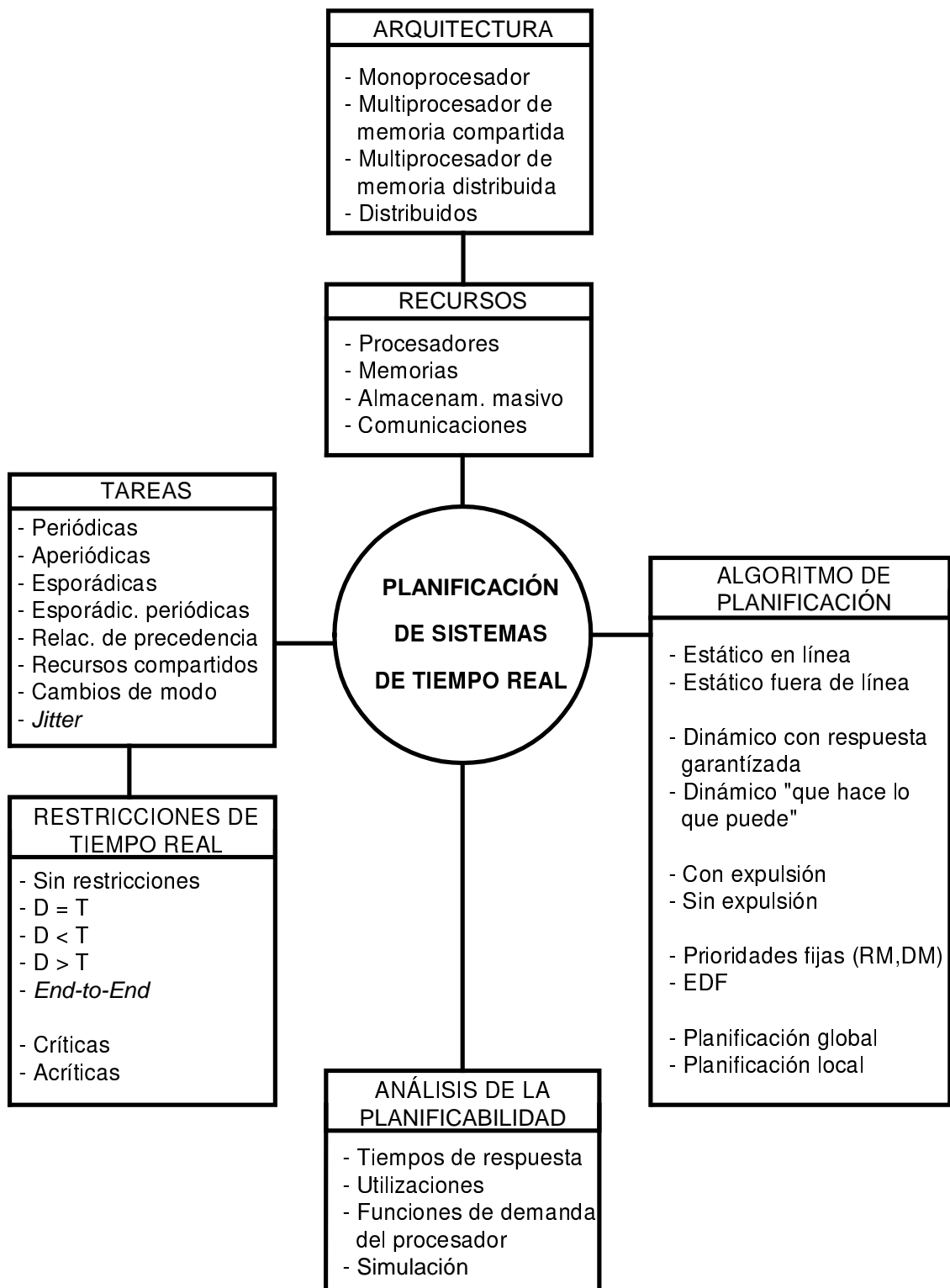


Figura 1.1: Los diferentes aspectos en la planificación de sistemas de tiempo real.

Los apartados 1.2.1 a 1.2.4 describen los cuatro aspectos anteriores, con las opciones asociadas.

1.2.1 Recursos

Los recursos incluyen el *hardware* del sistema, como son procesadores, redes de comunicación, elementos de almacenamiento masivo, etc. La memoria es un recurso que no suele considerarse en los sistemas de tiempo real. La asignación del recurso memoria se realiza de manera estática en los sistemas de tiempo real, por lo que puede ignorarse este recurso una vez el sistema comienza a funcionar. La asignación dinámica de memoria suele basarse en mecanismos de paginación, los cuales ocasionan una sobrecarga que dificulta enormemente el cumplimiento de las restricciones temporales. Los sistemas operativos de tiempo real suelen disponer de primitivas para evitar los efectos de la paginación. Por ejemplo, el estándar POSIX 1003.1b define la primitiva *mlockall* (Gallmeister, 1995).

El acceso de tiempo real a elementos de almacenamiento masivo, como por ejemplo los discos, es relevante en el caso de sistemas de tiempo real transaccionales o multimedia (A. Molano y Rajkumar, 1997). En los demás sistemas de tiempo real, el almacenamiento de información en discos no suele ser crítico. Un ejemplo de este punto son los sistemas de control.

El recurso más ampliamente considerado en la teoría de la planificación es el procesador. No obstante, muchos de los resultados referidos a este recurso son válidos para los recursos de comunicaciones debido a la analogía que existe entre los mismos (A. Ermedahl y Sjodin, 1997).

Los recursos como el procesador y las redes de comunicación son unitarios, en el sentido de que en un instante el recurso puede asignarse a lo sumo a una tarea.

Los recursos se organizan de acuerdo a una determinada arquitectura. Las arquitecturas más habituales son:

- Monoprocesador. El sistema dispone de un único procesador y no dispone de redes de comunicación.
- Multiprocesador de memoria compartida. El sistema dispone de dos o más procesadores que comparten una memoria principal. Los procesadores se comunican entre sí a través de la memoria compartida.
- Multiprocesador de memoria distribuida. El sistema dispone de dos mas procesadores. Cada procesador dispone de su memoria local, y no existe memoria compartida. Los procesadores se comunican entre sí a través de canales de comunicación específicos.
- Sistemas distribuidos. Conceptualmente son análogos a los multiprocesadores de memoria distribuida. La diferencia reside en la utilización de computadores autónomos en lugar de procesadores. La comunicación entre los computadores se realiza a través de redes de comunicación, habitualmente más lentas que los canales empleados en los multiprocesadores de memoria distribuida.

1.2.2 Consumidores de recursos

Los consumidores de recursos se denominan tareas. Las tareas no son más que abstracciones de los conceptos de proceso o hilo de los sistemas operativos multitarea. Las tareas vienen definidas por una secuencia de petición de recursos. A medida que se conceden los recursos pedidos por las tareas estas avanzan en su secuencia. Mientras esperan a que les sea concedido el recurso, las tareas se detienen en su secuencia.

Para que una tarea reciba un recurso tienen que darse simultáneamente tres circunstancias:

- Que la tarea haya pedido dicho recurso.
- Que la tarea no este pendiente del cumplimiento de ninguna condición de sincronización.
- Que la tarea haya sido seleccionada por el algoritmo de planificación para recibir el recurso, de entre todas las que cumplen las dos condiciones anteriores.

Lógicamente, si una tarea no solicita un recurso, no lo recibe.

Hay tareas que requieren el cumplimiento de ciertas condiciones de sincronización para avanzar en su secuencia de petición de recursos. Por ejemplo, bajo la presencia de relaciones de precedencia una tarea no puede recibir el recurso procesador hasta que las tareas que la preceden hayan terminado. El acceso a recursos compartidos protegidos es otro ejemplo típico, en el cual una tarea no puede entrar en un segmento de su ejecución mientras otras tareas no abandonen determinados segmentos de su ejecución. Aquellas tareas que no tienen asociada ninguna condición de sincronización se las conoce como independientes. Por el contrario, aquellas para las cuales existen condiciones de sincronización se las conoce como tareas dependientes. Obviamente, las dependencias entre tareas introducen complejidad en el sistema.

Aunque una tarea haya pedido un recurso y se cumplan todas sus condiciones de sincronización, puede suceder que la tarea no reciba el recurso. En el caso de existir dos o más tareas en esta situación el algoritmo de planificación sólo asignará el recurso a una de ellas, por lo que las demás no pueden recibirlo.

Habitualmente, las tareas vienen definidas por secuencias de petición de un único recurso, el procesador. En función del tipo de secuencia se distinguen los siguientes tipos de tareas:

- Tareas periódicas. Se piden C unidades del recurso procesador, denominado tiempo de computación, cada T unidades de tiempo. El parámetro T recibe el nombre de periodo de la tarea. Los instantes en los cuales se produce la petición del procesador se denominan instantes de activación. Estas tareas suelen activarse por medio del reloj del sistema (Buttazzo, 1997).
- Tareas aperiódicas. Se puede producir una petición del procesador en cualquier instante, desconocido a priori. Suelen originarse por eventos impredecibles, externos al sistema (Buttazzo, 1997).

- Tareas esporádicas. A pesar de no ser periódicas, se puede establecer un intervalo mínimo entre peticiones consecutivas del procesador. Pueden provenir de diversas fuentes, desde el muestreo periódico de eventos externos al sistema, a eventos externos que poseen intrínsecamente este carácter esporádico (Buttazzo, 1997).
- Tareas esporádicamente periódicas. Consisten en ráfagas de peticiones periódicas del procesador, de periodo fijo. La distancia mínima entre el comienzo de dos ráfagas está acotado. Este tipo de tareas son típicas de sistemas multimedia o transaccionales (K.W. Tindell y Wellings, 1994).
- Tareas con precedencias. Este tipo de tareas pueden comenzar a ejecutarse en el momento en el que todas las tareas que las preceden han terminado su ejecución. Este tipo de tareas suelen aparecer en multiprocesadores o sistemas distribuidos, en los cuales varias tareas cooperan para proporcionar la respuesta a los eventos del sistema. En esta situación, unas tareas emplean resultados intermedios proporcionados por las tareas que las preceden, de ahí que no puedan comenzar su ejecución hasta que las tareas que las preceden generen dichos resultados intermedios (Buttazzo, 1997).

Los tipos de tareas anteriores pueden generalizarse si se permite que las tareas accedan a recursos compartidos protegidos (L. Sha y Lehoczky, 1990; Baker, 1991). Esta generalización complica el algoritmo de planificación y el análisis de planificabilidad.

Otra generalización considera la existencia de *jitter* de activación. En general, los instantes de activación reales de las tareas no coinciden con los instantes de activación ideales considerados hasta ahora. Hay ocasiones en las cuales las activaciones reales sufren un cierto retraso debido a la granularidad temporal del reloj que activa el planificador, o debido a la espera de mensajes de sincronización. El retraso que existe entre la activación ideal y la real se la denomina *jitter* de activación. El *jitter* de activación de una tarea suele modelarse como un *jitter* máximo, denotado por J , en los sistemas de tiempo real crítico (K.W. Tindell y Wellings, 1994).

Otra de las generalizaciones del modelo de tareas considera las situaciones en las que el conjunto de tareas a planificar no es fijo, sino que varía a lo largo del tiempo para adaptarse a un entorno cambiante. Hay situaciones en las que el desconocimiento del entorno hace que el conjunto de tareas evolucione de una manera desconocida a priori. Este tipo de situaciones son objeto de estudio de la planificación dinámica (véase el apartado 1.2.3). Sin embargo, hay situaciones en las que el entorno puede encontrarse en un conjunto de estados claramente definidos y conocidos a priori, cada uno de los cuales requiere un conjunto de tareas también definido y conocido a priori. En este apartado se supone que el sistema se encuentra en esta última situación, asociada a lo que se denomina cambios de modo en la literatura (Pedro y Burns, 1998).

Las restricciones de tiempo real distinguen los sistemas de tiempo real de los sistemas computacionales tradicionales. Estas restricciones se suelen asociar a los instantes de terminación de las tareas.

En el caso de las tareas periódicas las restricciones temporales se suelen expresar como plazos relativos a los instantes de activación de las mismas. Estos plazos especifican el máximo intervalo de tiempo que puede transcurrir desde la activación de las tareas hasta que se completa el procesamiento pedido por dicha activación. El caso más ampliamente tratado es aquel en el que los plazos relativos, denotados por D , son iguales a los periodos de las tareas ($D = T$). Este caso es común entre las tareas que implementan sistemas de control, pues es necesario generar cada señal de control antes de comenzar la adquisición de la siguiente muestra. No obstante, hay generalizaciones (K.W. Tindell y Wellings, 1994; Manabe y Aoyagi, 1998) que consideran plazos menores que los periodos ($D < T$), y plazos mayores que los periodos ($D > T$). Las tareas periódicas normalmente suelen analizarse como tareas críticas.

Las tareas aperiódicas se modelan como tareas acríicas. No obstante, en los sistemas de tiempo real se intenta reducir el tiempo medio de respuesta de las mismas, sin comprometer el cumplimiento de las restricciones temporales de las demás tareas.

Las restricciones temporales de las tareas esporádicas suelen ser las mismas que las de las tareas periódicas, es decir, plazos relativos a los instantes de activación. Sin embargo, lo habitual no es que los plazos coincidan con el intervalo mínimo entre activaciones consecutivas, denotado como T , sino que lo habitual es que sea menor que dicho intervalo ($D < T$). El análisis de planificabilidad con este tipo de tareas coincide en muchos casos con el análisis de planificabilidad de las tareas periódicas (K.W. Tindell y Wellings, 1994; J.A. Stankovic y Buttazzo, 1998).

En cuanto a las tareas esporádicamente periódicas, las restricciones de tiempo real se suelen expresar como plazos relativos a cada una de las activaciones (K.W. Tindell y Wellings, 1994).

Las tareas con precedencias son “disparadas” por las tareas que las preceden. Esto hace que las restricciones de tiempo real se apliquen a una cadena de tareas con relaciones de precedencia que cooperan para generar un resultado. A cada una de estas cadenas de tareas se las conoce como tareas *end-to-end* en la literatura. En este caso las restricciones temporales suelen establecer el intervalo de tiempo máximo entre la activación de la primera tarea de la cadena de precedencias, y la terminación de la última tarea de la cadena (Natele y Stankovic, 1994).

1.2.3 Algoritmos de planificación

Los algoritmos de planificación realizan la administración de recursos, es decir, en todo instante deciden a que tarea se asigna cada uno de los recursos del sistema. Por supuesto puede haber recursos ociosos, debido a que no exista ninguna tarea que cumpliendo las condiciones de sincronización haya solicitado dichos recursos.

En las aplicaciones de tiempo real más simples la planificación se puede realizar mediante programación a muy bajo nivel, empleando por ejemplo interrupciones priorizadas. Sin embargo, es cada vez más habitual el empleo de micronucleos. En el desarrollo de sistemas de tiempo real de complejidad media o alta la opción más apropiada suele ser el empleo de sistemas operativos de tiempo real (Gallmeister, 1995), los cuales simplifican la programación y mejoran por lo tanto la fiabilidad del

sistema. En cualquier caso, explícita o implícitamente siempre se emplea un cierto algoritmo de planificación.

El objetivo fundamental de los algoritmos de planificación es lograr la correcta administración de los recursos del sistema, entendiéndose por correcta administración el cumplimiento de las restricciones temporales. El resultado del algoritmo de planificación es una planificación, o lo que es lo mismo, una asignación temporal de recursos a tareas.

El campo de los algoritmos de planificación para sistemas de tiempo real ha sido históricamente uno de los más tratados. Esto puede explicarse teniendo en cuenta el fuerte impacto que tiene el algoritmo de planificación empleado sobre el cumplimiento de las restricciones temporales.

Los algoritmos de planificación se pueden clasificar de acuerdo a diferentes criterios.

El primer criterio clasifica los algoritmos de planificación en

- estáticos y
- dinámicos

Los algoritmos de planificación estáticos obtienen una planificación basada en el conocimiento a priori del número y características de las tareas. Por el contrario, los algoritmos de planificación dinámicos generalmente no se basan en el conocimiento previo de las características de las tareas. Cada vez que llega una nueva tarea al sistema se genera una nueva planificación.

Los algoritmos de planificación estáticos tienen la ventaja de una menor sobrecarga en tiempo de ejecución con respecto a los algoritmos de planificación dinámicos. Sin embargo, no son aplicables a sistemas de tiempo real ubicados en entornos dinámicos o impredecibles, en los cuales no se sabe a priori que tareas van a llegar al sistema ni sus características. En este tipo de situaciones se emplean los algoritmos de planificación dinámicos.

Los algoritmos de planificación estáticos se pueden dividir a su vez en

- algoritmos de planificación en línea (*online*), y
- algoritmos de planificación fuera de línea (*offline*)

Un algoritmo de planificación se dice que es fuera línea cuando la planificación se calcula antes de empezar a funcionar el sistema de tiempo real. La planificación suele guardarse en una tabla que se emplea en tiempo de ejecución. Puesto que la tabla no puede tener una longitud ilimitada, es necesario que las tareas tengan algún tipo de periodicidad. De esta forma, la tabla contiene únicamente las asignaciones de recursos a tareas para el mínimo común múltiplo de los periodos.

Este tipo de planificación tiene varias ventajas. En primer lugar, la sobrecarga asociada a la planificación es muy baja. El planificador lo único que tiene que hacer es consultar la tabla para saber a que tarea debe le corresponde cada recurso. En segundo lugar, pueden emplearse técnicas computacionalmente muy costosas para la obtención de planificaciones, como puede ser la búsqueda de planificaciones óptimas

o cuasióptimas empleando técnicas de *branch and bound* (Xu y Parnas, 1990; D. Peng y Abdelzaher, 1997)

Sin embargo, la planificación estática fuera de línea tiene muchas desventajas provenientes de su rigidez. Por ejemplo, se emplea el tiempo máximo de computación de las tareas críticas para obtener la tabla con la planificación. Esto evita que tareas aperiódicas presentes en el sistema puedan aprovechar los huecos proporcionados por la selección conservadora de los tiempos de computación. Además, cada vez que se modifica el sistema para introducir una nueva tarea debe regenerarse la tabla.

En cuanto a los algoritmos de planificación dinámicos, estos se suelen dividir en dos tipos:

- algoritmos de planificación “que hacen lo que pueden” (*Best effort*), y
- algoritmos de planificación con respuesta garantizada (*Guaranteed*)

Los algoritmos de planificación “que hacen lo que pueden” no realizan ningún tipo de predicción acerca de posibles sobrecargas. Cada vez que llega una nueva tarea al sistema esta se acepta, pasando a la cola de tareas listas para ejecutar. El algoritmo de planificación intenta “hacer lo que puede” para cumplir las restricciones temporales de esta nueva tarea y de las anteriores. Este tipo de algoritmos son los algoritmos de planificación dinámica más simples de implementar y los que incurren en una menor sobrecarga. Sin embargo, también son los peores desde el punto de vista de la planificabilidad del sistema.

Los algoritmos de planificación con respuesta garantizada realizan un test de aceptación con cada nueva tarea que llega al sistema. El test consiste en decidir si es posible garantizar el cumplimiento de las restricciones temporales de la nueva tarea y las anteriormente aceptadas. Si tal garantía es posible la nueva tarea se acepta y se genera una nueva planificación. En caso contrario se rechaza. Respecto a los algoritmos de planificación “que hacen lo que pueden” tienen la desventaja de una mayor complejidad, con la sobrecarga en tiempo de ejecución que esto supone. Sin embargo, la planificabilidad del sistema mejora.

Para más información sobre planificación dinámica pueden consultarse las referencias Baruah y Wang (1992) y G. Buttazzo y Sensini (1995).

Otra clasificación de los algoritmos de planificación los divide en

- algoritmos de planificación con expulsión (*preemptive*), y
- algoritmos de planificación sin expulsión (*non-preemptive*)

La expulsión consiste en retirar el recurso a una tarea para dárselo a otra, antes de que se haya satisfecho completamente su petición sobre dicho recurso. En algunos casos, las tareas pueden definir secciones de las mismas en las cuales se prohíbe la expulsión de un recurso. De esta manera, incluso bajo la presencia de algoritmos de planificación con expulsión, ninguna otra tarea puede recibir el recurso hasta que la primera no lo libere. La definición de secciones en las que se prohíbe la expulsión influye negativamente en la planificabilidad del sistema, y dificulta enormemente su análisis. Sin embargo evitan la necesidad de utilizar protocolos de acceso a recursos

compartidos. En cualquier caso, este tipo de secciones en las que se prohíbe la expulsión deberían ser lo más pequeñas que sea posible.

Uno de los conceptos más ampliamente incorporados en los algoritmos de planificabilidad es el concepto de prioridad, el cual establece el orden de acceso de las tareas que compiten por un mismo recurso. Muchos algoritmos de planificación asignan una prioridad a cada activación de una tarea. El ejemplo más representativo es el del algoritmo de planificación EDF (Liu y Layland, 1973). Otros simplemente utilizan una prioridad predefinida, asignada de acuerdo a criterios como *Rate Monotonic* (RM) (Liu y Layland, 1973), o *Deadline Monotonic* (DM) (Leung y Whitehead, 1982).

El caso de la planificación multiprocesador, o en general en sistemas distribuidos, es un problema mucho más complejo que el de la planificación en multiprocesadores. En particular, muchos de los resultados disponibles para el caso monoprocesador no son válidos para el caso multiprocesador. Básicamente hay dos técnicas para atacar el problema multiprocesador (Oh y Son, 1995):

- planificación local, y
- planificación global

Empleando planificación local, una vez una tarea se asigna a un procesador, esta se ejecuta única y exclusivamente en ese procesador. Por el contrario, en las estrategias globales cualquier instancia de una tarea puede ejecutarse en un procesador diferente, o incluso ser expulsada y movida a un procesador diferente, antes de su terminación.

El empleo de estrategias de planificación globales mejora la planificabilidad. Sin embargo este tipo de estrategias tienen una serie de inconvenientes que se muestran en el apartado 1.5.

1.2.4 Análisis de planificabilidad

Una vez definidos los recursos a administrar con su organización, las tareas con sus restricciones temporales, y los algoritmos de planificación, el problema es como comprobar si se cumplen las restricciones de tiempo real. Esto es lo que se conoce como test de planificabilidad. Pero el análisis de la planificabilidad es más que comprobar el cumplimiento de las restricciones temporales. Debe establecerse que parámetros influyen sobre dicha planificabilidad, y en que magnitud, para así tomar las medidas correctoras necesarias en el caso de que el sistema no sea planificable.

En la literatura se encuentran fundamentalmente las siguientes técnicas para el test o análisis de planificabilidad:

- técnicas basadas en utilizaciones, (Liu y Layland, 1973; Lehoczky, 1990)
- técnicas basadas en el tiempo de respuesta, (K.W. Tindell y Wellings, 1994; J.A. Stankovic y Buttazzo, 1998; Lehoczky, 1999)
- el factor de carga o demanda del procesador, (Baruah y Rosier, 1990; I. Ripoll y Mok, 1996; J. Lehoczky y Ding, 1989; Manabe y Aoyagi, 1998), y

- simulación

Las técnicas basadas en utilizaciones obtienen el valor máximo de utilización total del sistema que asegura el cumplimiento de las restricciones temporales. A este máximo se le conoce como límite de utilización. El valor del límite de utilización se obtiene considerando el caso peor. Este tipo de técnicas se emplearon en un principio para estudiar la planificabilidad considerando únicamente el recurso procesador. No obstante, pueden emplearse y de hecho se emplean en el caso de otros recursos como puede ser una red de comunicaciones.

En el caso del recurso procesador, el factor de utilización de las tareas periódicas se define como el cociente del tiempo de computación y el periodo de las mismas ($u = C/T$). Considerando como periodo el tiempo mínimo entre dos activaciones consecutivas de una tarea esporádica, se define de forma análoga el factor de utilización de las tareas esporádicas.

La utilización del sistema se obtiene como la suma de los factores de utilización de las tareas. Intuitivamente es claro que todo sistema de utilización mayor que el número de procesadores no es planificable, pues se pide al sistema más tiempo de computación que el que pueden proporcionar todos los procesadores juntos. Esta es una de las propiedades interesantes de los análisis de planificabilidad basados en utilizaciones. Se sabe de antemano lo cerca que se está de la situación ideal en la cual la utilización total iguala al número de procesadores.

Otra de las ventajas de los análisis de planificabilidad basados en utilizaciones es la simplicidad del test de planificabilidad. Basta con comparar la utilización del sistema con el límite de utilización para determinar si es planificable.

El límite de utilización es una función analítica de parámetros del sistema. De esta manera resulta sencillo el análisis de influencia de los diferentes parámetros sobre la planificabilidad del sistema, al menos para el caso más desfavorable.

El mayor inconveniente de los análisis de planificabilidad basados en utilizaciones proviene de que no son exactos. Incluso aunque la utilización del sistema estuviese por encima del límite de utilización, el sistema podría ser planificable (J. Lehoczky y Ding, 1989). Además, la obtención de los límites de utilización suele ser relativamente compleja, de ahí que sea factible su obtención sólo en los casos más simples.

Una de las técnicas de análisis de planificabilidad más importante es la basada en los tiempos de respuesta. El objetivo de este tipo de técnicas es la obtención del tiempo de respuesta máximo asociado a cada tarea, entendiendo como tal el máximo tiempo que transcurre desde una activación de la tarea hasta el procesamiento completo de dicha activación. De esta forma, comparando el tiempo máximo de respuesta con el plazo de la tarea se puede determinar si se cumple el plazo de la misma.

Las técnicas de planificabilidad basadas en tiempos de respuesta tienen en la mayor parte de los casos un coste computacional razonable, son de amplia aplicabilidad y además proporcionan tests de planificabilidad exactos. Sin embargo, son poco útiles para realizar un análisis de planificabilidad, entendiendo por análisis la determinación de la influencia cuantitativa de diferentes parámetros del sistema sobre la planificabilidad. Este tipo de técnicas son como “máquinas” que reciben como entrada un sistema, y determinan si el sistema es planificable o no.

Por ejemplo, cuando los plazos de las tareas periódicas son pequeños con respecto a los periodos, la planificabilidad empleando prioridades fijas o prioridades variables (EDF) es similar. Este hecho se puede mostrar fácilmente observando la coincidencia de los límites de utilización en ambos casos, pero no es fácil hacerlo a partir de las expresiones del tiempo de respuesta.

Las técnicas basadas en el factor de carga (*load factor*), también denominado demanda del procesador (*processor demand*) se basan en comprobar que la demanda total del recurso procesador por parte de las tareas en determinados intervalos de tiempo no excede lo que el recurso puede dar de sí en dichos intervalos. Este tipo de técnicas definen tests de planificabilidad exactos con un coste computacional en algunos casos menor que el asociado a la obtención del tiempo de respuesta de tareas. No obstante, lo mismo que sucede con las técnicas basadas en tiempo de respuesta son de poca utilidad para realizar un análisis de planificabilidad en el sentido amplio de la palabra.

Finalmente la última técnica es la de simulación. La idea es “ejecutar” un modelo del sistema y a partir del resultado de la “ejecución” obtener toda la información requerida, como puede ser la comprobación de las restricciones temporales. La gran ventaja de esta técnica es que se puede aplicar prácticamente en cualquier situación. No obstante, no es ninguna “panacea”.

Cuando se trabaja con sistemas de tiempo real críticos se trata de comprobar que el sistema cumple las restricciones de tiempo real, incluso en las condiciones más desfavorables. El problema es que en ocasiones se desconoce cuales son esas condiciones más desfavorables, por lo que no es posible simularlas. Además, en aquellas ocasiones en que se conocen las condiciones más desfavorables, el coste computacional asociado a la simulación es mayor que el asociado a otras técnicas exactas, como puede ser la obtención de los tiempos de respuesta. Otro de los grandes problemas de la simulación es ¿hasta donde hay que simular? Por ejemplo, en el caso de tareas periódicas se tienen infinitas activaciones de las mismas, y por supuesto no se puede simular hasta el infinito. En estos casos lo habitual es obtener analíticamente un intervalo de simulación cumpliendo que si el sistema es planificable en dicho intervalo, entonces lo es en cualquier intervalo. En muchos casos la obtención de este intervalo es muy compleja.

Las técnicas basadas en simulación juegan un papel destacado en el caso de sistemas de tiempo real ubicados en entornos dinámicos, en los cuales no se conocen a priori las tareas que van a llegar al sistema.

Un gran numero de trabajos sobre planificación en sistemas de tiempo real recurren a la simulación para obtener el comportamiento del sistema bajo estudio frente bajo un entorno simulado que trata de parecerse a lo que puede aparecer en la realidad.

Las técnicas de análisis de planificabilidad anteriores pueden extenderse al caso multiprocesador con ciertos matices. Por ejemplo, las técnicas basadas en tiempo de respuesta se han extendido a sistemas distribuidos (Tindell y Clark, 1994; Lehoczky, 1999), y se han empleado en el caso de planificación local (S. Sáez y Crespo, 1998; S. Lauzac y Mossé, 1998). Sin embargo, pocos resultados se han encontrado basados en utilizaciones.

Recursos	Arquitectura	Tareas	Restricc. temporales	Planificación	Análisis
Procesador	Multiproc.	Periódicas Aperiódicas Esporádicas Rec. compart. Camb. modo <i>Jitter</i>	$D \leq T$	RM, EDF, planif. local	Utilizaciones

Tabla 1.1: Opciones del problema de planificación consideradas en la tesis.

Como resumen de las técnicas de análisis de planificabilidad conviene resaltar que no hay ninguna técnica perfecta, dependiendo del problema particular unas pueden ser mejores que otras. En realidad, las técnicas anteriores se complementan, aportando cada una sus propias virtudes.

1.3 Problemas de planificación en el contexto de la tesis

Si bien el problema de planificación de sistemas de tiempo real presentado en el apartado 1.2 proporciona una visión general, resulta conveniente ir enfocando hacia el problema de planificación tratado en la tesis.

La tabla 1.1 especifica las diferentes opciones del problema de planificación tratado en la tesis. A parte de las opciones del problema de planificación mostradas en la tabla 1.1, es interesante mostrar otras opciones del problema de planificación relevantes en el contexto de la tesis. El conjunto de las opciones tratadas en la tesis y las opciones adicionales se representan en la figura 1.2.

Entre las opciones adicionales se encuentra la arquitectura monoprocesador. Su inclusión viene justificada por el hecho de que los resultados multiprocesador proporcionados por la tesis se basan en resultados monoprocesador análogos.

La otra opción adicional es el análisis de planificabilidad basado en tiempos de respuesta. En el apartado anterior se indicaron algunas de las deficiencias de los análisis de planificabilidad basados en tiempos de respuesta, y como los análisis de planificabilidad basados en utilidades paliaban estas deficiencias. Por lo tanto, resulta interesante describir con mayor detalle las técnicas de planificabilidad basadas en utilidades.

La planificación global y la local son las dos alternativas fundamentales en la planificación de multiprocesadores, de ahí que sea interesante su comparación. Esto justifica su consideración.

A continuación en los apartados 1.4 y 1.5 se exponen los resultados más relevantes encontrados en la literatura para las opciones de planificación mostradas en la figura 1.2. Por conveniencia los resultados se han estructurado en resultados monoprocesador, proporcionados en el apartado 1.4, y resultados multiprocesador, proporcionados en el apartado 1.5.

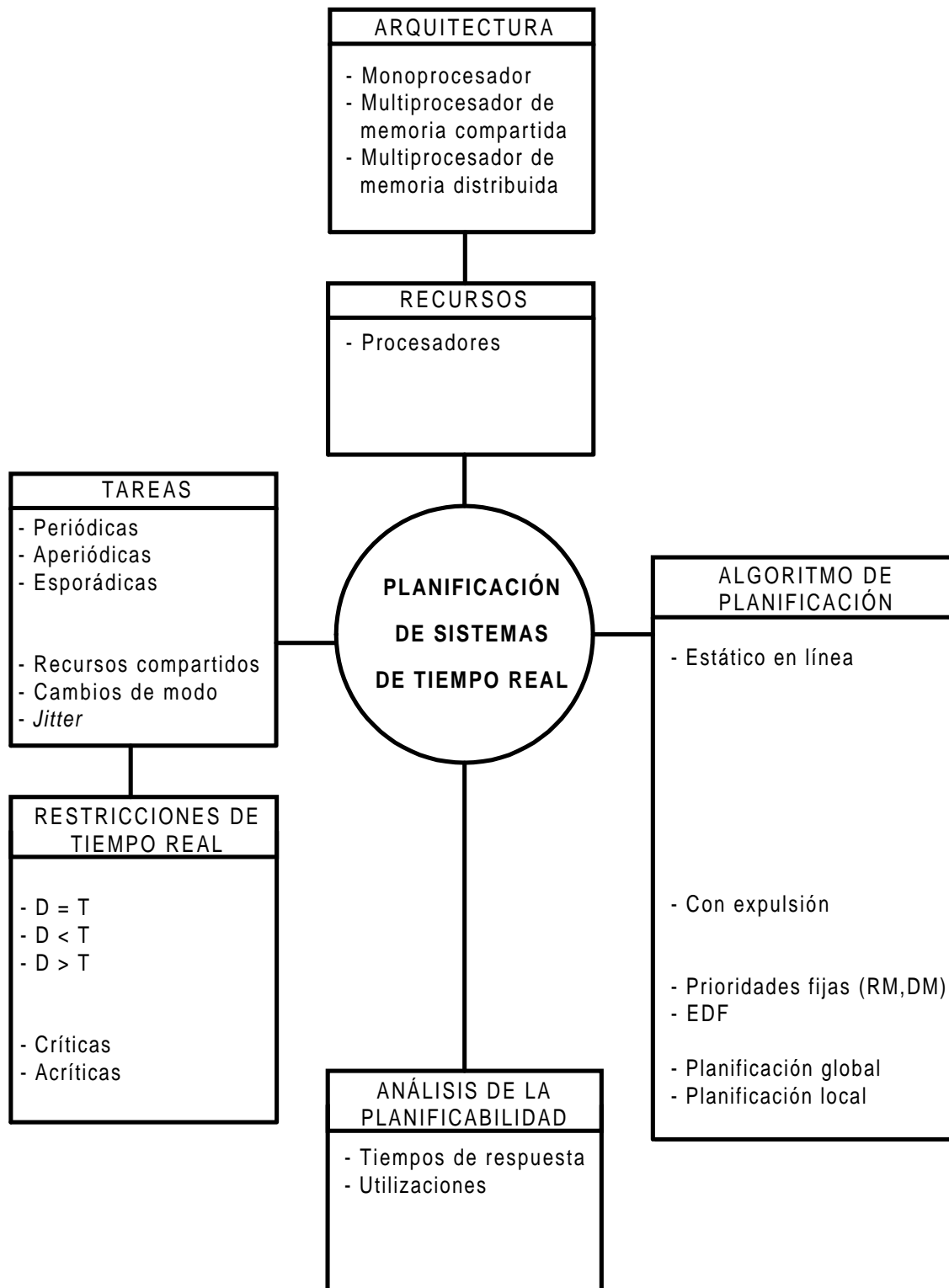


Figura 1.2: Problemas de planificación en el contexto de la tesis.

1.4 Planificación en sistemas monoprocesador

La planificación monoprocesador consiste en la administración de un único procesador para dar servicio a las tareas del sistema. El procesador representa el recurso a administrar, mientras que las tareas son las consumidoras del recurso.

La teoría de tiempo real monoprocesador se centra fundamentalmente en dos tipos de planificación:

- planificación basada en prioridades fijas, y
- planificación basada en prioridades variables.

En la planificación basada en prioridades fijas a cada tarea se le asigna una prioridad con la cual se ejecuta durante toda su existencia. Todas las activaciones² de la tarea se ejecutan con esa prioridad. Por el contrario, la planificación basada en prioridades variables asigna prioridades variables a las tareas, de tal forma que cada activación de una misma tarea puede recibir una prioridad diferente (Liu y Layland, 1973).

Para entender como influyen las prioridades en la planificación, debe tenerse en cuenta que en todo instante el procesador se concede a la activación de mayor prioridad. Si otra activación de menor prioridad estaba ejecutándose en ese instante es expulsada del procesador para pasar a ejecutar la de mayor prioridad. Una vez se ha completado la activación de mayor prioridad esta desaparece del sistema y pasa a ejecutarse la nueva activación de mayor prioridad. Si no hubiese ninguna activación pendiente de completarse, entonces el procesador permanece inactivo.

La planificación empleando prioridades fijas presenta la ventaja de una mayor simplicidad de implementación con respecto a la planificación empleando prioridades variables. Su mayor inconveniente proviene precisamente de su simplicidad, pues reduce muchos de los grados de libertad de la planificación. Debe tenerse en cuenta que empleando prioridades fijas hay tantos grados de libertad como prioridades a fijar, esto es, tantos como tareas. Por otro lado, en el caso de prioridades variables los grados de libertad son tantos como activaciones, y debe tenerse en cuenta que cada tarea está formada en general por un número infinito de activaciones.

Considérese un conjunto de m tareas periódicas $\Gamma = \{\tau_1, \dots, \tau_m\}$, de periodos $\{T_1, \dots, T_m\}$, tiempos de computación $\{C_1, \dots, C_m\}$, y plazos $\{D_1, \dots, D_m\}$.

Liu y Layland (1973) demostraron que si las tareas periódicas son independientes y tienen plazos iguales a los periodos la asignación de prioridades fijas denominada *Rate-Monotonic* (RM) es óptima. Es decir, si el conjunto de tareas no es planificable asignándole las prioridades RM, tampoco lo es asignándole cualquiera de las otras $(m! - 1)$ asignaciones de prioridades posibles. La asignación de prioridades RM consiste en ordenar las tareas en sentido de periodos no decrecientes, y asignar las prioridades en sentido estrictamente decreciente. Esta asignación óptima de prioridades fue posteriormente generalizada por Leung y Whitehead (1982) para el

²Las tareas siguen habitualmente un patrón de ejecución mediante el cual se despiertan al recibir un evento y se duermen al finalizar su procesamiento, esperando por un nuevo evento. Cada vez que una tarea recibe un evento se genera lo que se conoce como instancia, trabajo o activación de la tarea. En el caso de las tareas periódicas el evento es la expiración de un temporizador.

caso de plazos menores o iguales que los periodos. En este caso la asignación se denomina *Deadline Monotonic* (DM). La asignación de prioridades DM consiste en ordenar las tareas en sentido de plazos no decrecientes, y asignar las prioridades en sentido estrictamente decreciente. Se puede comprobar fácilmente que para el caso de plazos iguales a periodos la asignación de prioridades DM coincide con la asignación RM. La planificación basada en prioridades fijas asignadas de acuerdo con el criterio RM o DM se la conoce como planificación RM o DM respectivamente.

Liu y Layland (1973) demostraron que si las tareas periódicas son independientes y tienen plazos iguales a los periodos la asignación de prioridades variables *Earliest Deadline First* (EDF) es óptima. Más aún, Dertouzos (1974) probó que la planificación EDF (planificación basada en las prioridades variables asignadas de acuerdo con el criterio EDF) es óptima para cualquier conjunto de tareas, sean periódicas o no. Esto es, si no es posible cumplir los plazos del conjunto de tareas empleando planificación EDF, no existe ningún algoritmo de planificación capaz de hacer que se cumplan. La asignación de prioridades EDF consiste en asignar en todo instante la mayor prioridad a la activación de plazo absoluto menor, es decir, a la que tiene más cerca su plazo en el instante dado.

A continuación, en los apartados 1.4.1 y 1.4.2 se muestran una serie de resultados relacionados con la planificabilidad de conjuntos de tareas periódicas como los descritos, empleando prioridades fijas RM o prioridades variables EDF en monoprocesadores. Estos resultados se centran en los análisis de planificabilidad basados en utilizaciones y basados en tiempos de respuesta.

1.4.1 Planificación RM

En este apartado se trata el análisis de planificabilidad en monoprocesadores empleando técnicas basadas en utilizaciones y tiempos de respuesta, cuando se emplea planificación RM.

Ambas técnicas suponen que las tareas son periódicas e independientes. Las tareas esporádicas pueden analizarse como tareas periódicas cuando se considera el caso peor. Las tareas aperiódicas suelen planificarse empleando servidores periódicos, como son el *Sporadic Server* (SS) (B. Sprunt y Lehoczky, 1989), o el *Total Bandwidth Server* (TBS) (Spuri y Buttazzo, 1996).

Técnicas de análisis de planificabilidad basadas en utilizaciones

Quizá el resultado más conocido basado en utilizaciones para planificación en monoprocesadores es el obtenido por Liu y Layland (1973). Establecieron que cualquier conjunto de m tareas de utilización total menor o igual que $m(2^{1/m} - 1)$ es planificable en un monoprocesador empleando planificación RM. La condición de planificabilidad derivada de este límite es sólo suficiente pues existen conjuntos de tareas de utilización mayor que el límite que son planificables. Por ejemplo, el conjunto formado por dos tareas de tiempos de computación $\{C_1 = 10, C_2 = 20\}$ y periodos iguales a los plazos de valor $\{T_1 = 22, T_2 = 44\}$ es planificable, a pesar de que su utilización total de valor 0'91 sea superior al límite, de valor $2(2^{1/2} - 1) = 0'83$. Una característica muy interesante de este límite de utilización es que es ajustado, es

decir, no es posible encontrar un límite de utilización mayor. Cuando el número de tareas tiende a infinito este límite tiende al valor $\ln 2$.

Con la intención de eliminar alguna de las restricciones del conjunto de tareas, Lehoczky (1990) generalizó los plazos de las tareas, permitiendo plazos de valor $D_i = \Delta T_i$. Para $\Delta < 1$ se tienen plazos menores que los periodos, mientras que para $\Delta = 1$ se tienen plazos iguales a los periodos. La expresión del límite de utilización obtenida para un número de tareas arbitrario es demasiado complejo para ser generalizado al caso multiprocesador. Además la aplicabilidad de estos límites de utilización es muy reducida. Por ello, sólo se considera el caso simple de tener un número de tareas infinito, el cual se corresponde con el número de tareas más desfavorable. El límite de utilización viene dado por la ecuación (1.1).

$$\begin{cases} \Delta & \text{si } 0 < \Delta \leq 1/2 \\ \ln(2\Delta) + 1 - \Delta & \text{si } 1/2 \leq \Delta \leq 1 \end{cases} \quad (1.1)$$

Otra generalización del límite de Liu y Layland incluye la posibilidad de acceso a recursos compartidos. Esta generalización relaja la restricción de que las tareas sean independientes. El acceso a recursos compartidos como puede ser por ejemplo un área de memoria protegida por semáforos, puede provocar que una activación de alta prioridad se encuentre bloqueada mientras una activación de baja prioridad accede al recurso compartido. Esto contradice la hipótesis de que en todo momento se ejecuta la activación de mayor prioridad. Para evitar tiempos de bloqueo excesivos de las tareas de alta prioridad y otros serios inconvenientes se suelen emplear protocolos que controlan el acceso a recursos compartidos (Buttazzo, 1997).

Dos algoritmos muy representativos son el PCP (L. Sha y Lehoczky, 1990) y el SRP (Baker, 1991). El protocolo PCP se emplea con planificadores basados en prioridades fijas, mientras que el SRP se puede emplear tanto con planificadores basados en prioridades fijas como variables. Con relación al análisis de planificabilidad de la tesis, el efecto de estos algoritmos es un tiempo de bloqueo máximo, denotado por B , para cada una de las tareas del sistema.

El límite de utilización para planificación RM en un monoprocesador, considerando acceso a recursos compartidos controlado por el protocolo PCP o SRP, pasa a ser (L. Sha y Lehoczky, 1990)

$$m(2^{1/m} - 1) - \max_{1 \leq i \leq m} (B_i/T_i) \quad (1.2)$$

Técnicas de análisis de planificabilidad basadas en tiempos de respuesta

La obtención de los tiempos de respuesta es una de las técnicas más potentes para la determinación de la planificabilidad de sistemas de tiempo real.

En el caso más sencillo, en el cual se tienen tareas periódicas independientes, el máximo tiempo de respuesta de una tarea τ_i , se obtiene como el mínimo valor tr_i que cumple la ecuación recurrente (N. Audsley y Burns, 1991)

$$tr_i = C_i + \sum_{k=1}^{i-1} \lceil tr_i/T_k \rceil$$

En dicha ecuación se supone que el índice de las tareas es inversamente proporcional a su prioridad. Simplemente comparando el valor de tr_i con su plazo, D_i , se puede saber si la tarea τ_i es planificable o no.

La ecuación se basa en una premisa de que el peor tiempo de respuesta se produce cuando todas las tareas se activan simultáneamente, lo cual es cierto únicamente cuando los plazos de las tareas son menores o iguales que los periodos y las tareas son planificables. Por lo tanto, la ecuación recurrente anterior es aplicable también al caso de asignación de tareas DM. En general, la fase de las tareas no se suele fijar, por lo que podría suceder que no se alcanzase ese instante de activación simultánea. Sin embargo, se suele considerar el caso peor, en el cual todas las tareas se activan simultáneamente. Además, la consideración de las fases de las tareas proporciona poca mejora en la planificabilidad (J.M. López y García, 2000a).

La ecuación recurrente anterior se puede extender para tener en cuenta los efectos del *jitter*, o el acceso a recursos compartidos protegidos. Sin embargo, siempre que se tiene en cuenta el acceso a recursos compartidos el tiempo de respuesta obtenido es pesimista. La nueva ecuación recurrente pasa a ser (K.W. Tindell y Wellings, 1994)

$$tr_i = C_i + B_i + \sum_{k=1}^{i-1} [(tr_i + J_k)/T_k]$$

donde B_i es el máximo tiempo de bloqueo que puede experimentar la tarea τ_i , y donde J_k es el *jitter* de la tarea τ_k .

Esta ecuación recurrente se puede generalizar para permitir plazos mayores que los periodos, y otro tipo de extensiones que no son de relevancia para la tesis.

Cuando se consideran cambios de modo, el conjunto de tareas puede evolucionar dinámicamente pasando de un modo a otro. Todos los modos posibles se conocen a priori, así como las tareas asociadas a los mismos, lo cual permite un análisis de planificabilidad a priori. El análisis de los cambios de modo suele realizarse a través del cálculo de los tiempos de respuesta. La ecuación recurrente del tiempo de respuesta se complica enormemente. Para más información se puede consultar la referencia Pedro y Burns (1998).

1.4.2 Planificación EDF

En este apartado se trata el análisis de planificabilidad en monoprocesadores empleando técnicas basadas en utilidades y tiempos de respuesta, cuando se emplea planificación EDF.

Técnicas de análisis de planificabilidad basadas en utilidades

Liu y Layland (1973) también propusieron un límite de utilización para la planificación EDF, a la cual denominaron *Deadline Driven Scheduling*.

Cualquier conjunto de tareas periódicas independientes de plazos iguales a los periodos es planificable empleando el algoritmo EDF si y sólo si la utilización total del conjunto es menor o igual que uno. Por lo tanto, el límite de utilización en

este caso es uno. Además este límite de utilización proporciona una condición de planificabilidad exacta.

Para relajar la restricción de que los plazos sean iguales a los periodos se pueden definir estos como $D_i = \Delta T_i$, de igual forma que en el caso de planificación RM.

Una condición suficiente de planificabilidad bajo EDF de conjuntos de tareas periódicas de plazos menores o iguales que los periodos establece que el conjunto de tareas es planificable si se cumple (J.A. Stankovic y Buttazzo, 1998)

$$\sum_{i=1}^m \frac{C_i}{\min\{D_i, T_i\}} \leq 1 \quad (1.3)$$

Teniendo en cuenta que $D_i = \Delta T_i$ se deduce

$$\sum_{i=1}^m u_i \leq \Delta \quad (1.4)$$

Se puede encontrar un conjunto de tareas periódicas de plazos $D_i = \Delta T_i$ y utilización total $(\Delta + \epsilon)$, con $\epsilon \rightarrow 0$, el cual no es planificable por EDF. Por ejemplo un conjunto n de tareas idénticas de utilización total $(\Delta + \epsilon)$ no es planificable con EDF. Por lo tanto la condición suficiente de planificabilidad proporcionada por la ecuación (1.4) es ajustada.

Para relajar la hipótesis de tareas independientes se permite el acceso a recursos compartidos empleando el protocolo SRP. De esta forma el límite de utilización pasa a ser (Baker, 1991)

$$1 - \max_{1 \leq i \leq m} (B_i/T_i) \quad (1.5)$$

En el caso de que los plazos de las tareas sean menores o iguales que los periodos de las mismas, existe un límite que sin ser un límite de utilización se puede emplear de manera análoga. Este límite se deduce directamente de la ecuación 1.3, teniendo en cuenta que para el caso de plazos menores o iguales que los periodos $\min\{D_i, T_i\} = D_i$ para toda $i = 1, \dots, m$. La condición de planificación asociada viene dada por la siguiente ecuación

$$\sum_{i=1}^m \frac{C_i}{D_i} \leq 1$$

Técnicas de análisis de planificabilidad basadas en tiempos de respuesta

Desafortunadamente, el cálculo analítico de los tiempos de respuesta para el caso de planificación EDF no es tan simple como en el caso de planificación con prioridades fijas.

La obtención de los tiempos de respuesta es costosa, lo que elimina precisamente una de las virtudes de las técnicas basadas en tiempos de respuesta para el caso de planificación empleando prioridades fijas. En este sentido, es en general preferible el empleo de tests de planificabilidad basados en la demanda del procesador, como el descrito por I. Ripoll y Mok (1996).

Para más información sobre la obtención de los tiempos de respuesta para el caso

de planificación EDF puede consultarse el capítulo 3 de J.A. Stankovic y Buttazzo (1998).

1.5 Planificación en sistemas multiprocesador

En este apartado se describen resultados de análisis de planificabilidad multiprocesador relevantes en el contexto de la tesis, mostrando sus deficiencias.

En el caso de planificación multiprocesador el problema se complica de forma extraordinaria. Ahora en lugar de decidir que tarea se ejecuta en el procesador en cada instante, debe decidirse que tareas se ejecutan en cada uno de los n procesadores del multiprocesador en cada instante.

Básicamente hay dos estrategias en el tratamiento de este problema: estrategias locales, también denominadas estrategias de particionado, y estrategias globales (Oh y Son, 1995).

En las estrategias locales, una vez una tarea se asigna a un procesador, esta se ejecuta única y exclusivamente en ese procesador. Por el contrario, en las estrategias globales cualquier instancia de una tarea puede ejecutarse en un procesador diferente, o incluso ser expulsada y movida a un procesador diferente, antes de su terminación.

Se sabe que el algoritmo de planificación EDF es óptimo en sistemas monoprocesador (Dertouzos, 1974). Desafortunadamente la planificación EDF global no es óptima en el caso de un multiprocesador (Dall y Liu, 1978; Dertouzos y Mok, 1989). La razón reside en la imposibilidad de ejecutar estrictamente de forma simultánea una tarea en dos o más procesadores. Además, en el caso más desfavorable el rendimiento que se puede extraer del multiprocesador empleando planificación EDF global es el mismo que el de un monoprocesador (Dall y Liu, 1978), lo cual muestra la incapacidad de este planteamiento.

Otra estrategia de planificación habitual en el caso monoprocesador es el empleo de prioridades fijas. Sin embargo, la planificación global multiprocesador empleando prioridades fijas presenta serios inconvenientes. En primer lugar, la asignación óptima de prioridades a tareas es un problema *NP-hard* en el sentido estricto (Leung y Whitehead, 1982). Por lo tanto no es posible encontrar un algoritmo general que realice la asignación óptima, ni siquiera en tiempo pseudopolinomial. En segundo lugar, empleando la asignación de prioridades RM, la cual es óptima en monoprocesadores, el rendimiento puede llegar a ser tan bajo como el de un monoprocesador (Dall y Liu, 1978).

Por lo tanto queda patente que la planificación multiprocesador empleando estrategias globales debe seguir diferentes caminos que los seguidos en el caso monoprocesador. Desde la perspectiva de la planificación global los resultados conocidos para el caso monoprocesador son de poca utilidad sino inservibles.

En la literatura se pueden encontrar diferentes propuestas relacionadas con planificadores multiprocesador globales (Dertouzos y Mok, 1989; Han y Lin, 1989; F. Wang y Stankovich, 1992). Sin embargo, esta no es la dirección que se va a seguir en la tesis, debido a las desventajas que presentan las estrategias globales en comparación con las estrategias locales, algunas de las cuales se muestran a continuación.

- La sobrecarga debida a los planificadores globales es mayor que la debida a los planificadores locales. En primer lugar el proceso de toma de decisiones es mucho más complejo. En segundo lugar los flujos de información entre diferentes procesadores es elevado. Esta comunicación requiere por ejemplo el acceso a memoria compartida o la comunicación a través de canales *hardware*.
- La predecibilidad asociada a los planificadores globales es mucho menor que la asociada a planificadores locales, pues los elementos del sistema están más interrelacionados y los algoritmos de planificación son habitualmente más complejos.
- No se pueden aplicar los resultados conocidos para el caso monoprocesador.

La única ventaja que proporcionan las estrategias globales frente a las locales proviene de su generalidad. De hecho, las estrategias locales son un caso particular de estrategias globales, en las cuales todas las activaciones de una misma tarea se ejecutan en el mismo procesador. Por lo tanto, las estrategias globales permiten planificar los mismos conjuntos de tareas planificables con estrategias locales e incluso otros. No obstante el precio a pagar es muy alto. Además, cuando se planifican conjuntos de tareas periódicas independientes como las consideradas en la tesis, existen estrategias locales que proporcionan resultados cercanos a los obtenidos con estrategias globales.

La tesis sigue la estrategia de planificación local. Todo algoritmo de planificación multiprocesador local queda totalmente definido por dos elementos:

- El algoritmo de planificación monoprocesador. Normalmente se supone que todos los procesadores del multiprocesador ejecutan el mismo algoritmo de planificación monoprocesador. No obstante, en general podría ser diferente.
- El algoritmo de asignación de tareas a procesadores.

En cuanto al algoritmo de planificación monoprocesador se consideran los dos tipos de planificadores habituales en la teoría de planificación multiprocesador:

- Planificación basada en prioridades fijas. Este tipo de planificación requiere la asignación previa de prioridades a tareas.
- Planificación *Earliest Deadline First* (EDF).

Una vez se haya elegido el algoritmo de planificación monoprocesador el único grado de libertad del sistema es el algoritmo de asignación de tareas a procesadores.

En la literatura el estudio de la planificabilidad de sistemas de tiempo real multiprocesador empleando estrategias locales se ha centrado fundamentalmente en la comparación de distintas estrategias de asignación de tareas a procesadores. Esto es lógico, pues una vez elegido el algoritmo de planificación monoprocesador, bien sea empleando prioridades fijas o EDF, el único grado de libertad del sistema es el algoritmo de asignación. Para realizar esta comparación los trabajos anteriores han seguido las líneas de simulación y teórica.

La línea de simulación consiste en la generación aleatoria de conjuntos de tareas bajo ciertas restricciones. A continuación se obtiene el número medio de procesadores necesarios para planificar de forma factible los conjuntos de tareas de una utilización dada, o bien la utilización total media de los conjuntos de tareas que utilizan totalmente un número de procesadores dado³. Siguiendo la línea de simulación, la comparación de distintas estrategias de asignación de tareas a procesadores es directa. No obstante, los resultados de simulación deben tomarse con cierta cautela. La razón es clara, los conjuntos de tareas generados no tienen por que ser representativos de los que aparecen en la práctica. Por ejemplo, es muy habitual generar los periodos de las tareas siguiendo una distribución uniforme, cuando en la práctica los periodos están agrupados alrededor de determinados valores, y cuando además tienen cierta tendencia a ser unos múltiplos de otros.

Los trabajos encontrados en la literatura en la línea de simulación obvian parámetros de gran influencia en la planificabilidad de los conjuntos de tareas, como es el caso de la desviación típica de la distribución empleada en la generación de los conjuntos de tareas. Este tipo de deficiencias son paliadas en la tesis mediante un meticuloso proceso de simulación.

En cuanto a la línea teórica, ésta se centra casi exclusivamente en la obtención de la métrica (N_{AA}/N_{OPT}), para parejas algoritmo de planificación monoprocesador-algoritmo de asignación (Dall y Liu, 1978; Garey y Johnson, 1979a; Davari y Dhall, 1986b; Davari y Dhall, 1986a; Oh y Son, 1995; A. Burchard y Son, 1995). La métrica anterior indica el cociente entre el número de procesadores necesario para planificar un conjunto de tareas empleando el algoritmo de asignación AA , y el número de procesadores requerido por el algoritmo de asignación óptimo. La tabla 1.2 muestra el valor de esta métrica para diferentes parejas algoritmo de planificación monoprocesador-algoritmo de asignación, incluyendo la referencia del trabajo que la obtuvo.

Oh y Son (1995), y A. Burchard y Son (1995) consideran asimismo el caso de tareas con factores de utilización pequeños. En este caso se obtienen sustanciales mejoras de la métrica (N_{AA}/N_{OPT}).

Muchos de los algoritmos de asignación que aparecen en la tabla se describen en el apartado 2.1.1.

La métrica (N_{AA}/N_{opt}) es útil para comparar diferentes algoritmos de asignación de tareas, pero no para estudiar la planificabilidad del sistema o para su dimensionado. Por dimensionado se entiende la obtención del número de procesadores necesario para planificar un conjunto de tareas dado. Hay varias razones para ello.

En primer lugar, la métrica se obtiene habitualmente para conjuntos de tareas que requieren un número de procesadores que tiende a infinito.

En segundo lugar, la obtención de N_{OPT} requiere un tiempo de procesamiento exponencial tal como se indica en el apartado 2.1.1.

En tercer lugar, incluso si el valor de N_{OPT} fuese conocido el número mínimo de procesadores necesarios deducidos de las métricas para planificar un conjunto de tareas dado puede ser excesivo. Por ejemplo, considérese la planificación local

³El incremento del tiempo de computación de cualquiera de las tareas de un conjunto que utiliza totalmente el multiprocesador hace que el conjunto no quepa en este.

<i>Planificación multiprocesador</i>	(N_{AA}/N_{OPT})	<i>Referencia</i>
EDF-FF	1.7	Garey y Johnson (1979a)
EDF-BF	1.7	Garey y Johnson (1979a)
RM-NF	2.67	Dall y Liu (1978)
RM-FF	2.33	Oh y Son (1995)
RM-BF	2.33	Oh y Son (1995)
RM-NFM	2.34	Davari y Dhall (1986a)
RM-FFD	2.0	Davari y Dhall (1986b)
RRM-FF	2.0	Oh y Son (1995)
RRM-BF	2.0	Oh y Son (1995)
RM-GT	1.75	A. Burchard y Son (1995)

Tabla 1.2: Valores de la métrica (N_{AA}/N_{OPT}) cuando $N_{opt} \rightarrow \infty$, para diferentes parejas algoritmo de planificación monoprocesador-algoritmo de asignación de tareas.

EDF de un conjunto de $10n$ tareas de factores de utilización $(0.5 + \epsilon)$. En este caso, empleando el algoritmo óptimo de asignación de tareas a procesadores se obtiene $N_{OPT} = 10n$ procesadores, pues ningún procesador puede albergar dos o más tareas de factor de utilización $(0.5 + \epsilon)$. Empleando el algoritmo de planificación EDF y el algoritmo de asignación de tareas FF se obtiene a partir de la tabla 1.2 el valor $(N_{FF}/N_{OPT}) = 1.7$. Es decir, se necesitarían al menos $17n$ procesadores para garantizar la planificabilidad del conjunto de tareas empleando planificación monoprocesador EDF y el algoritmo de asignación de tareas FF. En realidad, 10 procesadores son suficientes para planificar el conjunto de tareas empleando planificación monoprocesador EDF y el algoritmo de asignación de tareas FF, lo cual muestra el pesimismo de la métrica (N_{FF}/N_{OPT}) . Este pesimismo también se puede mostrar en términos de utilización.

La utilización total del conjunto de tareas anterior es $5n$ cuando $\epsilon \rightarrow 0$. Empleando $(17n - 1)$ procesadores, la planificabilidad del conjunto de tareas no está asegurada, y por lo tanto el límite de utilización deducido de la métrica (N_{FF}/N_{OPT}) para $(17n - 1)$ procesadores es menor que $5n$, mucho más bajo que el límite de utilización ajustado $0.5(17n - 1 + 1)$ que se presenta en la tesis.

La razón del pesimismo de métricas como la anterior reside en la cercanía o incluso coincidencia del caso peor (es decir, del conjunto de tareas más difícil de asignar) empleando el algoritmo de asignación considerado en la métrica, y el algoritmo de asignación de tareas óptimo.

Todo esto justifica la búsqueda de límites de utilización ajustados para las parejas algoritmo de planificación monoprocesador-algoritmo de asignación de tareas. Estos límites de utilización servirán además para realizar un dimensionado menos pesimista del sistema.

Desde el punto de vista teórico, la tesis obtiene fundamentalmente los límites de utilización ajustados asociados a diferentes combinaciones de algoritmo de planificación monoprocesador y algoritmo de asignación de tareas a procesadores. Es-

tos límites de utilización conjuntamente con los resultados de simulación permiten comparar desde diversos puntos de vista los diferentes algoritmos de planificación multiprocesador empleando estrategias locales.

Cabe reseñar el límite de utilización obtenido por Oh y Baker (1998), en la línea de los límites de utilización obtenidos en la tesis. Estos autores demostraron que el límite de utilización para planificación RM y asignación FF, denotado por U_{wc}^{RM-FF} , se encuentra dentro del rango:

$$n(2^{1/2} - 1) < U_{wc}^{RM-FF}(n) \leq (n + 1)/(1 + 2^{1/(n+1)}) \quad (1.6)$$

De este resultado se pueden extraer dos consecuencias. La primera es el límite de utilización $n(2^{1/2} - 1)$ para planificación RM y asignación de tareas FF. Este límite de utilización no es ajustado, como se puede comprobar comparándolo con el límite de utilización ajustado proporcionado en la tesis.

El límite de utilización proporcionado por Oh y Baker (1998), aparte de no ser ajustado tiene otra carencia. No tiene en cuenta el tamaño de las tareas. Este parámetro influye drásticamente en la planificabilidad del sistema, tal como se demuestra en la tesis.

En el caso de planificación local, es posible realizar análisis de planificabilidad basados en tiempo de respuesta, de manera análoga a como se hace en el caso monoprocesador. Durante la asignación de tareas a procesadores, para determinar si una tarea cabe en un procesador se emplea la condición de planificabilidad exacta monoprocesador basada en tiempos de respuesta (S. Sáez y Crespo, 1998; S. Lauzac y Mossé, 1998). Obviamente, esta forma de proceder mejora la planificabilidad en el caso de planificación RM o DM con respecto al empleo de condiciones de planificabilidad monoprocesador basadas en utilidades. No obstante, aparecen las mismas carencias que en el caso monoprocesador. Carencias que suple el empleo de condiciones de planificación multiprocesador basadas en utilidades.

En resumen, hay un vacío importante en el análisis de planificabilidad de sistemas de tiempo real implementados sobre multiprocesadores. El objetivo de la tesis es llenar parte de este vacío, obteniendo los límites de utilización multiprocesador para diferentes combinaciones de algoritmos de planificación monoprocesador y algoritmos de asignación de tareas.

Capítulo 2

Definición del problema

El problema de planificabilidad de sistemas de tiempo real es un problema de decisión (Garey y Johnson, 1979b), en el cual debe responderse “SI” o “NO” a la pregunta: ¿el sistema de tiempo real es planificable?, o lo que es lo mismo, ¿el sistema cumple las restricciones de tiempo real?

Habitualmente los estudios de planificabilidad no se conforman con responder a la pregunta anterior, sino que comparan asimismo la planificabilidad asociada a diferentes alternativas, o la influencia de determinados parámetros. Todo problema de decisión se puede dividir en dos partes:

- Un conjunto de instancias del problema.
- Una pregunta a la cual debe responderse con “SI” o “NO”.

En el apartado 2.1 se definen las instancias posibles del problema, o más exactamente, cuales son las instancias que se pretende resolver. La pregunta del problema es la indicada anteriormente: ¿el sistema de tiempo real es planificable?

En el apartado 2.2 se replantea el problema de planificabilidad. Las instancias del problema son las mismas, pero la pregunta original del problema de planificabilidad se plantea ahora en términos de utilización.

2.1 Definición de las instancias del problema

La definición de todas las instancias a resolver del problema se estructura de acuerdo con los siguientes elementos:

- Definición del conjunto de procesadores.
- Definición de la carga computacional asociada al conjunto de tareas.
- Definición del algoritmo de planificación.
- Definición de las restricciones de tiempo real del conjunto de tareas.

El conjunto de procesadores representa el recurso a administrar o planificar. Desde un punto de vista cliente/servidor los procesadores pueden verse como servidores

de tiempo de computación, mientras que las tareas como clientes de tiempo de computación. El tiempo de computación de una determinada tarea en un determinado procesador indica el tiempo que necesita una activación de la tarea para completarse en dicho procesador. El caso más complejo aparece cuando todos los procesadores son diferentes, pues en este caso el tiempo de computación de cada tarea viene especificado por tantos valores como procesadores hay en el sistema. En el caso más simple los procesadores son idénticos, por lo que el tiempo de computación de cada tarea viene definido por un único valor.

La carga computacional asociada al conjunto de tareas especifica las peticiones de computación realizadas por las tareas a lo largo del tiempo. Por ejemplo, una tarea puede solicitar 15 unidades de tiempo de computación únicamente en el instante $t = 10$, mientras que otra puede solicitar 5 unidades de tiempo de computación cada 10 unidades de tiempo a partir del instante $t = 2$. En este último caso se dice que la tarea es periódica de periodo 10, tiempo de computación 5, y fase 2. Cada 10 unidades de tiempo se dice además que se produce una activación de la tarea. Los instantes $t = 2$, $t = 12$, $t = 22$, etc, son los instantes de activación de dicha tarea. Sin embargo en la práctica, el tiempo de computación de cada una de las activaciones de una tarea periódica real es diferente. Asimismo, el tiempo que transcurre entre dos activaciones consecutivas de una tarea periódica real en la práctica no es exactamente el periodo, pues puede aparecer una pequeña desviación denominada *jitter* de activación. No obstante, a pesar de estas aperiodicidades las tareas periódicas reales se suelen asimilar a tareas periódicas ideales de tiempos de computación y periodos fijos. Además, se suelen considerar fases nulas, todo ello con la intención de simplificar el análisis. Un parámetro de la tarea periódica ideal que proporciona una idea acerca de la carga computacional asociada a la misma es su factor de utilización. Este factor de utilización se obtiene dividiendo el tiempo de computación de la tarea por su periodo. De forma análoga, la suma de los factores de utilización de un conjunto de tareas periódicas ideales proporciona una idea acerca de la carga computacional asociada a dicho conjunto. Esta suma de utilidades recibe el nombre de utilización total del conjunto de tareas.

El algoritmo de planificación arbitra el acceso de las tareas al procesador. En todo momento sólo aquellas tareas que estén listas compiten por el procesador. Esto es, si en un determinado instante todas las activaciones de una determinada tarea han sido completamente procesadas, entonces esa tarea no compite por ninguno de los procesadores. El algoritmo de planificación es una función que asigna en todo momento una tarea diferente a cada uno de los procesadores que constituyen el multiprocesador. Si hay menos tareas listas que procesadores, entonces alguno de los procesadores permanece inactivo. Un ejemplo de algoritmo de planificación multiprocesador es el *Global Rate-Monotonic* (GRM), el cual asigna los procesadores a tareas basándose en prioridades fijas de acuerdo con el criterio RM (Dall y Liu, 1978).

Las tareas tienen unas restricciones de tiempo real, las cuales se expresan habitualmente en función de los instantes de terminación de sus activaciones. Por ejemplo, una restricción de tiempo real es que cada activación de una cierta tarea termine a lo sumo tres unidades de tiempo después de su instante de activación. En

este caso se dice que la tarea tiene un plazo de tres unidades de tiempo. Los plazos de las tareas periódicas ideales son idénticos para cada una de sus activaciones. Si todas las activaciones de todas las tareas del conjunto cumplen sus restricciones de tiempo real se dice que el conjunto de tareas es planificable. En caso contrario se dice que el conjunto de tareas no es planificable.

A partir de las descripciones anteriores se definen formalmente las instancias a resolver del problema.

- Conjunto de procesadores.

Se tiene un conjunto de n procesadores idénticos e independientes. El hecho de que sean independientes indica que la ejecución en uno de ellos no depende de la ejecución en cualquier otro.

- Carga computacional asociada al conjunto de tareas.

Se tiene un conjunto de m tareas periódicas independientes $\Gamma = \{\tau_1, \dots, \tau_m\}$, de periodos $\{T_1, \dots, T_m\}$, tiempos de computación $\{C_1, \dots, C_m\}$, y factores de utilización $\{u_1 = C_1/T_1, \dots, u_m = C_m/T_m\}$.

Los factores de utilización de las tareas verifican $0 < u_i \leq \alpha \leq 1$ para $i = 1, \dots, m$. El parámetro α es el máximo factor de utilización alcanzable. Esto significa que no tiene por qué haber una tarea cuyo factor de utilización tome el valor α . La única restricción para que un valor de α sea válido es que se cumpla $u_i \leq \alpha \leq 1$ para $i = 1, \dots, m$. El máximo factor de utilización de las tareas, $u_{max} = \max_{1 \leq i \leq m} u_i$, cumple $\alpha \leq u_{max}$. Para más información acerca de la diferencia entre α y u_{max} véase el apartado 5.1. La utilización total del conjunto de tareas se obtiene como $U = \sum_{i=1}^m u_i$.

La primera activación de cada tarea ocurre en el instante $t = 0$. Por lo tanto, cualquier tarea τ_i se activa en los instantes jT_i , con $j = 0, 1, \dots$

El hecho de que las tareas sean independientes indica que no existe sincronización entre las mismas. Además se supone que tampoco se sincronizan con ningún evento externo, salvo por supuesto cuando estas se activan.

El conjunto de tareas es fijo en el tiempo, esto es, no se crean ni destruyen tareas a lo largo del tiempo. Todas las tareas se inician en el instante $t = 0$, es decir, tienen fase cero, y permanecen continuamente en el sistema.

El modelo de tareas descrito se generaliza en el capítulo 6.

- Definición del algoritmo de planificación. Las tareas se asignan a los procesadores siguiendo la política de particionado. Esto es, todas las activaciones de una misma tarea se ejecutan en el mismo procesador. Cada procesador tiene un planificador independiente de los demás planificadores. Sólo se consideran algoritmos de asignación de tareas a procesadores pertenecientes a la clase de *algoritmos de asignación razonables* (RA), descritos en el apartado 2.1.1. Todos los procesadores tienen un planificador local basado en prioridades fijas o *Earliest Deadline First* (EDF) (Liu y Layland, 1973). En el caso de que el planificador esté basado en prioridades fijas, estas se suponen previamente asignadas a las tareas empleando la asignación RM. El planificador local de todos los procesadores es el mismo.

- Restricciones de tiempo real del conjunto de tareas.
Las tareas tienen plazos $\{D_1 = T_1, \dots, D_m = T_m\}$ respecto a los instantes de activación. El conjunto de tareas es planificable si toda activación que ocurre en el instante jT_i se ha ejecutado completamente en el instante $jT_i + D_i$, con $i = 1, \dots, m$ y $j = 0, 1, \dots$. En el capítulo 6 se extiende el modelo de restricciones de tiempo real, permitiendo plazos menores que los periodos.

2.1.1 Algoritmos de asignación de tareas a procesadores

Siguiendo la estrategia local en la planificación multiprocesador, también denominada estrategia de particionado, una vez elegido el planificador local el único grado de libertad del sistema es el algoritmo de asignación de tareas a procesadores.

Sería interesante disponer de un algoritmo de asignación óptimo. Esto es, un algoritmo cumpliendo la siguiente propiedad: si el algoritmo óptimo no encuentra ninguna asignación de tareas a procesadores que haga el conjunto de tareas planificable, entonces no existe ningún algoritmo capaz de generar una asignación de tareas a procesadores que haga el conjunto de tareas planificable. Sin embargo, en la mayor parte de los casos el algoritmo óptimo carece de utilidad práctica, tal como se muestra a continuación.

El problema de asignar un conjunto de tareas a un conjunto de procesadores es análogo al problema de *bin-packing* (Garey y Johnson, 1979a). En este caso, el procesador es el recipiente (*bin*), cuya capacidad viene dada por la condición de planificabilidad local. Es decir, la condición que establece cuando el conjunto de tareas asignadas a un procesador es planificable, ignorando las demás tareas y procesadores. El problema de *bin-packing* pertenece a la clase de problemas *NP-hard* en el sentido estricto. Por lo tanto, los algoritmos de asignación óptimos en general no pueden computarse ni siquiera en tiempo pseudopolinomial. Esto es, a medida que crece el número de tareas y el número de procesadores el coste computacional crece de forma exponencial. Este tipo de algoritmos suelen obtener una solución mediante la búsqueda exhaustiva a lo largo de todo el árbol de posibilidades. Por ejemplo, dado un conjunto de m tareas y un conjunto de n procesadores existen n^m asignaciones diferentes. No obstante $n!$ de estas son equivalentes, por lo que un algoritmo de asignación óptimo se podría obtener comprobando la planificabilidad de hasta $(n^m/n!)$ asignaciones no equivalentes. Este algoritmo óptimo podría mejorarse recortando el árbol de búsqueda mediante ciertas técnicas. No obstante, en el caso peor el coste computacional del algoritmo es inadmisiblemente.

Dada la inviabilidad de aplicar algoritmos óptimos, debe acudir a algoritmos heurísticos. Estos tienen la propiedad de tener un coste computacional admisible, pero pagando el precio en algunos casos de una planificabilidad menor. Es decir, un determinado algoritmo heurístico puede tachar el sistema de no planificable cuando en realidad existe una asignación de tareas a procesadores que hace el sistema planificable.

En la literatura se pueden encontrar algoritmos de asignación heurísticos propuestos por diferentes autores (Garey y Johnson, 1979a; Dall y Liu, 1978; Davari y Dhall, 1986b; Davari y Dhall, 1986a; A. Burchard y Son, 1995; Oh y Son, 1995;

S. Sáez y Crespo, 1998). Algunos de estos algoritmos se podrían calificar de “no razonables”.

Por ejemplo, considérese el algoritmo de asignación *Next Fit* (Dall y Liu, 1978). Este algoritmo asigna cada tarea al primer procesador del sistema con suficiente capacidad. Los procesadores se recorren en el sentido $\{P_1, \dots, P_n\}$. Si una tarea no cabe en P_1 , entonces sólo se consideran los procesadores $\{P_2, \dots, P_n\}$ en la asignación de las tareas siguientes, incluso aunque alguna de las tareas siguientes cupiese en P_1 . Es como si el procesador P_1 dejase de existir, por lo que el hueco que quede en P_1 se desaprovecha. De forma análoga, si una tarea no cabe en P_2 , las demás tareas se asignan a los procesadores $\{P_3, \dots, P_n\}$, como si el procesador P_2 no existiese, desaprovechando el posible hueco que hubiese en P_2 . Este desafortunado comportamiento puede aparecer con los demás procesadores. Por lo tanto es posible que el algoritmo indique que una tarea no cabe en el multiprocesador, por lo cual el sistema es considerado no planificable, cuando en realidad cabe en alguno procesadores. Lógicamente, los algoritmos de este tipo no se pueden considerar “razonables”. En la tesis sólo se consideran algoritmos de asignación de tareas “razonables”. A continuación se define formalmente el concepto de *algoritmo de asignación razonable*.

Definición 1 *Se dice que un algoritmo de asignación de tareas a procesadores es razonable cuando es capaz de asignar una tarea a alguno de los procesadores siempre que existe al menos un procesador en el que la tarea cabe.*

El concepto de caber o no caber una tarea en un procesador depende del algoritmo de planificación local y de la condición de planificabilidad local empleada en el procesador. Para cada algoritmo de planificación local y cada condición de planificabilidad local asociada al mismo, se supone la existencia de una función capaz de proporcionar la *capacidad residual* (CR) del procesador después de haberle asignado un subconjunto de tareas. La capacidad residual indica la capacidad del procesador que aún queda disponible para alojar más tareas.

Si la capacidad residual es positiva indica que el subconjunto de tareas asignado al mismo es planificable (es decir, cabe) en dicho procesador, y aún queda capacidad para más tareas. Valores positivos crecientes de la capacidad residual indican procesadores menos cargados, o lo que es lo mismo, capaces de albergar más y más “pesadas” tareas. Valores negativos de la capacidad residual indican que el conjunto de tareas asignadas al procesador no es planificable (es decir, no cabe) en el mismo, utilizando el algoritmo de planificación local dado, y la condición de planificabilidad local asociada.

Por ejemplo, si el planificador local es RM, y la condición de planificabilidad local es la de Liu y Layland (1973), la capacidad residual del procesador P_j se puede definir intuitivamente como $CR(j) = (m_j + 1)(2^{1/(m_j+1)} - 1) - U_j$, donde U_j es la utilización total de las tareas previamente asignadas a P_j , y m_j es el número de tareas previamente asignadas a P_j . Como se puede observar, la capacidad residual del procesador depende del número de tareas. Además en este caso, un algoritmo razonable de asignación es aquel incapaz de asignar una tarea τ_i sólo cuando se cumple

$$CR(j) = (m_j + 1)(2^{1/(m_j+1)} - 1) - U_j < u_i \quad \text{para } j = 1, \dots, n \quad (2.1)$$

donde n denota el número de procesadores.

El empleo de condiciones de planificabilidad más complejas, como es la condición exacta para RM exigiría la definición de capacidades residuales más elaboradas.

Si el planificador local es EDF, la capacidad residual del procesador P_j se puede definir intuitivamente como $CR(j) = 1 - U_j$. En este caso la capacidad residual del procesador no depende del número de tareas. Además en este caso, un algoritmo razonable de asignación es aquel incapaz de asignar una tarea τ_i sólo cuando se cumple

$$CR(j) = 1 - U_j < u_i \quad \text{para } j = 1, \dots, n. \quad (2.2)$$

Los algoritmos de asignación razonables constituyen una clase formada por los mejores algoritmos de asignación. Por ejemplo, se puede demostrar fácilmente que el algoritmo de asignación óptimo pertenece a esta clase. Además, a partir de cualquier algoritmo de asignación no razonable se puede obtener de manera obvia un algoritmo de asignación razonable, capaz de asignar los mismos conjuntos de tareas que el no razonable del que parte e incluso otros. Esto justifica que la tesis considere únicamente algoritmos de asignación razonables, olvidando otros algoritmos teóricamente posibles pero de poca utilidad práctica, como es el caso del algoritmo *Next Fit*.

A continuación se describen los algoritmos de asignación básicos que se emplearan a lo largo de la tesis, estos son: *First Fit* (FF), *Best Fit* (BF), *Worst Fit* (WF), y *Random Fit* (RF). Todos ellos son algoritmos razonables de asignación. La descripción se realiza mediante pseudocódigo. Debe tenerse en cuenta que el objetivo de la descripción es poner de relieve las diferencias entre los algoritmos, sin tener en cuenta criterios de rapidez en su implementación. Por ejemplo, el algoritmo FF proporcionado puede optimizarse fácilmente en cuanto a su coste computacional, calculando únicamente las capacidades residuales de los primeros procesadores hasta llegar a un procesador con capacidad residual suficiente para albergar la tarea considerada en cada momento.

El algoritmo FF asigna cada tarea al primer procesador con capacidad residual suficiente para albergar la tarea. Los procesadores se recorren en el sentido $\{P_1, \dots, P_n\}$. Si todas las tareas caben en los procesadores el conjunto de tareas es planificable. Si por el contrario existe al menos una tarea que no cabe en los procesadores el conjunto de tareas es no planificable. Debe notarse que para implementar el algoritmo FF no es estrictamente necesario haber definido una expresión de capacidad residual. Es suficiente disponer de un algoritmo que indique si la nueva tarea y las anteriormente asignadas a un procesador dado son planificables en dicho procesador. Por lo tanto en este caso la comparación $CR(j) \geq u_i$ se puede sustituir por una condición de planificabilidad monoprocesador.

El algoritmo BF asigna cada tarea al procesador con menor capacidad residual suficiente para albergar la tarea. Si dos o más procesadores tuviesen la mínima capacidad residual suficiente, se elegiría el primero de ellos que se encuentra al recorrer la lista en sentido $\{P_1, \dots, P_n\}$. Si todas las tareas caben en los procesadores el conjunto de tareas es planificable. Si por el contrario existe al menos una tarea que no cabe en los procesadores el conjunto de tareas es no planificable.

Al contrario de lo que sucede con el algoritmo de asignación FF, no es suficien-

```

Asignación FF
  desde  $i := 1$  hasta  $m$ 
    planificable := NO
    desde  $j := 1$  hasta  $n$ 
      calcular  $CR(j)$ 
      si  $CR(j) \geq u_i$ 
        planificable := SI
    fin de desde
  si planificable = NO
    retornar NO PLANIFICABLE
  si no
     $j := 1$ 
    mientras  $CR(j) < u_i$ 
       $j := j + 1$ 
    fin de mientras
    asignar  $\tau_i$  a  $P_j$ 
  fin de si
fin de desde
retornar PLANIFICABLE
fin de asignación FF

```

Algoritmo 2.1.1: Algoritmo de asignación de tareas FF.

te disponer de una condición de planificabilidad monoprosesor para realizar la asignación BF. Es necesario definir una expresión para la capacidad residual que permita saber no sólo si una tarea cabe en un procesador, sino además cual es el “espacio disponible” en un procesador.

El algoritmo WF asigna cada tarea al procesador con mayor capacidad residual de entre todos los que tienen suficiente capacidad residual para albergar la tarea. Si dos o más procesadores tuviesen la máxima capacidad residual positiva, se elegiría el primero de ellos que se encuentra al recorrer la lista en sentido $\{P_1, \dots, P_n\}$. Si todas las tareas caben en los procesadores el conjunto de tareas es planificable. Si por el contrario existe al menos una tarea que no cabe en los procesadores el conjunto de tareas no es planificable.

Al contrario de lo que sucede con el algoritmo de asignación FF, no es suficiente disponer de una condición de planificabilidad monoprosesor para realizar la asignación WF. Es necesario definir una expresión para la capacidad residual que permita saber no sólo si una tarea cabe en un procesador, sino además cual es el “espacio disponible” en un procesador.

El algoritmo RF asigna cada tarea de forma aleatoria a uno de los procesadores de entre todos los que tienen capacidad residual mayor o igual que el factor de utilización de la tarea. Si todas las tareas caben en los procesadores el conjunto de tareas es planificable. Si por el contrario existe al menos una tarea que no cabe en los procesadores el conjunto de tareas es no planificable.

```

Asignación BF
  desde  $i := 1$  hasta  $m$ 
    planificable = NO
    desde  $j := 1$  hasta  $n$ 
      calcular  $CR(j)$ 
      si  $CR(j) \geq u_i$ 
        planificable := SI
    fin de desde
  si planificable = NO
    retornar NO PLANIFICABLE
  si no
     $k := 1$ 
    desde  $j := 2$  hasta  $n$ 
      si  $(CR(j) \geq u_i)$  y  $(CR(k) > CR(j) \text{ ó } CR(k) < u_i)$ 
         $k := j$ 
      fin de si
    fin de desde
    asignar  $\tau_i$  a  $P_k$ 
  fin de si
fin de desde
retornar PLANIFICABLE
fin de asignación BF

```

Algoritmo 2.1.2: Algoritmo de asignación de tareas BF.

Al igual que sucedía con el algoritmo FF, en este caso la comparación $CR(j) \geq u_i$ se puede sustituir por una condición de planificación monoprosesor.

Los algoritmos de asignación mostrados no son más que distintas formas de elegir entre los procesadores con capacidad residual suficiente para albergar cada tarea, es decir, de capacidad residual mayor o igual que el factor de utilización de la tarea. El FF asigna la tarea al primer procesador con capacidad residual suficiente, el BF al procesador con la menor capacidad residual pero suficiente, el WF al procesador con la mayor capacidad residual suficiente, y finalmente el RF a un procesador cualquiera de entre los que tienen capacidad residual suficiente. En principio, la intuición nos indica que el algoritmo BF debería ser el que proporcionase mejores resultados de entre el FF, BF, WF y RF. Sin embargo se demostrará que el algoritmo FF, intuitivamente inferior, proporciona resultados idénticos que el algoritmo BF en términos de límite de utilización. Este comportamiento idéntico de los algoritmos FF y BF también se observó desde el punto de vista de las métricas (N_{AA}/N_{opt}) que aparecen en la tabla 1.2, en la página 23.

El algoritmo RF puede parecer artificial, sin embargo representa el peor de los algoritmos de asignación razonables en términos de límite de utilización. Además, tal como se muestra en el apartado 6.5, se emplea para obtener una condición de planificabilidad bajo la presencia de cambios de modo en multiprocesadores.

```

Asignación WF
  desde  $i := 1$  hasta  $m$ 
    planificable := NO
    desde  $j := 1$  hasta  $n$ 
      calcular  $CR(j)$ 
      si  $CR(j) \geq u_i$ 
        planificable = SI
    fin de desde
  si planificable = NO
    retornar NO PLANIFICABLE
  si no
     $k := 1$ 
    desde  $j := 2$  hasta  $n$ 
      si  $(CR(j) \geq u_i)$  y  $(CR(k) < CR(j))$ 
         $k := j$ 
      fin de si
    fin de desde
    asignar  $\tau_i$  a  $P_k$ 
  fin de si
fin de desde
retornar PLANIFICABLE
fin de asignación WF

```

Algoritmo 2.1.3: Algoritmo de asignación de tareas WF.

El coste computacional de los cuatro algoritmos de asignación anteriores es similar en el caso peor. Sin embargo, el algoritmo FF optimizado presenta en general un menor coste computacional, pues en muchos casos es suficiente obtener la capacidad residual para los primeros procesadores, mientras que el empleo de los algoritmos BF, WF o RF exige en todos los casos el cálculo de las capacidades residuales de los n procesadores.

Existen variaciones de estos algoritmos que realizan un paso previo de ordenación de las tareas. Así los algoritmos FFD, BFD, WFD y RFD realizan la ordenación de las tareas en sentido decreciente de algún parámetro de las mismas, antes de realizar la asignación FF, BF, WF o RF. Asimismo, si se realiza un paso previo de ordenación de las tareas en sentido creciente de algún parámetro se obtienen los algoritmos FFI, BFI, WFI y RFI. Normalmente el parámetro elegido para la ordenación es un parámetro que varía de forma monótona creciente con la carga asociada a la tarea, esto es, un parámetro proporcional a la reducción de la carga residual cuando la tarea se asigna a un procesador. Por ejemplo, si se emplea el factor de utilización para representar la carga computacional asociada a una tarea, el algoritmo FFD ordena las tareas en sentido de utilidades decrecientes antes de realizar la asignación FF. Por lo tanto, se asigna primero con FF la tarea de factor de utilización mayor, y en último lugar la tarea de factor de utilización menor.

```

Asignación RF
  desde  $i := 1$  hasta  $m$ 
    planificable := NO
    desde  $j := 1$  hasta  $n$ 
      calcular  $CR(j)$ 
      si  $CR(j) \geq u_i$ 
        planificable := SI
    fin de desde
  si planificable = NO
    retornar NO PLANIFICABLE
  si no
     $k = \text{aleatorio}(j \mid CR(j) \geq u_i)$ 
    asignar  $\tau_i$  a  $P_k$ 
  fin de si
fin de desde
retornar PLANIFICABLE
fin de asignación RF

```

Algoritmo 2.1.4: Algoritmo de asignación de tareas RF.

2.2 Definición del problema en términos de utilización

Para clarificar la definición del problema en términos de utilización se establecerá un paralelismo entre el caso monoprocesador empleando planificación RM, y el caso multiprocesador.

Liu y Layland (1973) encontraron el límite de utilización ajustado $U_{wc} = m(2^{1/m} - 1)$. Todo conjunto de tareas de de utilización total menor o igual que U_{wc} es planificable empleando planificación RM. Por otra parte, para cualquier valor de utilización total mayor o igual que U_{wc} existe un conjunto de tareas con esa utilización que no es planificable. La importancia de este tipo de condiciones de planificabilidad es que permiten establecer rápidamente la planificabilidad del conjunto de tareas, considerando el caso peor que puede ocurrir. El inconveniente de este tipo de condiciones de planificabilidad es que son sólo condiciones suficientes. De hecho, conjuntos de tareas de utilización total mayor que U_{wc} pueden ser planificables.

El problema de planificabilidad en el caso monoprocesador con las restricciones anteriores se puede replantear como: ¿la utilización total del conjunto de tareas es menor o igual que U_{wc} ?

La idea es llevar este resultado al caso multiprocesador. En este caso el problema de planificabilidad en multiprocesadores se podría replantear de igual forma: ¿la utilización total del conjunto de tareas es menor o igual que U_{wc} ? Para ello debe obtenerse el límite de utilización U_{wc} multiprocesador, análogo al límite $m(2^{1/m} - 1)$ asociado a monoprocesadores.

A lo largo de la tesis se obtiene el valor de U_{wc} para para diferentes combinaciones

de algoritmo de planificación monoprocesador y algoritmo de asignación.

Para la obtención del límite de utilización del multiprocesador se consideran únicamente los factores de utilización de las tareas, ignorando sus tiempos de computación y periodos. La razón es que el límite de utilización viene dado para los tiempos de computación y periodos más desfavorables que cumplen la restricción $u_i = C_i/T_i$ impuesta por los factores de utilización.

Al venir definidas las tareas simplemente por sus factores de utilización, a lo largo de la tesis se empleará para evaluar la capacidad residual las expresiones

$$CR(j) = (m_j + 1)(2^{1/(m_j+1)} - 1) - U_j \quad (2.3)$$

para RM, y

$$CR(j) = 1 - U_j \quad (2.4)$$

para EDF, tal como se indicó en los ejemplos del apartado anterior.

Además, cualquier ordenación de las tareas ya sea en sentido creciente o decreciente supone implícitamente una ordenación por factores de utilización.

Capítulo 3

Límites de utilización empleando planificación EDF

En este capítulo se presentan los límites de utilización para planificación Earliest Deadline First (EDF) sobre multiprocesadores homogéneos, empleando técnicas de particionado. Las tareas se planifican con expulsión sobre cada monoprocesador de acuerdo con el algoritmo EDF, y se asignan a los procesadores de acuerdo con los algoritmos de asignación presentados en el apartado 2.1.1. Las restricciones de tiempo real se modelan como plazos críticos de valor igual a los periodos de las tareas. Los límites de utilización para plazos menores que los periodos y planificación EDF se presentan en el apartado 6.3.

En primer lugar, se define en el apartado 3.1 el concepto de límite de utilización para planificación EDF y un algoritmo arbitrario de asignación de tareas AA dado. Al final del apartado 3.1 se demuestra la existencia de dicho límite para el caso de algoritmos de asignación razonables.

El límite de utilización asociado a cualquier algoritmo razonable de asignación de tareas y planificación EDF se encuentra dentro de un intervalo $[L_{EDF}, H_{EDF}]$. El término L_{EDF} del intervalo se corresponde con el mínimo de los límites de utilización, evaluado entre todos los algoritmos RA de asignación razonables. De forma análoga, el término H_{EDF} del intervalo se corresponde con el máximo de los límites de utilización, evaluado entre todos los algoritmos RA de asignación razonables. El intervalo $[L_{EDF}, H_{EDF}]$ se calcula en el apartado 3.2.1.

El límite de utilización para los algoritmos de asignación FF, BF, WF, RF y OPT se calcula en los apartados 3.2.2 a 3.2.6.

La ordenación de las tareas por factores de utilización antes de realizar la asignación FF, BF, WF, o RF, modifica los límites de utilización de la forma indicada en el apartado 3.2.7.

Las expresiones matemáticas de los límites de utilización obtenidas en los apartados 3.2.1 a 3.2.7 son analizadas en el apartado 3.3.

Con la intención de mostrar el pesimismo de los límites de utilización (absolutos), en el apartado 3.4 se presentan los límites de utilización estadísticos, obtenidos mediante simulación exhaustiva. Los límites de utilización absolutos, denominados simplemente límites de utilización, se obtienen considerando el caso peor. Estos son el resultado de un enfoque puramente teórico del problema. Los límites de

utilización estadísticos permiten un tanto por ciento de conjuntos de tareas generados aleatoriamente que no son planificables. Estos últimos son el resultado del enfoque de simulación. Por lo tanto, la información proporcionada por los enfoques teórico y de simulación es complementaria.

El apartado 3.5 compara los diferentes algoritmos de asignación de tareas. La conclusión es que el algoritmo FFD es superior en términos de límite de utilización a todos los demás algoritmos heurísticos considerados en la tesis. Además proporciona resultados cercanos a los del algoritmo de asignación óptimo. El inconveniente del algoritmo de asignación FFD es que no puede aplicarse en línea pues ordena las tareas.

El apartado 3.5 muestra asimismo que el límite de utilización estadístico asociado al algoritmo FF es generalmente inferior pero cercano al asociado al algoritmo FFD. No obstante, al contrario de lo que sucede con el algoritmo FFD, el algoritmo FF puede aplicarse en línea.

3.1 Definición y existencia de los límites de utilización

En este apartado se define el concepto de límite de utilización para planificación multiprocesador con particionado, empleando EDF como algoritmo de planificación local (en cada procesador), y un algoritmo de asignación de tareas AA arbitrario (razonable o no). Una vez definido el concepto de límite de utilización se demuestra su existencia para el caso de algoritmos de asignación razonables.

Por convenio, la planificación multiprocesador con particionado empleando EDF como planificador local, y AA como algoritmo de asignación de tareas, se representa como EDF-AA. El límite de utilización asociado se representa como U_{wc}^{EDF-AA} . Cuando el algoritmo de asignación es razonable se sustituyen las iniciales AA por las iniciales RA.

Definición 2 *El límite de utilización para EDF-AA se define como el número real U_{wc}^{EDF-AA} que verifica las siguientes propiedades:*

- *Cualquier conjunto de tareas de utilización total $U \leq U_{wc}^{EDF-AA}$ cabe en el multiprocesador, empleando planificación EDF en cada procesador y asignación de tareas AA. Por lo tanto, es planificable empleando planificación multiprocesador EDF-AA.*
- *Para todo valor de utilización $U > U_{wc}^{EDF-AA}$ existe al menos un conjunto de tareas con esa utilización que no cabe en el multiprocesador, empleando planificación EDF en cada procesador y asignación de tareas AA. Por lo tanto, el conjunto de tareas con esa utilización puede ser planificable o no empleando planificación multiprocesador EDF-AA.*

Por consiguiente, el límite de utilización puede emplearse como una condición suficiente de planificabilidad.

Una tarea cabe en un procesador si y sólo si la suma de los factores de utilización de la tarea en cuestión y de las tareas previamente asignadas al procesador acumulan una utilización total de valor uno o por debajo de uno. Este resultado es una consecuencia del límite de utilización para planificación EDF en monoprocesadores (Liu y Layland, 1973).

A continuación se demuestra la existencia del límite de utilización U_{wc}^{EDF-RA} asociado a la planificación EDF-RA, para cualquier algoritmo razonable de asignación de tareas RA dado.

Lema 1 *Existe el límite de utilización asociado a la planificación EDF-RA para cualquier algoritmo razonable de asignación de tareas RA, el cual es una función (a lo sumo) del número de tareas, del número de procesadores, y del máximo factor de utilización alcanzable, α .*

Demostración:

Sea $\Pi(m, n, \alpha)$ el conjunto de todos los números reales positivos π que cumplen la condición siguiente: cualquier conjunto de m tareas, de factores de utilización $0 < u_i \leq \alpha$, y utilización total $U \leq \pi$ cabe en n procesadores empleando la asignación de tareas RA y teniendo en cuenta que la capacidad de cada procesador es uno. El conjunto $\Pi(m, n, \alpha)$ es no vacío, pues cualquier conjunto de tareas de utilización menor o igual que uno cabe en un procesador, y por lo tanto en n procesadores. Además todos los elementos de $\Pi(m, n, \alpha)$ son menores o iguales que el valor finito n , pues cualquier conjunto de tareas de utilización total mayor que n no cabe en n procesadores. Por lo tanto, existe un máximo en el conjunto $\Pi(m, n, \alpha)$, denominado $\pi_{\max}(m, n, \alpha)$, que es una función de m , n y α . A continuación probaremos que $\pi_{\max}(m, n, \alpha)$ es el límite de utilización, esto es, cumple las dos propiedades de la definición 2.

Cualquier conjunto de tareas de utilización menor o igual que $\pi_{\max}(m, n, \alpha)$ cabe en n procesadores, pues $\pi_{\max}(m, n, \alpha)$ es un elemento de $\Pi(m, n, \alpha)$. Esto demuestra la primera propiedad de la definición 2. Además, al ser $\pi_{\max}(m, n, \alpha)$ el máximo de Π resulta que existe al menos un conjunto de m tareas, de utilización total $\pi_{\max}(m, n, \alpha) + \epsilon$, con $\epsilon \rightarrow 0^+$, que no cabe en los procesadores¹. Si no fuese así, $\pi_{\max}(m, n, \alpha) + \epsilon$ sería un elemento de Π mayor que el máximo, $\pi_{\max}(m, n, \alpha)$, lo cual no es posible. Además, para cualquier utilización total $U > \pi_{\max}(m, n, \alpha) + \epsilon$ aún mayor, es posible encontrar un conjunto de m tareas que no cabe en los n procesadores. Basta con incrementar (cumpliendo la restricción $u_i \leq \alpha$) los factores de utilización de la última o últimas tareas del conjunto de utilización total $\pi_{\max}(m, n, \alpha) + \epsilon$, hasta alcanzar el valor de utilización. Lógicamente, el conjunto de tareas obtenido es más difícil de asignar que el conjunto original de utilización total $\pi_{\max}(m, n, \alpha) + \epsilon$. Por lo tanto, para cualquier utilización total mayor que $\pi_{\max}(m, n, \alpha)$, existe al menos un conjunto de m tareas, el cual no cabe en n procesadores. Esto demuestra la última propiedad de la definición 2. ■

¹La expresión $\epsilon \rightarrow 0^+$ es equivalente a $\epsilon \rightarrow 0$, y $\epsilon > 0$.

El lema 1 demuestra entre otras cosas que el límite de utilización para planificación EDF en un multiprocesador, empleando un determinado algoritmo razonable RA de asignación de tareas depende a lo sumo del número de tareas m , del número de procesadores n , y del máximo factor de utilización alcanzable α . Por esta razón el límite de utilización se representa como

$$U_{wc}^{EDF-RA}(m, n, \alpha)$$

3.2 Cálculo de los límites de utilización

Una vez definido el concepto de límite de utilización, y después de haber demostrado su existencia para el caso de algoritmos de asignación razonables se procede al cálculo del mismo. Debe tenerse en cuenta que cada algoritmo de asignación de tareas tendrá asociado un límite de utilización en principio distinto.

En general, los límites de utilización son una función del número de tareas m , el número de procesadores n , y el máximo factor de utilización alcanzable α , tal como se demostró en el lema 1.

Antes de calcular los límites de utilización, se introducirá un nuevo parámetro denominado β_{EDF} .

Definición 3 *El parámetro β_{EDF} se define como el máximo número de tareas de factor de utilización α que caben en un procesador.*

Este parámetro es un concepto clave en la obtención de los límites de utilización. El parámetro β_{EDF} es una función del máximo factor de utilización alcanzable α , tal como demuestra el lema 2.

Lema 2

$$\beta_{EDF} = \lfloor 1/\alpha \rfloor \tag{3.1}$$

Demostración:

A partir de la definición del parámetro β_{EDF} , se deduce que β_{EDF} tareas de factor de utilización α caben en un procesador. De esta forma, a partir del límite de utilización para planificación EDF en monoprocesadores se obtiene $\beta_{EDF}\alpha \leq 1$. Despejando β_{EDF} resulta $\beta_{EDF} \leq 1/\alpha$. Se sabe que β_{EDF} es un número entero, y por lo tanto se cumple $\beta_{EDF} \leq \lfloor 1/\alpha \rfloor$. Por otra parte, β_{EDF} es el máximo número de tareas de factor de utilización α que caben en un procesador. Por lo tanto, $(\beta_{EDF} + 1)$ tareas de factor de utilización α no caben en un procesador. Es decir, $(\beta_{EDF} + 1)\alpha > 1$. Despejando β_{EDF} se obtiene $\beta_{EDF} > 1/\alpha - 1$. Puesto que β_{EDF} es un número entero resulta $\beta_{EDF} \geq \lfloor 1/\alpha \rfloor$. En conclusión, $\beta_{EDF} = \lfloor 1/\alpha \rfloor$. ■

El valor de β_{EDF} puede emplearse para establecer la planificabilidad de ciertos conjuntos de tareas. A partir de la definición de β_{EDF} se tiene que β_{EDF} tareas de factor de utilización α caben en cada procesador. Puesto que todas las tareas tienen factores de utilización menores o iguales que α , al menos β_{EDF} tareas de factores de utilización arbitrarios (menores o iguales que α) caben en cada procesador. Como

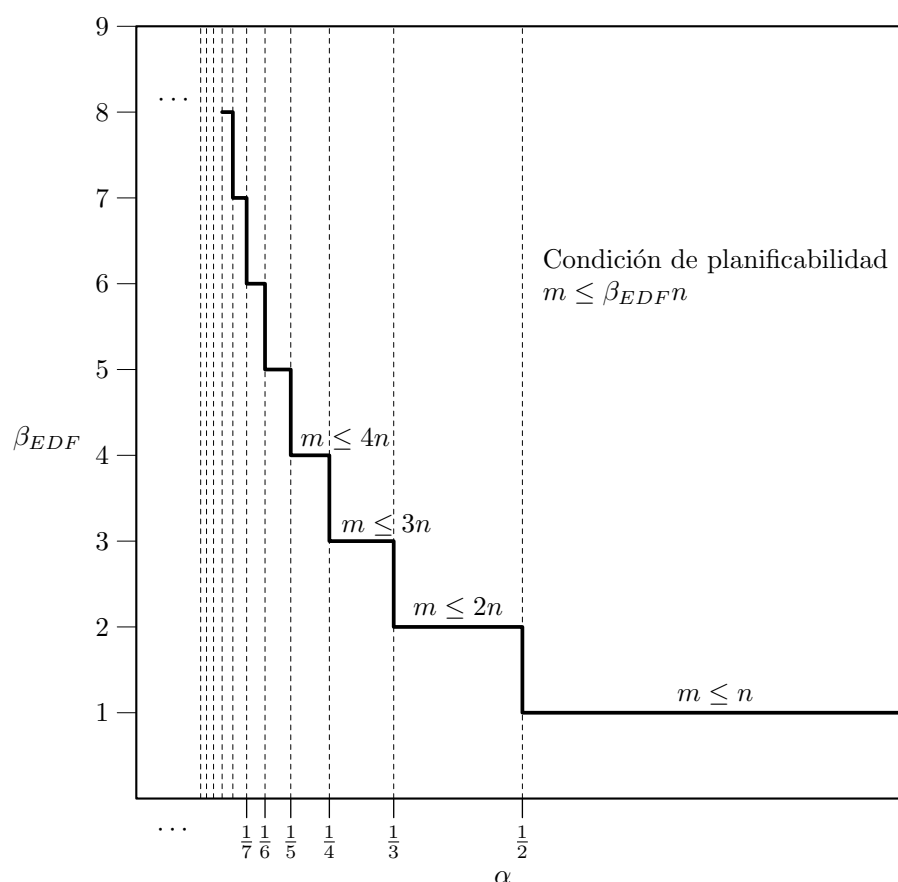


Figura 3.1: Representación de la función $\beta_{EDF}(\alpha)$, y la condición de planificabilidad asociada.

consecuencia, un multiprocesador formado por n procesadores puede alojar al menos $\beta_{EDF}n$ tareas. Por consiguiente, cualquier conjunto de m tareas de factores de utilización $u_i \leq \alpha$, con $m \leq \beta_{EDF}n$, es planificable empleando EDF como planificador monoprocador, y empleando cualquier algoritmo razonable de asignación de tareas a procesadores.

La figura 3.1 muestra β_{EDF} en función de α , mostrando asimismo la condición suficiente de planificabilidad $m \leq \beta_{EDF}n$. Por ejemplo, si α se encuentra en el intervalo $(1/3, 1/2]$ entonces $\beta_{EDF} = 2$. En este caso, si el conjunto de tareas está formado por $2n$ tareas o menos es planificable.

Otra consecuencia de la condición de planificabilidad $m \leq \beta_{EDF}n$ es que merece la pena obtener el límite de utilización sólo si $m > \beta_{EDF}n$. En caso contrario, el conjunto de tareas es directamente planificable.

A partir de ahora se supone que $m > \beta_{EDF}n$, de tal forma que la condición de planificabilidad anterior no puede aplicarse.

Antes de calcular los límites de utilización asociados a los diferentes algoritmos de asignación de tareas se obtendrá el rango dentro del cual se encuentran. Este resultado es significativo desde tres puntos de vista. En primer lugar, establece los valores mínimo y máximo entre los que se encuentran los límites de utilización de todos los algoritmos de asignación razonables. En segundo lugar, permite obtener

de forma sencilla el límite de utilización correspondiente a ciertos algoritmos de asignación razonables. En tercer y último lugar, servirá como comprobación de la validez de los límites de utilización obtenidos para los diferentes algoritmos de asignación de tareas, pues deben encontrarse de forma forzosa dentro del rango definido.

3.2.1 Rango de los límites de utilización

El objetivo es obtener la pareja de valores L_{EDF} y H_{EDF} que verifican la siguiente condición:

$$L_{EDF} \leq U_{wc}^{EDF-RA} \leq H_{EDF}$$

donde U_{wc}^{EDF-RA} es el límite de utilización para planificación EDF empleando cualquier algoritmo razonable RA de asignación de tareas a procesadores. El límite de utilización para cualquier algoritmo de asignación razonable es positivo (mayor o igual que uno), y por lo tanto a partir de cualquier pareja (L_{EDF}, H_{EDF}) que cumpla la condición anterior se pueden obtener infinitas parejas que también la cumplen, simplemente decrementando el valor de L_{EDF} o incrementando el valor de H_{EDF} . Resulta conveniente entonces calcular la pareja de valores ajustada. Esto es, debe existir al menos un algoritmo razonable de asignación de tareas cuyo límite de utilización para planificación EDF tome el valor L_{EDF} . Por lo tanto

$$L_{EDF} = \min_{RA} U_{wc}^{EDF-RA} \quad (3.2)$$

De la misma forma, debe existir al menos un algoritmo razonable de asignación de tareas cuyo límite de utilización para planificación EDF tome el valor H_{EDF} . Por lo tanto

$$H_{EDF} = \max_{RA} U_{wc}^{EDF-RA} \quad (3.3)$$

A partir de ahora, cualquier referencia al rango de los límites de utilización en este capítulo se referirá a los valores L_{EDF} y H_{EDF} ajustados, definidos por las ecuaciones (3.2) y (3.3).

El teorema 1 proporciona una cota inferior del límite de utilización, válida para todos los algoritmos de asignación razonables. En el apartado 3.2.4 se encuentra un algoritmo razonable de asignación de tareas cuyo límite de utilización coincide con esa cota. Por lo tanto L_{EDF} será igual a la cota dada por el teorema 1.

Por otra parte, el teorema 2 proporciona una cota superior del límite de utilización, válida para todos los algoritmos de asignación razonables. En el apartado 3.2.2 se encuentra un algoritmo razonable de asignación de tareas cuyo límite de utilización coincide con esa cota. Por lo tanto H_{EDF} será igual a la cota dada por el teorema 2.

Teorema 1 *Sea RA un algoritmo razonable de asignación de tareas a procesadores arbitrario. Si $m > \beta_{EDF}n$ se cumple*

$$U_{wc}^{EDF-RA} \geq n - (n - 1)\alpha \quad (3.4)$$

Demostración:

Sea $\{\tau_1, \dots, \tau_m\}$ un conjunto de m tareas que no cabe en el multiprocesador. Sea τ_k la primera tarea del conjunto que no cabe en el multiprocesador. Puesto que el algoritmo de asignación es razonable, aplicando la ecuación (2.2) de la página 30 se tiene

$$U_j > 1 - u_k \quad (3.5)$$

para $j = 1, \dots, n$. Donde U_j es la utilización total de las tareas previamente asignadas al procesador P_j , y u_k es el factor de utilización de la tarea τ_k .

Al ser τ_k una tarea del conjunto de m tareas resulta que la utilización total del conjunto de tareas denotada por U verifica

$$U = \sum_{i=1}^m u_i \geq \sum_{i=1}^k u_i = \sum_{j=1}^n U_j + u_k \quad (3.6)$$

A partir de la ecuación (3.5) se deduce

$$\sum_{j=1}^n U_j > \sum_{j=1}^n (1 - u_k) = n(1 - u_k)$$

Sustituyendo esta expresión en la ecuación (3.6) resulta

$$U > n(1 - u_k) + u_k = n - (n - 1)u_k$$

Como todos los factores de utilización son menores o iguales que α se tiene $u_k \leq \alpha$, y por lo tanto

$$U > n - (n - 1)\alpha$$

Por lo tanto, una condición necesaria que debe cumplir un conjunto de tareas que no cabe en n procesadores es que su utilización total U verifique

$$U > n - (n - 1)\alpha$$

de donde se deduce que todo conjunto de tareas de utilización total U cumpliendo

$$U \leq n - (n - 1)\alpha$$

cabe en n procesadores.

Debe tenerse en cuenta que no se ha hecho ninguna suposición con respecto al algoritmo de asignación de tareas a procesadores, salvo que este sea razonable. Por consiguiente se cumple

$$U_{wc}^{EDF-RA} \geq n - (n - 1)\alpha$$

donde RA es un algoritmo razonable de asignación de tareas arbitrario. ■

A continuación se obtiene una cota superior del límite de utilización asociado a cualquier algoritmo de asignación de tareas bajo planificación EDF. Por lo tanto, dicha cota es aplicable a los algoritmos razonables de asignación.

Teorema 2 *Sea AA un algoritmo de asignación de tareas a procesadores arbitrario (razonable o no). Si $m > \beta_{EDF}n$ se cumple*

$$U_{wc}^{EDF-AA} \leq \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \quad (3.7)$$

Demostración:

Se demostrará que existe un conjunto de m tareas $\{\tau_1, \dots, \tau_m\}$, con factores de utilización $0 < u_i \leq \alpha$ para $i = 1, \dots, m$, y utilización total

$$U = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} + \epsilon$$

donde $\epsilon \rightarrow 0^+$, el cual no cabe en n procesadores empleando cualquier algoritmo de asignación de tareas a procesadores.

Se construirá este conjunto de m tareas a partir de dos subconjuntos de tareas: un primer subconjunto de $(m - \beta_{EDF}n - 1)$ tareas, y un segundo subconjunto con $(\beta_{EDF}n + 1)$ tareas.

Todas las tareas del primer subconjunto tienen el mismo factor de utilización de valor ϵ/m , donde $i = 1, \dots, (m - \beta_{EDF}n - 1)$.

Todas las tareas del segundo subconjunto tienen el mismo factor de utilización de valor

$$u_i = \frac{1}{\beta_{EDF} + 1} + \epsilon/m$$

donde $i = (m - \beta_{EDF}n), \dots, m$.

Se puede comprobar que la utilización total del conjunto de tareas completo es

$$U = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} + \epsilon$$

En primer lugar, es necesario comprobar que los factores de utilización de ambos subconjuntos son válidos, es decir, se cumple $0 < u_i \leq \alpha$ para $i = 1, \dots, m$.

Comprobación de los factores de utilización del primer subconjunto. Eligiendo ϵ suficientemente pequeño se obtiene

$$0 < u_i = \frac{\epsilon}{m} \leq \alpha$$

Comprobación de los factores de utilización del segundo subconjunto. A partir de la definición de β_{EDF} se tiene que $(\beta_{EDF} + 1)$ tareas de factor de utilización

α no caben en un procesador. Por lo tanto, $(\beta_{EDF} + 1)\alpha > 1$, y se tiene

$$\alpha > \frac{1}{\beta_{EDF} + 1}$$

Siempre es posible encontrar un número real entre dos números reales. Por consiguiente existe un real positivo ϵ verificando

$$\alpha > \frac{1}{\beta_{EDF} + 1} + \frac{\epsilon}{m} = u_i$$

lo cual demuestra que los factores de utilización del segundo subconjunto son menores o iguales que α cuando $\epsilon \rightarrow 0^+$. Además, los factores de utilización del segundo subconjunto son obviamente mayores que cero.

A partir de los resultados anteriores se concluye que el conjunto de tareas propuesto es válido. A continuación se demostrará que dicho conjunto de tareas no cabe en n procesadores empleando planificación EDF en cada procesador, cualquiera que sea el algoritmo de asignación de tareas empleado.

El conjunto contiene $(\beta_{EDF}n + 1)$ tareas que pertenecen al segundo subconjunto. Por consiguiente, al menos un procesador de los n disponibles debe alojar $(\beta_{EDF} + 1)$ o más de estas tareas. Sin embargo, ningún procesador puede alojar $(\beta_{EDF} + 1)$ o más tareas del segundo subconjunto, pues $(\beta_{EDF} + 1)$ de estas tareas en total acumulan una utilización por encima de uno.

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

De esta forma se concluye que el conjunto de tareas propuesto de utilización total

$$U = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} + \epsilon$$

no cabe en n procesadores cualquiera que sea el algoritmo de asignación AA, cuando $\epsilon \rightarrow 0^+$, luego se cumple

$$U_{wc}^{EDF-AA} \leq \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1}$$

NOTA: las tareas del primer subconjunto son necesarias únicamente dentro de la demostración para cumplir la restricción de que el conjunto de tareas esté formado por m tareas. ■

En el apartado 3.2.4 se demostrará que el límite de utilización para EDF-WF toma el valor

$$U_{wc}^{EDF-WF} = n - (n - 1)\alpha$$

el cual coincide con la cota inferior dada en el teorema 1. Por lo tanto se tiene

$$L_{EDF} = L_{EDF}(n, \alpha) = n - (n - 1)\alpha \quad (3.8)$$

De forma análoga, en el apartado 3.2.2 se demostrará que el límite de utilización para EDF-FF toma el valor

$$U_{wc}^{EDF-FF} = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1}$$

el cual coincide con la cota superior dada en el teorema 2. Por lo tanto se tiene

$$H_{EDF} = H_{EDF}(n, \beta_{EDF}) = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \quad (3.9)$$

Cualquier algoritmo razonable de asignación de tareas RA cumple entonces

$$n - (n - 1)\alpha \leq U_{wc}^{EDF-RA} \leq \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \quad (3.10)$$

donde

$$\beta_{EDF} = \lfloor 1/\alpha \rfloor$$

α es el máximo factor de utilización alcanzable, y n es el número de procesadores que constituyen el multiprocesador.

3.2.2 Límite de utilización para EDF-FF

El algoritmo de asignación de tareas denominado FF asigna cada tarea al primer procesador que encuentra con capacidad residual suficiente para albergar la tarea. Los procesadores se recorren en el sentido $\{P_1, \dots, P_n\}$. La descripción detallada de este algoritmo de asignación de tareas puede encontrarse en el apartado 2.1.1.

La demostración del límite de utilización para EDF-FF es bastante compleja, por lo que se indicará en primer lugar la estructura de la demostración. Los pasos que se seguirán son los siguientes:

1. Se demuestra el lema 3. Este lema relaciona los límites de utilización empleando el mismo algoritmo razonable de asignación, el mismo número de procesadores pero distinto número de tareas. Este lema se emplea para demostrar el teorema 3.
2. Se demuestra el teorema 3, el cual relaciona el límite de utilización asociado a m tareas y n procesadores con el límite de utilización asociado a $(m - \beta_{EDF})$ tareas y $(n - 1)$ procesadores, para planificación EDF-FF.
3. A partir del resultado dado en el paso 2 y el límite de utilización para planificación EDF en un monoprocesador, el teorema 4 obtiene una cota inferior del límite de utilización para EDF-FF.
4. La cota inferior dada en el paso 3, y la cota superior proporcionada por el teorema 2 coinciden. Por lo tanto, ambas cotas coinciden además con el límite de utilización para EDF-FF, tal como muestra finalmente el teorema 4.

La demostración del teorema 3 requiere el lema 3 que se demuestra a continuación. Este lema relaciona los límites de utilización asociados a un mismo número de procesadores pero distinto número de tareas, empleando un algoritmo de asignación razonable arbitrario, siempre y cuando no realice ordenación alguna de tareas antes de hacer la asignación. El resultado es válido para la asignación de tareas FF, por ser un algoritmo razonable de asignación de tareas que no lleva a cabo ninguna ordenación de tareas previa a la asignación.

Lema 3 *Sea RA un algoritmo razonable de asignación de tareas a procesadores que no realiza ordenación previa de las tareas antes de realizar la asignación. Se cumple*

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

Demostración:

El lema se demostrará por reducción al absurdo. Supóngase que existe una pareja de enteros q y m , tal que $q < m$ y

$$U_{wc}^{EDF-RA}(q, n, \alpha) < U_{wc}^{EDF-RA}(m, n, \alpha)$$

Entre dos números reales siempre es posible encontrar otro número real, de tal forma que se puede encontrar un $\epsilon > 0$ verificando

$$U_{wc}^{EDF-RA}(q, n, \alpha) < U_{wc}^{EDF-RA}(m, n, \alpha) - \epsilon < U_{wc}^{EDF-RA}(m, n, \alpha)$$

De acuerdo con la definición de límite de utilización, existe al menos un conjunto de q tareas, $\{\tau_1, \dots, \tau_q\}$, de utilización total

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

mayor que el límite de utilización para q tareas, el cual no cabe en n procesadores. A continuación se demostrará que esto da lugar a una contradicción.

Si se añade a este conjunto de q tareas $(m - q)$ nuevas tareas, $\{\tau_{q+1}, \dots, \tau_m\}$, cada una de factor de utilización $\epsilon/(m - q)$, se obtiene un nuevo conjunto de tareas formado por m tareas, de utilización total

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

el cual cabe en n procesadores. Por consiguiente, las primeras q tareas caben en n procesadores, lo que constituye una contradicción. ■

El matiz de que el algoritmo de asignación no realizase ningún tipo de ordenación previa antes de asignar las tareas es importante. En la demostración del lema 3 se han añadido $(m - q)$ tareas de factor de utilización arbitrariamente bajo al final de un conjunto de q tareas. Por lo tanto, las últimas tareas son las que tienen los factores de utilización más bajos, lo que iría en contra de una ordenación de tareas

en sentido de factores de utilización crecientes, como la que realiza el algoritmo FFI. Por lo tanto el lema anterior no sería aplicable empleando asignación FFI.

A continuación se demostrará una expresión que relaciona los límites de utilización asociados a multiprocesadores con n y $(n - 1)$ procesadores, empleando asignación de tareas FF. Esto permitirá obtener una cota inferior del límite de utilización para asignación de tareas FF, yendo del caso $n = 1$ (caso monoprocesador) hasta un multiprocesador general con un número arbitrario n de procesadores.

Teorema 3 *Si $m > \beta_{EDF}n$ se cumple*

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

Demostración

Se probará que cualquier conjunto de m tareas $\{\tau_1, \dots, \tau_m\}$, con factores de utilización $0 < u_i \leq \alpha$ para $i = 1, \dots, m$, y utilización total menor o igual que

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

cabe en n procesadores empleando planificación EDF en cada procesador y asignación de tareas FF.

Hay dos casos posibles:

Caso 1: Las primeras $(m - \beta_{EDF})$ tareas tienen utilización total menor o igual que $U_{wc}^{EDF-FF}(m - \beta_{EDF}, n - 1, \alpha)$, esto es

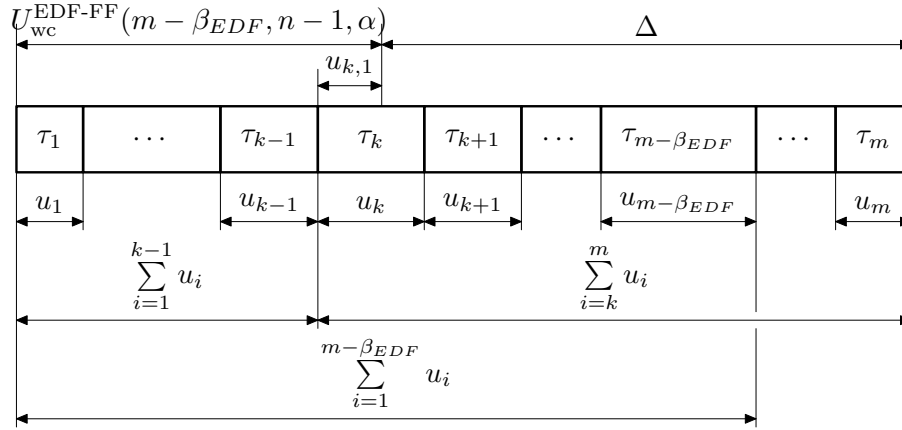
$$\sum_{i=1}^{m-\beta_{EDF}} u_i \leq U_{wc}^{EDF-FF}(m - \beta_{EDF}, n - 1, \alpha)$$

Las primeras $(m - \beta_{EDF})$ tareas caben en los primeros $(n - 1)$ procesadores, pues su utilización total está por debajo del límite de utilización. Por lo tanto, empleando el algoritmo FF se asignan a los $(n - 1)$ primeros procesadores. Las restantes β_{EDF} tareas caben en el último procesador, pues la definición de β_{EDF} implica que al menos β_{EDF} tareas siempre caben en un procesador. El hecho de que quepan en el último procesador no significa que se asignen al último procesador. De todas formas, el caso en el cual alguna de estas β_{EDF} se asignase a alguno de los primeros $(n - 1)$ procesadores es más favorable, pues habría menos tareas que asignar al último procesador.

Caso 2: Las primeras $(m - \beta_{EDF})$ tareas tienen una utilización total mayor que $U_{wc}^{EDF-FF}(m - \beta_{EDF}, n - 1, \alpha)$, esto es,

$$\sum_{i=1}^{m-\beta_{EDF}} u_i > U_{wc}^{EDF-FF}(m - \beta_{EDF}, n - 1, \alpha)$$

En este caso se demostrará que el conjunto de las m tareas aún cabe en n procesa-


 Figura 3.2: Situación general en el *caso 2* del teorema 3.

dores si su utilización total es igual a

$$U_{wc}^{EDF-FF}(m - \beta_{EDF}, n - 1, \alpha) + \Delta$$

donde $\Delta \in \mathbb{R}$, y

$$\Delta \leq \frac{\beta_{EDF}}{\beta_{EDF} + 1}$$

Debe existir una tarea τ_k , cuyo factor de utilización u_k , sumado a los previos factores de utilización u_i , hace que el límite $U_{wc}^{EDF-FF}(m - \beta_{EDF}, n - 1, \alpha)$ se supere. Esta situación se muestra en la figura 3.2, la cual representa gráficamente los factores de utilización de todas las tareas y las relaciones que existen entre diferentes cantidades y sumatorios empleados a lo largo de la demostración. El valor del subíndice k se obtiene como el entero que cumple la siguiente condición:

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

Debe notarse que $k \leq m - \beta_{EDF}$, pues de otra forma se estaría en el *caso 1*.

Se puede demostrar que las primeras $(k - 1)$ tareas caben en los primeros $(n - 1)$ procesadores. La utilización total de las primeras $(k - 1)$ tareas cumple

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

Aplicando el lema 3 con $(m - \beta_{EDF}) > (k - 1)$ y con RA=FF, se obtiene

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

y por lo tanto

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

De esta forma, las primeras $(k-1)$ tareas caben en los $(n-1)$ primeros procesadores. Por lo tanto, empleando el algoritmo de asignación FF estas primeras $(k-1)$ tareas se asignan a los $(n-1)$ primeros procesadores.

Falta por demostrar que las restantes $(m-k+1)$ tareas se pueden asignar a alguno de los huecos que quedan en los primeros $(n-1)$ procesadores, o bien al último procesador que de momento está vacío.

Analicemos la situación actual, representada en la figura 3.2. Se tienen $(m-k+1)$ tareas de factores de utilización comprendidos entre cero y α . Se tiene que $(m-k+1) > \beta_{EDF}$, pues en caso contrario se estaría en el *caso 1*. La utilización total de estas $(m-k+1)$ tareas es $u_{k,1} + \Delta$, y por lo tanto es menor o igual que $u_{k,1} + \beta_{EDF}/(\beta_{EDF} + 1)$. El valor de $u_{k,1}$ es menor que α , pues $u_{k,1} < u_k \leq \alpha$. Se dispone de huecos en los primeros $(n-1)$ procesadores suficientes para albergar al menos $(m-k+1 - \beta_{EDF})$ tareas de utilización total menor o igual que $u_{k,1}$, además de un procesador, el último, completamente vacío.

Bajo estas restricciones debe demostrarse que los huecos disponibles en los primeros $(n-1)$ procesadores, y el último procesador proporcionan espacio suficiente para albergar las últimas $(m-k+1)$ tareas.

Para realizar la demostración el conjunto de todas las posibilidades definido por las restricciones anteriores se reducirá, introduciendo una restricción adicional que llevará al caso más desfavorable. La restricción adicional consiste en considerar que ninguna de las últimas $(m-k+1)$ tareas se asigna a alguno de los $(n-1)$ primeros procesadores. Este resultado es intuitivamente claro, pero se puede demostrar de una manera formal. Supóngase que alguna de las $(m-k+1)$ tareas se asignase a uno de los $(n-1)$ primeros procesadores, habría dos posibilidades. En la primera posibilidad, el nuevo número de tareas sin asignar, esto es $(m-k)$, es igual a β_{EDF} por lo que las $(m-k)$ tareas caben en el último procesador y finaliza la demostración. La segunda posibilidad proporciona el caso más desfavorable, en el cual $(m-k)$ es mayor que β_{EDF} . En este último caso la situación es análoga a la representada en la figura 3.2, pero con valores de $u_{k,1}$ y Δ aún más restringidos, y con k una unidad mayor. Este razonamiento se puede repetir con todas las tareas en $\{\tau_k, \dots, \tau_m\}$ que se asignasen a los $(n-1)$ primeros procesadores obteniéndose que el conjunto de tareas cabe en los procesadores, o bien se obtiene un nuevo conjunto $\{\tau_k, \dots, \tau_m\}$ análogo al de la figura 3.2 en el que ninguna de estas tareas se asigna a alguno de los $(n-1)$ primeros procesadores.

La consideración del caso más desfavorable descrito por la restricción anterior tiene efecto sobre los factores de utilización de las tareas $\{\tau_k, \dots, \tau_m\}$. Estos factores de utilización deben ser mayores que $u_{k,1}$. La demostración es simple, en caso contrario el algoritmo FF asignaría alguna de estas tareas a alguno de los primeros $(n-1)$ procesadores incumpliendo la restricción del caso peor impuesta.

Para probar que las últimas $(m-k+1)$ tareas caben en el último procesador se debe probar que la utilización total de esas tareas es menor o igual que uno, esto es,

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

La figura 3.2 muestra que

$$\sum_{i=k}^m u_i = u_{k,1} + \Delta \quad (3.11)$$

Como se comentó anteriormente, todos los factores de utilización en ese sumatorio son mayores que o iguales a $u_{k,1}$, y por lo tanto

$$\begin{aligned} (m - k + 1)u_{k,1} &< u_{k,1} + \Delta \\ &< u_{k,1} + \frac{\beta_{EDF}}{\beta_{EDF} + 1} \quad \text{por definición de } \Delta \end{aligned}$$

Despejando $u_{k,1}$.

$$u_{k,1} < \frac{\beta_{EDF}}{(\beta_{EDF} + 1)(m - k)} \quad (3.12)$$

Sustituyendo el valor de $u_{k,1}$ dado por la ecuación (3.12) en la ecuación (3.11) se obtiene

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

Se sabe que $k \leq (m - \beta_{EDF})$ en el caso 2, y por lo tanto $(m - k) \geq \beta_{EDF}$. Finalmente se obtiene

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

Esta ecuación muestra que las últimas $(m - k + 1)$ tareas cumplen la condición de planificabilidad de EDF, luego caben en el último procesador.

Se ha probado que cualquier conjunto de m tareas y utilización total

$$\begin{aligned} U_{wc}^{\text{EDF-FF}}(m - \beta_{EDF}, n - 1, \alpha) + \Delta &\leq \\ U_{wc}^{\text{EDF-FF}}(m - \beta_{EDF}, n - 1, \alpha) + \frac{\beta_{EDF}}{\beta_{EDF} + 1} & \end{aligned}$$

cabe en n procesadores, de tal forma que el límite de utilización $U_{wc}^{\text{EDF-FF}}(m, n, \alpha)$ debe cumplir

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

El límite de utilización para planificación EDF y asignación de tareas FF se obtiene a partir de los teoremas 2 y 3, tal como muestra el teorema 4.

Teorema 4 *Si $m > \beta_{EDF}n$ se cumple que el límite de utilización para EDF-FF es una función única y exclusivamente de n y β_{EDF} , y viene dado por*

$$U_{wc}^{EDF-FF}(n, \beta_{EDF}) = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \quad (3.13)$$

Demostración:

En primer lugar se obtendrá una cota inferior del límite de utilización para un conjunto de m tareas sobre un multiprocesador con n procesadores.

El teorema 3 relaciona el límite de utilización de conjuntos de m tareas en multiprocesadores de n procesadores, con el límite de utilización de conjuntos de $(m - \beta_{EDF})$ tareas en multiprocesadores de $(n - 1)$ procesadores.

$$U_{wc}^{EDF-FF}(m, n, \alpha) \geq \frac{\beta_{EDF}}{\beta_{EDF} + 1} + U_{wc}^{EDF-FF}(m - \beta_{EDF}, n - 1, \alpha) \quad (3.14)$$

Pero el teorema 3 también relaciona el límite de utilización de conjuntos de $(m - \beta_{EDF})$ tareas en multiprocesadores de $(n - 1)$ procesadores, con el límite de utilización de conjuntos de $(m - 2\beta_{EDF})$ tareas en multiprocesadores de $(n - 2)$ procesadores.

$$U_{wc}^{EDF-FF}(m - \beta_{EDF}, n - 1, \alpha) \geq \frac{\beta_{EDF}}{\beta_{EDF} + 1} + U_{wc}^{EDF-FF}(m - 2\beta_{EDF}, n - 2, \alpha) \quad (3.15)$$

Sustituyendo la ecuación (3.15) en la ecuación (3.14) se obtiene

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

Este procedimiento puede repetirse, hasta finalmente relacionar el límite de utilización de conjuntos de m tareas en multiprocesadores de n procesadores, con el límite de utilización de conjuntos de $(m - (n - 1)\beta_{EDF})$ tareas en un monoprocesador.

$$U_{wc}^{EDF-FF}(m, n, \alpha) \geq \frac{(n - 1)\beta_{EDF}}{\beta_{EDF} + 1} + U_{wc}^{EDF-FF}(m - (n - 1)\beta_{EDF}, 1, \alpha) \quad (3.16)$$

El límite de utilización para $(m - (n - 1)\beta_{EDF})$ tareas y un procesador es uno, independientemente del valor de α .

$$U_{wc}^{EDF-FF}(m - (n - 1)\beta_{EDF}, 1, \alpha) = 1 \quad (3.17)$$

Sustituyendo la ecuación (3.17) en la ecuación (3.16) se obtiene una cota inferior del límite de utilización de m tareas en n procesadores.

$$U_{wc}^{\text{EDF-FF}}(m, n, \alpha) \geq \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \quad (3.18)$$

El teorema 2 demostró que

$$U_{wc}^{\text{EDF-AA}}(m, n, \alpha) \leq \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \quad (3.19)$$

donde AA es un algoritmo de asignación cualquiera. Por lo tanto, es aplicable al algoritmo de asignación FF. A partir de la ecuación (3.18) y de la ecuación (3.19) se tiene

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

Se observa que $U_{wc}^{\text{EDF-FF}}$ sólo depende del número de procesadores, n , y del valor de β_{EDF} , luego finalmente se concluye con el resultado

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

■

El límite de utilización empleando planificación EDF-FF es el más alto posible de entre todos los algoritmos de asignación de tareas, pues coincide con la cota superior proporcionada por el teorema 2.

3.2.3 Límite de utilización para EDF-BF

El algoritmo de asignación *Best Fit* asigna cada tarea al procesador con menor capacidad residual, pero suficiente para albergar la tarea. Si dos o más procesadores tienen la misma capacidad residual se elige entre estos empleando el algoritmo FF, recorriendo los procesadores en el sentido $\{P_1, \dots, P_n\}$. La descripción detallada de este algoritmo de asignación de tareas puede encontrarse en el apartado 2.1.1.

La demostración del límite de utilización para planificación EDF-BF es análoga a la vista en el apartado 3.2.2 para la planificación EDF-FF.

Teorema 5 *Si $m > \beta_{EDF}n$ se cumple*

$$U_{wc}^{\text{EDF-BF}}(m, n, \alpha) \geq \frac{\beta_{EDF}}{\beta_{EDF} + 1} + U_{wc}^{\text{EDF-BF}}(m - \beta_{EDF}, n - 1, \alpha)$$

Demostración

Se probará que cualquier conjunto de m tareas $\{\tau_1, \dots, \tau_m\}$, con factores de utilización $0 < u_i \leq \alpha$ para $i = 1, \dots, m$, y utilización total menor o igual que

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

cabe en n procesadores empleando planificación EDF en cada procesador y asignación de tareas BF.

Hay dos casos posibles:

Caso 1: Las primeras $(m - \beta_{EDF})$ tareas tienen utilización total menor o igual que $U_{wc}^{EDF-BF}(m - \beta_{EDF}, n - 1, \alpha)$, esto es

$$\sum_{i=1}^{m-\beta_{EDF}} u_i \leq U_{wc}^{EDF-BF}(m - \beta_{EDF}, n - 1, \alpha)$$

Las primeras $(m - \beta_{EDF})$ tareas caben en los primeros $(n - 1)$ procesadores, pues su utilización total está por debajo del límite de utilización. Por lo tanto, empleando el algoritmo BF se asignan a los $(n - 1)$ primeros procesadores. Las restantes β_{EDF} tareas caben en el último procesador, pues la definición de β_{EDF} implica que al menos β_{EDF} tareas siempre caben en un procesador. El hecho de que quepan en el último procesador no significa que se asignen al último procesador. De todas formas, el caso en el cual alguna de estas β_{EDF} se asignase a alguno de los primeros $(n - 1)$ procesadores es más favorable, pues habría menos tareas que asignar al último procesador.

Caso 2: Las primeras $(m - \beta_{EDF})$ tareas tienen una utilización total mayor que $U_{wc}^{EDF-BF}(m - \beta_{EDF}, n - 1, \alpha)$, esto es,

$$\sum_{i=1}^{m-\beta_{EDF}} u_i > U_{wc}^{EDF-BF}(m - \beta_{EDF}, n - 1, \alpha)$$

En este caso se demostrará que el conjunto de las m tareas aún cabe en n procesadores si su utilización total es igual a

$$U_{wc}^{EDF-BF}(m - \beta_{EDF}, n - 1, \alpha) + \Delta$$

donde $\Delta \in \mathbb{R}$, y

$$\Delta \leq \frac{\beta_{EDF}}{\beta_{EDF} + 1}$$

Debe existir una tarea τ_k , cuyo factor de utilización u_k , sumado a los previos factores de utilización u_i , hace que el límite $U_{wc}^{EDF-BF}(m - \beta_{EDF}, n - 1, \alpha)$ se supere. Esta situación se muestra en la figura 3.3, la cual representa gráficamente los factores de utilización de todas las tareas y las relaciones que existen entre diferentes cantidades y sumatorios empleados a lo largo de la demostración. El valor del subíndice k se obtiene como el entero que cumple la siguiente condición:

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

Debe notarse que $k \leq m - \beta_{EDF}$, pues de otra forma se estaría en el *caso 1*.

Se puede demostrar que las primeras $(k - 1)$ tareas caben en los primeros $(n - 1)$

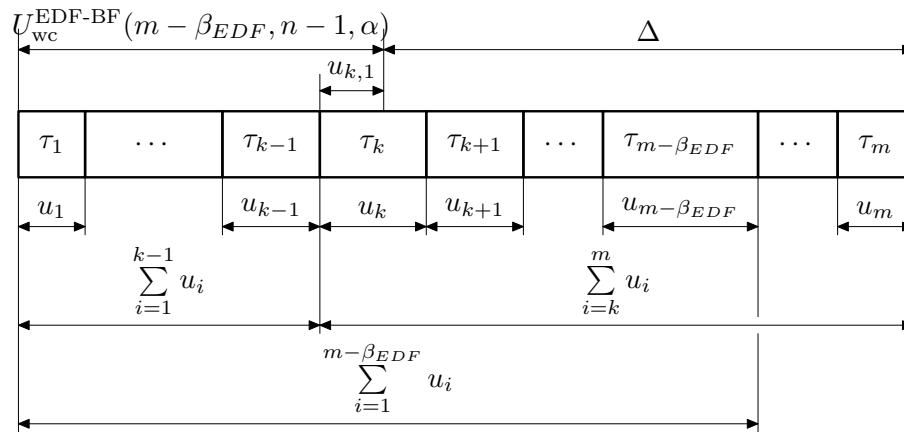


Figura 3.3: Situación general en el *caso 2* del teorema 5.

procesadores. La utilización total de las primeras $(k - 1)$ tareas cumple

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

Aplicando el lema 3 con $(m - \beta_{EDF}) > (k - 1)$ y con RA=BF, se obtiene

$$U_{wc}^{EDF-BF}(m - \beta_{EDF}, n - 1, \alpha) \leq U_{wc}^{EDF-BF}(k - 1, n - 1, \alpha)$$

y por lo tanto

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

De esta forma, las primeras $(k - 1)$ tareas caben en los $(n - 1)$ primeros procesadores. Por lo tanto, empleando el algoritmo de asignación BF estas primeras $(k - 1)$ tareas se asignan a los $(n - 1)$ primeros procesadores.

Falta por demostrar que las restantes $(m - k + 1)$ tareas se pueden asignar a alguno de los huecos que quedan en los primeros $(n - 1)$ procesadores, o bien al último procesador que de momento está vacío.

Analicemos la situación actual, representada en la figura 3.3. Se tienen $(m - k + 1)$ tareas de factores de utilización comprendidos entre cero y α . Se tiene que $(m - k + 1) > \beta_{EDF}$, pues en caso contrario se estaría en el *caso 1*. La utilización total de estas $(m - k + 1)$ tareas es $u_{k,1} + \Delta$, y por lo tanto es menor o igual que $u_{k,1} + \beta_{EDF}/(\beta_{EDF} + 1)$. El valor de $u_{k,1}$ es menor que α , pues $u_{k,1} < u_k \leq \alpha$. Se dispone de huecos en los primeros $(n - 1)$ procesadores suficientes para albergar al menos $(m - k + 1 - \beta_{EDF})$ tareas de utilización total menor o igual que $u_{k,1}$, además de un procesador, el último, completamente vacío.

Bajo estas restricciones debe demostrarse que los huecos disponibles en los primeros $(n - 1)$ procesadores, y el último procesador proporcionan espacio suficiente para albergar las últimas $(m - k + 1)$ tareas.

Para realizar la demostración el conjunto de todas las posibilidades definido

por las restricciones anteriores se reducirá, introduciendo restricciones adicionales que llevarán al caso más desfavorable. Estas restricciones adicionales se indican a continuación:

- Se tiene un único procesador entre los $(n - 1)$ primeros con capacidad residual $u_{k,1}$ capaz de albergar hasta $(m - k + 1 - \beta_{EDF})$ tareas. Los demás procesadores de entre los $(n - 1)$ primeros están completamente llenos, es decir, su capacidad residual es cero.
- Ninguna de las últimas $(m - k + 1)$ tareas se asigna a alguno de los $(n - 1)$ primeros procesadores. Este resultado es intuitivamente claro, pero se puede demostrar de una manera formal. Supóngase que alguna de las $(m - k + 1)$ tareas se asignase a uno de los $(n - 1)$ primeros procesadores, habría dos posibilidades. En la primera posibilidad, el nuevo número de tareas sin asignar, esto es $(m - k)$, es igual a β_{EDF} por lo que las $(m - k)$ tareas caben en el último procesador y finaliza la demostración. La segunda posibilidad proporciona el caso más desfavorable, en el cual $(m - k)$ es mayor que β_{EDF} . En este último caso la situación es análoga a la representada en la figura 3.3, pero con valores de $u_{k,1}$ y Δ aún más restringidos, y con k una unidad mayor. Este razonamiento se puede repetir con todas las tareas en $\{\tau_k, \dots, \tau_m\}$ que se asignasen a los $(n - 1)$ primeros procesadores obteniéndose que el conjunto de tareas cabe en los procesadores, o bien un nuevo conjunto $\{\tau_k, \dots, \tau_m\}$ análogo al de la figura 3.3 en el que ninguna de estas tareas se asigna a alguno de los $(n - 1)$ primeros procesadores.

La consideración del caso más desfavorable descrito por las restricciones anteriores tiene efecto sobre los factores de utilización de las tareas $\{\tau_k, \dots, \tau_m\}$. Se demostrará que con asignación BF se cumple en el caso más desfavorable que el factor de utilización de las tareas $\{\tau_k, \dots, \tau_{k+\beta_{EDF}-1}\}$ es mayor que el valor $u_{k,1}$.

La demostración se realizará por reducción al absurdo. Para ello, sea τ_l la primera tarea en $\{\tau_k, \dots, \tau_{k+\beta_{EDF}-1}\}$ de factor de utilización $u_l \leq u_{k,1}$. La única posibilidad para que τ_l no se asigne a los $(n - 1)$ primeros procesadores, tal como sucede en el caso peor, es que la capacidad residual del último procesador sea menor que $u_{k,1}$ una vez asignadas a este procesador las tareas $\{\tau_k, \dots, \tau_{l-1}\}$. Esto es, debe cumplirse

$$1 - \sum_{i=k}^{l-1} u_i < u_{k,1} \quad (3.20)$$

Teniendo en cuenta que $u_i \leq \alpha$ y que $l \leq (k + \beta_{EDF} - 1)$ se deduce

$$1 - \sum_{i=k}^{l-1} u_i \geq 1 - (l - k)\alpha \geq 1 - (\beta_{EDF} - 1)\alpha$$

Por definición de β_{EDF} se tiene $\beta_{EDF} = \lfloor 1/\alpha \rfloor$, y por lo tanto $\beta_{EDF} \leq 1/\alpha$. Sustit-

tuyendo esta relación en la expresión anterior se tiene

$$1 - \sum_{i=k}^{l-1} u_i \geq \alpha$$

Como $u_{k,1} < \alpha$ resulta que

$$1 - \sum_{i=k}^{l-1} u_i > u_{k,1}$$

lo cual contradice la ecuación (3.20).

Una vez se ha demostrado que el factor de utilización de las primeras β_{EDF} tareas del subconjunto $\{\tau_k, \dots, \tau_m\}$ es mayor que $u_{k,1}$, la demostración de que las tareas $\{\tau_k, \dots, \tau_m\}$ caben en el último procesador es relativamente simple.

Si $u_{k,1}$ fuese mayor que $1/(\beta_{EDF} + 1)$, entonces las primeras β_{EDF} tareas de $\{\tau_k, \dots, \tau_m\}$ acumularían una utilización total de valor

$$\sum_{i=k}^{k+\beta_{EDF}-1} u_i > \beta_{EDF} u_{k,1} > \frac{\beta_{EDF}}{\beta_{EDF} + 1}$$

Se deduciría entonces que la utilización total de las tareas $\{\tau_{k+\beta_{EDF}}, \dots, \tau_m\}$ es menor que $u_{k,1}$ por lo que tienen que caber en el último procesador junto con las tareas $\{\tau_k, \dots, \tau_{k+\beta_{EDF}-1}\}$. Si no cupiesen en el último procesador, alguna de estas tareas se asignaría a los $(n-1)$ primeros procesadores, lo cual contradice la hipótesis del caso peor en el que todas las tareas $\{\tau_k, \dots, \tau_m\}$ se asignan al último procesador.

Por lo tanto en el caso más desfavorable $u_{k,1} \leq 1/(\beta_{EDF} + 1)$. Como la utilización total de las tareas $\{\tau_k, \dots, \tau_m\}$ es igual a $u_{k,1} + \Delta$, y $\Delta < \beta_{EDF}/(\beta_{EDF} + 1)$, resulta que la utilización total de estas tareas es menor o igual que uno. Por lo tanto, las tareas $\{\tau_k, \dots, \tau_m\}$ caben en el último procesador.

Se ha probado que cualquier conjunto de m tareas y utilización total

$$U_{wc}^{EDF-BF}(m - \beta_{EDF}, n - 1, \alpha) + \Delta \leq U_{wc}^{EDF-BF}(m - \beta_{EDF}, n - 1, \alpha) + \frac{\beta_{EDF}}{\beta_{EDF} + 1}$$

cabe en n procesadores, de tal forma que el límite de utilización $U_{wc}^{EDF-BF}(m, n, \alpha)$ debe cumplir

$$U_{wc}^{EDF-BF}(m, n, \alpha) \geq \frac{\beta_{EDF}}{\beta_{EDF} + 1} + U_{wc}^{EDF-BF}(m - \beta_{EDF}, n - 1, \alpha)$$

■

Teorema 6 Si $m > \beta_{EDF}n$ se cumple que el límite de utilización para EDF-BF es una función única y exclusivamente de n y β_{EDF} , y viene dado por

$$U_{wc}^{EDF-BF}(n, \beta_{EDF}) = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \quad (3.21)$$

Demostración:

La demostración es idéntica a la del teorema 4. Es suficiente cambiar las iniciales FF por las iniciales BF. ■

3.2.4 Límite de utilización para EDF-WF

El algoritmo de asignación *Worst Fit* (WF) asigna cada tarea al procesador con mayor capacidad residual, suficiente para albergar la tarea. Si dos o más procesadores tienen la misma capacidad residual se elige entre estos empleando el algoritmo FF, recorriendo los procesadores en el sentido $\{P_1, \dots, P_n\}$. La descripción detallada de este algoritmo de asignación de tareas puede encontrarse en el apartado 2.1.1.

Para obtener el límite de utilización U_{wc}^{EDF-WF} se obtendrá en primer lugar una cota superior de dicho límite de utilización.

Esta cota superior se verá que coincide con la cota inferior que proporciona el teorema 1 para cualquier algoritmo razonable de asignación de tareas a procesadores. Como el algoritmo de asignación WF es razonable, resulta que el límite de utilización asociado al mismo coincide con la cota inferior dada por el teorema 1, tal como indica el corolario 1.

Teorema 7 Si $m > \beta_{EDF}n$ se cumple

$$U_{wc}^{EDF-WF}(m, n, \alpha) \leq n - (n - 1)\alpha$$

Demostración:

Se demostrará que existe un conjunto de m tareas $\{\tau_1, \dots, \tau_m\}$, de factores de utilización menores o iguales que α , y utilización total

$$U = n - (n - 1)\alpha + \epsilon$$

con $\epsilon \rightarrow 0^+$, el cual no cabe en n procesadores empleando el algoritmo WF de asignación de tareas a procesadores.

Se construirá este conjunto de m tareas a partir de tres subconjuntos: un primer subconjunto de $(m - \beta_{EDF}n - 1)$ tareas, un segundo subconjunto de $\beta_{EDF}n$ tareas, y un tercer subconjunto formado por una única tarea.

Todas las tareas del primer subconjunto tienen el mismo factor de utilización de valor

$$u_i = \frac{\epsilon}{2(m - \beta_{EDF}n - 1)}$$

con $i = 1, \dots, (m - \beta_{EDF}n - 1)$.

Todas las tareas del segundo subconjunto tienen el mismo factor de utilización de valor

$$u_i = \frac{1 - \alpha}{\beta_{EDF}} + \frac{\epsilon}{2\beta_{EDF}n}$$

con $i = (m - \beta_{EDF}n), \dots, (m - 1)$.

La tarea del tercer subconjunto tiene un factor de utilización de valor

$$u_m = \alpha$$

Se puede comprobar que la utilización total obtenida a partir de los tres subconjuntos anteriores es

$$U = n - (n - 1)\alpha + \epsilon$$

A continuación se comprobará que el conjunto de tareas propuesto, obtenido a partir de la unión de los subconjuntos anteriores, es válido. Esto es, se comprobará que

$$0 < u_i \leq \alpha$$

para $i = 1, \dots, m$.

Teniendo en cuenta que $0 < \alpha \leq 1$, $m > \beta_{EDF}n$, y $\beta_{EDF} \geq 1$ resulta que los factores de utilización de todas las tareas son mayores que cero. El factor de utilización de las tareas del primer y tercer subconjunto son obviamente menores o iguales que α eligiendo un ϵ positivo suficientemente pequeño. Para demostrar la validez del conjunto de tareas resta por demostrar que el factor de utilización de las tareas del segundo subconjunto es menor o igual que α .

El lema 2 proporcionaba la siguiente expresión que permitía calcular el valor de β_{EDF} a partir del valor de α

$$\beta_{EDF} = \lceil 1/\alpha \rceil$$

Por lo tanto se cumple

$$\beta_{EDF} > \frac{1}{\alpha} - 1 = \frac{1 - \alpha}{\alpha}$$

y

$$\frac{1 - \alpha}{\beta_{EDF}} < \alpha$$

Entre dos números reales siempre es posible encontrar otro número real, por lo tanto existe un ϵ positivo suficientemente pequeño tal que

$$\frac{1 - \alpha}{\beta_{EDF}} + \frac{\epsilon}{2\beta_{EDF}n} < \alpha$$

lo cual demuestra que el factor de utilización de las tareas del segundo subconjunto también es menor o igual que α .

Una vez demostrado que el conjunto de tareas propuesto es válido, se demostrará que no cabe en n procesadores empleando la asignación de tareas WF, y empleando planificación EDF en cada procesador.

Empleando el algoritmo de asignación WF, las tareas del primer subconjunto se

reparten de equitativamente entre los n procesadores. Equitativamente dentro de lo posible, pues el número de tareas de este subconjunto no tiene por que se un múltiplo de n . Estas tareas no tienen ninguna influencia sobre el resto de la demostración, por lo que pueden ignorarse. Las tareas de este subconjunto son necesarias únicamente dentro de la demostración para cumplir la restricción de que el conjunto de tareas esté formado por m tareas.

Aplicando el algoritmo WF, las tareas del segundo subconjunto se asignan de tal forma que cada procesador del multiprocesador recibe β_{EDF} tareas. Por lo tanto, la utilización total de cualquier procesador P_j es

$$U_j = 1 - \alpha + \frac{\epsilon}{2n}$$

La tarea del tercer subconjunto tiene un factor de utilización de valor α , la cual no puede ser asignada a ninguno de los procesadores pues entonces la utilización total del procesador P_j al cual se asignase pasaría a ser

$$U_j = 1 + \frac{\epsilon}{2n} > 1$$

Es decir, se ha encontrado un conjunto de m tareas válido de utilización total

$$U = n - (n - 1)\alpha + \epsilon$$

con $\epsilon \rightarrow 0$, el cual no cabe en n procesadores empleando asignación de tareas WF y planificación EDF en cada procesador. Como consecuencia, se cumple:

$$U_{wc}^{EDF-WF}(m, n, \alpha) \leq n - (n - 1)\alpha$$

■

Corolario 1 *Si $m > \beta_{EDF}n$ se cumple que el límite de utilización para EDF-WF es una función única y exclusivamente de n y α , y viene dado por*

$$U_{wc}^{EDF-WF}(n, \alpha) = n - (n - 1)\alpha \quad (3.22)$$

Demostración

A partir del teorema 7 se tiene

$$U_{wc}^{EDF-WF}(m, n, \alpha) \leq n - (n - 1)\alpha$$

Por otra parte, como WF es un algoritmo razonable de asignación de tareas, aplicando el teorema 1 se deduce

$$U_{wc}^{EDF-WF}(m, n, \alpha) \geq n - (n - 1)\alpha$$

y por lo tanto

$$U_{wc}^{EDF-WF}(m, n, \alpha) = n - (n - 1)\alpha$$

Se observa que U_{wc}^{EDF-WF} sólo depende del número de procesadores, n , y del parámetro

α , luego finalmente se concluye con el resultado

$$U_{wc}^{EDF-WF} = U_{wc}^{EDF-WF}(n, \alpha) = n - (n - 1)\alpha$$

■

El límite de utilización empleando planificación EDF-WF es el más bajo posible de entre todos los algoritmos razonables de asignación de tareas, pues coincide con la cota inferior proporcionada por el teorema 1.

3.2.5 Límite de utilización para EDF-RF

El algoritmo de asignación *Random Fit* (RF) asigna de forma aleatoria cada tarea entre todos los procesadores con capacidad residual suficiente para albergar la tarea. Este algoritmo es el peor de los algoritmos de asignación razonables en términos de límite utilización. La explicación es simple, la asignación de tareas a procesadores realizada por cualquier algoritmo razonable de asignación podría ser el resultado también de la asignación de tareas empleando el algoritmo RF. La probabilidad de que RF realice la misma asignación puede ser mayor o menor, pero en cualquier caso esa posibilidad siempre existe y debe ser tenida en cuenta.

A continuación, el teorema 8 proporciona el límite de utilización empleando planificación EDF-RF.

Teorema 8 *Si $m > \beta_{EDF}n$ se cumple que el límite de utilización para EDF-RF es una función única y exclusivamente de n y α , y viene dada por*

$$U_{wc}^{EDF-RF}(n, \alpha) = n - (n - 1)\alpha \quad (3.23)$$

Demostración

En el apartado 3.2.4 se obtuvo el límite de utilización empleando el algoritmo razonable de asignación de tareas WF. Como el límite de utilización empleando asignación RF no puede ser mayor que el límite de utilización empleando asignación WF resulta

$$U_{wc}^{EDF-RF}(m, n, \alpha) \leq U_{wc}^{EDF-WF} = n - (n - 1)\alpha \quad (3.24)$$

Sin embargo, a partir del teorema 1 se sabe que el límite de utilización asociado a todos los algoritmos razonables de asignación de tareas debe ser mayor o igual que

$$n - (n - 1)\alpha$$

Como RF es un algoritmo razonable de asignación de tareas, a partir del resultado anterior se tiene

$$U_{wc}^{EDF-RF}(m, n, \alpha) \geq n - (n - 1)\alpha \quad (3.25)$$

A partir de las ecuaciones (3.24) y (3.25) se deduce

$$U_{wc}^{EDF-RF}(m, n, \alpha) = n - (n - 1)\alpha$$

Se observa que U_{wc}^{EDF-RF} sólo depende del número de procesadores, n , y del valor de α , luego finalmente se concluye con el resultado

$$U_{wc}^{EDF-RF} = U_{wc}^{EDF-RF}(n, \alpha) = n - (n - 1)\alpha$$

■

El algoritmo de asignación RF puede emplearse para obtener los límites de utilización en el caso de la creación y destrucción dinámica de tareas que ocurre en los cambios de modo, tal como se indica en el apartado 6.5.

3.2.6 Límite de utilización para EDF-OPT

El algoritmo de asignación de tareas a procesadores denominado *Optimal* (OPT) es el algoritmo óptimo de asignación de tareas a procesadores. La obtención de la asignación óptima de tareas es análoga al problema denominado *bin-packing*. Este problema pertenece a la clase de problemas *NP-hard* en el sentido estricto, por lo que no puede resolverse en general en tiempo pseudopolinomial. Aunque la búsqueda de la asignación óptima de tareas a procesadores no resulta práctica, sirve como referencia para conocer lo lejos que está un algoritmo heurístico de asignación del algoritmo óptimo de asignación. Por ejemplo, si los resultados obtenidos para un determinado algoritmo de asignación son cercanos a los obtenidos por el algoritmo óptimo, entonces no merece la pena la búsqueda de algoritmos heurísticos más complejos pues la mejora que puede obtenerse es pequeña.

Una descripción más detallada del algoritmo de asignación OPT puede encontrarse en el capítulo 2.

El algoritmo de asignación OPT asigna de forma factible todos los conjuntos de tareas que pueden asignarse con otros algoritmos. Por lo tanto se deduce que el límite de utilización para EDF-OPT es mayor o igual que el límite de utilización empleando otros algoritmos de asignación y planificación EDF en cada procesador. En particular, el algoritmo FF proporciona el máximo límite de utilización de entre todos los algoritmos razonables de asignación. Como el algoritmo OPT es también razonable, resulta que el límite de utilización asociado al mismo debe coincidir con el límite de utilización asociado al algoritmo FF. Es decir,

$$U_{wc}^{EDF-OPT} = U_{wc}^{EDF-OPT}(n, \beta_{EDF}) = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \quad (3.26)$$

Esto no significa que los algoritmos FF y OPT sean capaces de asignar los mismos conjuntos de tareas, sino que comparten el caso peor. En realidad, hay conjuntos de tareas que pueden ser asignados empleando el algoritmo OPT pero no el FF. El resultado inverso no es cierto por definición del algoritmo óptimo de asignación. Por ejemplo, considérese el conjunto formado por cuatro tareas $\{\tau_1, \tau_2, \tau_3, \tau_4\}$, de factores de utilización $\{0'35, 0'35, 0'6, 0'6\}$. Este conjunto no puede asignarse de forma factible a dos procesadores empleando el algoritmo FF. Sin embargo el algoritmo OPT puede asignarlos de forma factible, asignando a cada procesador una tarea de factor de utilización 0'35 y otra de factor de utilización 0'6.

3.2.7 Límites de utilización empleando ordenación

Los algoritmos de asignación FF, BF, WF y RF no son más que distintas formas de elegir un procesador de entre todos los procesadores en los que cabe la tarea. Estos algoritmos no suponen una determinada ordenación de las tareas antes de hacer la asignación. Las tareas vienen definidas únicamente por sus factores de utilización, por lo que la ordenación de tareas consiste en la ordenación basada en factores de utilización. Se consideran dos tipos de ordenación:

- Factores de utilización crecientes. Este tipo de ordenación se denota por I (*Increasing*). Después de realizar la ordenación los factores de utilización de las tareas cumplen la propiedad:

$$u_1 \leq u_2 \leq \dots \leq u_{m-1} \leq u_m$$

donde τ_1 es la primera tarea que se asigna, y τ_m la última.

- Factores de utilización decrecientes. Este tipo de ordenación se denota por D (*Decreasing*). Después de realizar la ordenación los factores de utilización de las tareas cumplen la propiedad:

$$u_1 \geq u_2 \geq \dots \geq u_{m-1} \geq u_m$$

donde τ_1 es la primera tarea que se asigna, y τ_m la última.

Intuitivamente, para llenar un recipiente con objetos es preferible al principio situar en el mismo los objetos más grandes, y finalmente los más pequeños. Llevado esto a los algoritmos de asignación de tareas hace pensar que lo mejor es ordenar las tareas en sentido de factores de utilización decrecientes antes de realizar la asignación. Sin embargo la intuición a veces “nos engaña” por lo que debe comprobarse este punto.

Si se considera ningún tipo de ordenación los algoritmos de asignación conservan los nombres anteriores, esto es, FF, BF, WF y RF. No obstante si llevan a cabo un determinado tipo de ordenación antes de hacer la asignación llevan una letra adicional que hace referencia a la misma. Para ordenación en sentido de factores de utilización crecientes se tienen los algoritmos FFI, BFI, WFI y RFI. Para ordenación en sentido decreciente de factores de utilización se tienen los algoritmos FFD, BFD, WFD y RFD.

En general, la ordenación de tareas tiene efecto sobre los límites de utilización, tal como se muestra en este apartado y en el apartado 3.4. Sin embargo hay un precio a pagar. En primer lugar, el algoritmo de ordenación tiene una complejidad $O(m \log m)$, donde m es el número de tareas. Este algoritmo tiene en general un coste computacional no despreciable, aunque asumible en muchos casos. En segundo lugar, el proceso de ordenación es incompatible con una asignación en línea de tareas a procesadores. En la asignación en línea no se conoce a priori el conjunto completo de tareas a asignar, por lo que cada vez que llega una nueva tarea, el proceso de ordenación puede dar lugar a que una tarea previamente asignada a un procesador

deba reasignarse a otro procesador. Esto va en contra de la filosofía de la planificación multiprocesador con particionado, donde se trata de evitar la migración de tareas entre procesadores. La asignación en línea de tareas plantea además otros problemas que se tratan en el apartado 6.5.

A continuación se muestra el efecto de la ordenación de tareas sobre los límites de utilización. Para ello se analizan de forma sucesiva los algoritmos derivados del FF, los derivados del BF, los derivados del WF, y finalmente los derivados del RF.

Deliberadamente se ha excluido el análisis de los algoritmos derivados del OPT. La razón es simple, el algoritmo OPT es óptimo sean cuales sean los factores de utilización de la tareas, e independientemente de si están ordenados o no.

Algoritmos de asignación FFD y FFI

Si el límite de utilización asociado a un planificador monoprocesador dado y a un algoritmo de asignación dado existe, este se puede obtener restándole un valor arbitrariamente pequeño a la mínima utilización total evaluada entre los conjuntos de tareas que no son planificables.

La idea es clara. Todos los conjuntos de tareas de utilización total por debajo del mínimo son planificables. Además, siempre es posible encontrar un conjunto de tareas de utilización total mayor que el mínimo que no es planificable.

Si de alguna manera se restringen los conjuntos de tareas sobre los cuales se evalúa el mínimo, este mínimo aumenta, o al menos no decrece.

Todo proceso de ordenación de las tareas restringe los conjuntos posibles que se pueden asignar con respecto al caso general en que no se realiza la ordenación. Cuando no se realiza la ordenación el mínimo anterior se evalúa sobre los mismos conjuntos de tareas que en el caso de existir ordenación, e incluso otros.

La planificación EDF-FF proporciona el límite de utilización más alto posible, tal como se deduce de los teoremas 2 y 4. Como por otra parte el límite de utilización para EDF-FFD no puede ser menor al restringirse los conjuntos de tareas posibles, el límite de utilización para EDF-FFD coincide con el límite de utilización para EDF-FF. Esto es,

$$U_{wc}^{EDF-FFD} = U_{wc}^{EDF-FFD}(n, \beta_{EDF}) = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \quad (3.27)$$

El razonamiento empleado para obtener el límite de utilización para planificación EDF-FFD podría emplearse para planificación EDF-FFI, obteniéndose el mismo resultado.

$$U_{wc}^{EDF-FFI} = U_{wc}^{EDF-FFI}(n, \beta_{EDF}) = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \quad (3.28)$$

Algoritmos de asignación BFD y BFI

El límite de utilización para EDF-BF coincide con el límite de utilización para EDF-FF tal como muestran los teoremas 4 y 6. Por lo tanto, aplicando razonamientos

análogos a los utilizados en el apartado anterior para planificación EDF-FFD y EDF-FFI se deduce

$$U_{wc}^{EDF-BFD} = U_{wc}^{EDF-BFD}(n, \beta_{EDF}) = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \quad (3.29)$$

y

$$U_{wc}^{EDF-BFI} = U_{wc}^{EDF-BFI}(n, \beta_{EDF}) = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \quad (3.30)$$

Algoritmos de asignación WFD y WFI

En primer lugar se obtiene el límite de utilización para EDF-WFI, y se dedicará el resto de este apartado a la obtención del límite de utilización para EDF-WFD.

En el teorema 7 se demostró que

$$U_{wc}^{EDF-WF}(n, \alpha) \leq n - (n - 1)\alpha$$

para lo cual se empleó un conjunto de tareas dividido en tres subconjuntos. Un primer subconjunto formado por $(m - \beta_{EDF}n - 1)$ tareas de factor de utilización

$$u_i = \frac{\epsilon}{2(m - \beta_{EDF}n - 1)}$$

para $i = 1, \dots, (m - \beta_{EDF}n - 1)$.

Un segundo subconjunto de $\beta_{EDF}n$ tareas de factor de utilización

$$u_i = \frac{1 - \alpha}{\beta_{EDF}} + \frac{\epsilon}{2\beta_{EDF}n}$$

con $i = (m - \beta_{EDF}n), \dots, (m - 1)$, y un tercer subconjunto formado por una única tarea de factor de utilización

$$u_m = \alpha$$

La utilización total obtenida a partir de los tres subconjuntos anteriores es

$$U = n - (n - 1)\alpha + \epsilon$$

Como se puede observar, los factores de utilización de las tareas del conjunto completo están ordenadas en sentido creciente de utilidades cuando $\epsilon \rightarrow 0$. Éste conjunto de tareas no cabía en n procesadores empleando la asignación de tareas WF. Por lo tanto se deduce

$$U_{wc}^{EDF-WFI}(m, n, \alpha) \leq n - (n - 1)\alpha$$

Obsérvese como la expresión del límite de utilización para EDF-WFI depende en principio de m , n , y α . A lo sumo depende de los tres parámetros, de ahí que a priori no se descarte ningún parámetro.

El teorema 1 demostraba que el límite de utilización empleando planificación

EDF y un algoritmo de asignación razonable arbitrario RA verificaba

$$U_{wc}^{EDF-RA}(m, n, \alpha) \geq n - (n - 1)\alpha$$

WFI es un algoritmo razonable de asignación, y por lo tanto

$$U_{wc}^{EDF-WFI}(m, n, \alpha) \geq n - (n - 1)\alpha$$

De tal forma que finalmente se concluye con el resultado

$$U_{wc}^{EDF-WFI} = U_{wc}^{EDF-WFI}(n, \alpha) = n - (n - 1)\alpha \quad (3.31)$$

donde se observa que el límite de utilización para EDF-WFI depende únicamente del número de procesadores y del máximo factor de utilización alcanzable.

Una vez calculado el límite de utilización para EDF-WFI se calculará a continuación el límite de utilización para EDF-WFD. Sin embargo el proceso es algo más complejo.

En primer lugar se obtendrá una cota inferior del límite de utilización aplicable a todos los algoritmos razonables de asignación que emplean ordenación de factores de utilización en sentido decreciente.

Teorema 9 *Sea RA un algoritmo razonable de asignación de tareas a procesadores que no realiza internamente ninguna ordenación de los factores de utilización. Sea RAD el algoritmo de asignación de tareas a procesadores obtenido ordenando previamente las tareas en sentido decreciente de factores de utilización y aplicando posteriormente el algoritmo de asignación RA. Si $m > \beta_{EDF}n$ se cumple*

$$U_{wc}^{EDF-RAD} \geq \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \quad (3.32)$$

Demostración:

Para $n = 1$ se cumple de forma trivial el resultado anterior pues la ecuación (3.32) proporciona un límite de utilización de valor uno, el cual se corresponde con el límite de utilización para el caso monoprocesador. Durante la parte restante de la demostración se supone que $n \geq 2$.

Sea τ_k la primera tarea del conjunto de m tareas $\{\tau_1, \dots, \tau_m\}$ que no cabe en el multiprocesador. Como el algoritmo de asignación RA es razonable se cumple

$$U_j > 1 - u_k \quad (3.33)$$

para $j = 1, \dots, m$. Donde U_j es la utilización total de las tareas previamente asignadas al procesador P_j , y u_k es el factor de utilización de la tarea τ_k .

La utilización total de las k primeras tareas del conjunto se puede obtener como

$$\sum_{i=1}^k u_i = \sum_{j=1}^n U_j + u_k \quad (3.34)$$

A partir de la ecuación (3.33) se deduce

$$\sum_{j=1}^n U_j > \sum_{j=1}^n (1 - u_k) = n(1 - u_k)$$

Sustituyendo esta expresión en la ecuación (3.34) resulta

$$\sum_{i=1}^k u_i > n(1 - u_k) + u_k = n - (n - 1)u_k \quad (3.35)$$

Al estar las tareas ordenadas en sentido de factores de utilización decrecientes resulta que el factor de utilización de la tarea τ_k es menor o igual que el factor de utilización de las tareas $\{\tau_1, \dots, \tau_{k-1}\}$. Por lo tanto se cumple:

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

Sustituyendo esta expresión en la ecuación (3.35) se obtiene

$$\sum_{i=1}^k u_i > n - (n - 1) \frac{\sum_{i=1}^k u_i}{k}$$

Despejando $\sum_{i=1}^k u_i$ en la ecuación anterior resulta

$$\sum_{i=1}^k u_i > \frac{kn}{k + n - 1}$$

Al ser τ_k una tarea del conjunto de m tareas resulta que la utilización total del conjunto de tareas denotada por U verifica

$$U = \sum_{i=1}^m u_i \geq \sum_{i=1}^k u_i > \frac{kn}{k + n - 1} \quad (3.36)$$

A continuación se calculará el valor mínimo que puede tomar la expresión

$$f(k, n) = \frac{kn}{k + n - 1} \quad (3.37)$$

cuando k varía dentro de su rango permitido. Si k fuese menor o igual que $\beta_{EDF}n$, entonces las k primeras tareas cabrían en los procesadores, lo que contradeciría la hipótesis de que la tarea τ_k no cabe en el multiprocesador. Por lo tanto, $k > \beta_{EDF}n$.

Se tiene que $k \leq m$. Por lo tanto k puede tomar cualquier valor entero dentro del rango $[\beta_{EDF}n + 1, m]$. Calculando la derivada primera con respecto a k de la función $f(k, n)$ se tiene

$$\frac{\partial f(k, n)}{\partial k} = \frac{n(n - 1)}{(k + n - 1)^2}$$

Esta derivada es positiva para $n \geq 2$, lo que se corresponde con una de las hipótesis (recuérdese que para $n = 1$ el teorema se demostraba de forma trivial). Por lo tanto la función $f(k, n)$ es monótona creciente en la variable k . El valor más pequeño que puede tomar esta función compatible con la restricción $k \in [\beta_{EDF}n + 1, m]$ se obtiene para $k = (\beta_{EDF}n + 1)$. Sustituyendo k por el valor $(\beta_{EDF}n + 1)$ en la ecuación (3.36) se obtiene

$$U > \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1}$$

Por lo tanto, una condición necesaria que debe cumplir un conjunto de tareas que no cabe en n procesadores empleando el algoritmo de asignación RAD y planificación EDF es que su utilización U cumpla

$$U > \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1}$$

de donde se deduce que todo conjunto de utilización U tal que

$$U \leq \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1}$$

cabe en n procesadores.

Debe tenerse en cuenta que no se ha hecho ninguna suposición con respecto al algoritmo de asignación de tareas a procesadores RAD, salvo que RA sea razonable, no realice internamente ordenación de tareas, y se realice la asignación RA justo después de la ordenación de las tareas en sentido de factores de utilización decrecientes. Por consiguiente se cumple

$$U_{wc}^{EDF-RAD} \geq \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1}$$

■

La restricción de que el algoritmo RA no realice ninguna ordenación interna de tareas trata de evitar la siguiente situación. Antes de realizar la asignación RA, el algoritmo de asignación RA internamente realiza una ordenación diferente a la ordenación en sentido de utilidades decrecientes, lo que anula la ordenación externa, previa al algoritmo RA.

Aplicando el teorema 2 al algoritmo de asignación RAD se tiene

$$U_{wc}^{EDF-RAD}(m, n, \alpha) \leq \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1}$$

A partir de este resultado y del teorema 9 se obtiene

$$U_{wc}^{EDF-RAD} = U_{wc}^{EDF-RAD}(n, \beta_{EDF}) = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \quad (3.38)$$

Lo cual demuestra que el límite de utilización es el mismo para todos los algoritmos de asignación obtenidos a partir de un algoritmo de asignación razonable sin

ordenación interna, añadiendo un paso previo de ordenación de tareas en sentido de factores de utilización decrecientes. Este límite de utilización común viene dado por la ecuación (3.38). Debe notarse además que este límite de utilización depende únicamente del número de procesadores y del parámetro β_{EDF} .

Como el algoritmo WF es razonable y no realiza internamente ninguna ordenación de tareas, para la planificación EDF-WFD se tiene

$$U_{wc}^{EDF-WFD} = U_{wc}^{EDF-WFD}(n, \beta_{EDF}) = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \quad (3.39)$$

Algoritmos de asignación RFD y RFI

El límite de utilización para EDF-RFD toma el valor

$$U_{wc}^{EDF-RFD} = U_{wc}^{EDF-RFD}(n, \beta_{EDF}) = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \quad (3.40)$$

dado por la ecuación (3.38), pues el algoritmo RF a partir del cual se obtiene el algoritmo RFD es razonable y no realiza ninguna ordenación de tareas internamente. Debe notarse que este límite de utilización depende únicamente del número de procesadores y del parámetro β_{EDF} .

El límite de utilización para planificación EDF-RFI se obtiene teniendo en cuenta que ha de ser el peor de entre todos los límites de utilización asociados a algoritmos razonables de asignación con ordenación previa de tareas en sentido de factores de utilización crecientes. En el apartado 3.2.7 se obtuvo

$$U_{wc}^{EDF-WFI}(n, \alpha) = n - (n - 1)\alpha$$

Por lo tanto empleando el algoritmo de asignación RFI se tiene

$$U_{wc}^{EDF-WFI}(m, n, \alpha) \leq U_{wc}^{EDF-WFI}(n, \alpha) = n - (n - 1)\alpha$$

Sin embargo, aplicando el teorema 1 no es posible encontrar ningún algoritmo razonable de asignación de tareas razonable con un límite de utilización menor que

$$n - (n - 1)\alpha$$

Como el algoritmo de asignación RFI es razonable (por ser RF razonable), se tiene finalmente

$$U_{wc}^{EDF-RFI} = U_{wc}^{EDF-RFI}(n, \alpha) = n - (n - 1)\alpha \quad (3.41)$$

Debe notarse además que este límite de utilización depende únicamente del número de procesadores y del máximo factor de utilización alcanzable.

3.3 Análisis de los resultados teóricos

En este apartado se analizan los resultados teóricos obtenidos en el apartado 3.2 bajo la hipótesis $m > \beta_{EDF}n$. Estos resultados son los siguientes:

- El límite de utilización U_{wc}^{EDF-RA} , asociado a la planificación EDF y a una asignación de tareas razonable cualquiera RA, se encuentra dentro del rango

$$L_{EDF}(n, \alpha) = n - (n - 1)\alpha \leq U_{wc}^{EDF-RA} \leq \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} = H_{EDF}(n, \beta_{EDF})$$

donde

$$\beta_{EDF} = \lfloor 1/\alpha \rfloor$$

α es el máximo factor de utilización alcanzable, y n es el número de procesadores que constituyen el multiprocesador.

- El límite de utilización $U_{wc}^{EDF-RAD}$, asociado a la planificación EDF y a una asignación de tareas RAD, toma el valor

$$U_{wc}^{EDF-RAD} = H_{EDF}(n, \beta_{EDF}) = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1}$$

donde RAD representa un algoritmo de asignación de tareas obtenido mediante ordenación previa de los factores de utilización en sentido decreciente, y posterior asignación empleando un algoritmo razonable cualquiera RA que no realice internamente ningún tipo de ordenación de tareas.

- El límite de utilización asociado a la planificación EDF, y a las asignaciones de tareas FF, FFD, FFI, BF, BFD, BFI, WFD, RFD y OPT toma el valor

$$H_{EDF}(n, \beta_{EDF}) = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1}$$

- El límite de utilización asociado a la planificación EDF, y a las asignaciones de tareas WF, WFI, RF, y RFI toma el valor

$$L_{EDF}(n, \alpha) = n - (n - 1)\alpha$$

Se puede ver que todos los resultados teóricos giran alrededor de dos expresiones matemáticas

$$L_{EDF}(n, \alpha) = n - (n - 1)\alpha$$

y

$$H_{EDF}(n, \beta_{EDF}) = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1}$$

que se analizarán en los dos apartados siguientes.

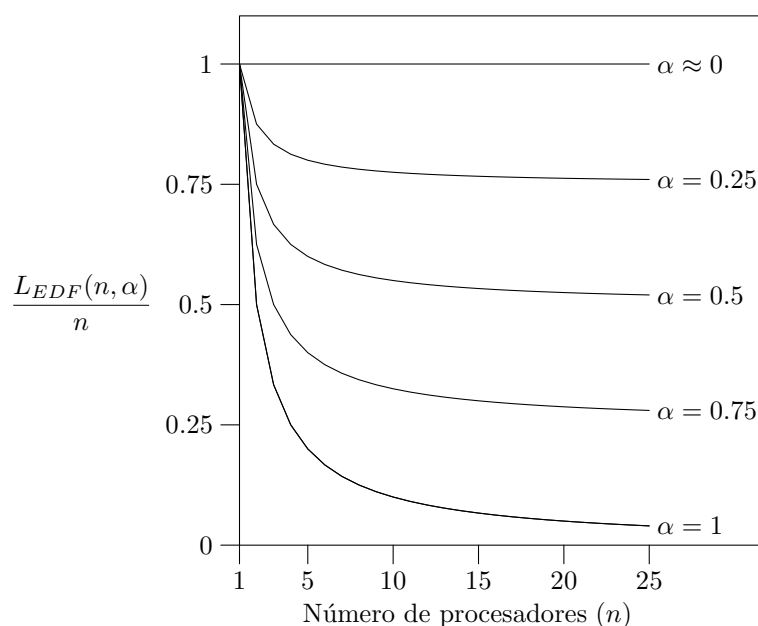


Figura 3.4: Representación del límite $L_{EDF}(n, \alpha)$.

3.3.1 Análisis del límite $L_{EDF}(n, \alpha)$

En este apartado se analiza la función $L_{EDF}(n, \alpha)$ que se corresponde con el límite de utilización empleando planificación EDF y asignación de tareas WF, WFI, RF y RFI. Además también coincide con el límite de utilización más bajo de entre todos los algoritmos de asignación razonables bajo planificación EDF.

La figura 3.4 representa la función $L_{EDF}(n, \alpha)$. La representación es continua a pesar de ser n una variable discreta, con el objetivo de mejorar la visualización. Además la representación ha sido normalizada dividiendo $L_{EDF}(n, \alpha)$ por el número de procesadores, mostrando así la utilización total promedio de los procesadores.

Cada vez que se añade un procesador más al sistema el valor de $L_{EDF}(n, \alpha)$ se incrementa en la cantidad $(1 - \alpha)$. Por lo tanto, dependiendo del valor de α este incremento varía en gran medida. Por ejemplo, en el caso más general se tiene $\alpha = 1$, lo que trae consigo un incremento nulo de $L_{EDF}(n, \alpha)$, se sumen los procesadores que se sumen. En este caso el límite de utilización coincide con el del monoprocesador y es por tanto igual a uno.

Esta desafortunada situación puede ilustrarse con un ejemplo. Considérese una asignación de tareas WF, para la cual se tiene un límite de utilización empleando planificación EDF igual a $L_{EDF}(n, \alpha)$. Considérese un conjunto de 20 tareas, de las cuales las 19 primeras tienen un factor de utilización $0'000001$, y la última un factor de utilización $1'0$. Este conjunto de tareas no cabe en ningún multiprocesador verificando $m > \beta n$, es decir, $n < 20$. La utilización de este conjunto de tareas es aproximadamente $1'0$, y sin embargo no cabe en 19 procesadores.

Si bien para $\alpha = 1$ se tiene una situación muy desafortunada, para $\alpha \rightarrow 0$ se tiene el mejor caso posible. Cualquier procesador que se añada al sistema incrementa el valor de $L_{EDF}(n, \alpha)$ en uno. La explicación es clara, cuando los factores de utilización

son muy pequeños se pueden llenar los procesadores casi hasta su capacidad. En este caso un multiprocesador de n procesadores se comporta como un monoprocesador n veces más rápido.

La función $L_{EDF}(n, \alpha)$ proporciona el límite de utilización empleando asignación WF, WFI, RF y RFI. Sin embargo, hay casos prácticos en los cuales se está interesado en el número mínimo de procesadores n_{wc}^{EDF} que aseguran la planificabilidad de un conjunto de m tareas de utilización total U y máximo factor de utilización alcanzable α . En el caso de los algoritmos de asignación cuyo límite de utilización asociado viene dado por $L_{EDF}(n, \alpha)$, el número mínimo de procesadores se representa por n_{wc}^{EDF-L} y se obtiene tal como se explica a continuación.

En la tesis se han presentado dos condiciones de planificabilidad complementarias empleando planificación EDF. La primera de ellas proviene de la definición de β_{EDF} dada en el apartado 3.2 (véase la figura 3.1)

$$m \leq \beta_{EDF} n_{wc}^{EDF-L}$$

Despejando n_{wc}^{EDF-L} y eligiendo el mínimo valor se obtiene

$$n_{wc}^{EDF-L} = \left\lceil \frac{m}{\beta_{EDF}} \right\rceil \quad (3.42)$$

La segunda condición de planificabilidad se deduce del empleo del límite de utilización

$$U \leq L_{EDF}(n_{wc}^{EDF-L}, \alpha) = n_{wc}^{EDF-L} - (n_{wc}^{EDF-L} - 1)\alpha$$

Despejando n_{wc}^{EDF-L} y eligiendo el mínimo valor se obtiene

$$n_{wc}^{EDF-L} = \left\lceil \frac{U - \alpha}{1 - \alpha} \right\rceil \quad (3.43)$$

La ecuación (3.43) no es válida para $U \leq \alpha$. Sin embargo, en este caso un único procesador es suficiente para planificar de forma factible el conjunto de tareas. Teniendo en cuenta esta observación, la ecuación (3.42) y la ecuación (3.43) se obtiene

$$n_{wc}^{EDF-L}(m, U, \alpha) = \begin{cases} 1 & \text{si } U \leq \alpha \\ \min \left\{ \left\lceil \frac{m}{\beta_{EDF}} \right\rceil, \left\lceil \frac{U - \alpha}{1 - \alpha} \right\rceil \right\} & \text{si } U > \alpha \end{cases} \quad (3.44)$$

Por ejemplo, para garantizar la planificabilidad de 100 tareas de utilización total 15, y factores de utilización menores o iguales que 0'25 (es decir $\beta_{EDF} = 4$) empleando asignación de tareas WF, WFI, RF o RFI y planificación EDF se necesitan 20 procesadores

$$n_{wc}^{EDF-L}(100, 15, 0'25) = \min \left\{ \left\lceil \frac{100}{4} \right\rceil, \left\lceil \frac{15 - 0'25}{1 - 0'25} \right\rceil \right\} = 20$$

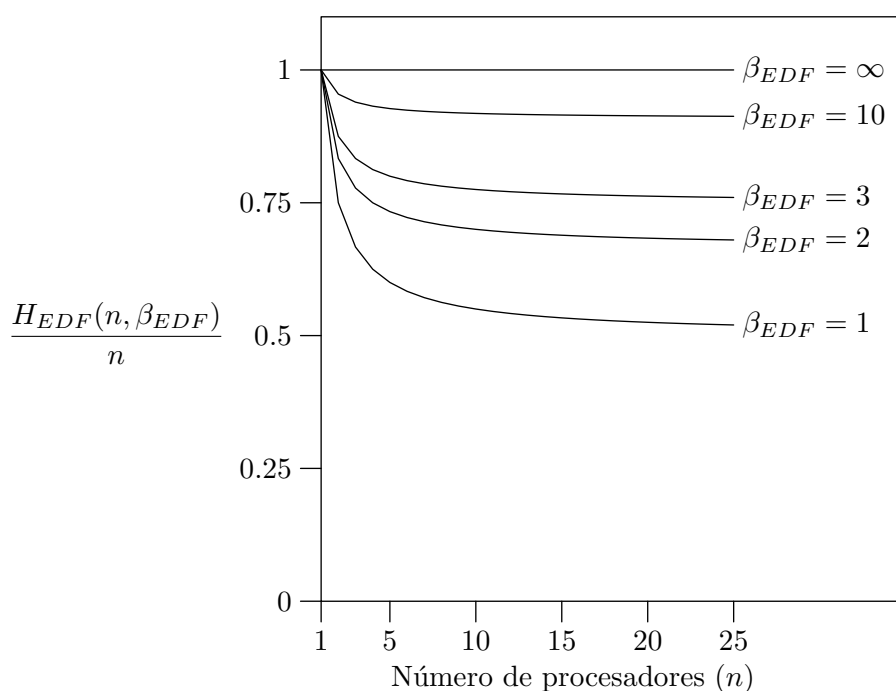


Figura 3.5: Representación del límite $H_{EDF}(n, \beta_{EDF})$.

3.3.2 Análisis del límite $H_{EDF}(n, \beta_{EDF})$

En este apartado se analiza la función $H_{EDF}(n, \beta_{EDF})$ que se corresponde con el límite de utilización empleando planificación EDF y asignación de tareas FF, FFD, FFI, BF, BFD, BFI, WFD, RFD y OPT. Además también coincide con el límite de utilización más alto de entre todos los algoritmos de asignación razonables bajo planificación EDF.

La figura 3.5 representa la función $H_{EDF}(n, \beta_{EDF})$. La representación es continua a pesar de ser n una variable discreta, con el objetivo de mejorar la visualización. Además la representación ha sido normalizada dividiendo $H_{EDF}(n, \beta_{EDF})$ por el número de procesadores, mostrando así la utilización total promedio de los procesadores.

Cada vez que se añade un procesador más al sistema el valor de $H_{EDF}(n, \beta_{EDF})$ se incrementa en la cantidad $\beta_{EDF}/(\beta_{EDF} + 1)$. Por lo tanto, dependiendo del valor de β_{EDF} este incremento varía en gran medida. Por ejemplo, en el caso más general se tiene $\beta_{EDF} = 1$, lo que trae consigo un incremento de 0'5 por procesador.

Para $\beta_{EDF} \rightarrow \infty$ se tiene el mejor caso posible. Cualquier procesador que se añada al sistema incrementa el valor de $H_{EDF}(n, \beta_{EDF})$ en uno. La explicación es clara, en este caso $\alpha \rightarrow 0$. Cuando los factores de utilización son muy pequeños se pueden llenar los procesadores casi hasta su capacidad. En este caso un multiprocesador de n procesadores se comporta como un monoprocesador n veces más rápido.

La función $H_{EDF}(n, \beta_{EDF})$ proporciona el límite de utilización empleando asignación FF, FFD, FFI, BF, BFD, BFI, WFD, RFD y OPT. Sin embargo, hay casos prácticos en los cuales se está interesado en el número mínimo de procesadores n_{wc}^{EDF} que aseguran la planificabilidad de un conjunto de m tareas de utilización total U .

En el caso de los algoritmos de asignación cuyo límite de utilización asociado viene dado por $H_{EDF}(n, \beta_{EDF})$, el número mínimo de procesadores se representa por n_{wc}^{EDF-H} y se obtiene tal como se explica a continuación.

En la tesis se han presentado dos condiciones de planificabilidad complementarias empleando planificación EDF. La primera de ellas proviene de la definición de β_{EDF} dada en el apartado 3.2 (véase la figura 3.1)

$$m \leq \beta_{EDF} n_{wc}^{EDF-H}$$

Despejando n_{wc}^{EDF-H} y eligiendo el mínimo valor se obtiene

$$n_{wc}^{EDF-H} = \left\lceil \frac{m}{\beta_{EDF}} \right\rceil \quad (3.45)$$

La segunda condición de planificabilidad se deduce del empleo del límite de utilización

$$U \leq H_{EDF}(n_{wc}^{EDF-H}, \beta_{EDF}) = \frac{\beta_{EDF} n_{wc}^{EDF-H} + 1}{\beta_{EDF} + 1}$$

Despejando n_{wc}^{EDF-H} y eligiendo el mínimo valor se obtiene

$$n_{wc}^{EDF-H} = \left\lceil \frac{(\beta_{EDF} + 1)U - 1}{\beta_{EDF}} \right\rceil \quad (3.46)$$

La ecuación (3.46) no es válida para $U \leq 1/(\beta_{EDF} + 1)$. Sin embargo, en este caso un único procesador es suficiente para planificar de forma factible el conjunto de tareas. Teniendo en cuenta esta observación, la ecuación (3.45) y la ecuación (3.46) se obtiene

$$n_{wc}^{EDF-H}(m, U, \beta_{EDF}) = \begin{cases} 1 & \text{si } U \leq 1 \\ \min \left\{ \left\lceil \frac{m}{\beta_{EDF}} \right\rceil, \left\lceil \frac{(\beta_{EDF} + 1)U - 1}{\beta_{EDF}} \right\rceil \right\} & \text{en otro caso} \end{cases} \quad (3.47)$$

Por ejemplo, para garantizar la planificabilidad de 100 tareas de utilización total 15, y factores de utilización menores o iguales que 0'25 (es decir $\beta_{EDF} = 4$) empleando asignación de tareas FF, FFD, FFI, BF, BFD, BFI, WFD, RFD o OPT, y planificación EDF se necesitan 19 procesadores

$$n_{wc}^{EDF-H}(100, 15, 0'25) = \min \left\{ \left\lceil \frac{100}{4} \right\rceil, \left\lceil \frac{(4 + 1)15 - 1}{4} \right\rceil \right\} = 19$$

Un procesador menos que el necesario empleando asignación WF, WFI, RF o RFI y planificación EDF, tal como se obtuvo al final del apartado 3.3.1.

Llegados a este punto resulta interesante comparar el límite $L_{EDF}(n, \alpha)$ con el límite $H_{EDF}(n, \beta_{EDF})$. En primer lugar se sabe que

$$L_{EDF}(n, \alpha) \leq H_{EDF}(n, \beta_{EDF})$$

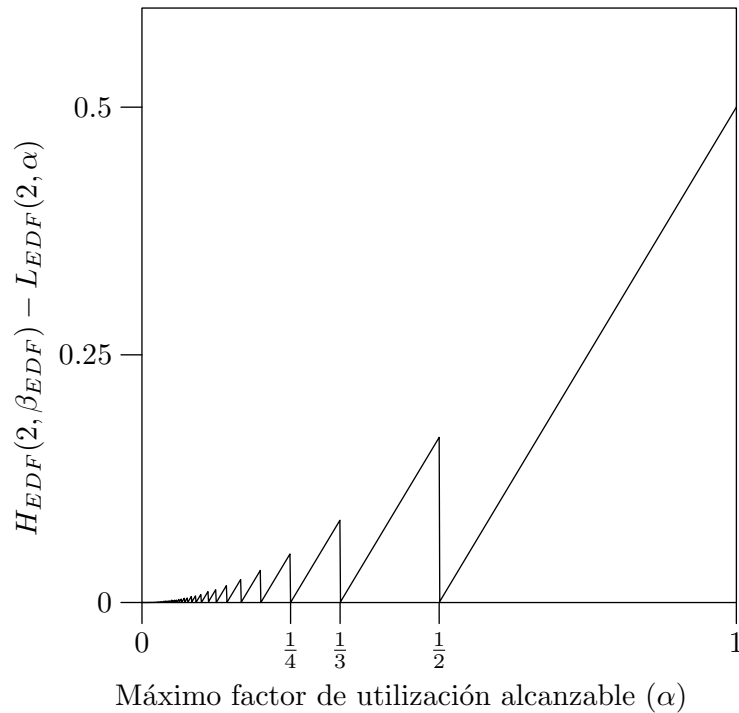


Figura 3.6: Representación de la diferencia $H_{EDF}(n, \beta_{EDF}) - L_{EDF}(n, \alpha)$ para el caso $n = 2$.

pues $L_{EDF}(n, \alpha)$ es el mínimo de los límites de utilización evaluado entre todos los algoritmos de asignación razonables, y $H_{EDF}(n, \beta_{EDF})$ es el máximo de los límites de utilización evaluado entre todos los algoritmos de asignación.

La diferencia entre ambos límites es una función de α y n , pues $\beta_{EDF} = \lfloor 1/\alpha \rfloor$.

$$H_{EDF}(n, \beta_{EDF}) - L_{EDF}(n, \alpha) = \frac{\lfloor \frac{1}{\alpha} \rfloor n + 1}{\lfloor \frac{1}{\alpha} \rfloor + 1} - n + (n - 1)\alpha \quad (3.48)$$

La máxima diferencia se obtiene para $\alpha = 1$. En este caso se tiene

$$H_{EDF}(n, \beta_{EDF}) - L_{EDF}(n, \alpha) = \frac{n - 1}{2} \quad (3.49)$$

A medida que α decrece la diferencia va disminuyendo, aunque no de forma monótona. En el límite, cuando $\alpha \rightarrow 0$ la diferencia tiende a cero.

La diferencia entre los dos límites no sólo tiende a cero cuando $\alpha \rightarrow 0$. También tiende a cero cuando

$$\alpha \rightarrow \left(\frac{1}{k}\right)^+$$

donde k es un entero mayor que uno. Esto puede comprobarse en la figura 3.6, la cual representa la diferencia entre los límites de utilización para el caso $n = 2$.

El hecho de que coincidan las cotas inferior y superior de los límites de utilización

para algoritmos de asignación razonables cuando

$$\alpha \rightarrow \left(\frac{1}{k}\right)^+$$

demuestra que el límite de utilización asociado a cualquier algoritmo razonable de asignación de tareas empleando planificación EDF verifica

$$\lim_{\epsilon \rightarrow 0^+} U_{wc}^{EDF-RA}(m, n, (1/k) + \epsilon) = \frac{(k-1)n + 1}{k} \quad (3.50)$$

donde k es un entero mayor que uno.

Por ejemplo, supóngase que se ha diseñado un nuevo algoritmo razonable de asignación de tareas a procesadores denominado RA. De antemano se puede afirmar que el límite de utilización asociado al mismo empleando planificación EDF toma el valor

$$U_{wc}^{EDF-RA} = \frac{(k-1)n + 1}{k}$$

para $\alpha = (1/k) + \epsilon$, con $\epsilon \rightarrow 0$ y $k > 1$.

3.4 Obtención de los límites de utilización estadísticos

El límite de utilización (absoluto) se obtiene para conjuntos de tareas que pueden ser infrecuentes en la práctica. Por ejemplo, considerando sólo el peor caso, los algoritmos de asignación FF y OPT (el algoritmo óptimo) serían equivalentes bajo planificación EDF, pues ambos llevan asociado el mismo límite de utilización. Sin embargo, está claro que el algoritmo OPT es superior al FF en términos de planificabilidad. Esto indica que el caso peor es más pesimista para asignación OPT que para asignación FF.

Con la intención de percibir el pesimismo asociado al límite de utilización (absoluto) se define el concepto de límite de utilización estadístico, denotado por $U = U_{ac}^{EDF-AA}(m, n, p)$, tal como sigue.

Definición 4 *Un conjunto de m tareas, de utilización total $U = U_{ac}^{EDF-AA}(m, n, p)$ es planificable sobre n procesadores con probabilidad p empleando planificación EDF-AA.*

Si se generasen aleatoriamente conjuntos de m tareas sin ninguna restricción sobre su factor de utilización ($\alpha = 1$), de utilización total $U = U_{ac}^{EDF-FF}(m, n, p)$, el $p\%$ de los conjuntos de tareas serían planificables sobre n procesadores empleando planificación EDF-AA. El $(100\% - p)$ restante se correspondería con conjuntos de tareas que no caben en los n procesadores.

Para el caso $p = 100\%$ se tiene que el límite de utilización estadístico y absoluto coinciden. Es decir,

$$U_{wc}^{EDF-RA}(m, n, \alpha = 1) = U_{ac}^{EDF-RA}(m, n, p = 100\%)$$

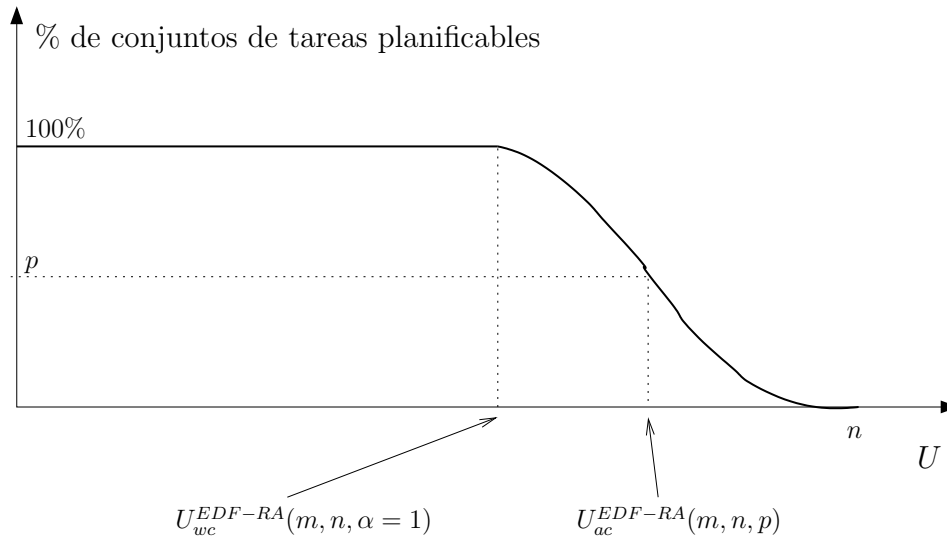


Figura 3.7: Relación entre los conceptos de límite de utilización absoluto y estadístico para unos valores de m y n fijos, y para una distribución estadística dada.

donde RA es un algoritmo de asignación de tareas razonable cualquiera.

La figura 3.7 muestra de forma intuitiva los conceptos de límite de utilización absoluto y estadístico para unos valores de m y n fijos. En realidad para cada pareja de valores m y n se obtendría una curva diferente, suponiendo que los parámetros de la distribución estadística son fijos.

La distribución estadística que se ha elegido para generar los conjuntos de tareas es la *Distribución Beta* (Canavos, 1988). Se trata de una distribución continua de probabilidad dada por la función de densidad

$$f(x; a, b) = \begin{cases} \frac{\Gamma(a+b)}{\Gamma(a)\Gamma(b)} x^{a-1} (1-x)^{b-1} & 0 < x < 1 \\ 0 & \text{en otro caso} \end{cases}$$

Esta distribución tiene dos parámetros positivos que permiten seleccionar la media μ , y la desviación típica σ de la distribución.

$$\mu = \frac{a}{a+b}; \quad \sigma^2 = \frac{ab}{(a+b)^2(a+b+1)}$$

Además es apropiada para generar los factores de utilización de las tareas, pues sólo genera valores en el intervalo $(0, 1)$. El máximo valor de σ para un valor dado de μ es $\sigma_{\max} = \sqrt{\mu(1-\mu)}$, el cual se obtiene considerando las restricciones de la distribución Beta. Al variar la desviación típica desde cero hasta el máximo, los factores de utilización pasan de ser iguales a ser muy diferentes. Por lo tanto, la desviación típica de la distribución afecta la planificabilidad del sistema, por lo que este factor debe ser tenido en cuenta. Por ejemplo, cuando la desviación típica toma un valor cercano a su máximo todos los factores de utilización tienden a cero o a

uno. Por el contrario, cuando la desviación típica de la distribución tiende a cero, los factores de utilización de todas las tareas toman el mismo valor. En este último caso el límite de utilización proporciona una condición de planificabilidad exacta independiente del algoritmo razonable de asignación de tareas, tal como demuestra el teorema 10.

Teorema 10 *Sea RA un algoritmo razonable de asignación de tareas a procesadores arbitrario. Sea $\{\tau_1, \dots, \tau_m\}$ un conjunto de m tareas de idénticos factores de utilización y utilización total U .*

El conjunto de tareas es planificable empleando planificación EDF-RA sobre n procesadores si y solo si

$$U \leq \frac{m}{\lceil \frac{m}{n} \rceil} \quad (3.51)$$

Demostración:

En primer lugar se demostrará que si

$$U \leq \frac{m}{\lceil \frac{m}{n} \rceil}$$

entonces el conjunto de m tareas cabe en los n procesadores.

El factor de utilización, común a todas las tareas, verifica

$$u_i \leq \frac{1}{\lceil \frac{m}{n} \rceil}$$

Por lo tanto, al menos $\lceil \frac{m}{n} \rceil$ de estas tareas caben en cada procesador. En los n procesadores caben por lo tanto $n \lceil \frac{m}{n} \rceil$ tareas de factor de utilización u_i . Como $n \lceil \frac{m}{n} \rceil \geq m$, resulta que las m tareas caben en los n procesadores.

Finalmente se demostrará que si

$$U > \frac{m}{\lceil \frac{m}{n} \rceil}$$

no caben las m tareas en los n procesadores. En este caso el factor de utilización, común a todas las tareas, verifica

$$u_i > \frac{1}{\lceil \frac{m}{n} \rceil}$$

Por lo tanto, en cada procesador caben a lo sumo $\lceil \frac{m}{n} \rceil - 1$ tareas, y en los n procesadores a lo sumo $n (\lceil \frac{m}{n} \rceil - 1)$. Pero este número es menor que m , el número de tareas del conjunto, y por lo tanto el conjunto de tareas no es planificable. ■

La figura 3.8 muestra la representación gráfica de la función $\frac{m}{\lceil \frac{m}{n} \rceil}$. Cabe destacar las siguientes características de la ecuación (3.51):

- El límite de utilización coincide con el ideal para un número de tareas múltiplo del número de procesadores.

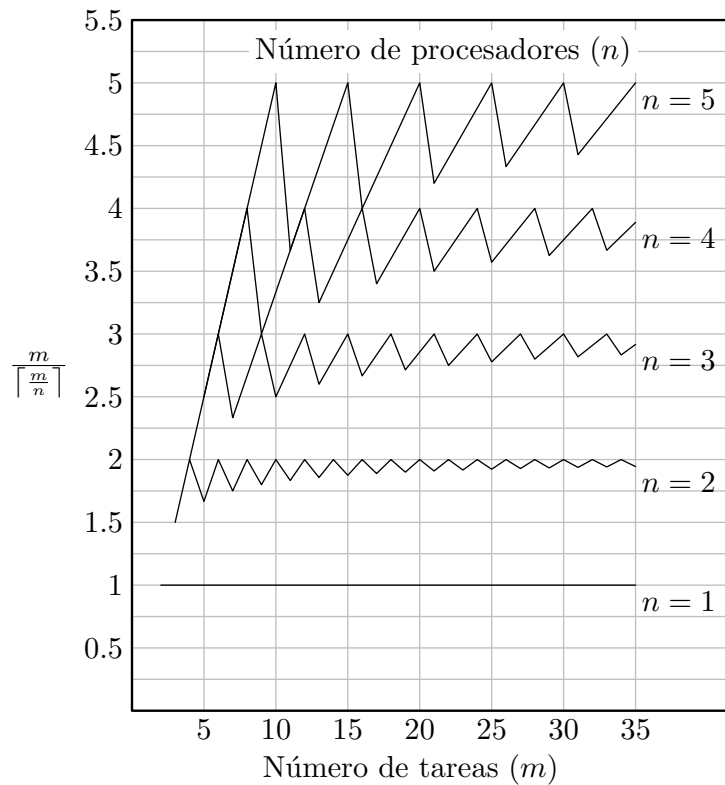


Figura 3.8: Representación del límite de utilización exacto para el caso de factores de utilización iguales y planificación EDF-RA.

- Aparecen unos ciclos de amplitud proporcional al número de procesadores al variar el número de tareas.
- La amplitud de los ciclos decrece a medida que aumenta el número de tareas.

Con la intención de establecer el pesimismo de los límites de utilización absolutos empleando planificación EDF se van llevar a cabo experimentos de simulación. Antes de realizar dichos experimentos es conveniente identificar con claridad que resultado se pretende obtener y en función de que parámetros. El resultado que se desea obtener es el límite de utilización estadístico. Una vez elegida la distribución Beta para generar los factores de utilización de las tareas, el límite de utilización estadístico depende de los siguientes parámetros:

- Número de procesadores n . El número de procesadores variará entre 2 y 10 en incrementos de 1. No se considera el caso de un procesador pues en este caso se sabe directamente que el límite de utilización estadístico coincide con el absoluto y toma el valor uno.
- Número de tareas m . El número de tareas empleado será $m = 2n$, $m = 3n$, $m = 4n$ y $m = 10n$, donde n puede tomar los valores especificados anteriormente.
- Probabilidad p . Este parámetro tomará los valores 50%, 75%, 90%, y 99%.

- Desviación típica de la distribución σ . Los valores que se consideran son $\sigma = 0'001\sigma_{max}$, $\sigma = 0'1\sigma_{max}$, $\sigma = 0'2\sigma_{max}$, y así sucesivamente hasta $\sigma = 0'9\sigma_{max}$.
- Algoritmo razonable de asignación de tareas RA. Puede ser cualquiera de los algoritmos considerados en la tesis, esto es, FF, FFD, FFI, BF, BFD, BFI, WF, WFD, WFI, RF, RFD y RFI.

Para la obtención de los límites de utilización estadísticos se barre a intervalos de 0'01 la utilización total U dentro del rango $[1, 0'9n]$. El número de conjuntos de tareas N_c generados para cada vector (n, m, σ, U) es como mínimo $N_c = 1000$, para todos los valores de p , y para todos los algoritmos de asignación.

El mecanismo de obtención de los límites estadísticos es el siguiente:

1. Generar N_c conjuntos de tareas para cada vector (n, m, σ, U) empleando una distribución Beta de media U/m y desviación típica σ .
2. Obtener el porcentaje de conjuntos de tareas planificables empleando la asignación RA, para cada uno de los valores de U anteriores.
3. El límite de utilización estadístico asociado al vector (n, m, p, σ, RA, U) se obtiene como el valor de U para el cual el tanto por ciento de tareas planificables es mayor o igual que p , y además para $(U + 0'01)$ el tanto por ciento de tareas planificables es menor que p .

Debe observarse que no se establece ninguna restricción sobre el tamaño máximo de las tareas. Es decir, $0 < u_i \leq 1$ para toda tarea τ_i de los conjuntos generados.

En el apartado siguiente se muestran los límites de utilización estadísticos para cada uno de los algoritmos de asignación analizados en la tesis, a excepción del algoritmo OPT. La obtención de los límites de utilización estadísticos asociados al algoritmo OPT requiere un tiempo computacional excesivo. No obstante, en el apartado 3.5 se compara el algoritmo óptimo con el mejor de los algoritmos heurísticos presentados en la tesis, para valores bajos del número de tareas y del número de procesadores.

Por una parte, se representan los límites estadísticos para las desviaciones típicas $\sigma = 0'1\sigma_{max}$, $\sigma = 0'3\sigma_{max}$, $\sigma = 0'5\sigma_{max}$, $\sigma = 0'7\sigma_{max}$, y $\sigma = 0'9\sigma_{max}$. Además se representa un límite estadístico independiente de la desviación típica que se obtiene como el mínimo de los límites estadísticos evaluado entre 10 las desviaciones típicas consideradas entre $\sigma = 0'001\sigma_{max}$ y $\sigma = 0'9\sigma_{max}$.

Por otra parte, se representa además el límite de utilización absoluto para el caso $\alpha = 1$. Esto permite comparar los límites de utilización absolutos y estadísticos. Finalmente las figuras muestran la función $U = n$, la cual representa el comportamiento ideal del multiprocesador. Esto es, cuando el multiprocesador se comporta como un monoprocesador n veces más rápido.

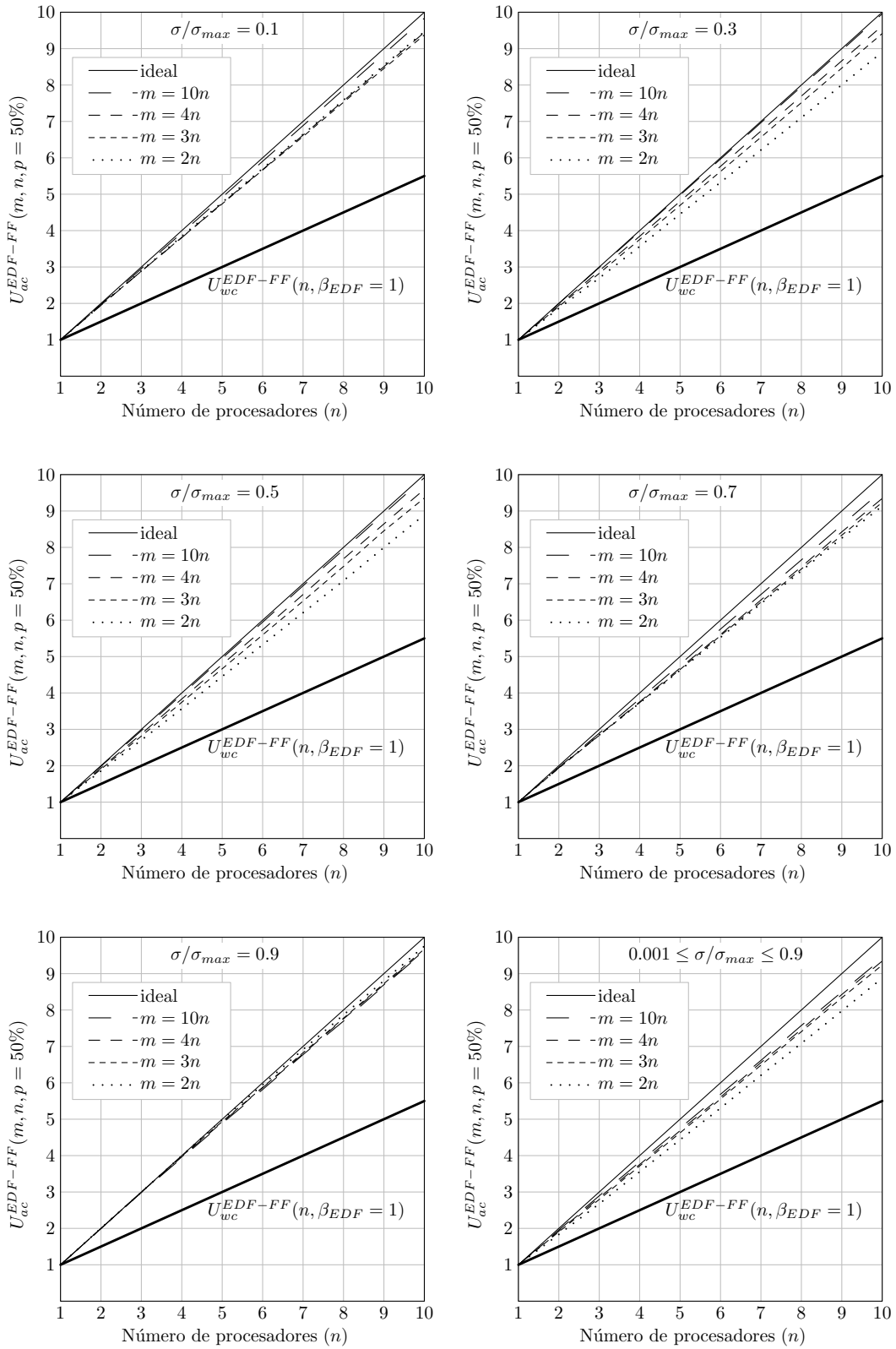


Figura 3.9: Límites de utilización para planificación EDF-FF y $p = 50\%$.

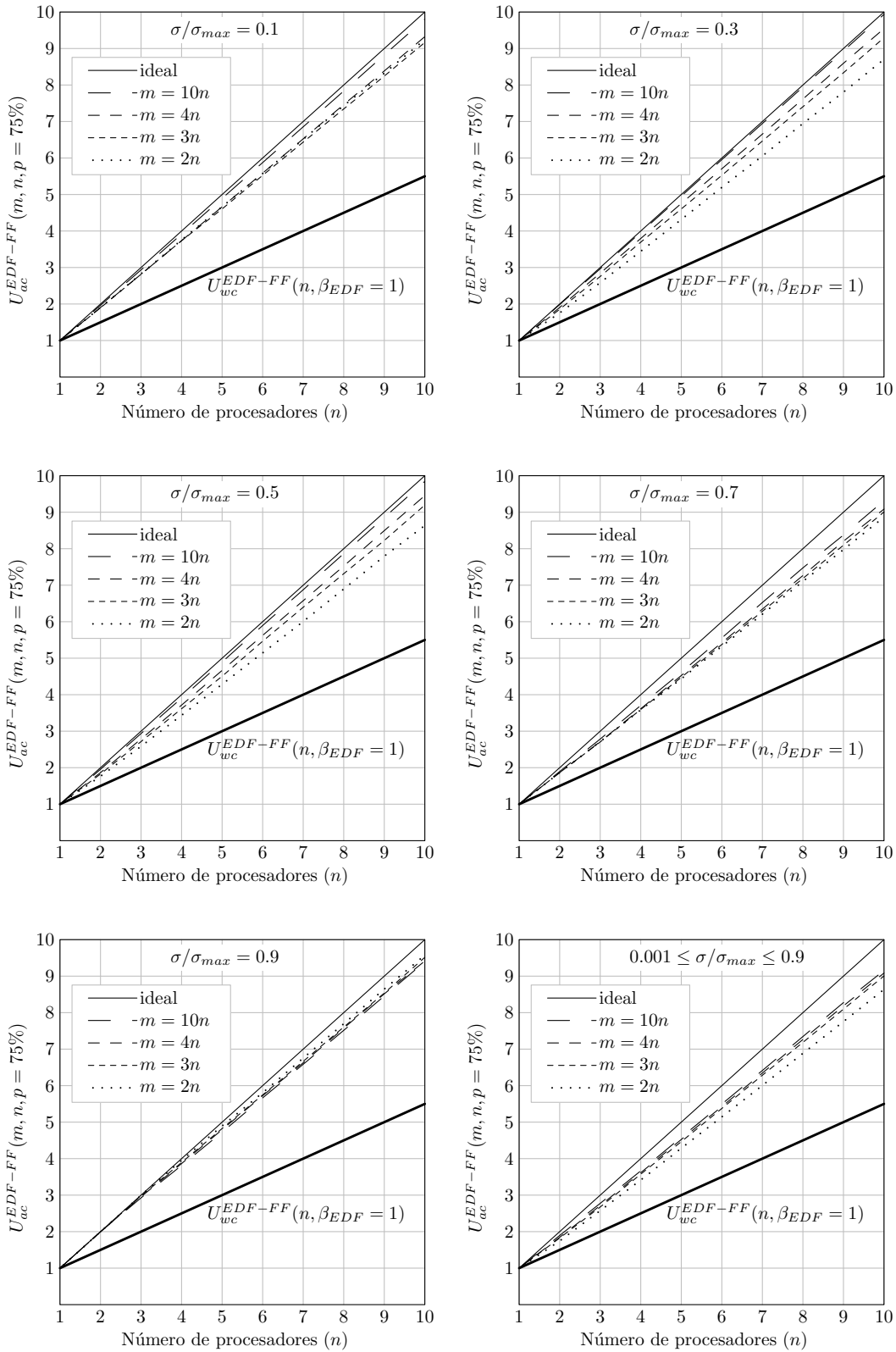


Figura 3.10: Límites de utilización para planificación EDF-FF y $p = 75\%$.

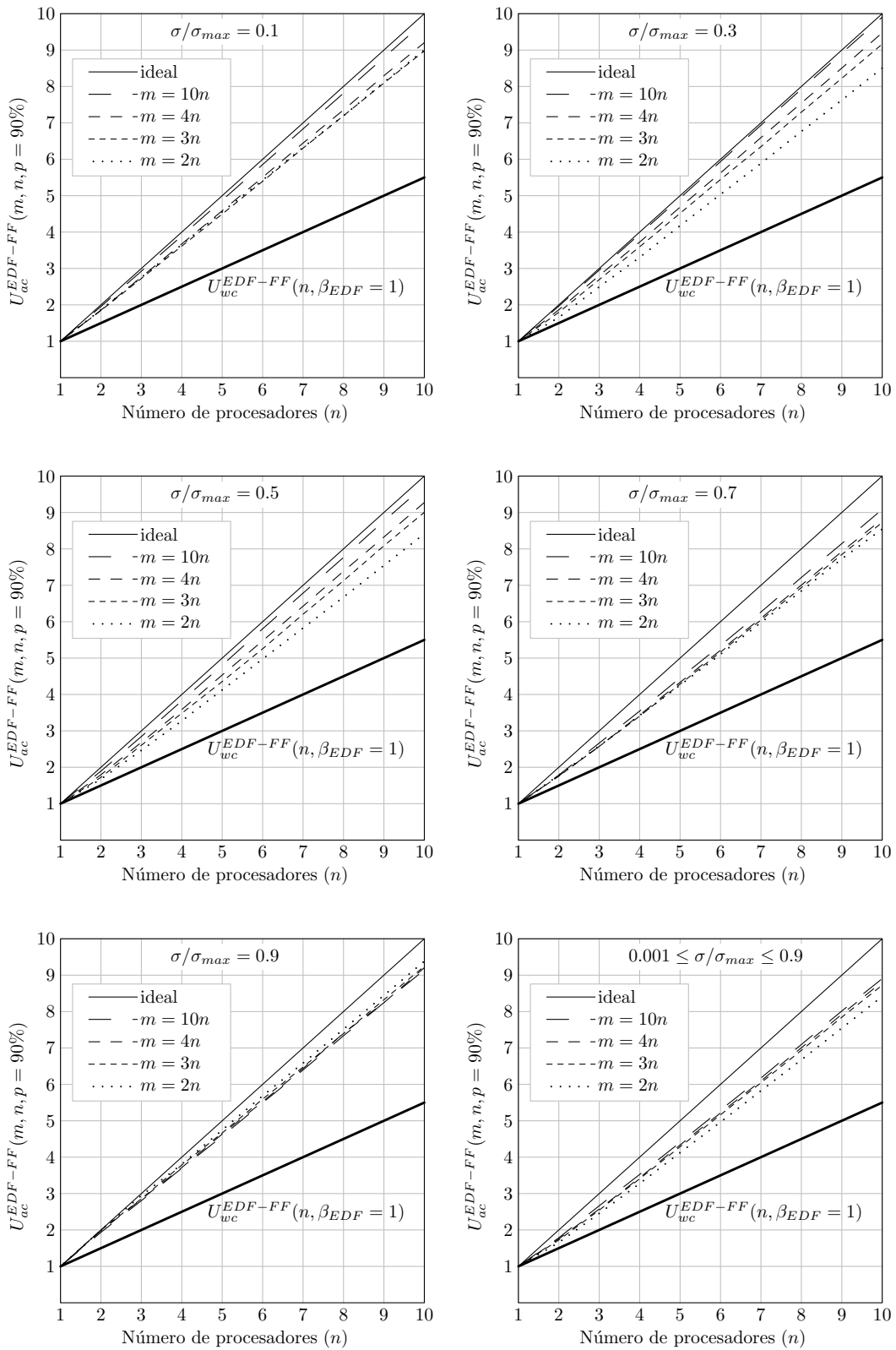


Figura 3.11: Límites de utilización para planificación EDF-FF y $p = 90\%$.

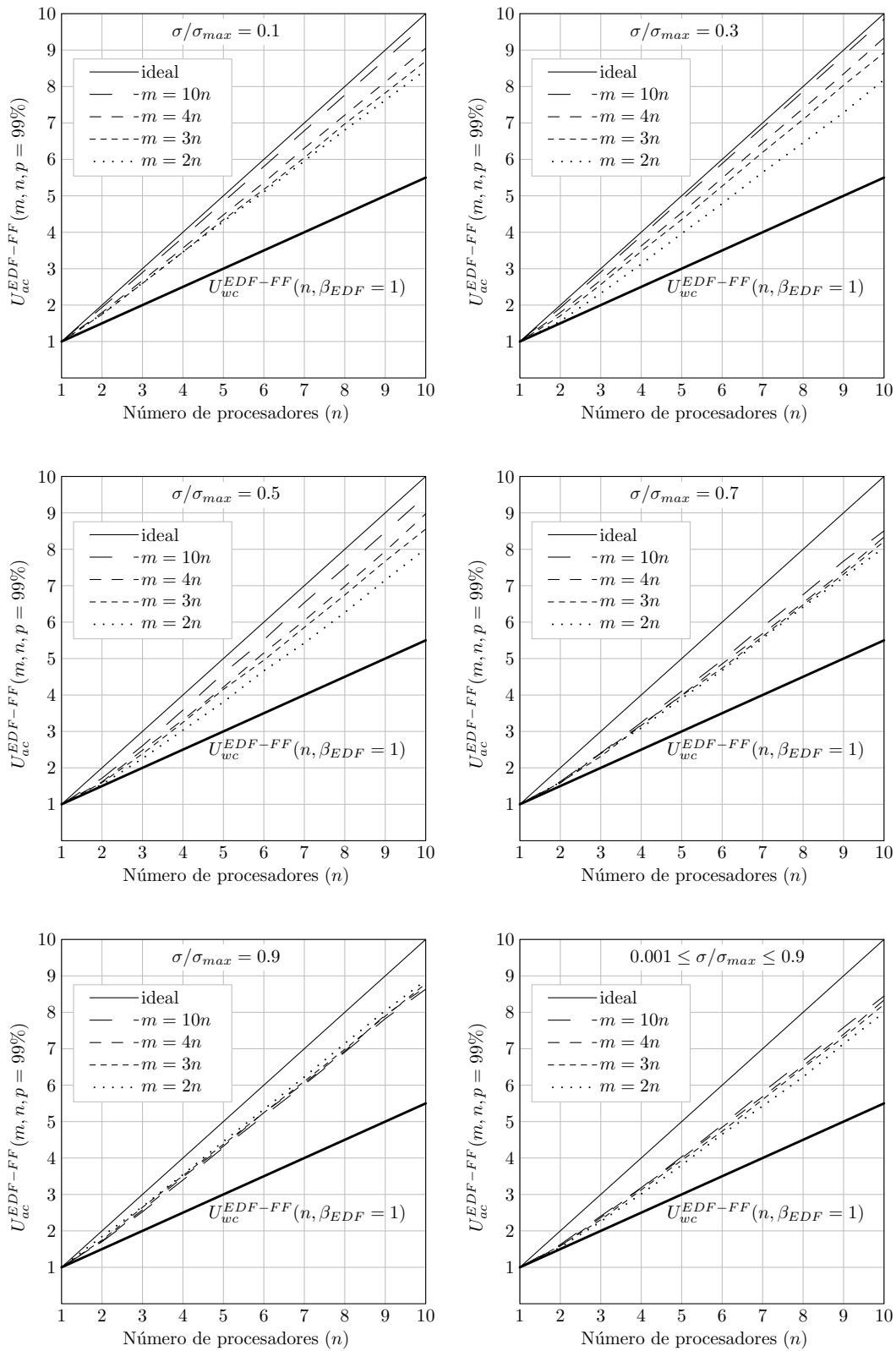


Figura 3.12: Límites de utilización para planificación EDF-FF y $p = 99\%$.

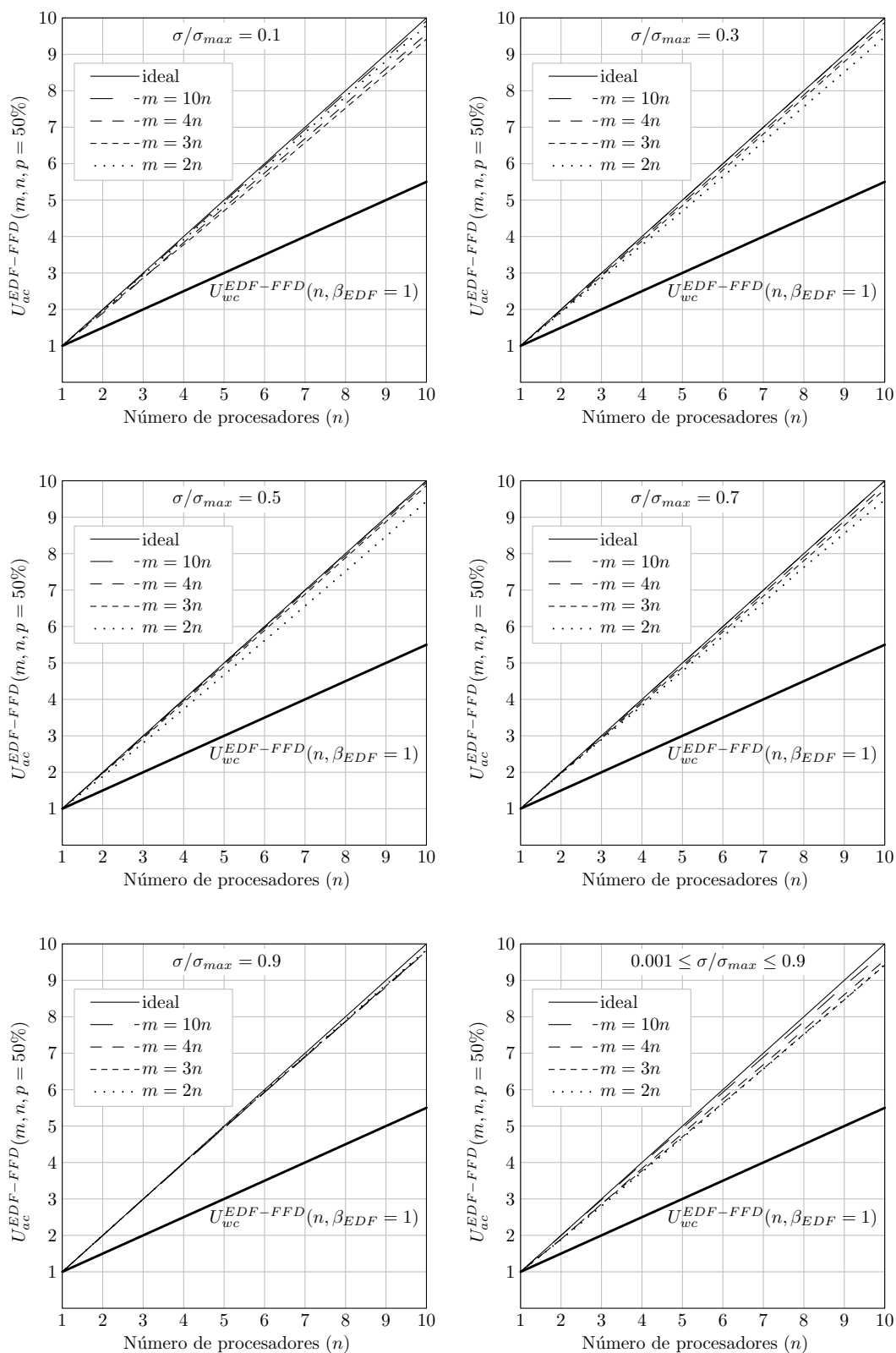


Figura 3.13: Límites de utilización para planificación EDF-FFD y $p = 50\%$.

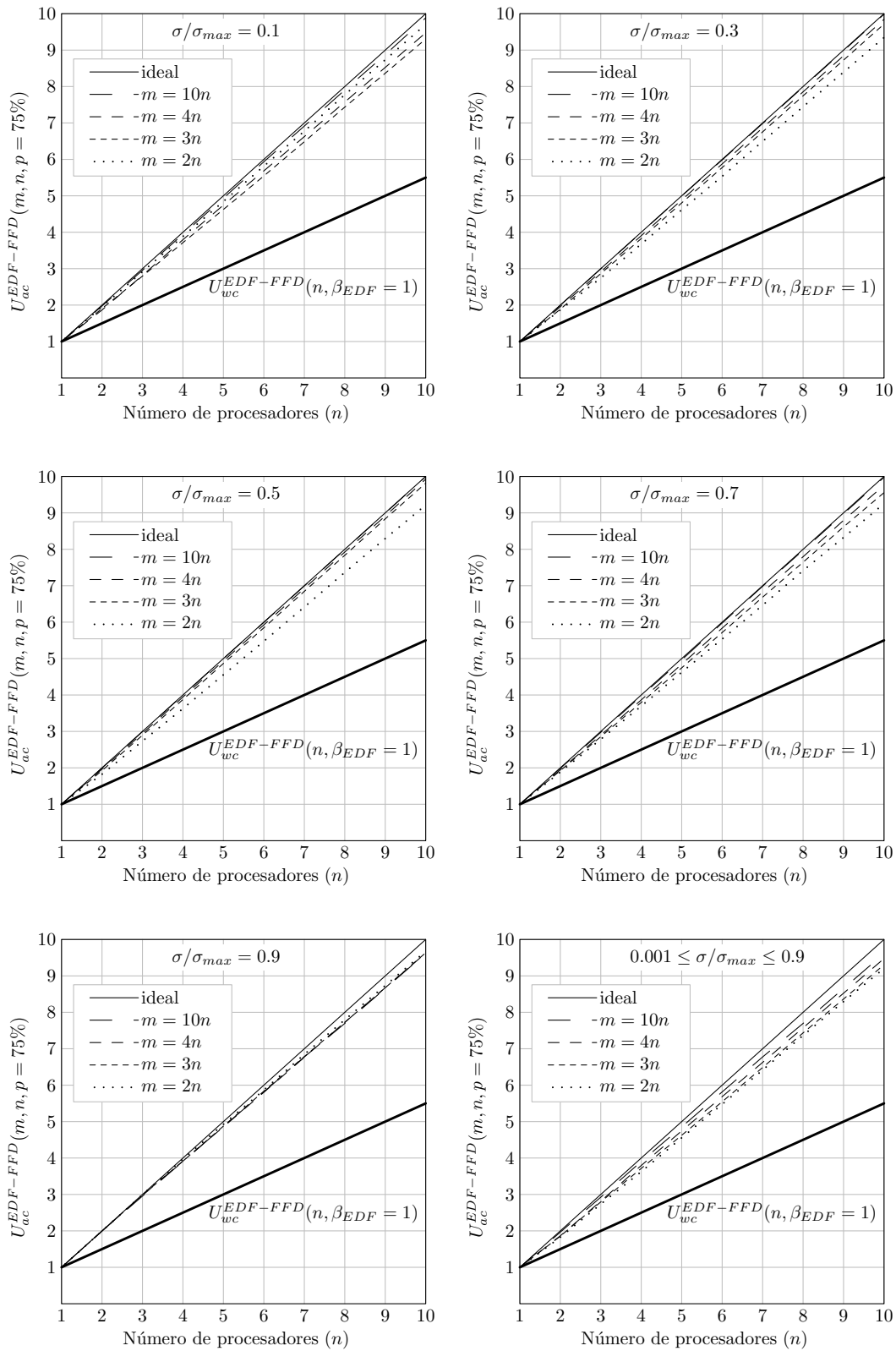


Figura 3.14: Límites de utilización para planificación EDF-FFD y $p = 75\%$.

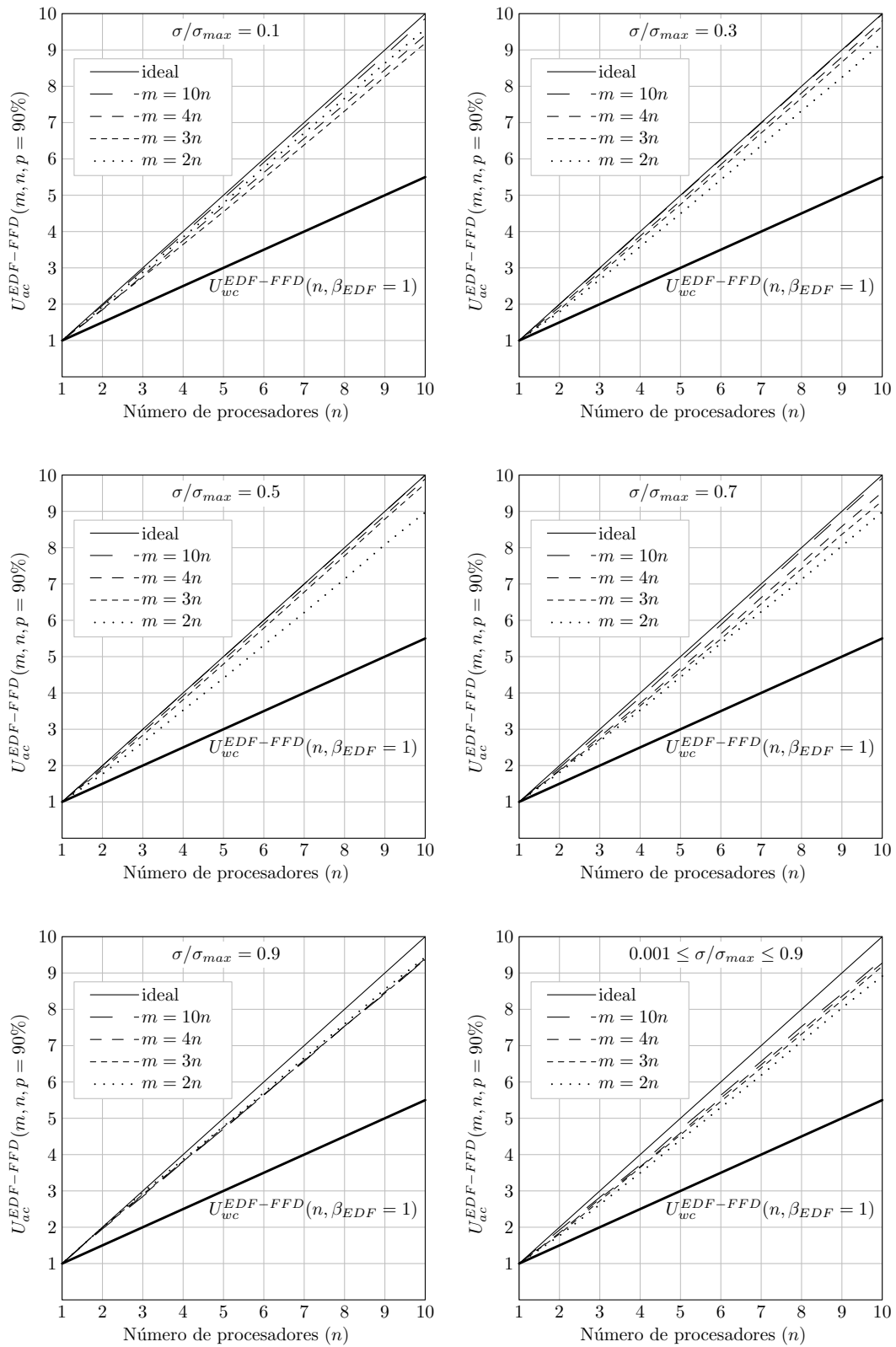


Figura 3.15: Límites de utilización para planificación EDF-FFD y $p = 90\%$.

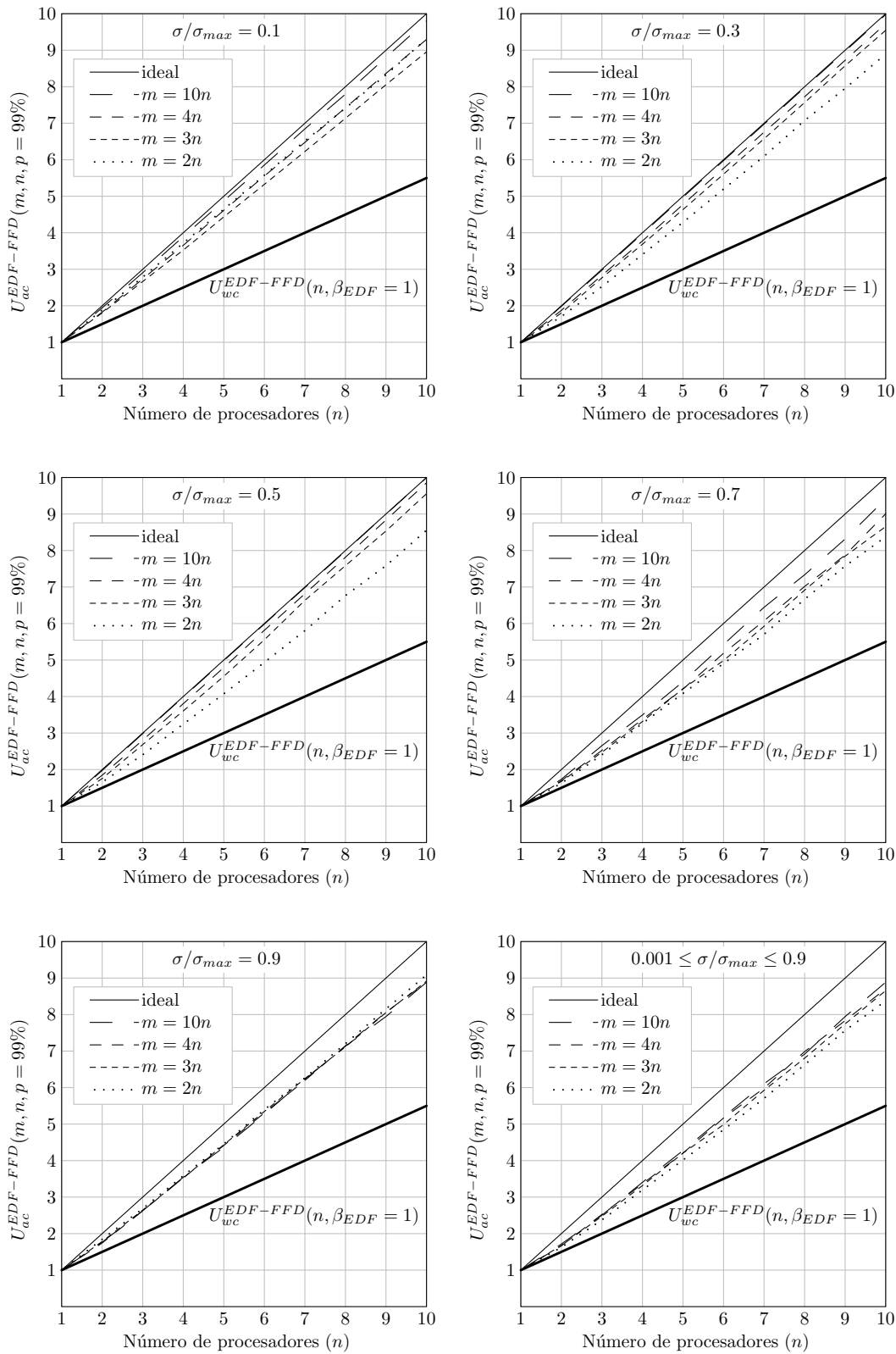


Figura 3.16: Límites de utilización para planificación EDF-FFD y $p = 99\%$.

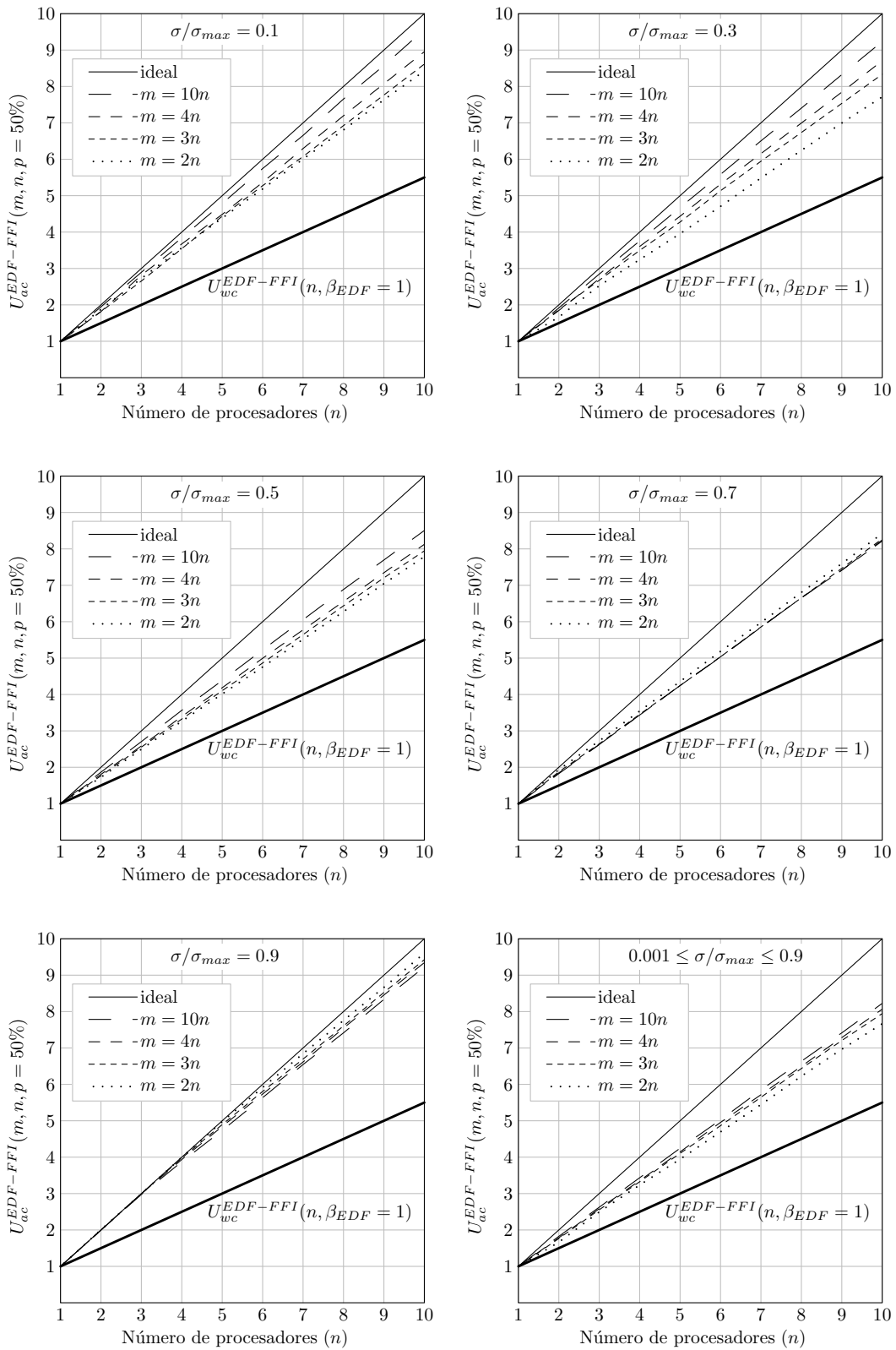


Figura 3.17: Límites de utilización para planificación EDF-FFI y $p = 50\%$.

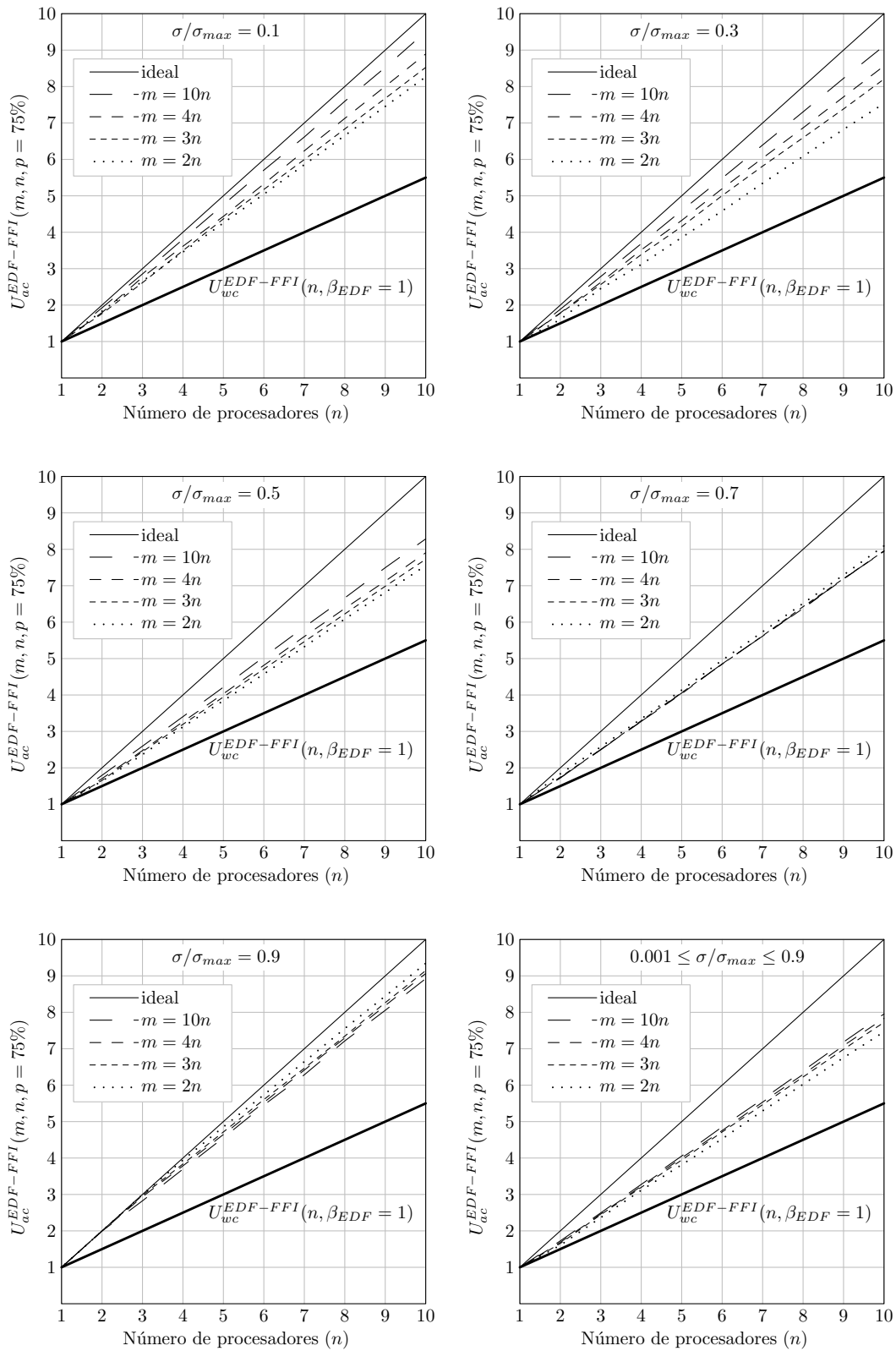


Figura 3.18: Límites de utilización para planificación EDF-FFI y $p = 75\%$.

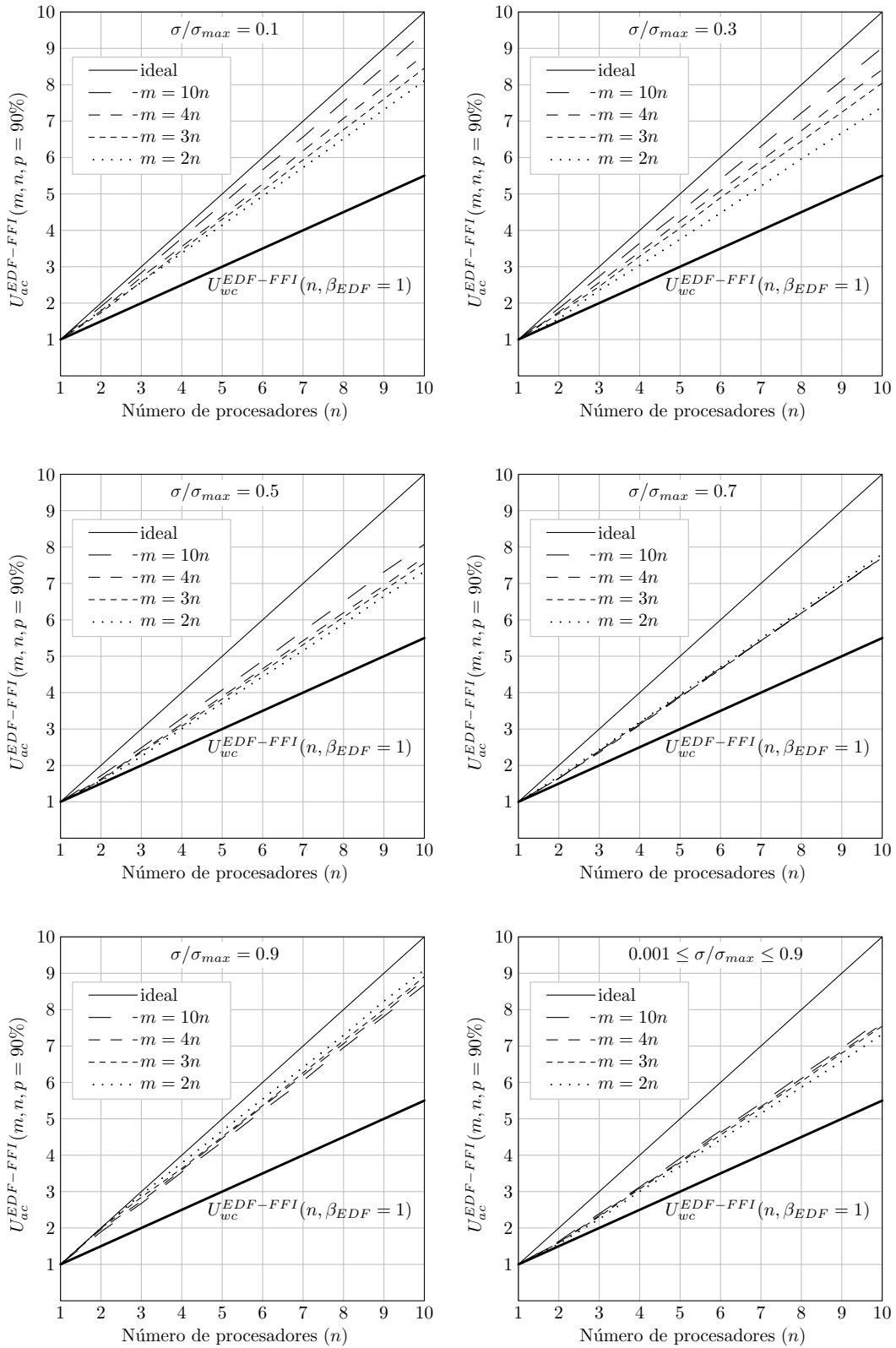


Figura 3.19: Límites de utilización para planificación EDF-FFI y $p = 90\%$.

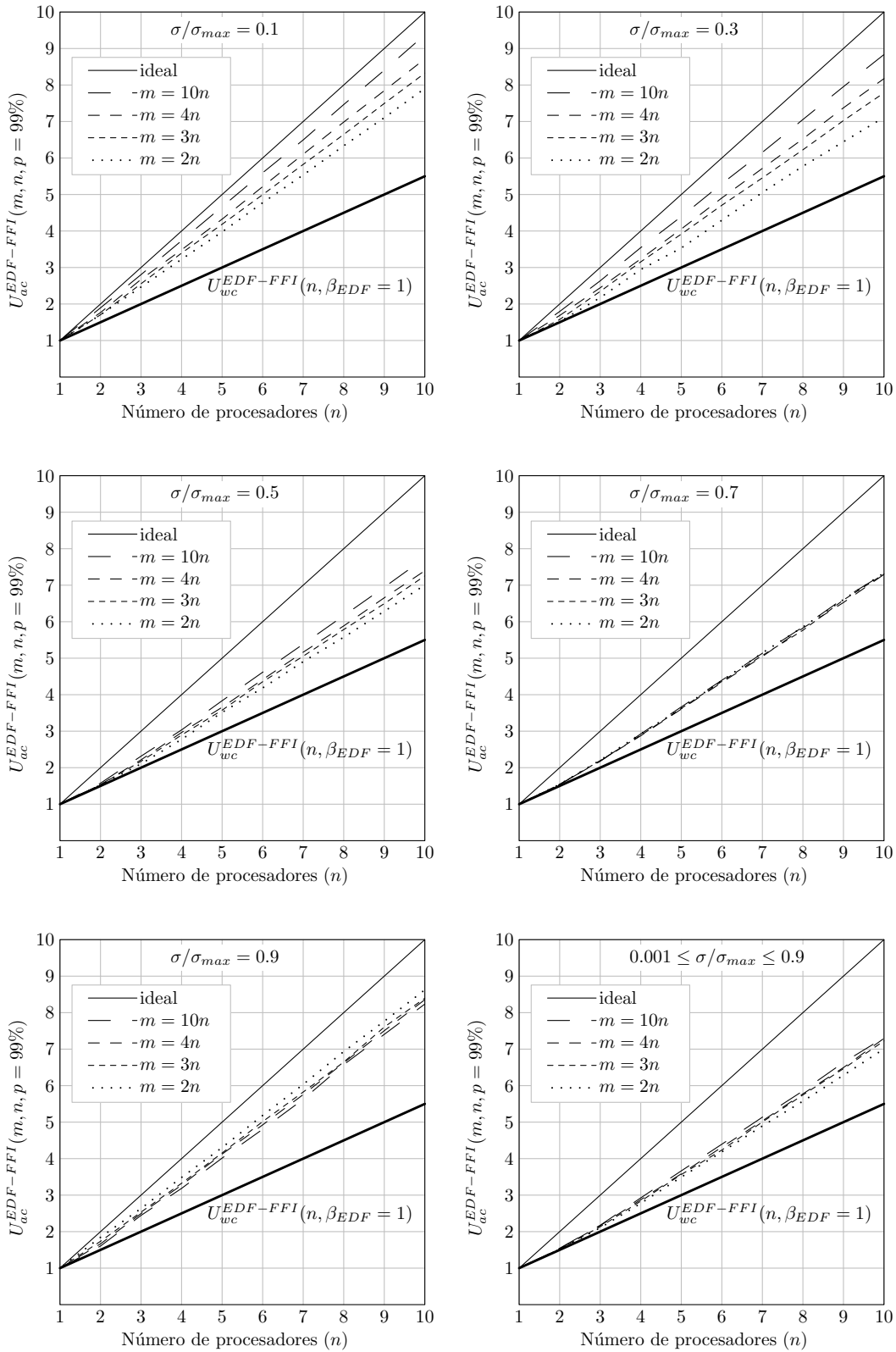


Figura 3.20: Límites de utilización para planificación EDF-FFI y $p = 99\%$.

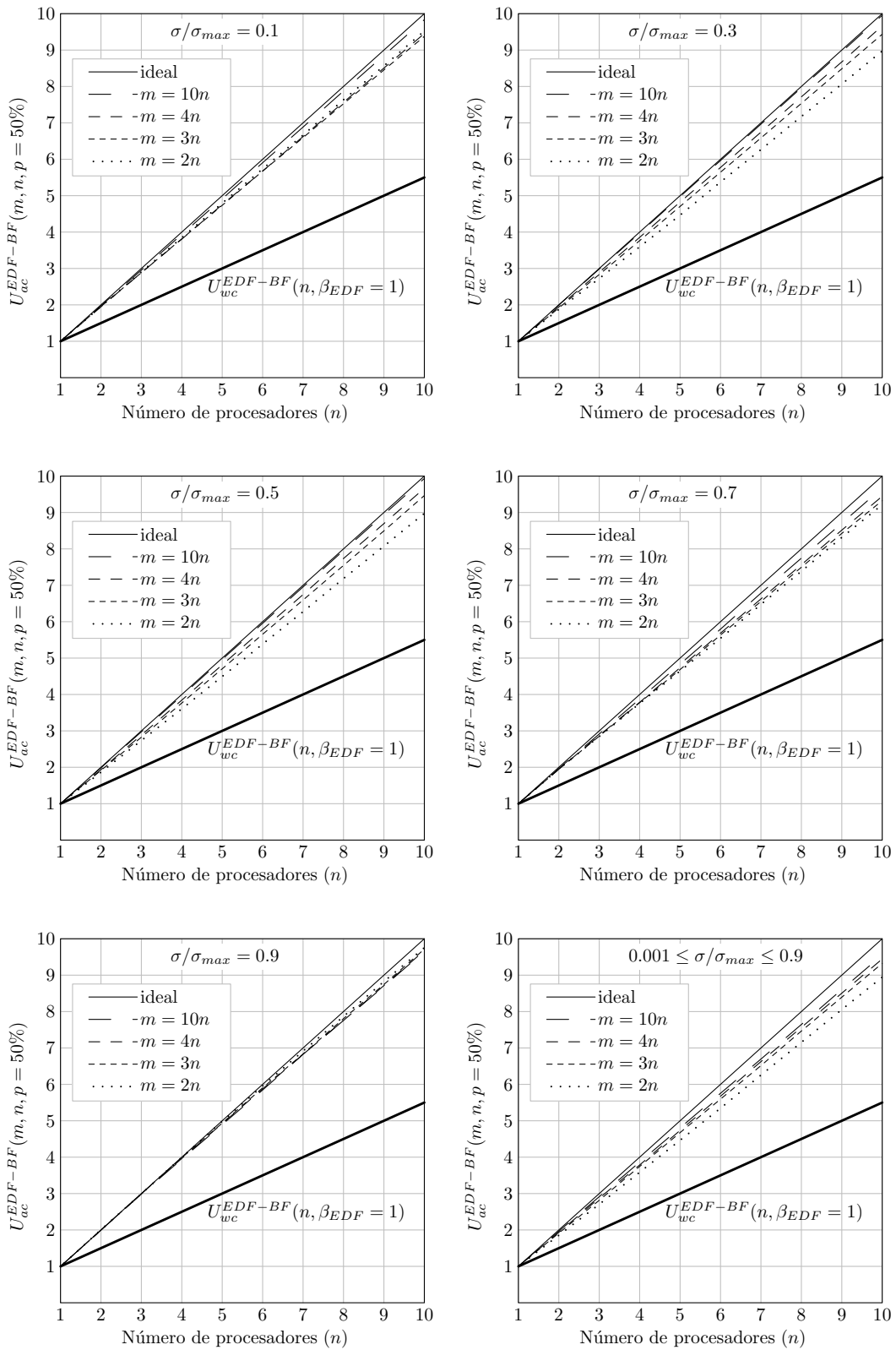


Figura 3.21: Límites de utilización para planificación EDF-BF y $p = 50\%$.

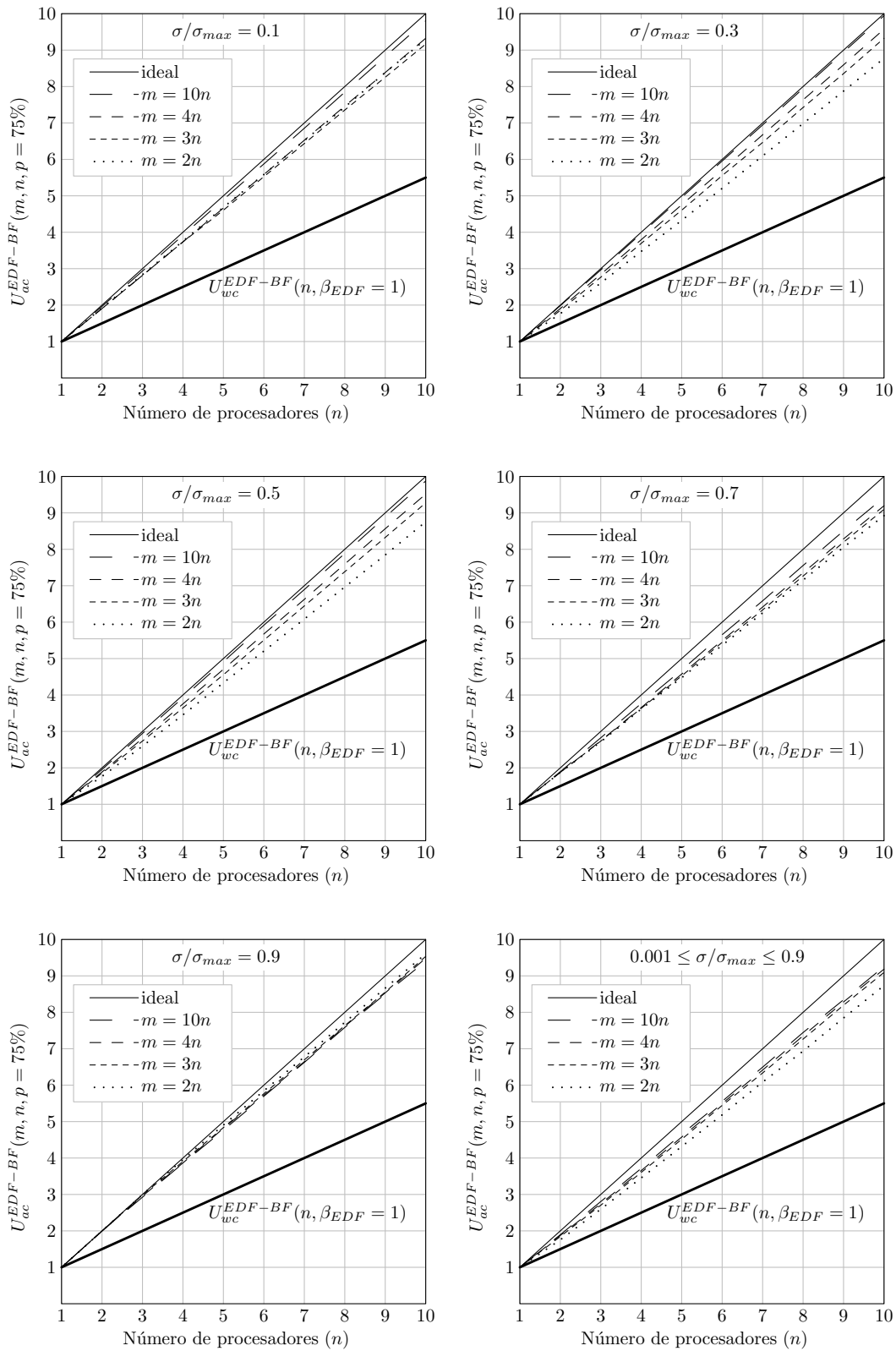


Figura 3.22: Límites de utilización para planificación EDF-BF y $p = 75\%$.

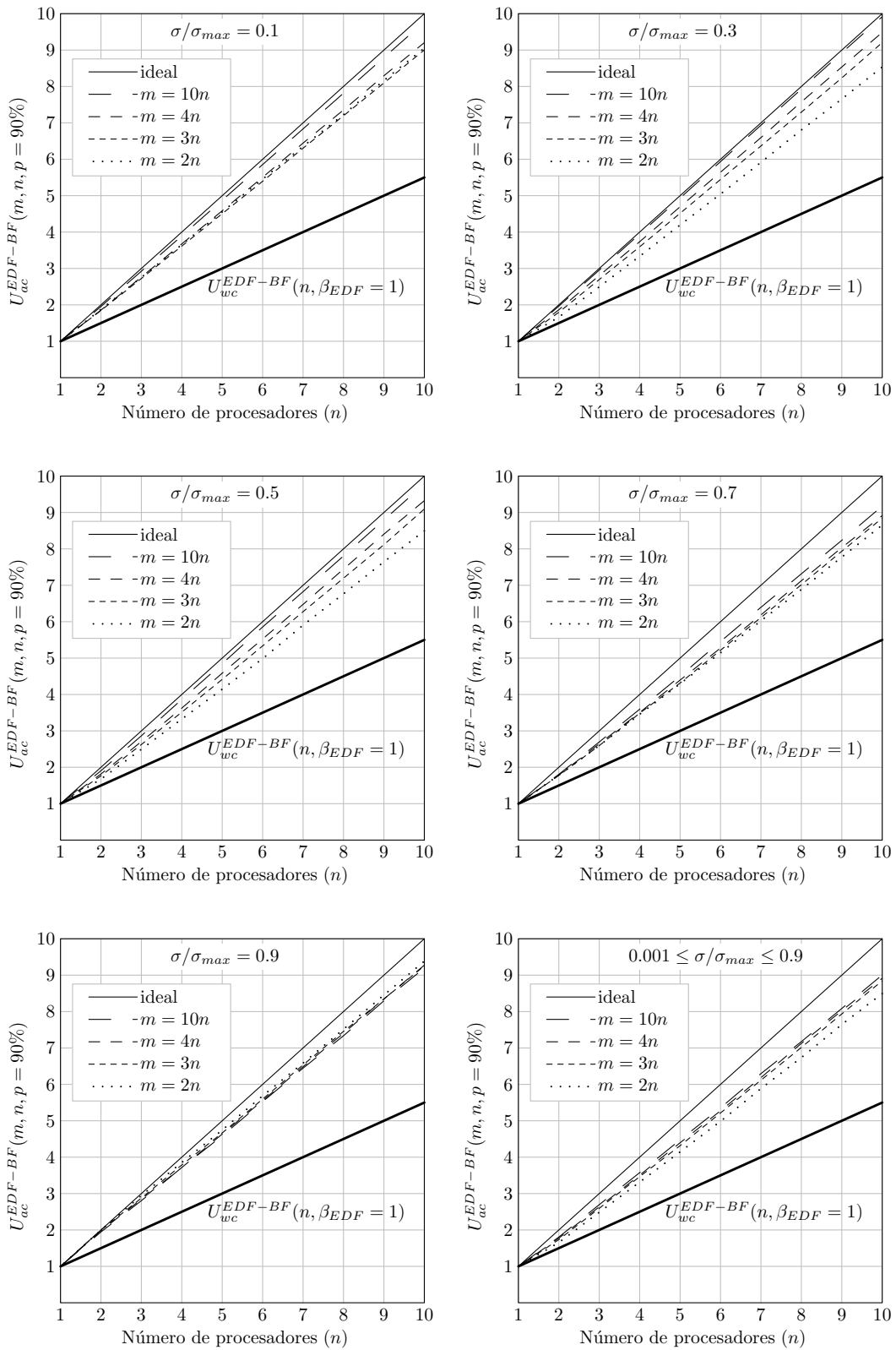


Figura 3.23: Límites de utilización para planificación EDF-BF y $p = 90\%$.

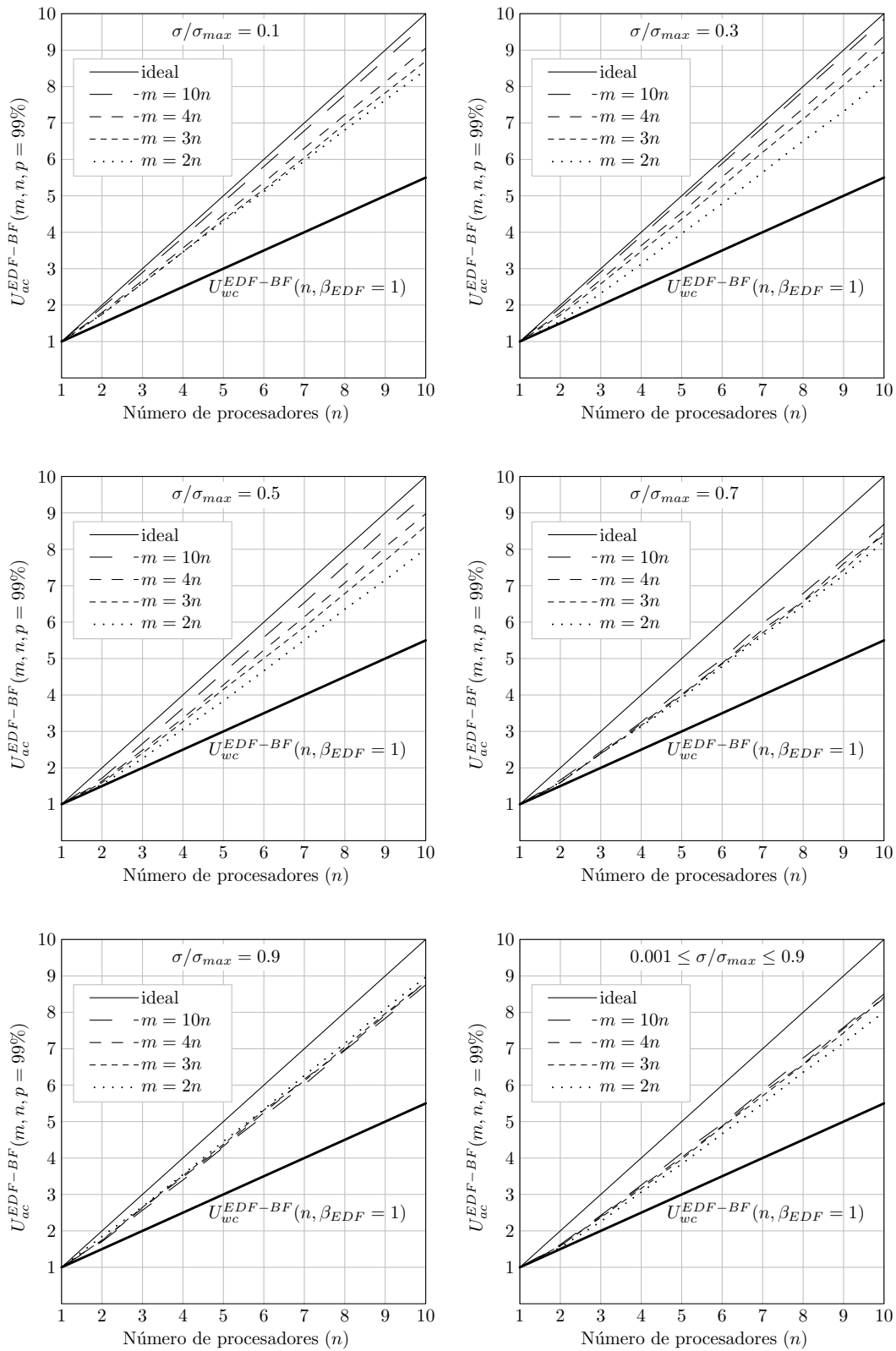


Figura 3.24: Límites de utilización para planificación EDF-BF y $p = 99\%$.

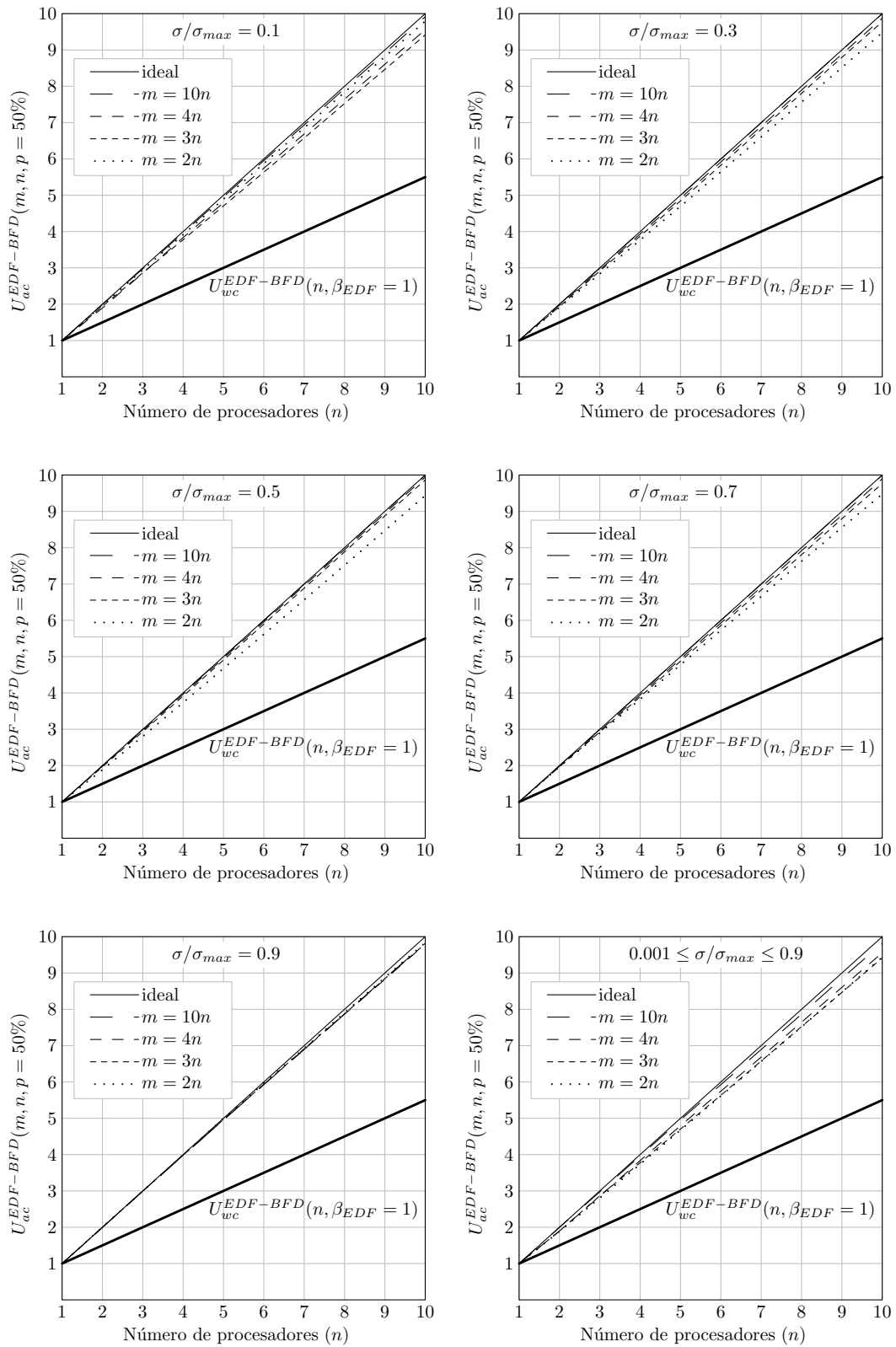


Figura 3.25: Límites de utilización para planificación EDF-BFD y $p = 50\%$.

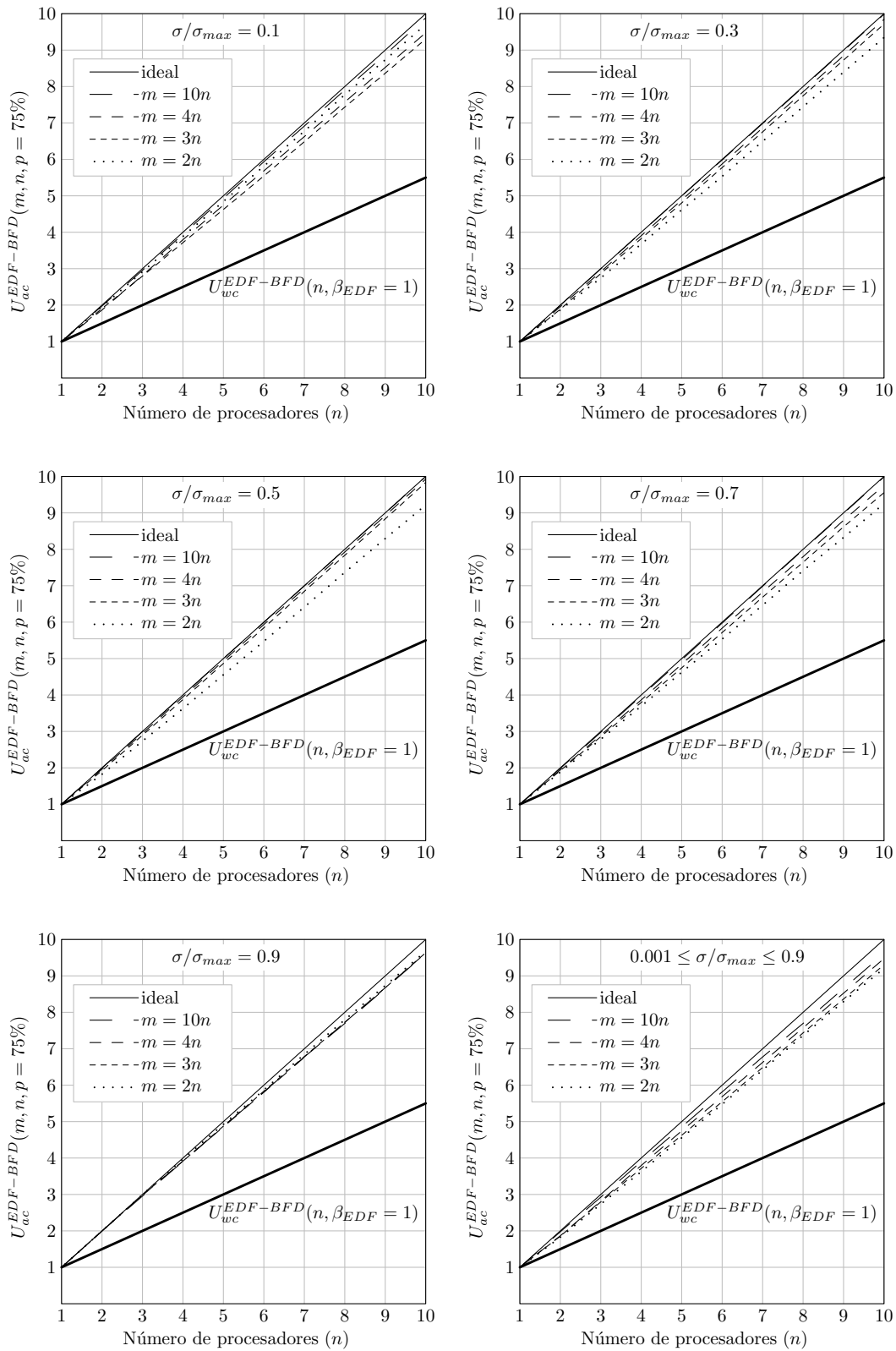


Figura 3.26: Límites de utilización para planificación EDF-BFD y $p = 75\%$.

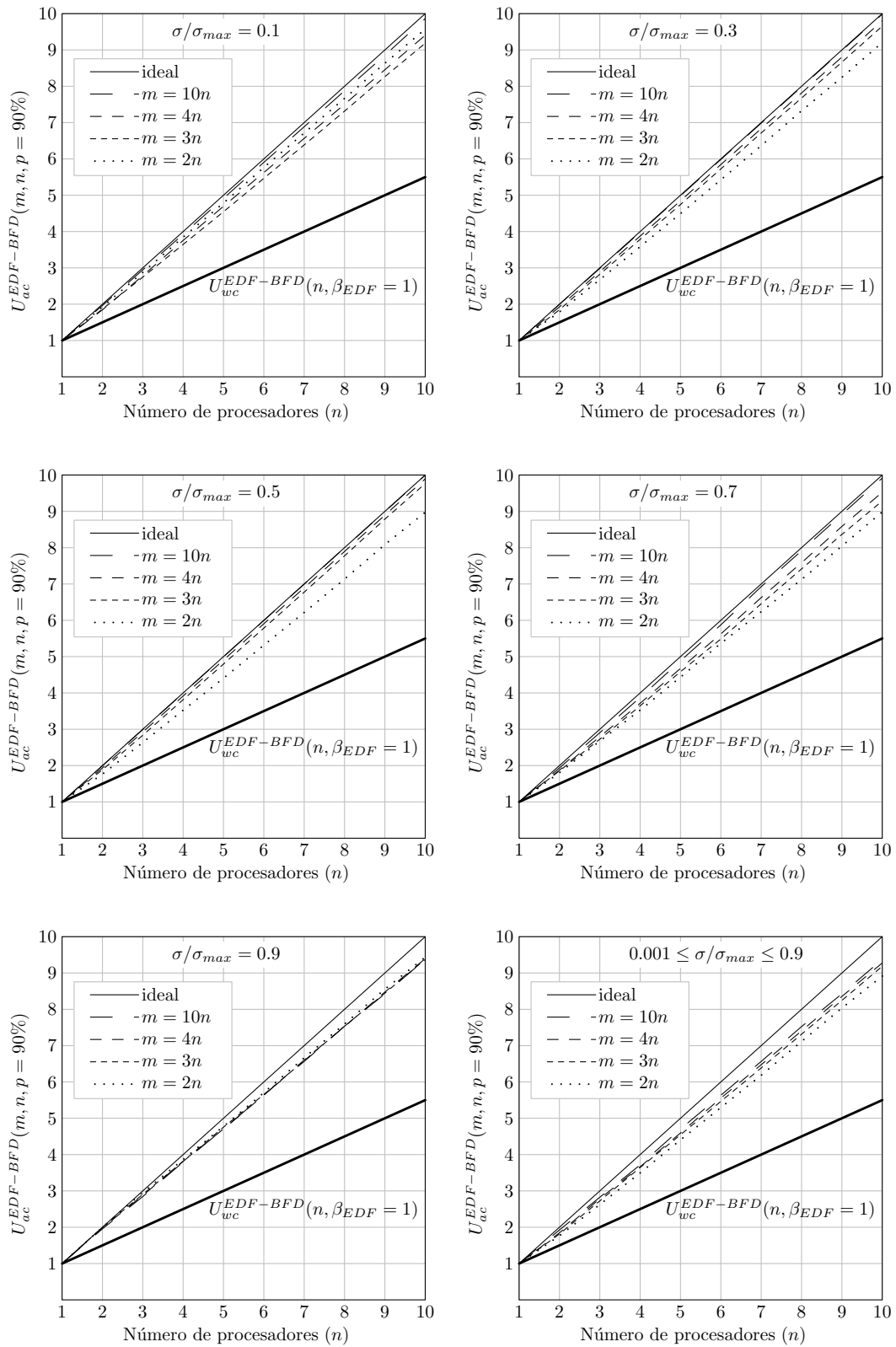


Figura 3.27: Límites de utilización para planificación EDF-BFD y $p = 90\%$.

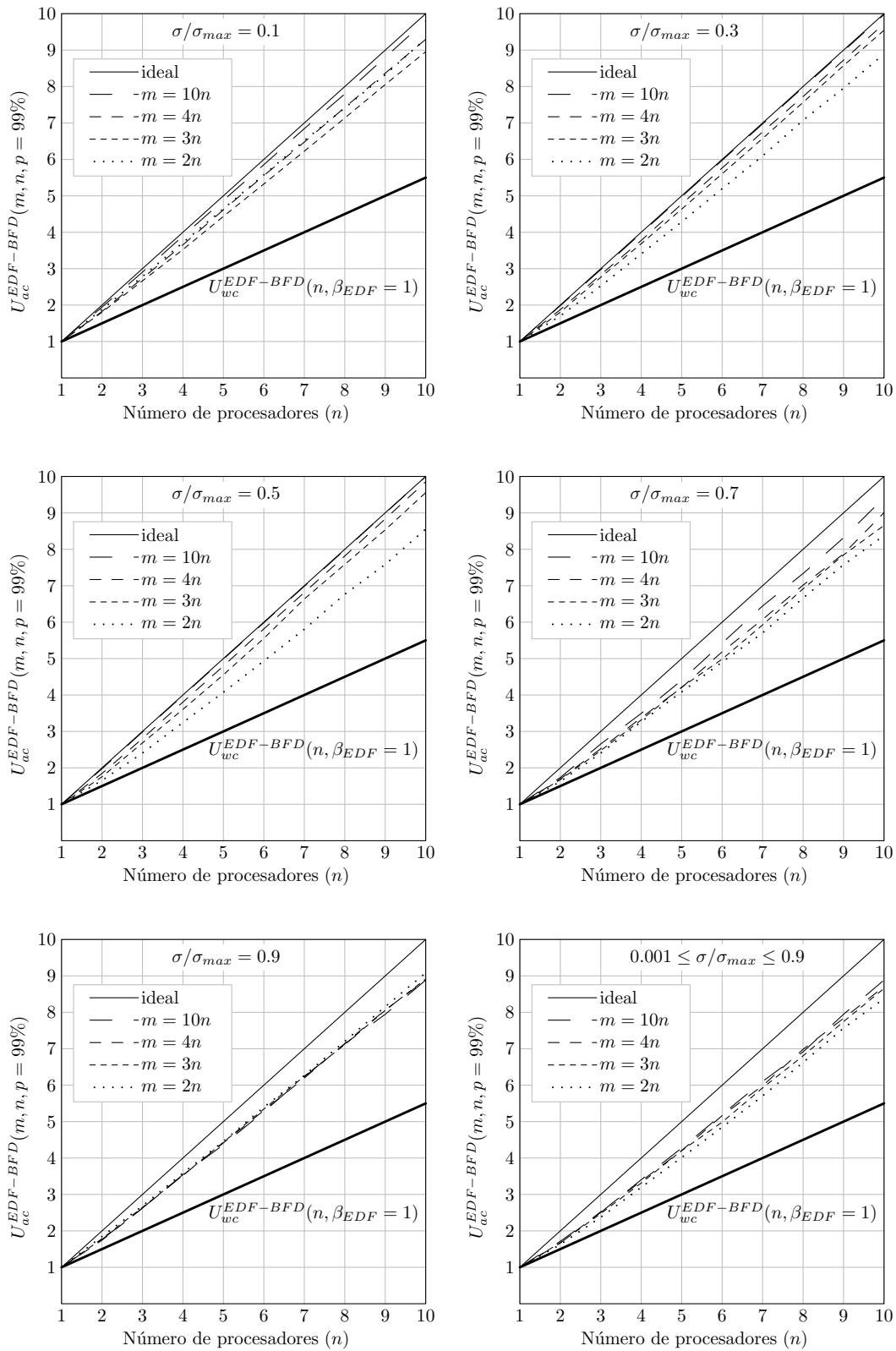


Figura 3.28: Límites de utilización para planificación EDF-BFD y $p = 99\%$.

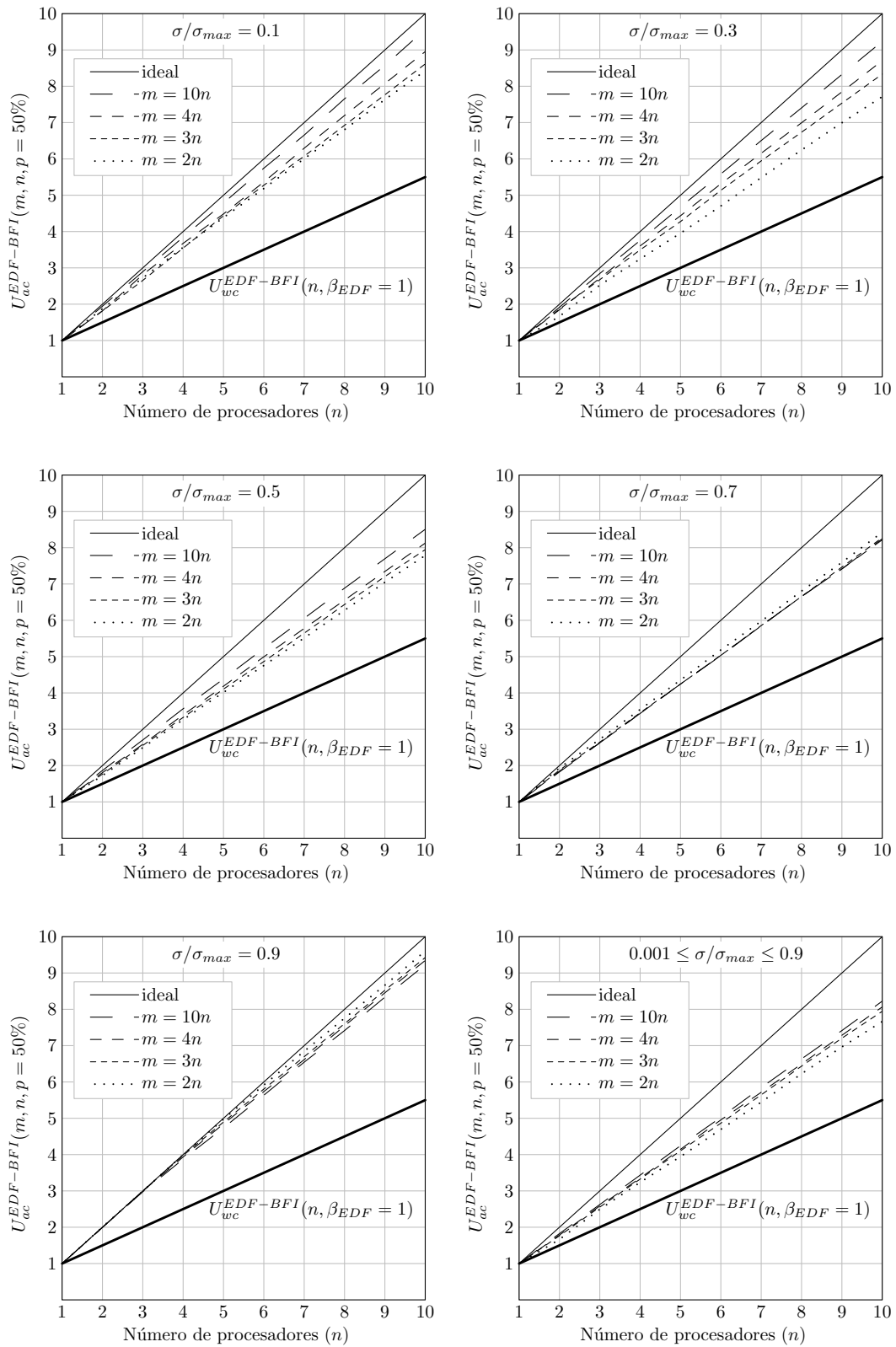


Figura 3.29: Límites de utilización para planificación EDF-BFI y $p = 50\%$.

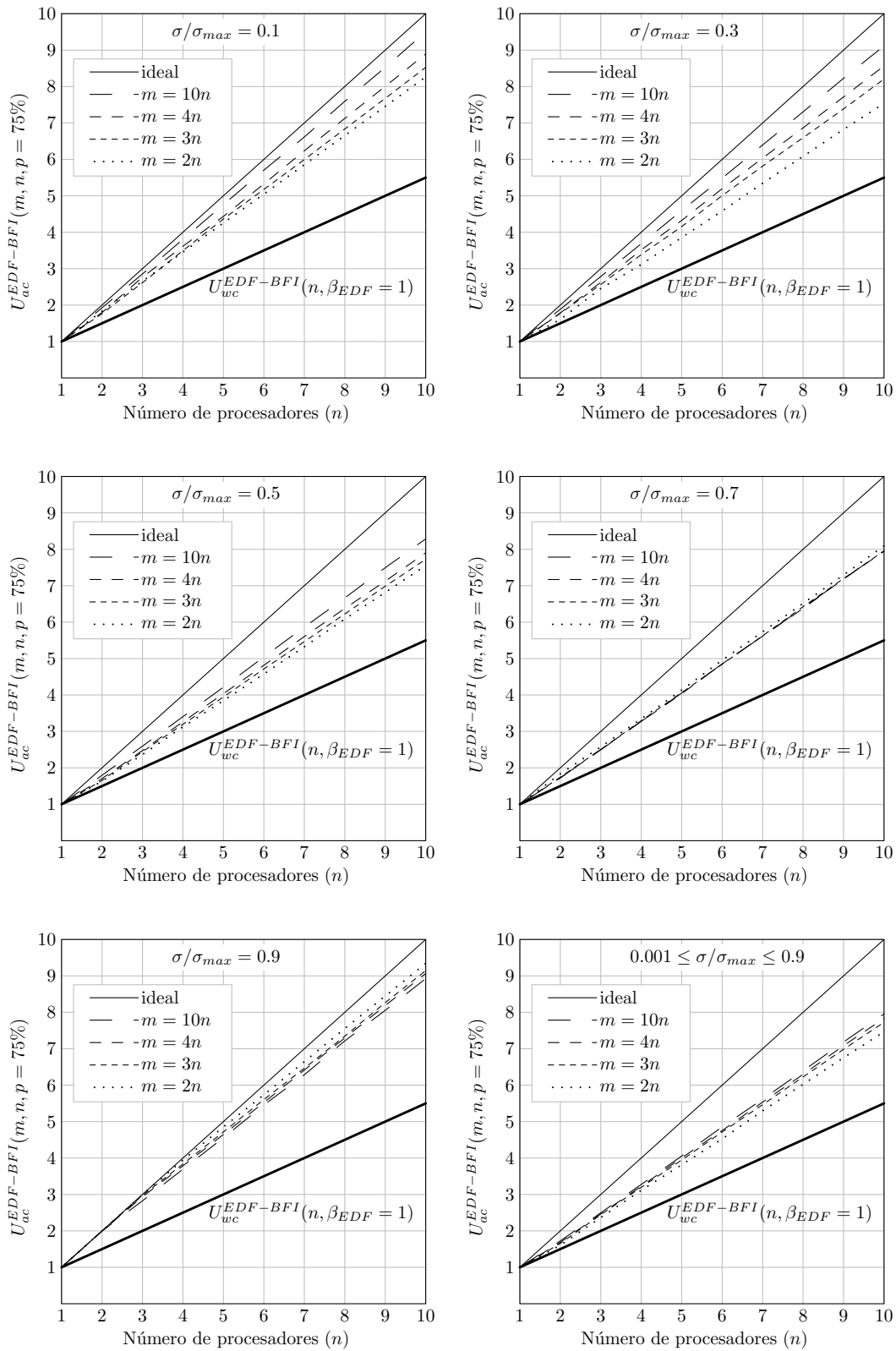


Figura 3.30: Límites de utilización para planificación EDF-BFI y $p = 75\%$.

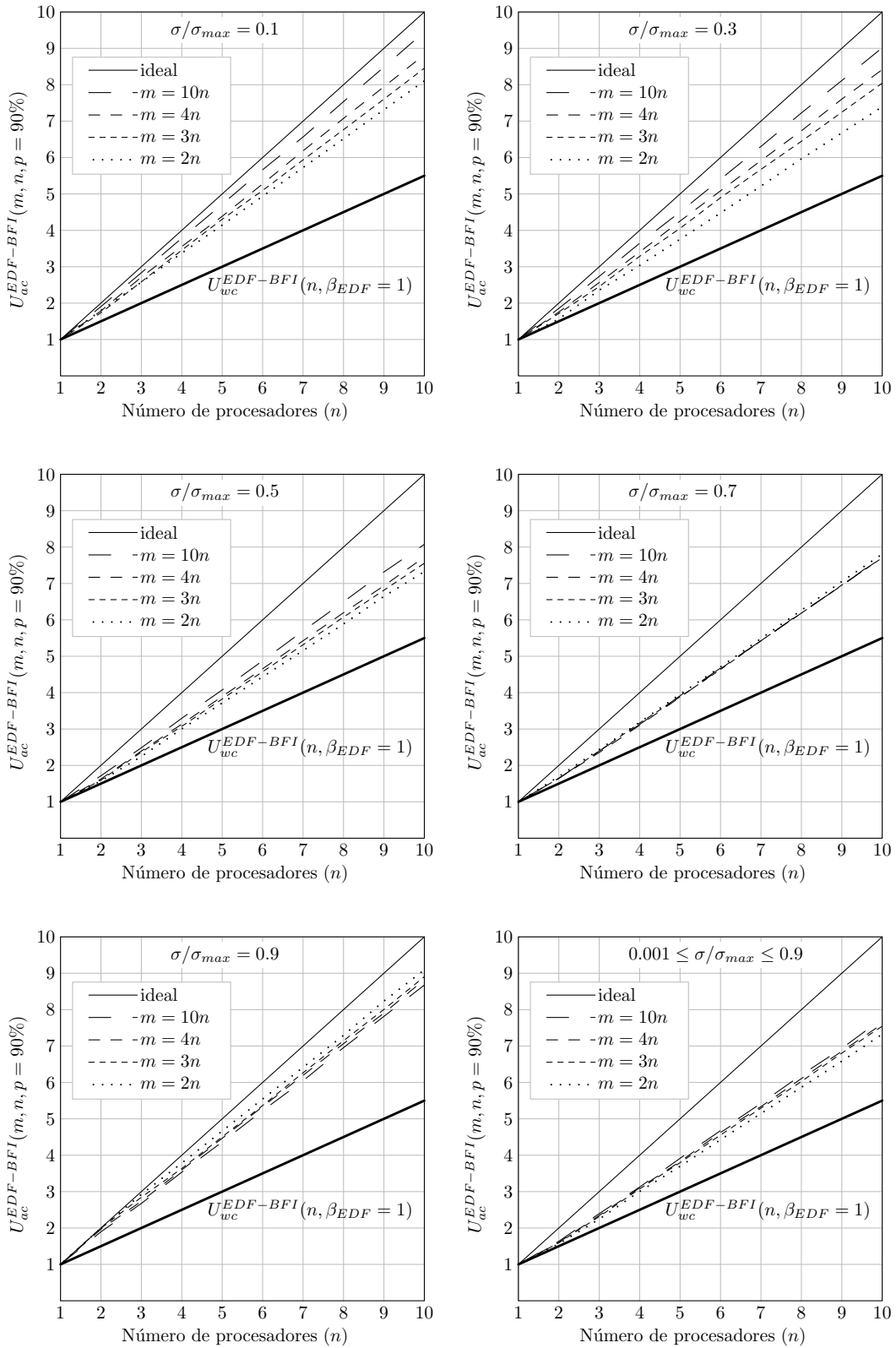


Figura 3.31: Límites de utilización para planificación EDF-BFI y $p = 90\%$.

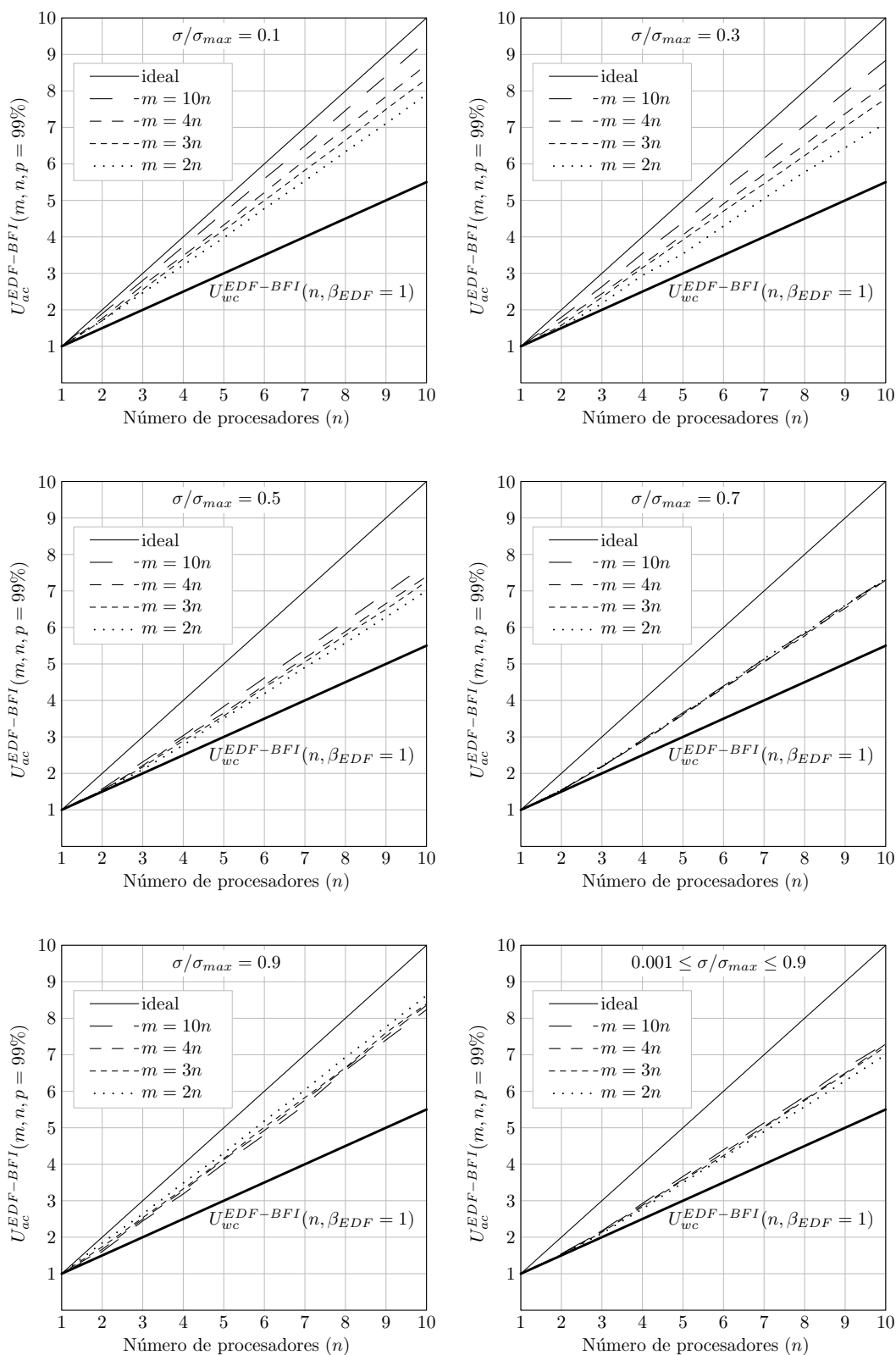


Figura 3.32: Límites de utilización para planificación EDF-BFI y $p = 99\%$.

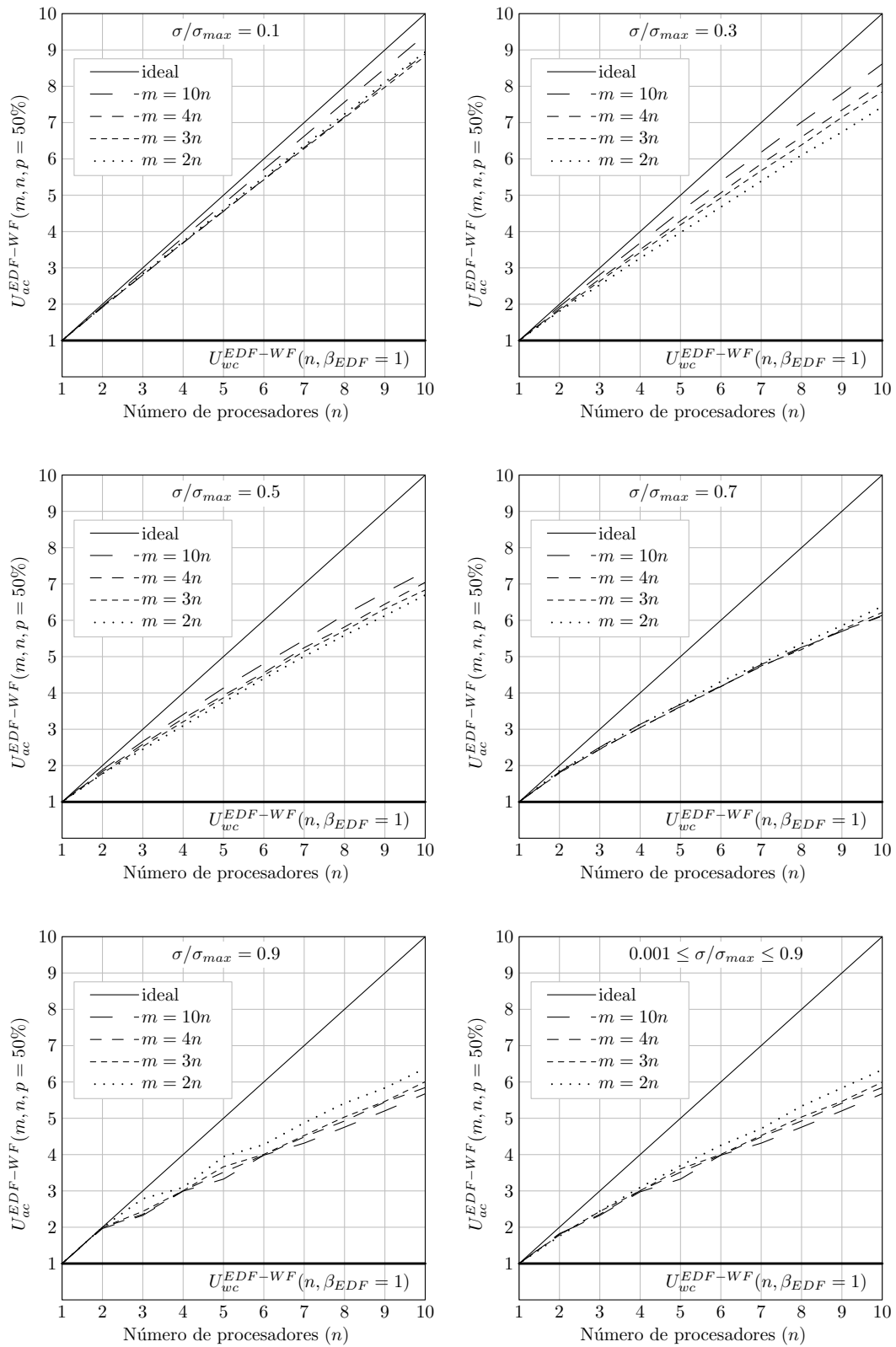


Figura 3.33: Límites de utilización para planificación EDF-WF y $p = 50\%$.

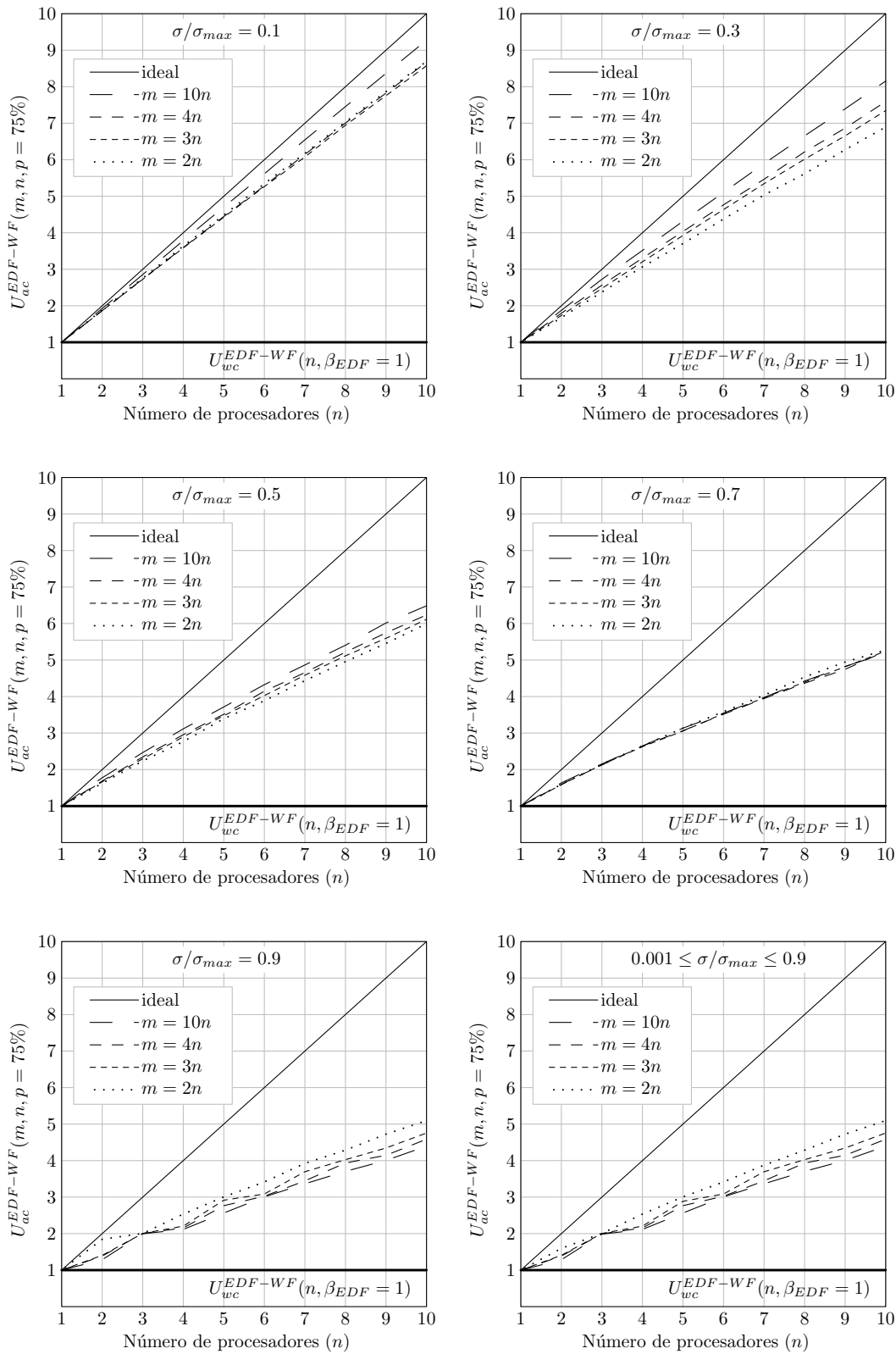


Figura 3.34: Límites de utilización para planificación EDF-WF y $p = 75\%$.

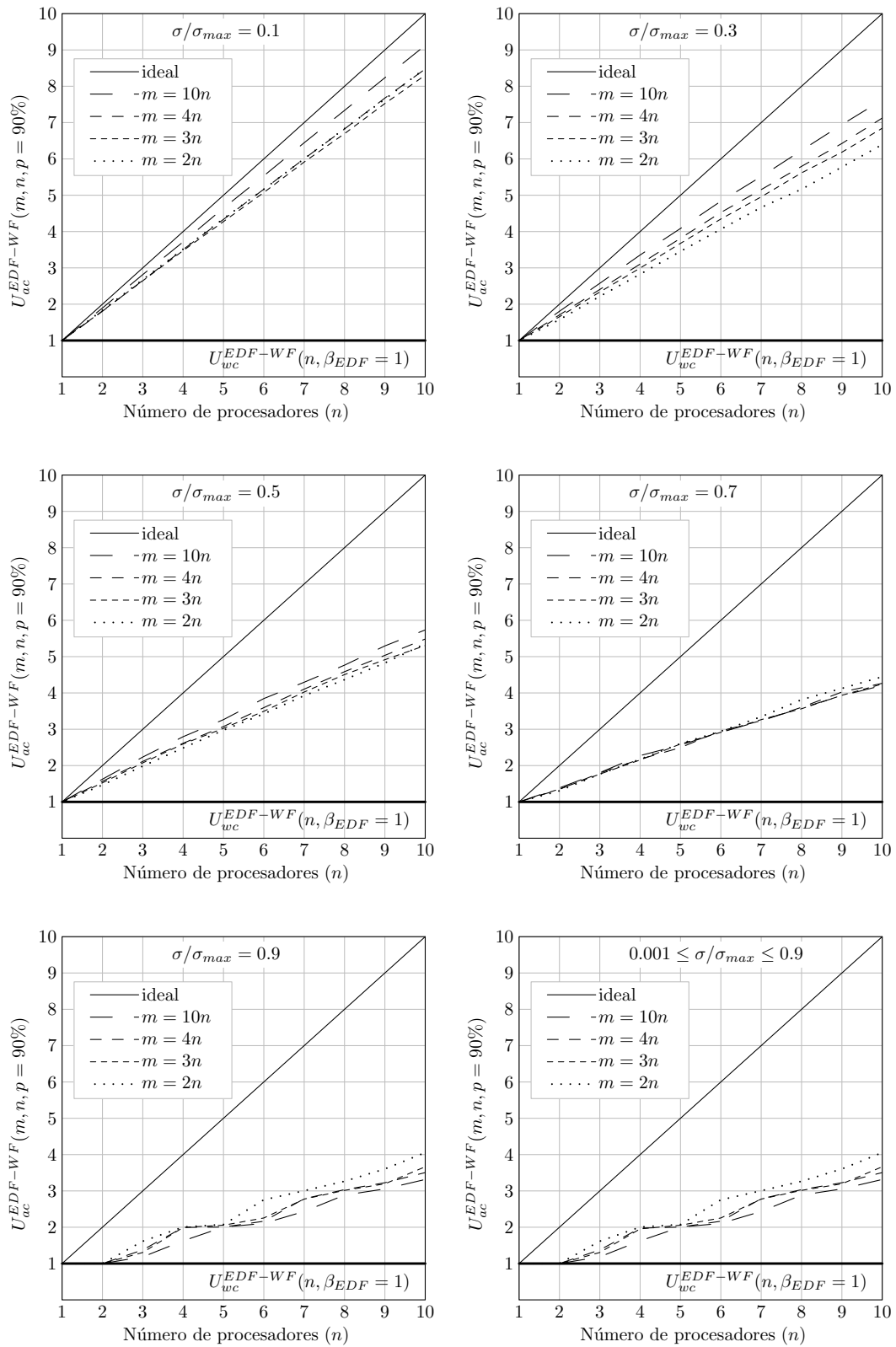


Figura 3.35: Límites de utilización para planificación EDF-WF y $p = 90\%$.

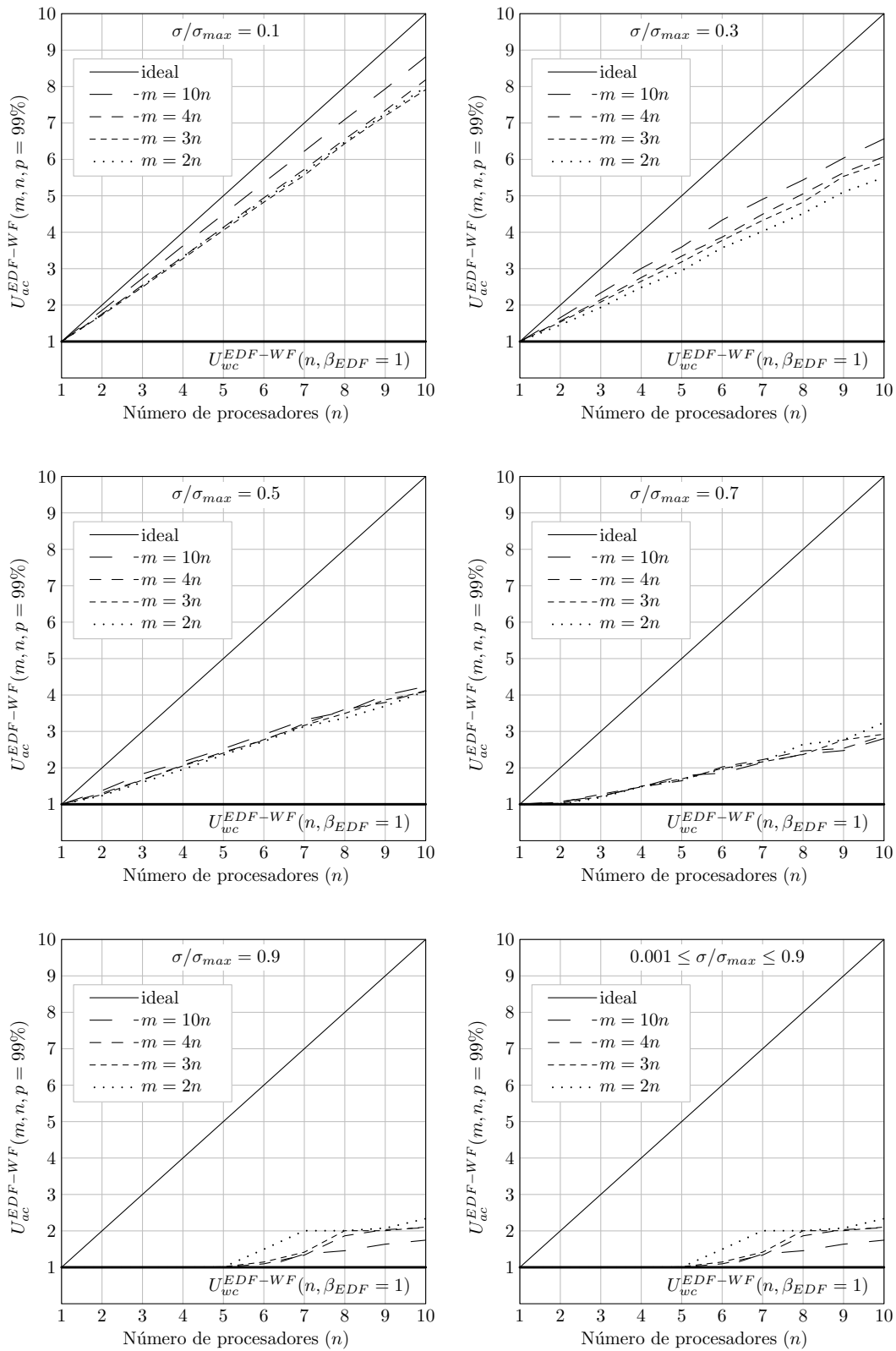


Figura 3.36: Límites de utilización para planificación EDF-WF y $p = 99\%$.

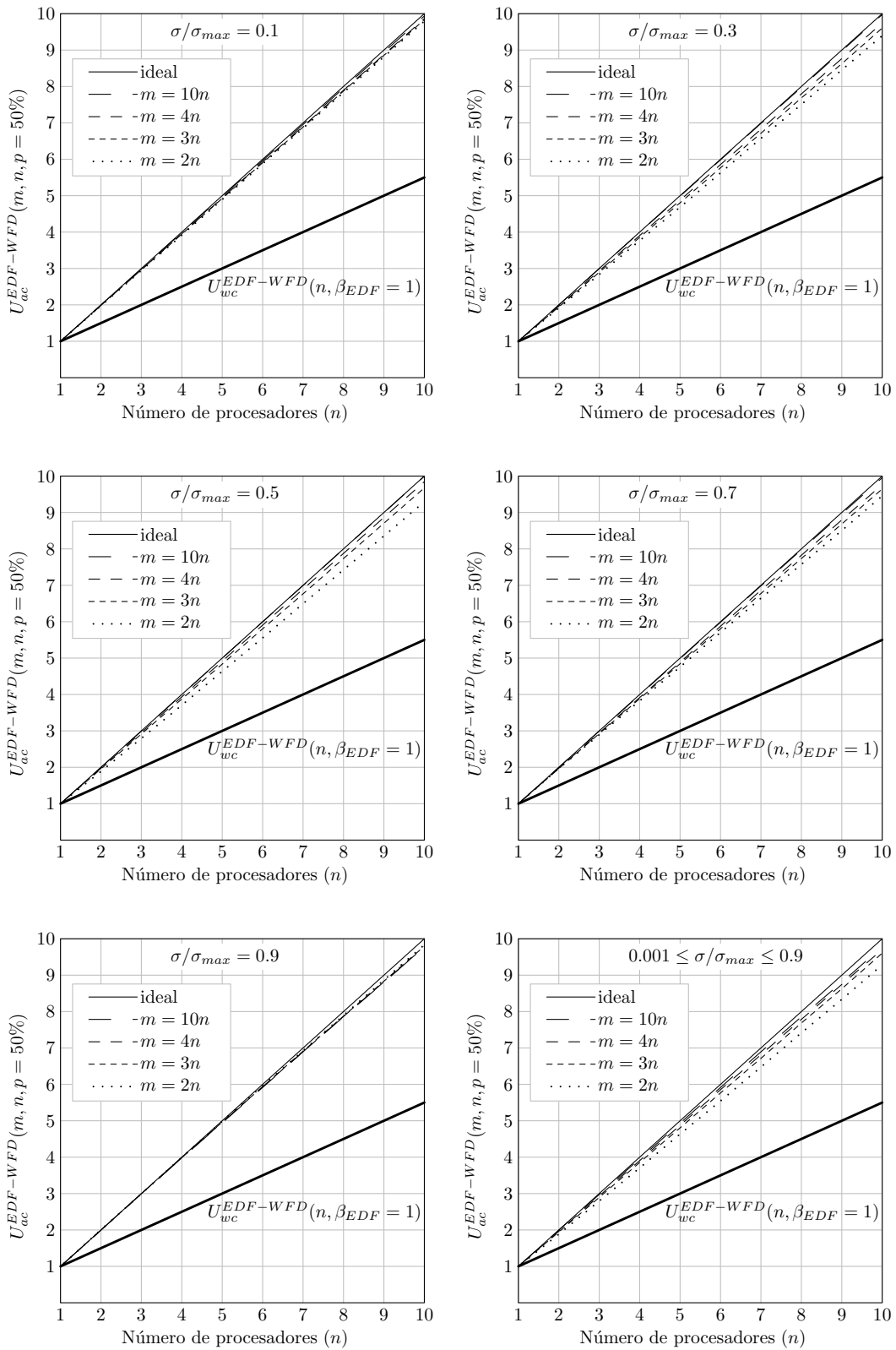


Figura 3.37: Límites de utilización para planificación EDF-WFD y $p = 50\%$.

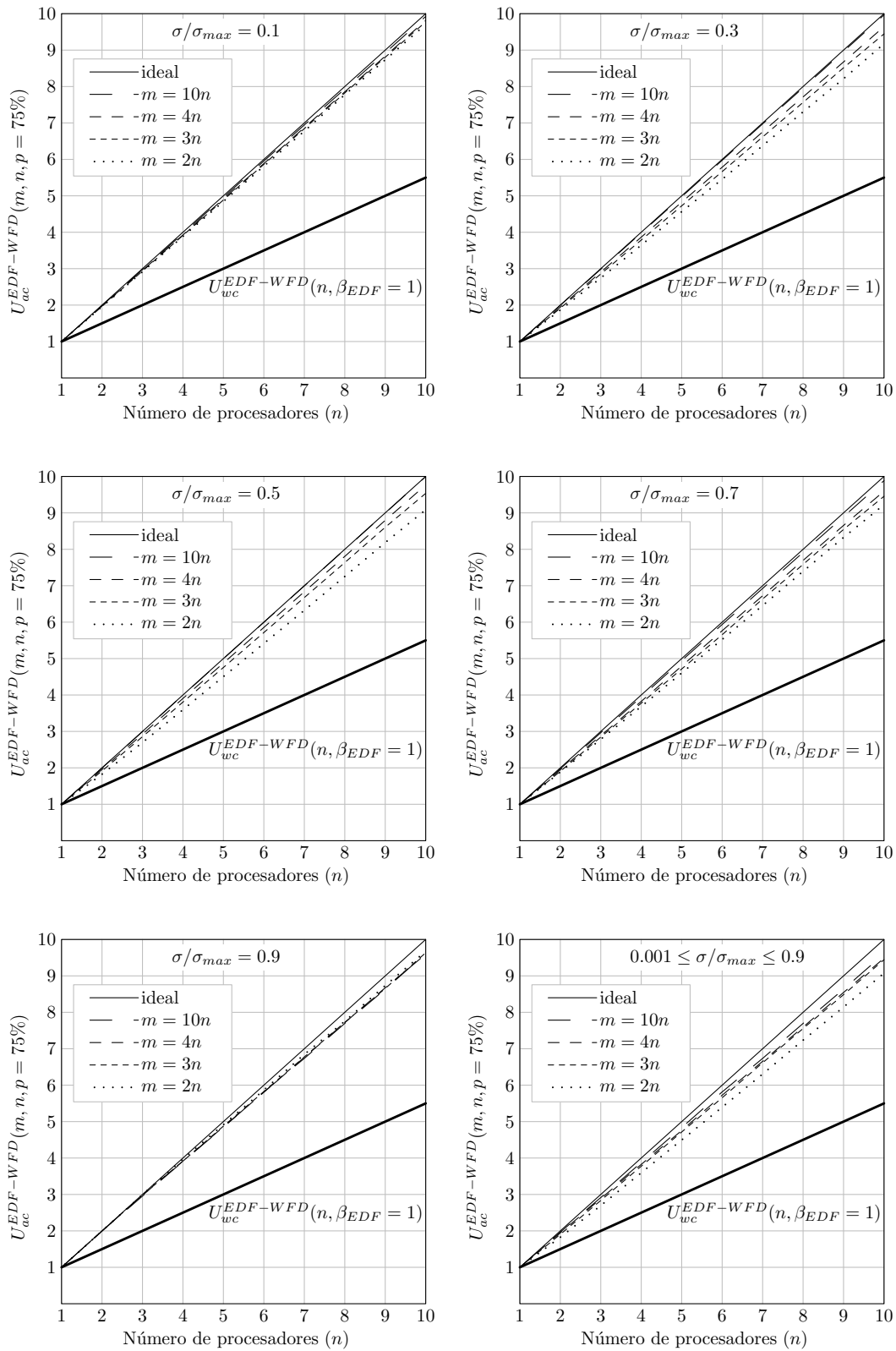


Figura 3.38: Límites de utilización para planificación EDF-WFD y $p = 75\%$.

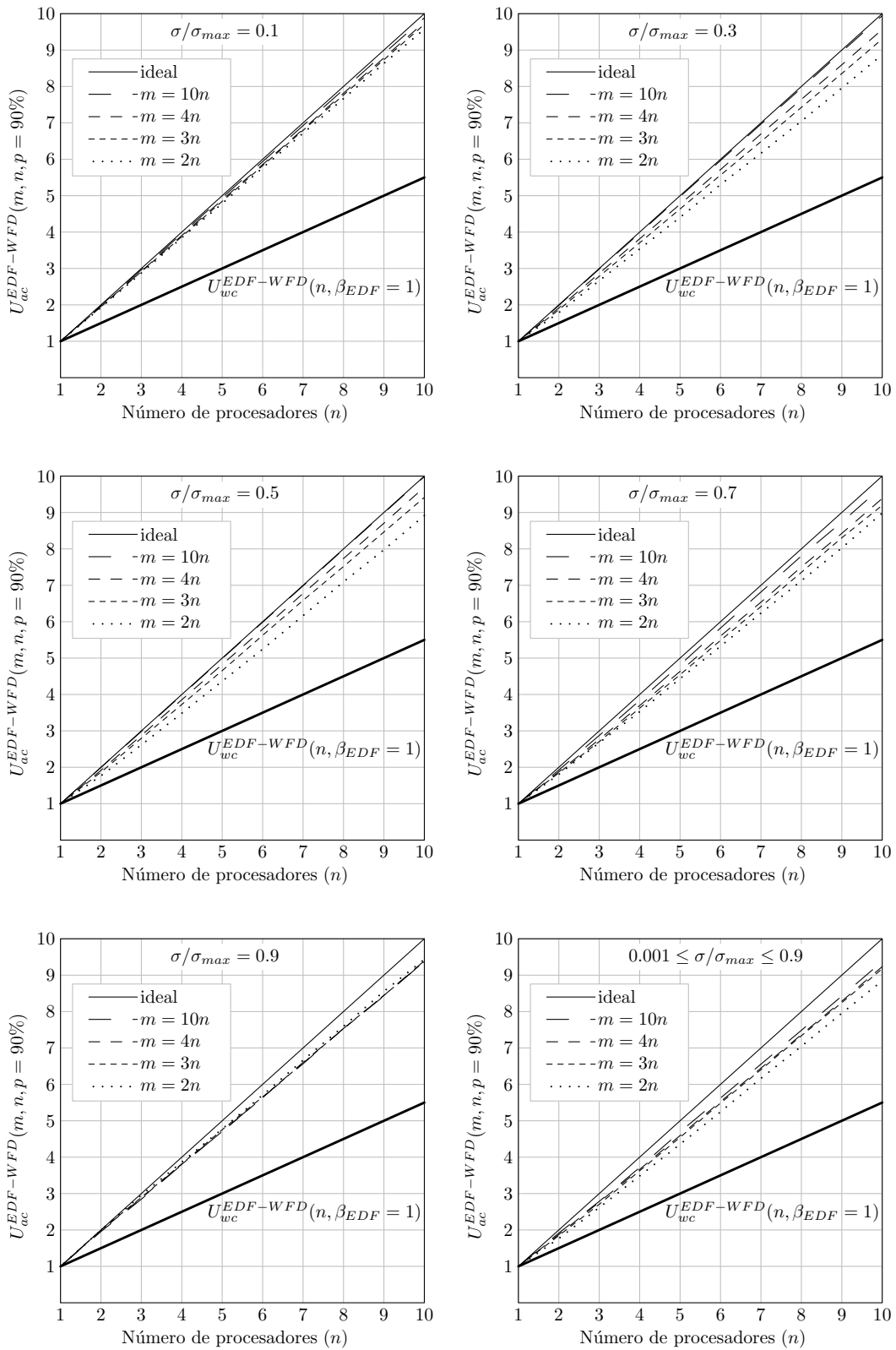


Figura 3.39: Límites de utilización para planificación EDF-WFD y $p = 90\%$.

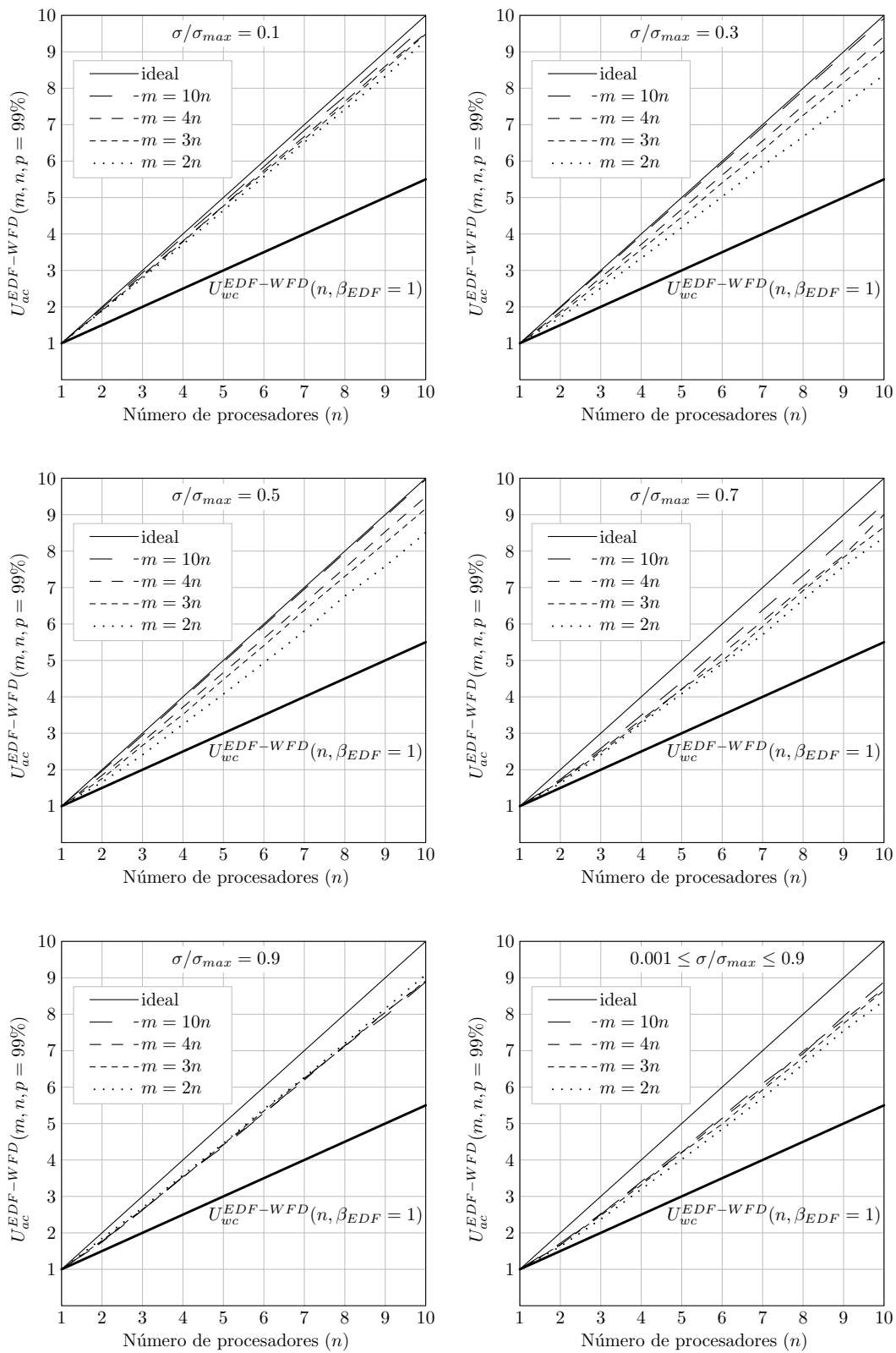


Figura 3.40: Límites de utilización para planificación EDF-WFD y $p = 99\%$.

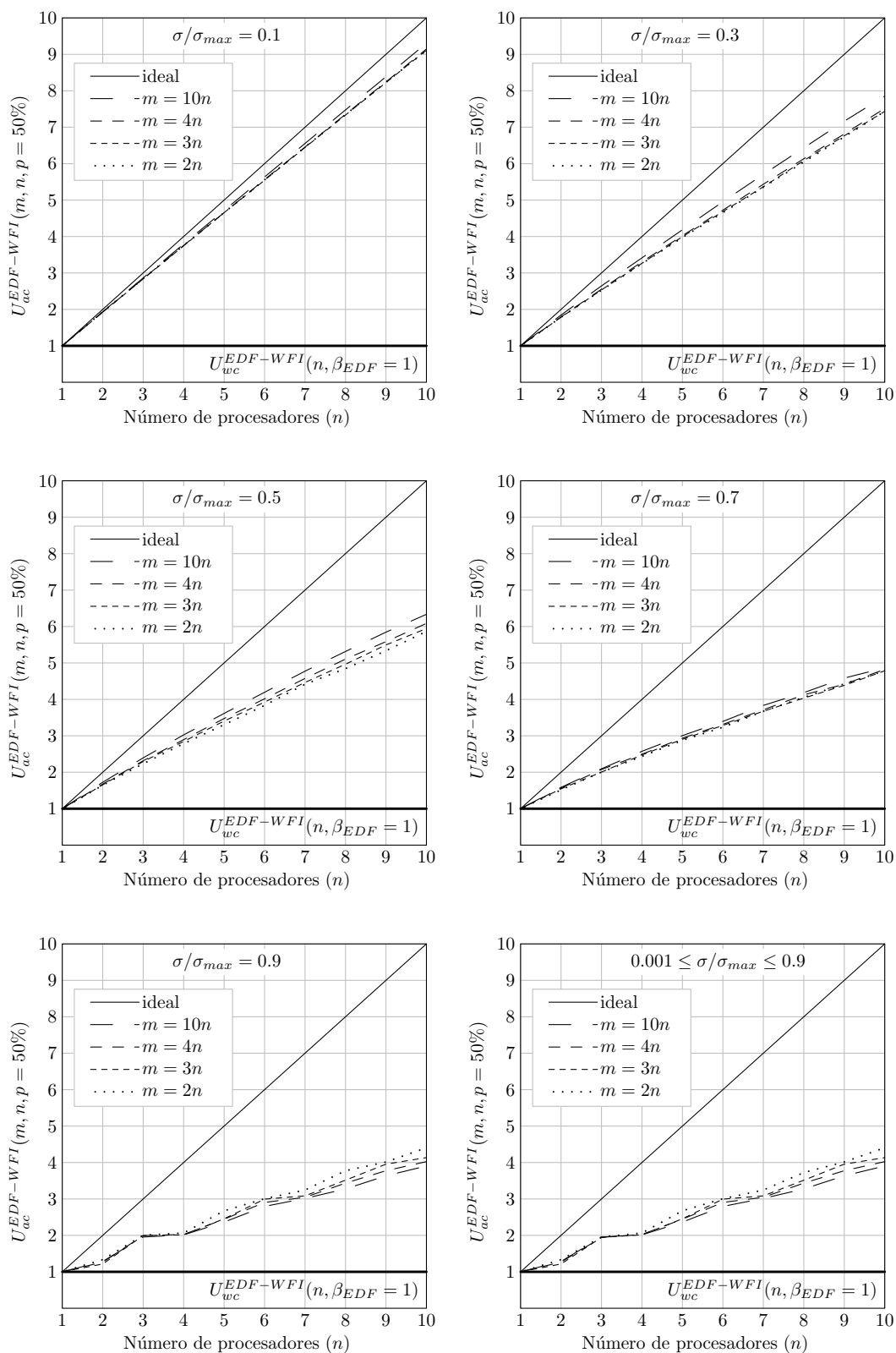


Figura 3.41: Límites de utilización para planificación EDF-WFI y $p = 50\%$.

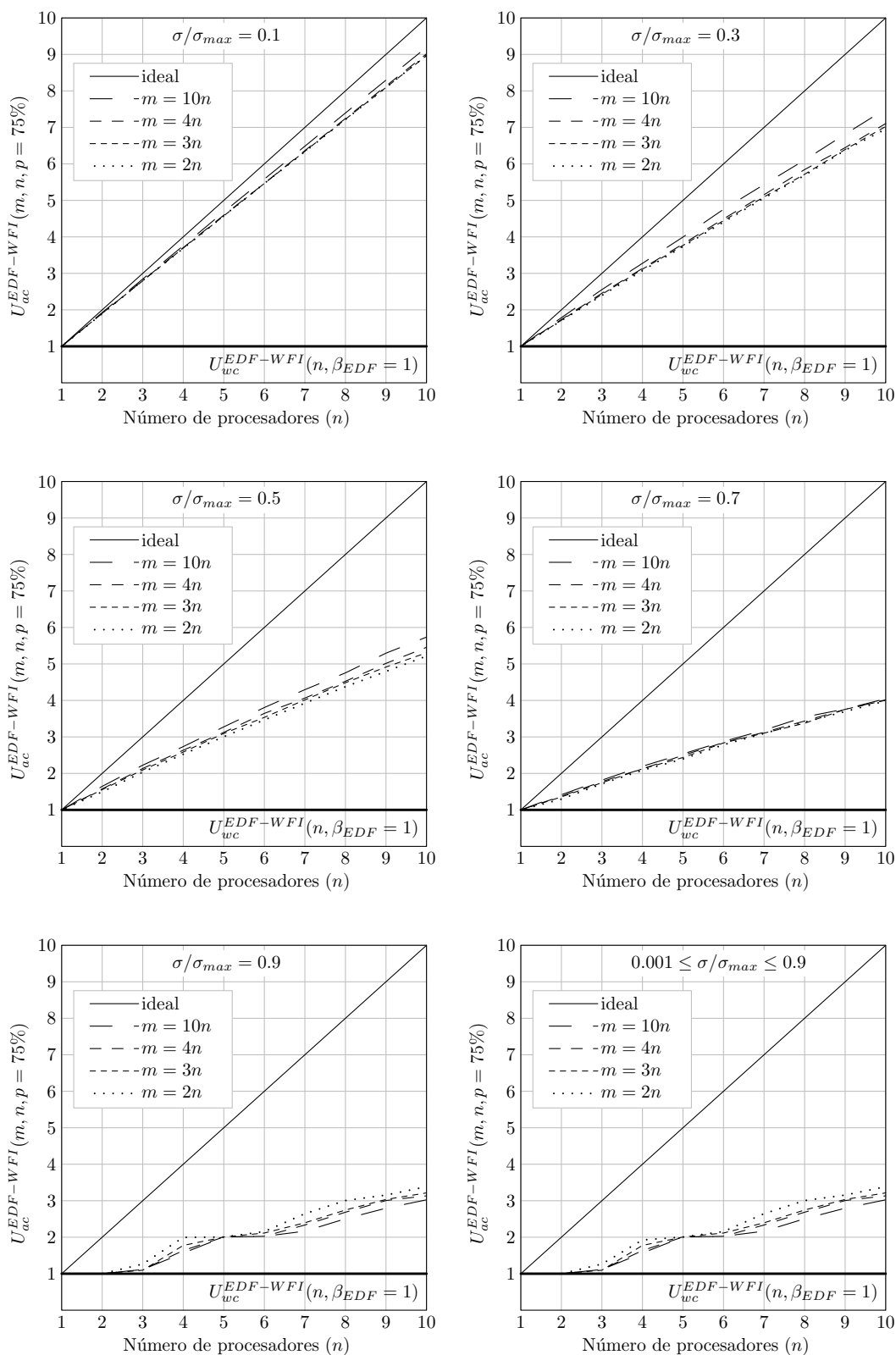


Figura 3.42: Límites de utilización para planificación EDF-WFI y $p = 75\%$.

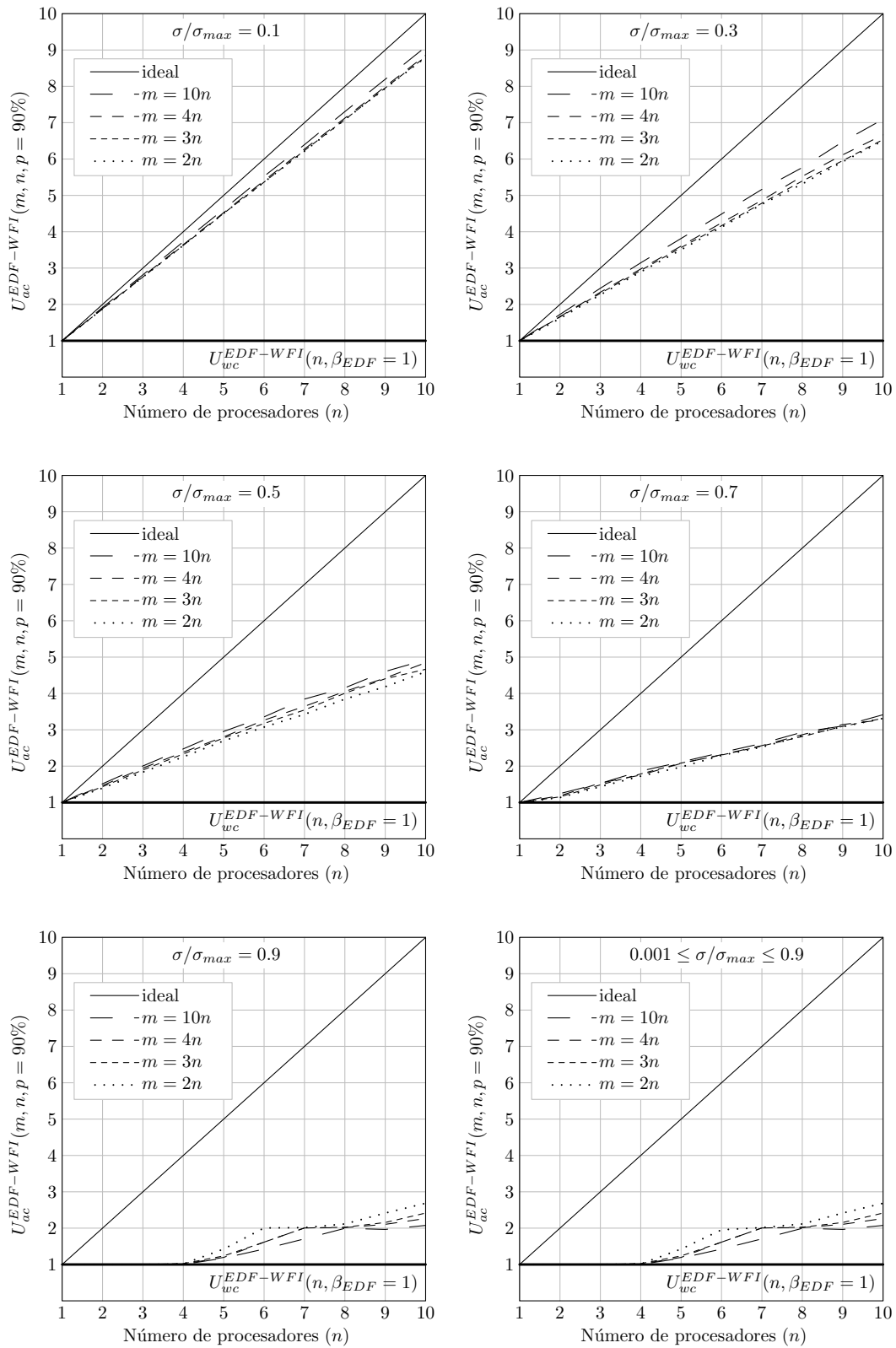


Figura 3.43: Límites de utilización para planificación EDF-WFI y $p = 90\%$.

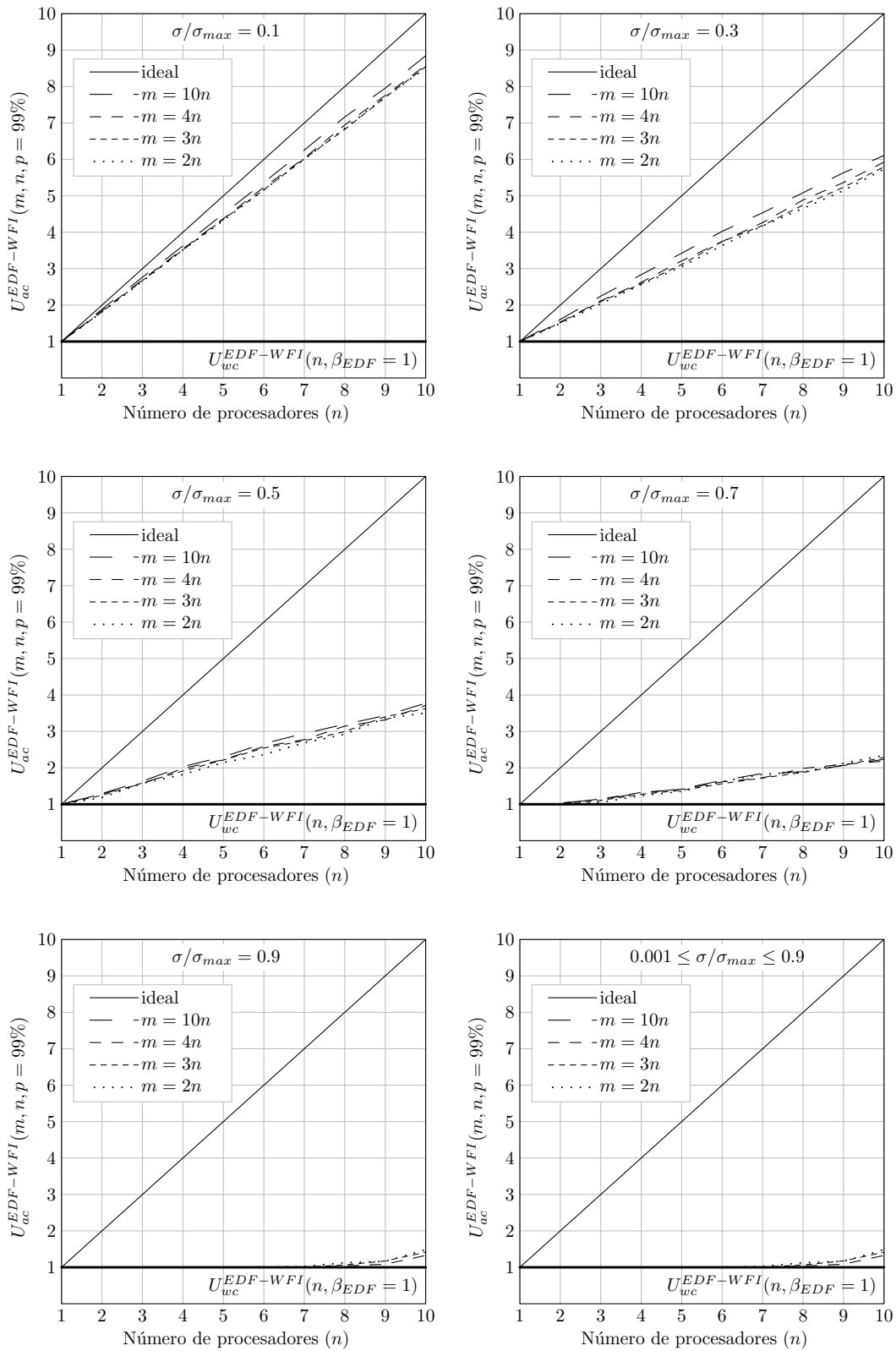


Figura 3.44: Límites de utilización para planificación EDF-WFI y $p = 99\%$.

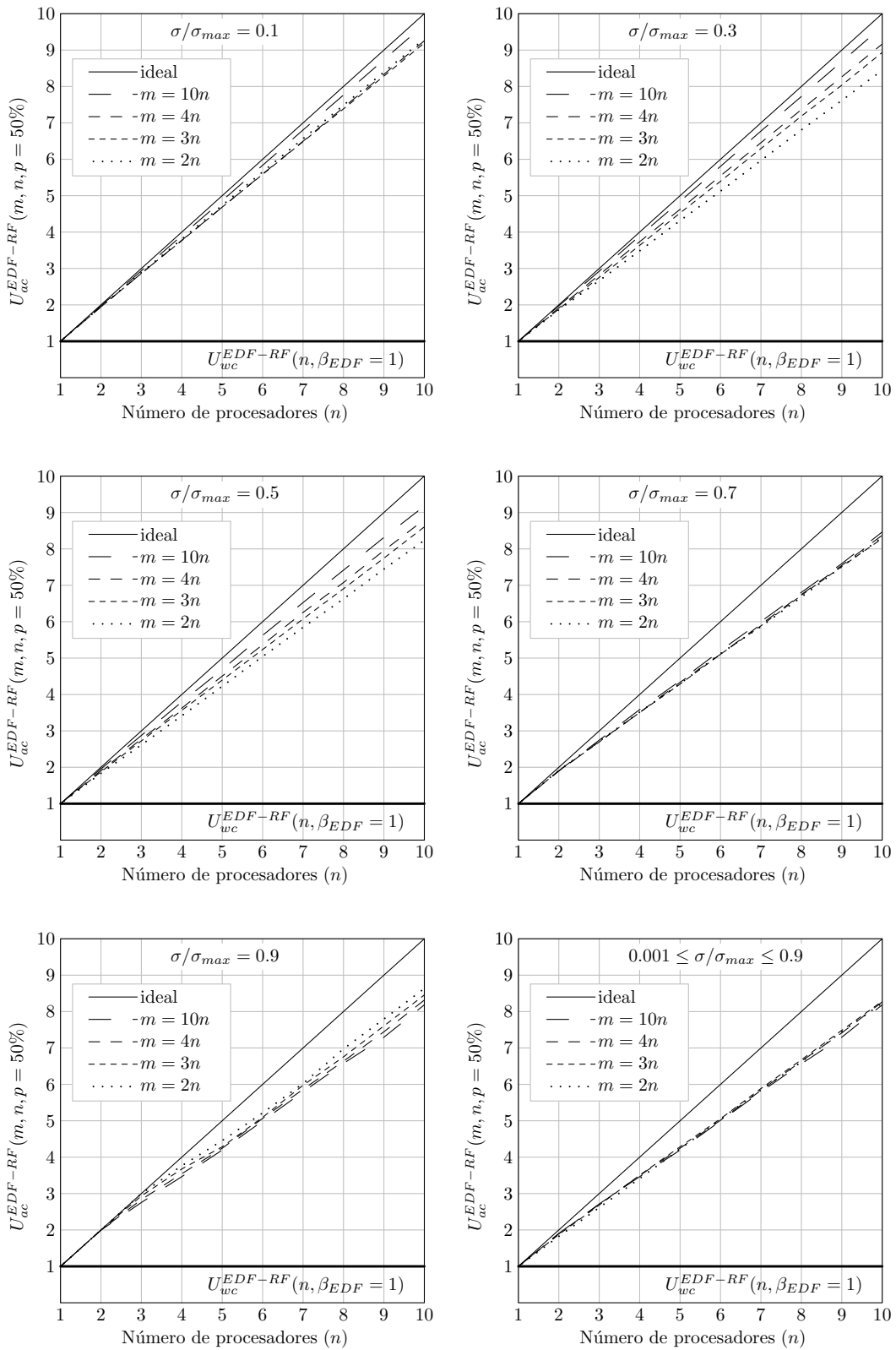


Figura 3.45: Límites de utilización para planificación EDF-RF y $p = 50\%$.

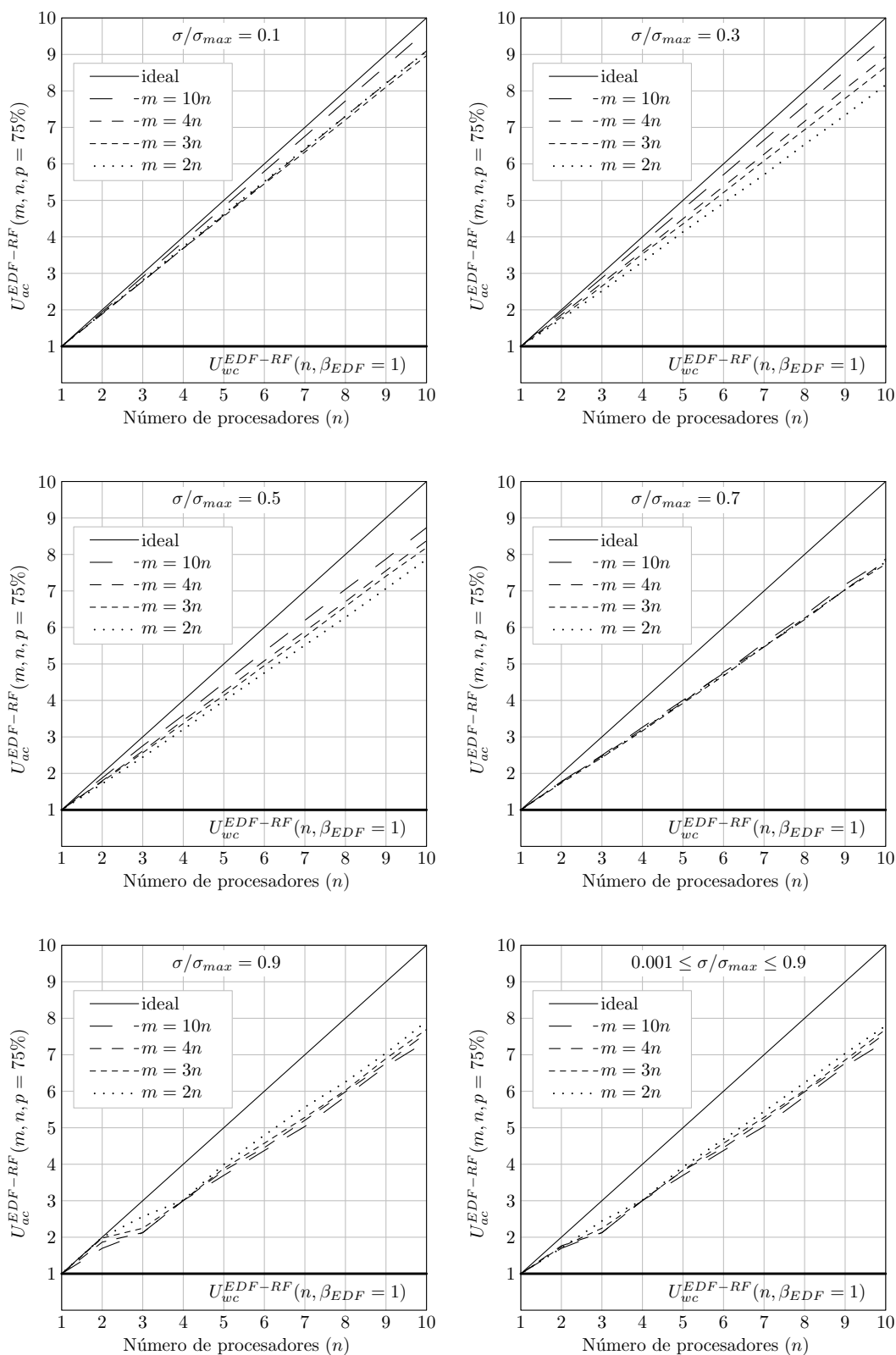
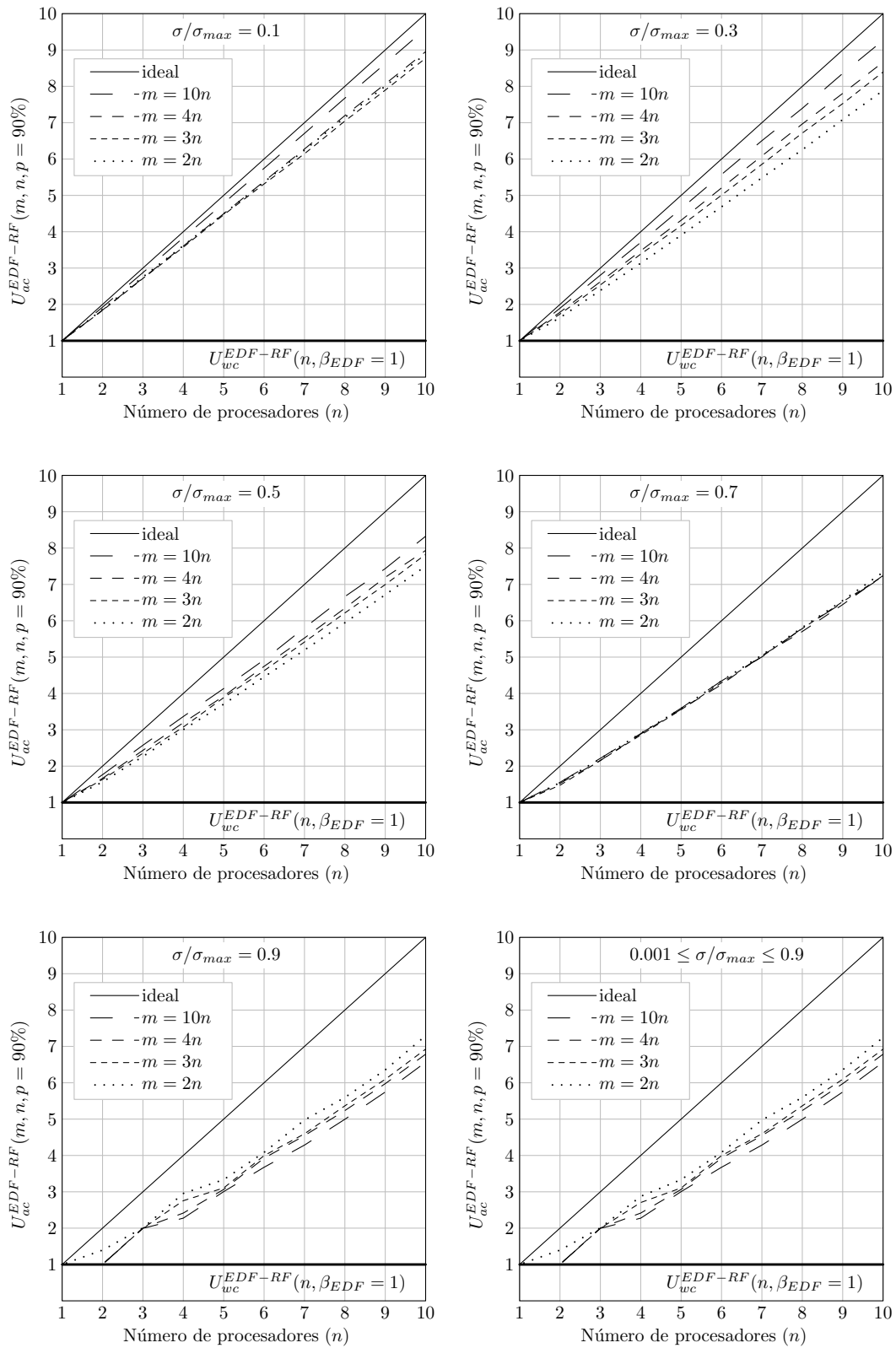


Figura 3.46: Límites de utilización para planificación EDF-RF y $p = 75\%$.

Figura 3.47: Límites de utilización para planificación EDF-RF y $p = 90\%$.

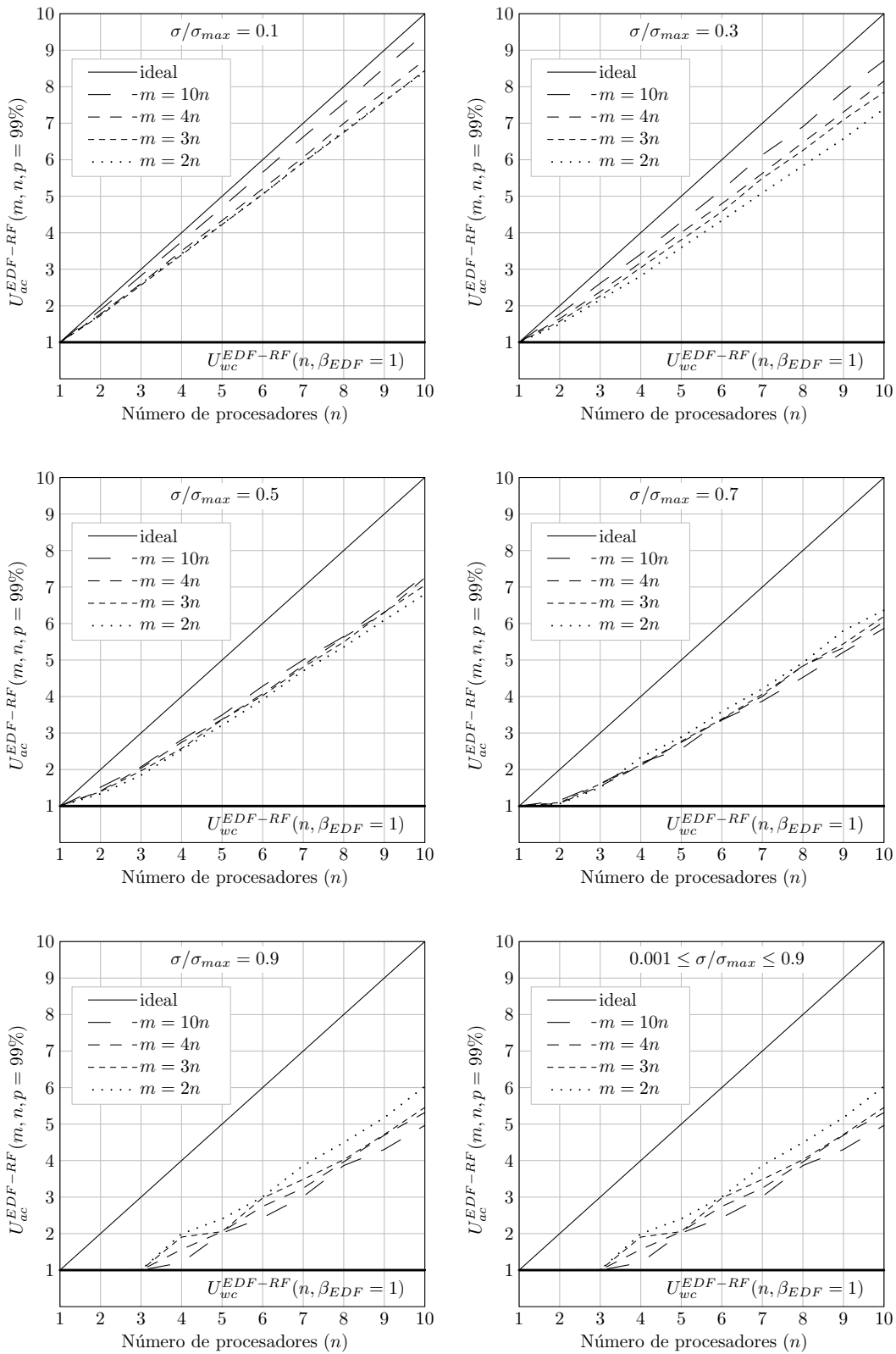


Figura 3.48: Límites de utilización para planificación EDF-RF y $p = 99\%$.

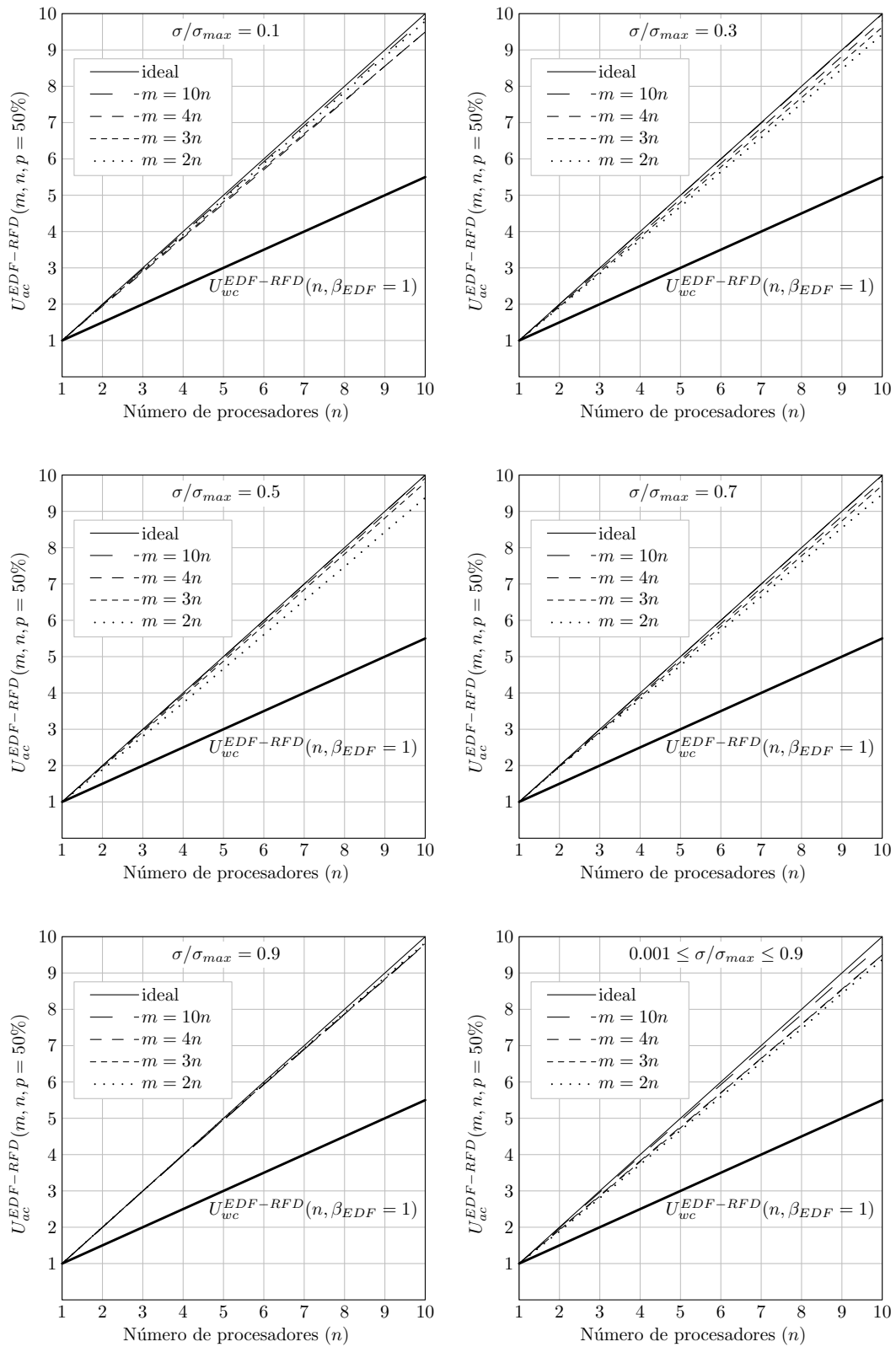


Figura 3.49: Límites de utilización para planificación EDF-RFD y $p = 50\%$.

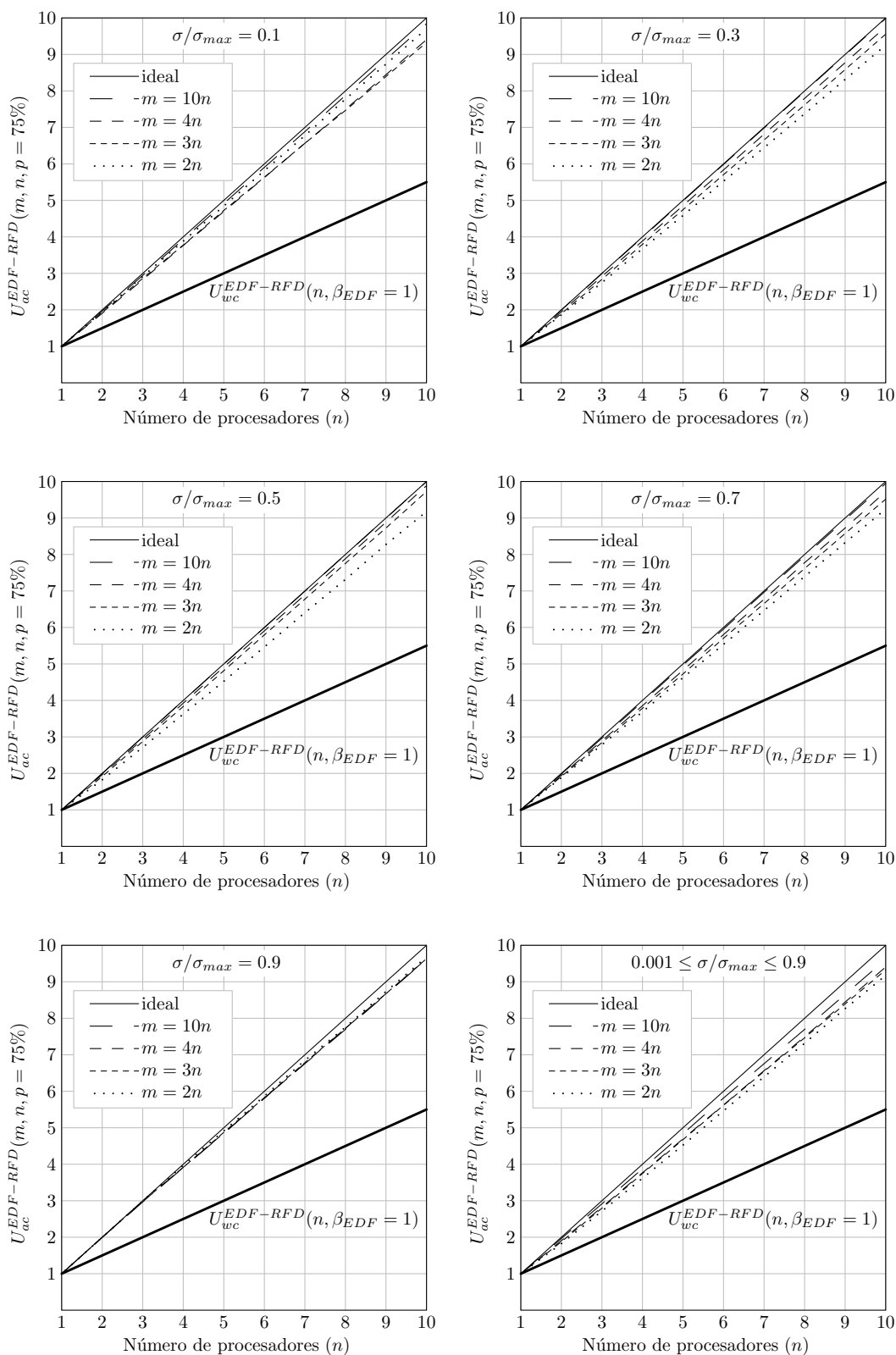


Figura 3.50: Límites de utilización para planificación EDF-RFD y $p = 75\%$.

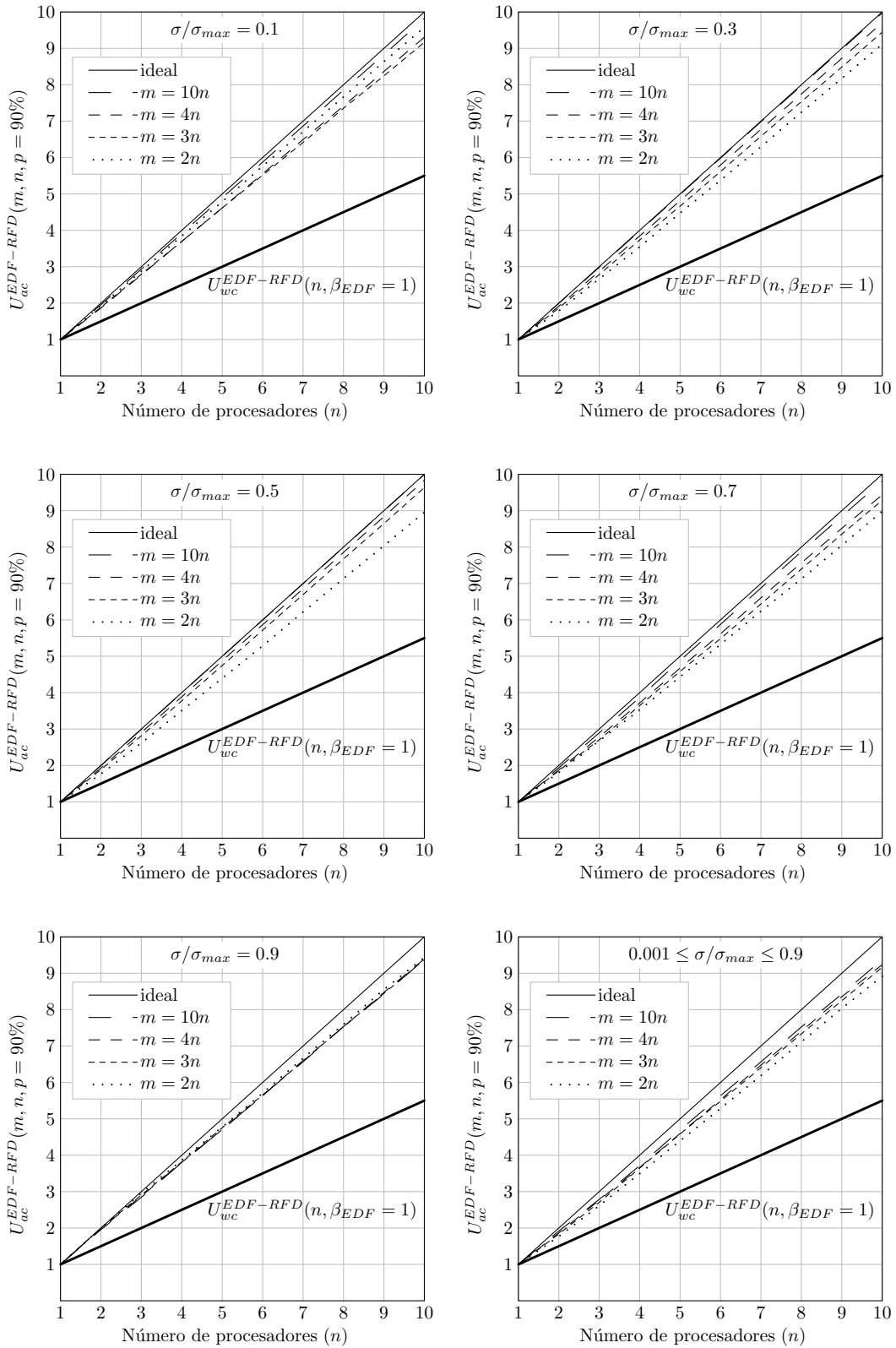


Figura 3.51: Límites de utilización para planificación EDF-RFD y $p = 90\%$.

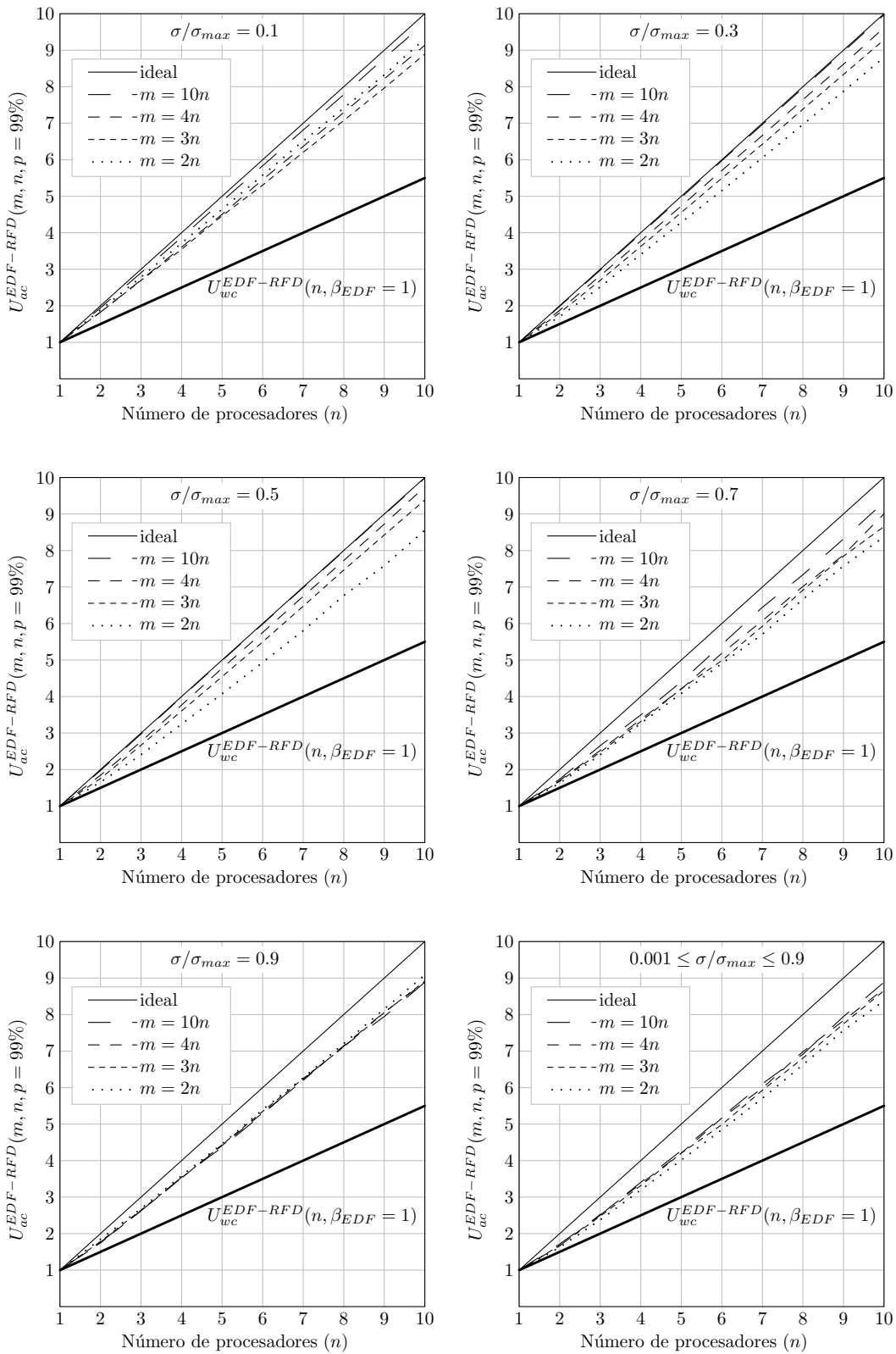


Figura 3.52: Límites de utilización para planificación EDF-RFD y $p = 99\%$.

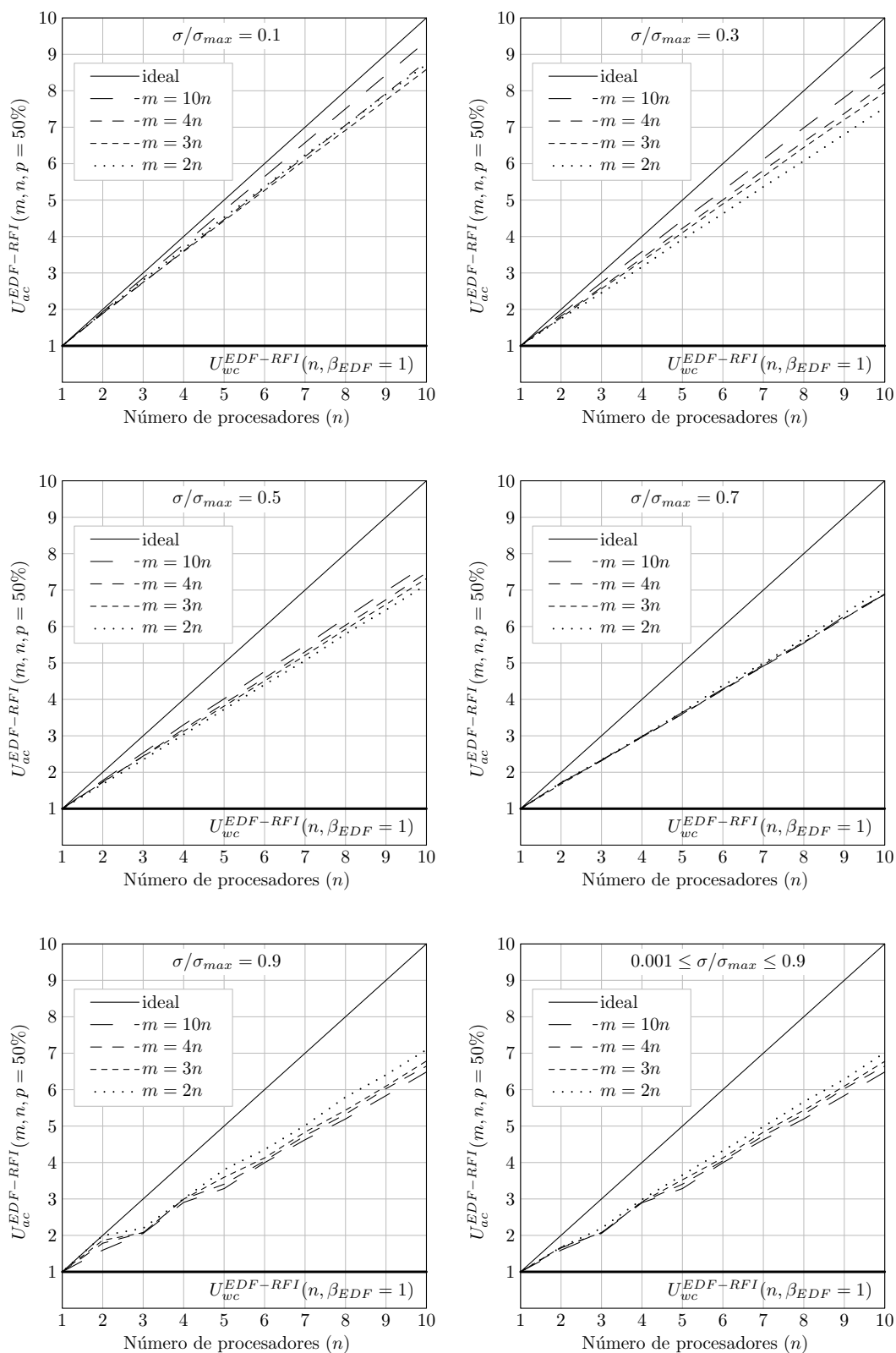


Figura 3.53: Límites de utilización para planificación EDF-RFI y $p = 50\%$.

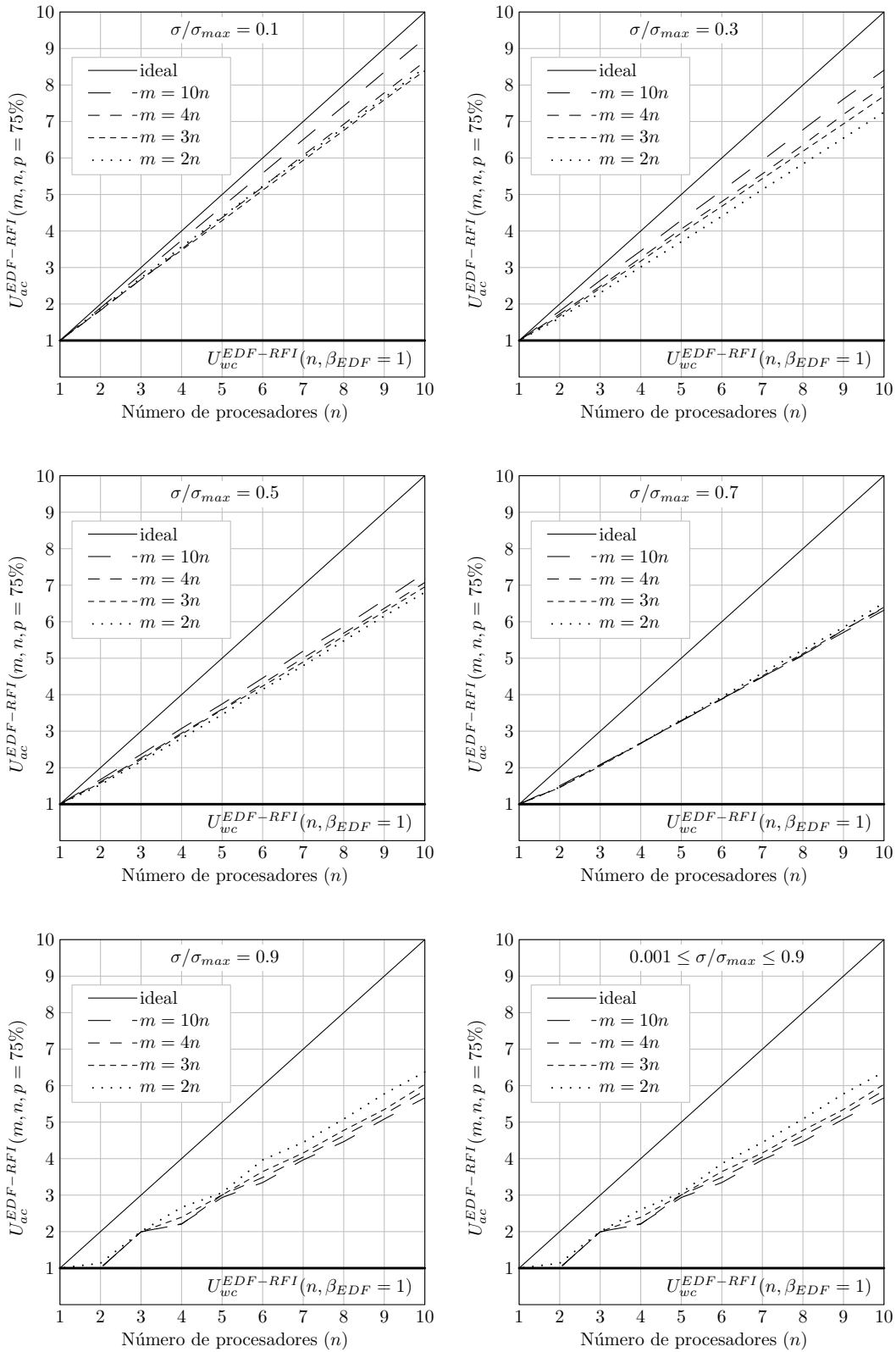


Figura 3.54: Límites de utilización para planificación EDF-RFI y $p = 75\%$.

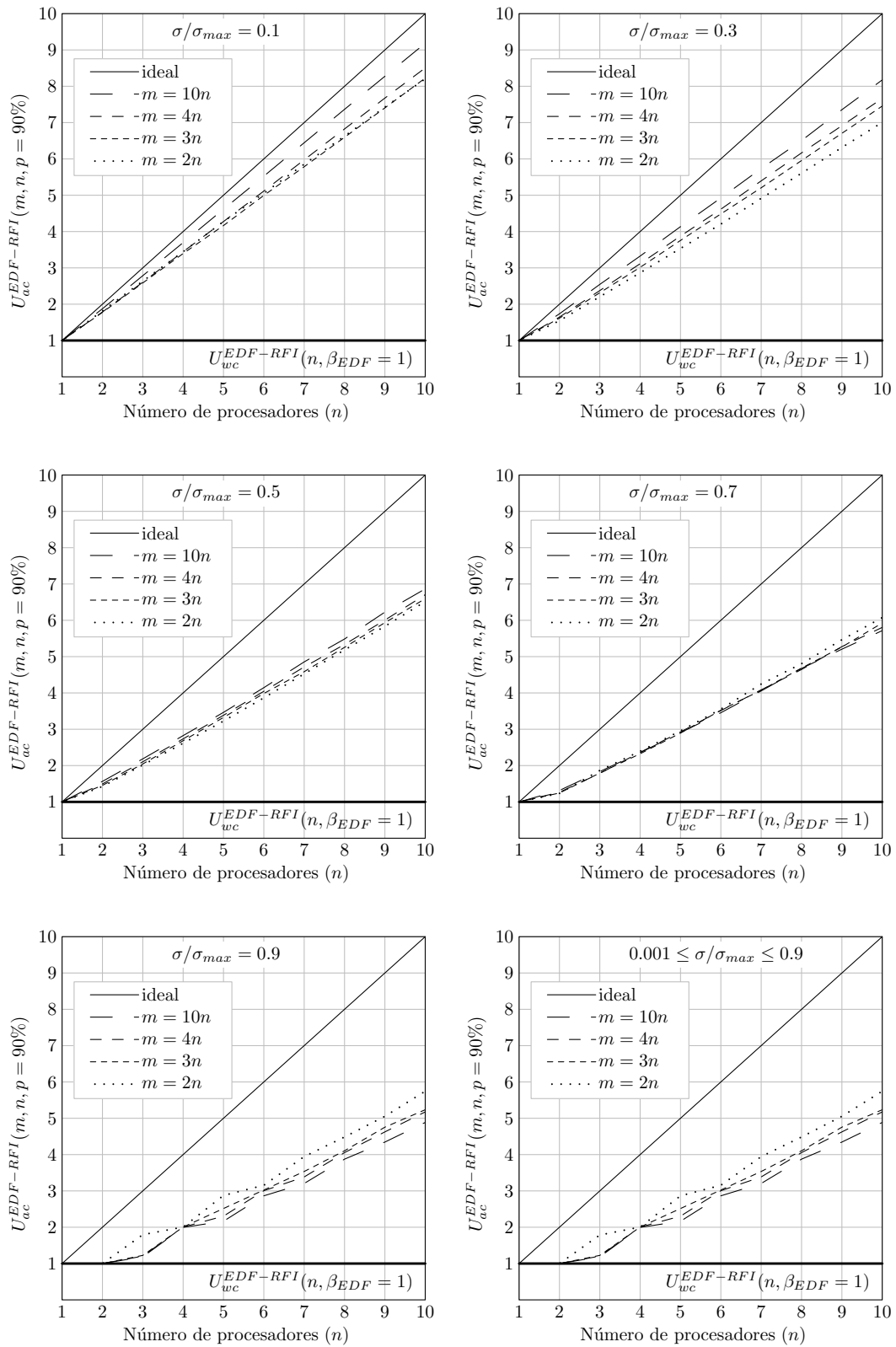


Figura 3.55: Límites de utilización para planificación EDF-RFI y $p = 90\%$.

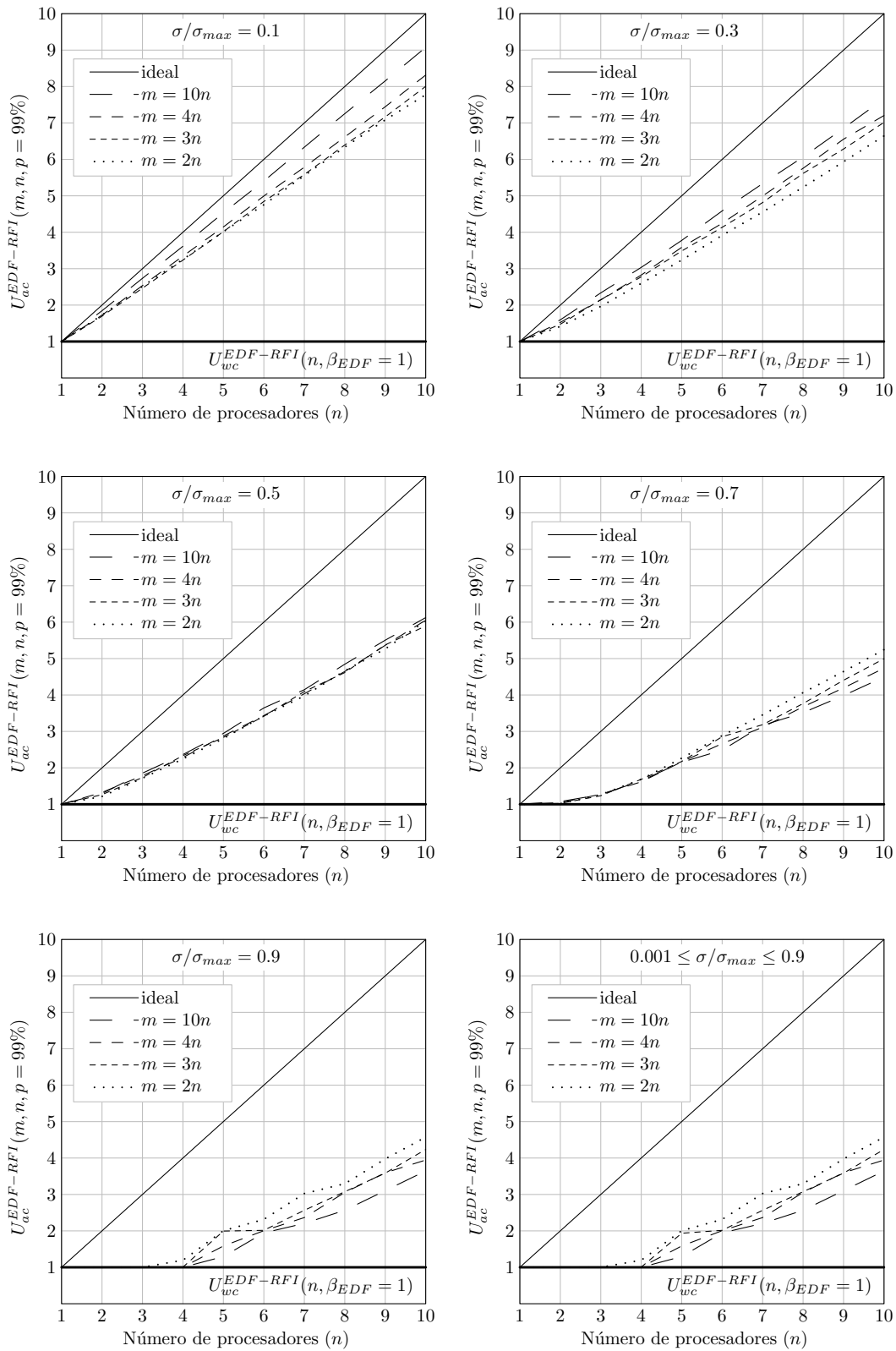


Figura 3.56: Límites de utilización para planificación EDF-RFI y $p = 99\%$.

3.5 Comparación de los algoritmos de asignación de tareas

A continuación se muestra la información mas relevante obtenida mediante observación de las gráficas obtenidas en el apartado anterior. Esta información es útil para la comparación de los diferentes algoritmos de asignación de tareas.

- El límite de utilización estadístico más elevado se obtiene para los algoritmos FFD, BFD, WFD y RFD. Por lo tanto, la ordenación de las tareas en sentido de factores de utilización decrecientes mejora la planificabilidad. Este resultado se ajusta a la idea intuitiva de que es preferible colocar los “objetos grandes” antes que los “objetos pequeños”. El inconveniente de la ordenación de tareas es que no puede realizarse en línea.
- El límite de utilización estadístico es prácticamente idéntico para todos los algoritmos estudiados con ordenación de tareas en el sentido de factores de utilización decrecientes. La consecuencia de esta observación es la superioridad del algoritmo FFD sobre los algoritmos BFD, WFD y RFD, pues estos últimos llevan asociado en general un coste computacional mayor.
- La ordenación de tareas en sentido de factores de utilización crecientes reduce el límite de utilización estadístico de forma apreciable, en comparación con la asignación de tareas sin ordenación. Por lo tanto, en este sentido los algoritmos FF, BF, WF y RF son superiores a los algoritmos FFI, BFI, WFI y RFI. Como además estos últimos llevan asociado en general un coste computacional mayor, y no son aplicables en línea, definitivamente son inferiores a sus versiones sin ordenación.
- De todos los algoritmos sin ordenación presentados, los algoritmos FF y BF poseen el límite de utilización mayor, especialmente cuando los factores de utilización de las tareas son bastante diferentes (es decir, para valores elevados de σ). Además, los límites de utilización para FF son un “calco” de los límites de utilización para BF. Teniendo en cuenta el mayor coste computacional asociado en general a los algoritmos BF, WF y RF, resulta que el algoritmo FF es superior a todos ellos.
- Se observa que todos los límites de utilización estadísticos se acercan a los límites de utilización absolutos a medida que el parámetro p crece.
- Los límites de utilización estadísticos asociados a los algoritmos WF, RF, WFI y RFI tienden a uno cuando la desviación típica tiende a su valor máximo. La explicación es clara, con desviaciones típicas elevadas los factores de utilización son todos o casi cero o casi uno. Por lo tanto se está cerca del caso peor, presentado en el teorema 7.
- Salvo para desviaciones típicas elevadas con asignación WF, RF, WFI o RFI, los límites de utilización estadísticos son una función lineal del número de

procesadores. Esta linealidad también está presente en todos los límites de utilización absolutos.

La conclusión fundamental que puede extraerse de los resultados de simulación está en la superioridad de los algoritmos de asignación FF y FFD sobre todos los demás. El algoritmo FFD es superior en términos de planificabilidad al algoritmo FF, pero no obstante su coste computacional es mayor y no puede aplicarse en línea. Además, los algoritmos FF y FFD tienen el límite de utilización absoluto más elevado, idéntico al del algoritmo de asignación de tareas óptimo.

Antes de finalizar este capítulo quedan dos cuestiones por resolver:

- ¿Merece la pena buscar algoritmos de asignación de tareas que mejoren el límite de utilización estadístico asociado al algoritmo FFD?
- ¿Merece la pena buscar algoritmos en línea de asignación de tareas que mejoren el límite de utilización estadístico asociado al algoritmo FF?

Para dar una respuesta a la primera pregunta se han realizado experimentos de simulación análogos a los en el apartado anterior, empleando como algoritmo de asignación de tareas el algoritmo óptimo. No obstante, en los experimentos de simulación se ha restringido las posibles combinaciones de número de tareas y número de procesadores a las siguientes: $(m = 4, n = 2)$, $(m = 6, n = 2)$, $(m = 8, n = 2)$, $(m = 6, n = 3)$, $(m = 9, n = 3)$, $(m = 12, n = 3)$, $(m = 8, n = 4)$, y $(m = 12, n = 4)$. El objetivo de la restricción es limitar el coste computacional a valores razonables. Las figuras 3.57 a 3.60 muestran los resultados obtenidos.

En el eje de ordenadas se representa la diferencia entre los límites de utilización estadísticos empleando los algoritmos de asignación FFD y OPT (óptimo). El resultado se ha dividido por el número de procesadores con la intención de mostrar la mejora por procesador obtenida empleando el algoritmo de asignación óptimo. Cada una de las subfiguras se corresponde con una pareja de valores (σ, p) diferente. La excepción se encuentra en las subfiguras etiquetadas con $0'001 \leq \sigma/\sigma_{max} \leq 0'9$, las cuales se corresponden con el máximo de las curvas obtenidas para las desviaciones típicas $\sigma/\sigma_{max} = 0'001, 0'1, 0'2, \dots, 0'9$ y el mismo valor de p .

Las conclusiones obtenidas a partir de las figuras 3.57 a 3.60 se indican a continuación:

- La máxima mejora obtenida empleando el algoritmo óptimo con respecto al algoritmo FFD es del orden de 0'05, 0'06, 0'07, y 0'09 para los valores $p = 50\%$, $p = 75\%$, $p = 90\%$, y $p = 99\%$ respectivamente.
- La mejora obtenida es prácticamente nula para $m = 2n$. Lo que hace ver que la mejora es baja para conjuntos de pocas tareas.
- Se observa una tendencia general de aumento de la mejora a medida que aumenta el número de procesadores, aunque cada vez con un ritmo de crecimiento menor.

- La mejora obtenida empleando distribuciones de factores de utilización con desviación típica mayor de 0'3 veces la máxima es insignificante. Esto hace ver que el algoritmo óptimo mejora al algoritmo FFD sólo en aquellos casos en los que los factores de utilización de las tareas son relativamente parecidos.

Para conjuntos de muchas tareas el algoritmo FFD tiene un comportamiento casi ideal, pues el factor de utilización de las tareas es probabilísticamente muy pequeño. Por lo tanto, en este caso el algoritmo óptimo no mejoraría casi nada los resultados obtenidos empleando el algoritmo de asignación FFD. No obstante, este resultado no se puede observar en las figuras, debido a las severas restricciones impuestas en cuanto al número de tareas y procesadores.

Como conclusión, se puede decir que la mejora obtenida empleando el algoritmo de asignación óptimo es en general escasa. Esta es significativa, del orden del 10% por procesador, únicamente cuando todos los factores de utilización son muy similares y se tiene un elevado número de procesadores. No obstante, este resultado debe tomarse con cautela, pues es básicamente la consecuencia de resultados de simulación con fuertes restricciones en cuanto al número de tareas y procesadores.

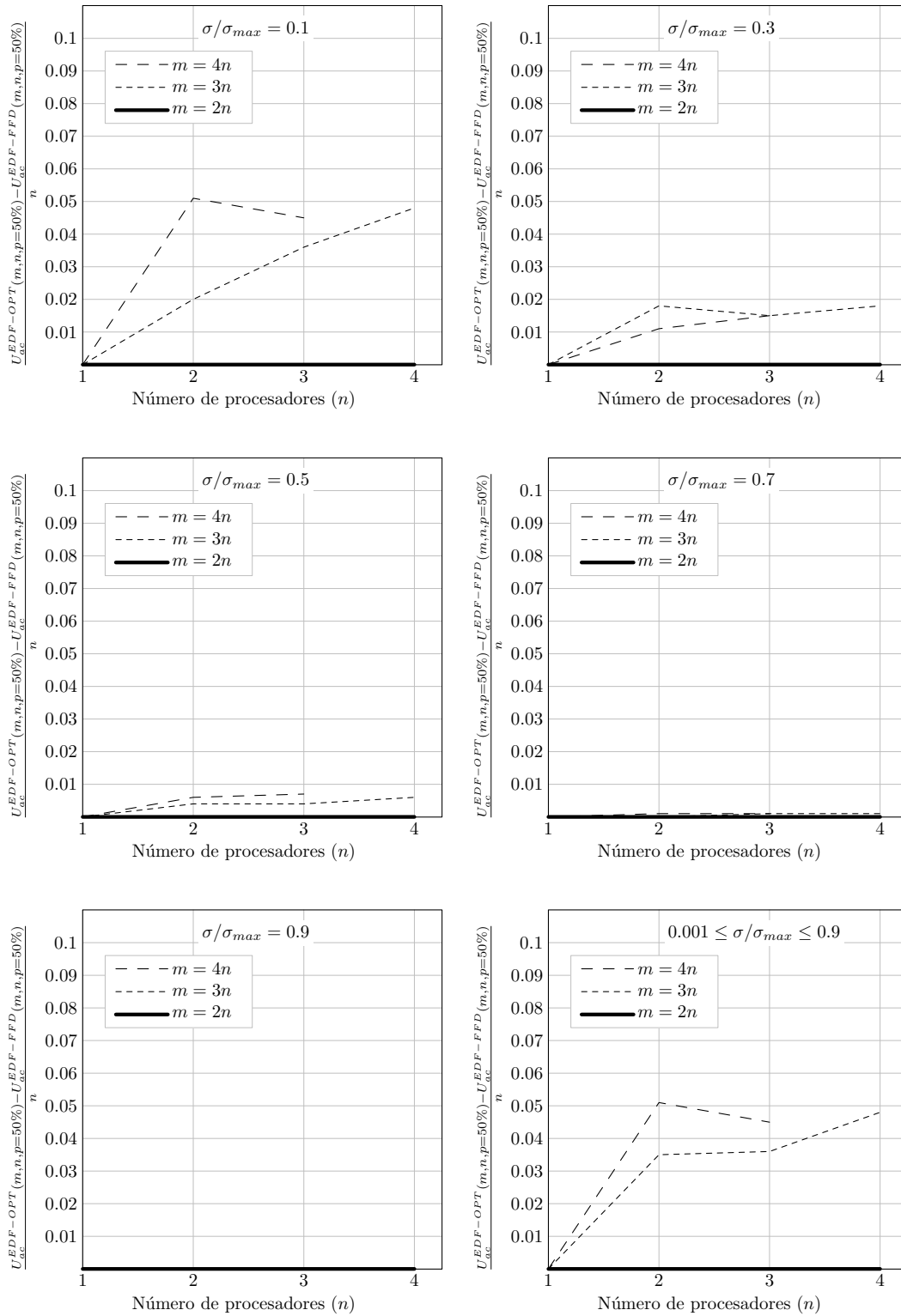


Figura 3.57: Comparación de los límites de utilización estadísticos para planificación EDF-OPT y EDF-FFD para $p = 50\%$.

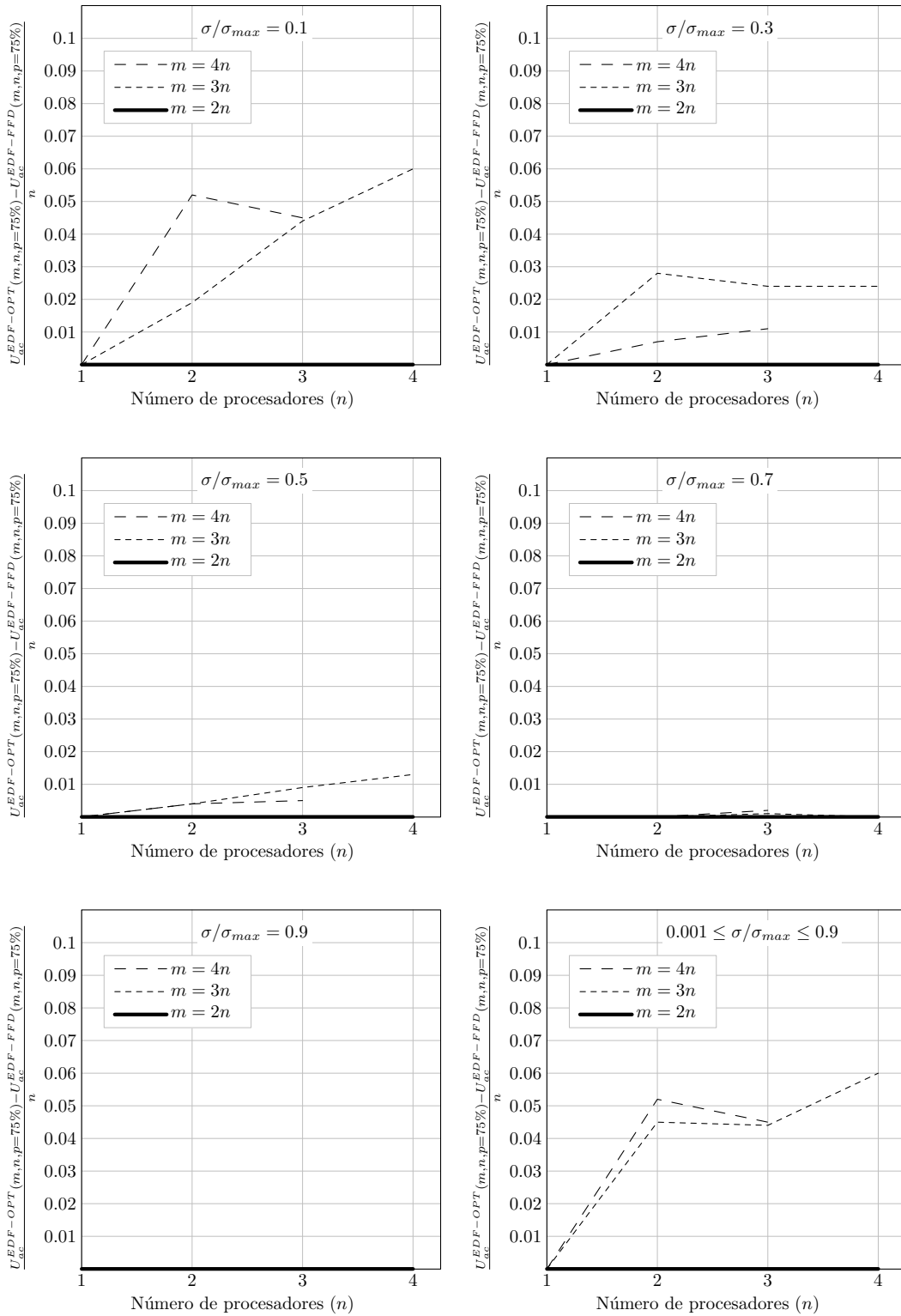


Figura 3.58: Comparación de los límites de utilización estadísticos para planificación EDF-OPT y EDF-FFD para $p = 75\%$.

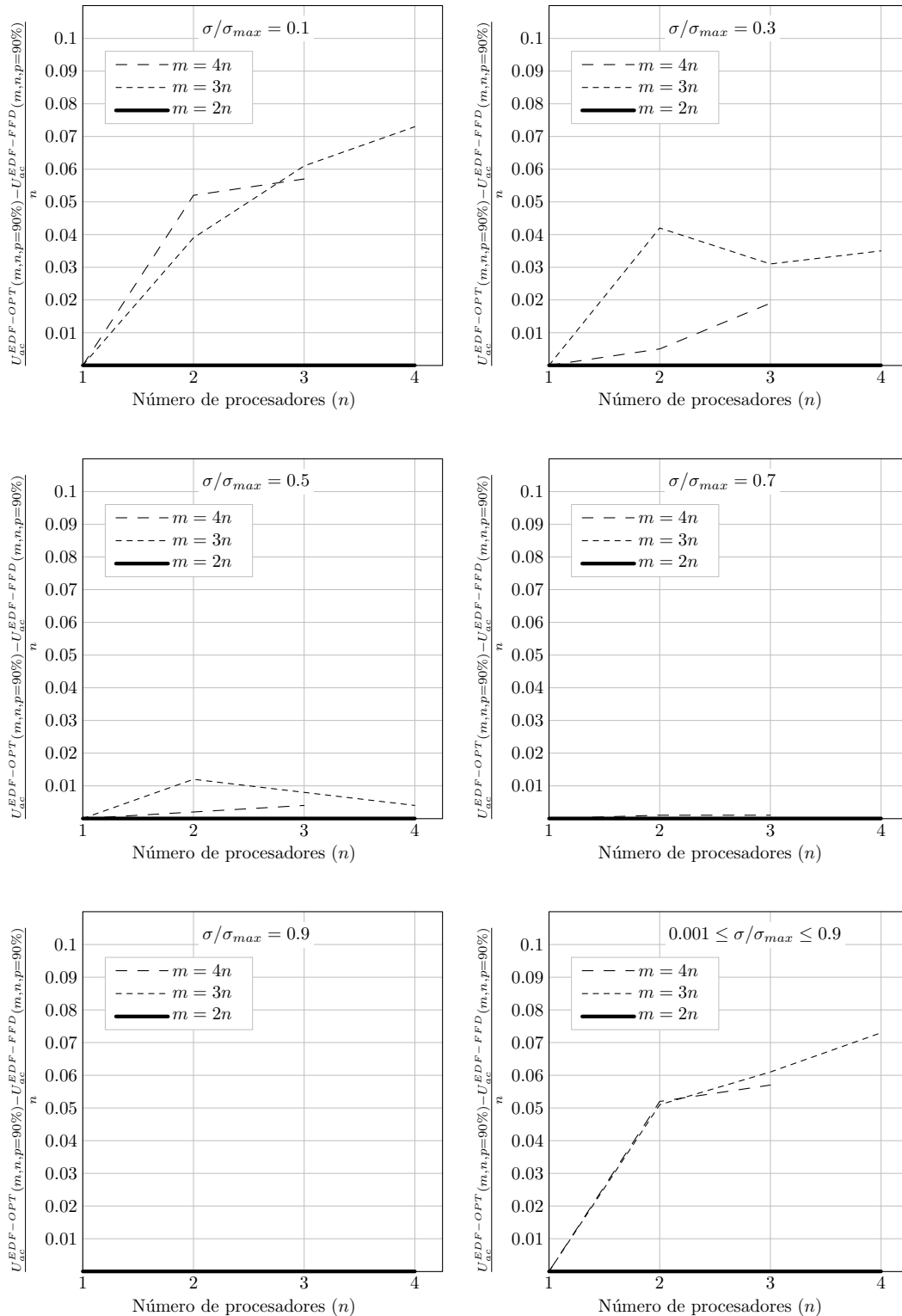


Figura 3.59: Comparación de los límites de utilización estadísticos para planificación EDF-OPT y EDF-FFD para $p = 90\%$.

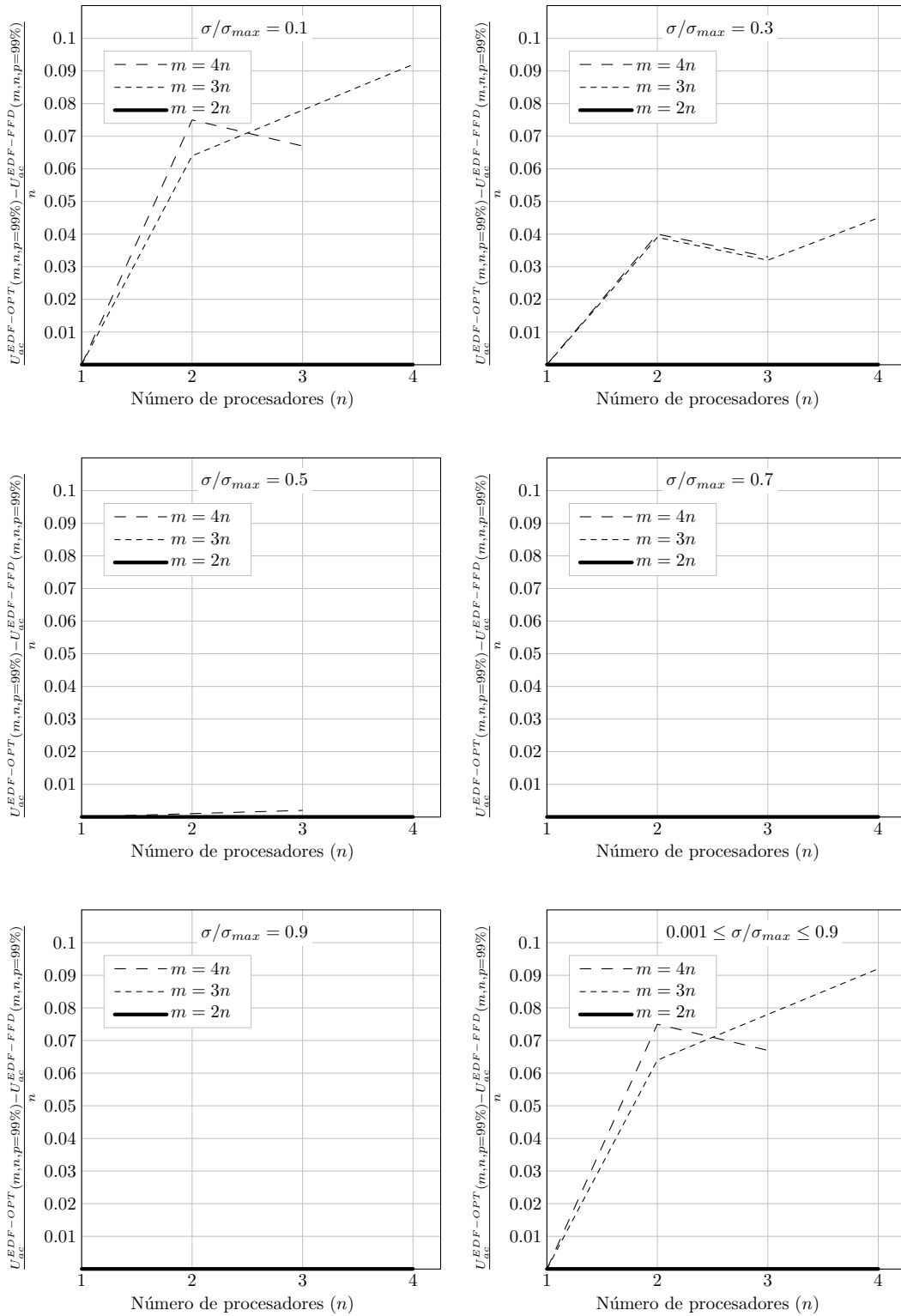


Figura 3.60: Comparación de los límites de utilización estadísticos para planificación EDF-OPT y EDF-FFD para $p = 99\%$.

Una vez se ha respondido a la primera cuestión de forma negativa, se puede replantear la segunda cuestión de la siguiente forma:

¿Merece la pena buscar algoritmos en línea de asignación de tareas que mejoren el límite de utilización estadístico asociado al algoritmo FF, acercándolo al límite de utilización estadístico asociado al algoritmo FFD?

Se puede responder a esta cuestión representando la función $U_{ac}^{EDF-FFD}(m, n, p) - U_{ac}^{EDF-FF}(m, n, p)$ para diferentes valores de σ y p , tal como se muestra en las figuras 3.61 a 3.64.

Cada una de las subfiguras se corresponde con una pareja de valores (σ, p) diferente. La excepción se encuentra en las subfiguras etiquetadas con $0'001 \leq \sigma/\sigma_{max} \leq 0'9$, las cuales se corresponden con el máximo de las curvas obtenidas para las desviaciones típicas $\sigma/\sigma_{max} = 0'001, 0'1, 0'2, \dots, 0'9$ y el mismo valor de p .

Las conclusiones obtenidas a partir de las figuras 3.61 a 3.64 se indican a continuación:

- La máxima mejora obtenida empleando el algoritmo FFD con respecto al algoritmo FF es del orden de 0'05, 0'06, 0'08, y 0'09 para los valores $p = 50\%$, $p = 75\%$, $p = 90\%$, y $p = 99\%$.
- La mejora obtenida es bastante independiente del número de procesadores.
- La mejora obtenida empleando distribuciones de factores de utilización con desviación típica menor de 0'3 veces la máxima y mayor que 0'7 la máxima es insignificante.

Para conjuntos de muchas tareas el algoritmo FF tiene un comportamiento casi ideal, pues el factor de utilización de las tareas es probabilísticamente muy pequeño. Por lo tanto, en este caso el algoritmo FFD no mejoraría casi nada los resultados obtenidos empleando el algoritmo de asignación FFD. No obstante, este resultado no se puede observar en las figuras, debido a la limitación en el número de tareas a $m \leq 10n$.

En resumen, se puede decir que la mejora obtenida empleando el algoritmo de asignación FFD es relativamente escasa con respecto al algoritmo FF. Esta es significativa, del orden del 10% por procesador, sólo en algunos casos. No obstante, este resultado debe tomarse con cautela, pues es básicamente la consecuencia de resultados de simulación con fuertes restricciones en cuanto al número de tareas y procesadores.

Como conclusión final de este apartado, los algoritmos de asignación FF y FFD son dos excelentes opciones para la asignación de tareas en línea y fuera de línea respectivamente, pues presentan un comportamiento razonablemente cercano al del algoritmo de asignación óptimo y presentan asimismo un bajo coste computacional.

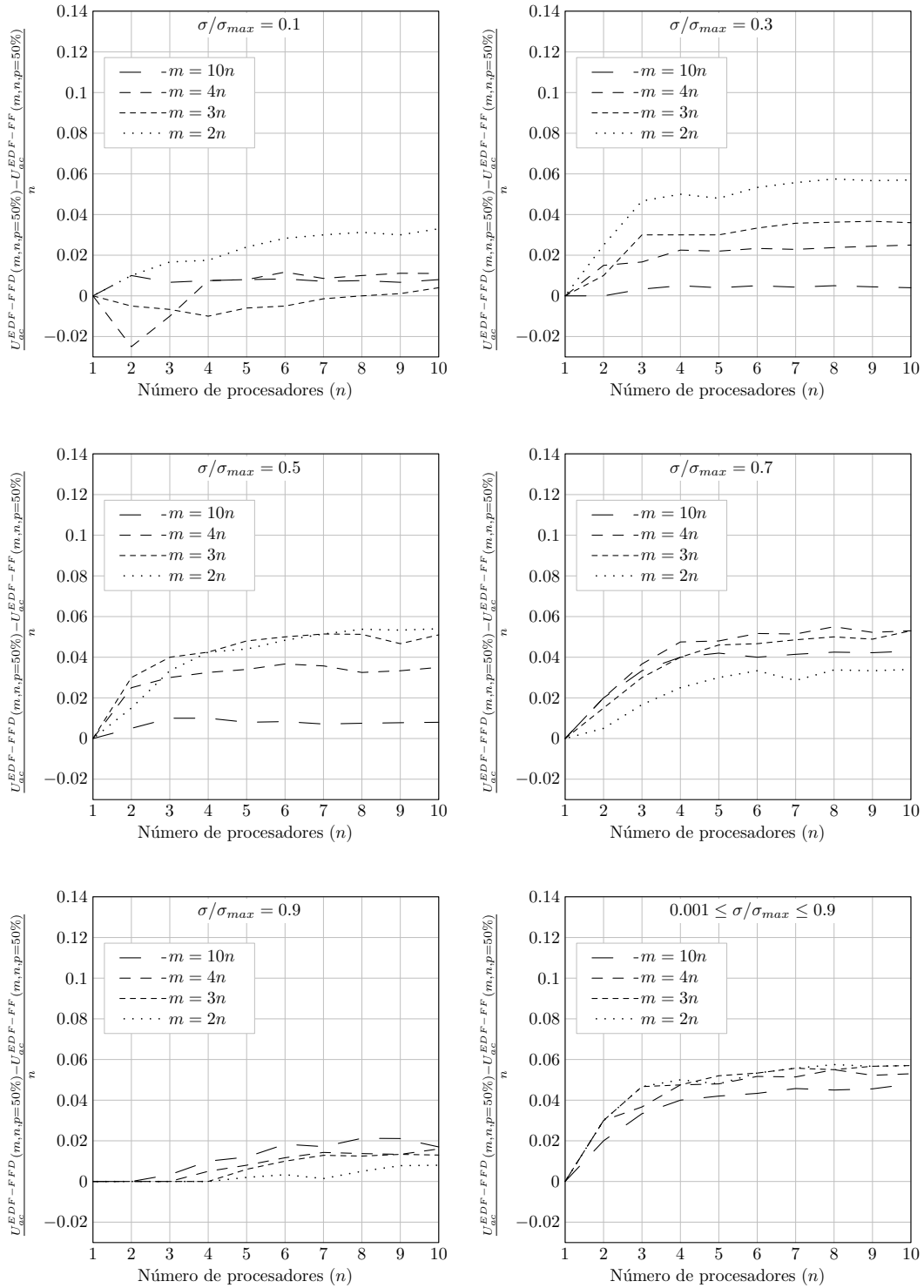


Figura 3.61: Comparación de los límites de utilización estadísticos para planificación EDF-FFD y EDF-FF para $p = 50\%$.

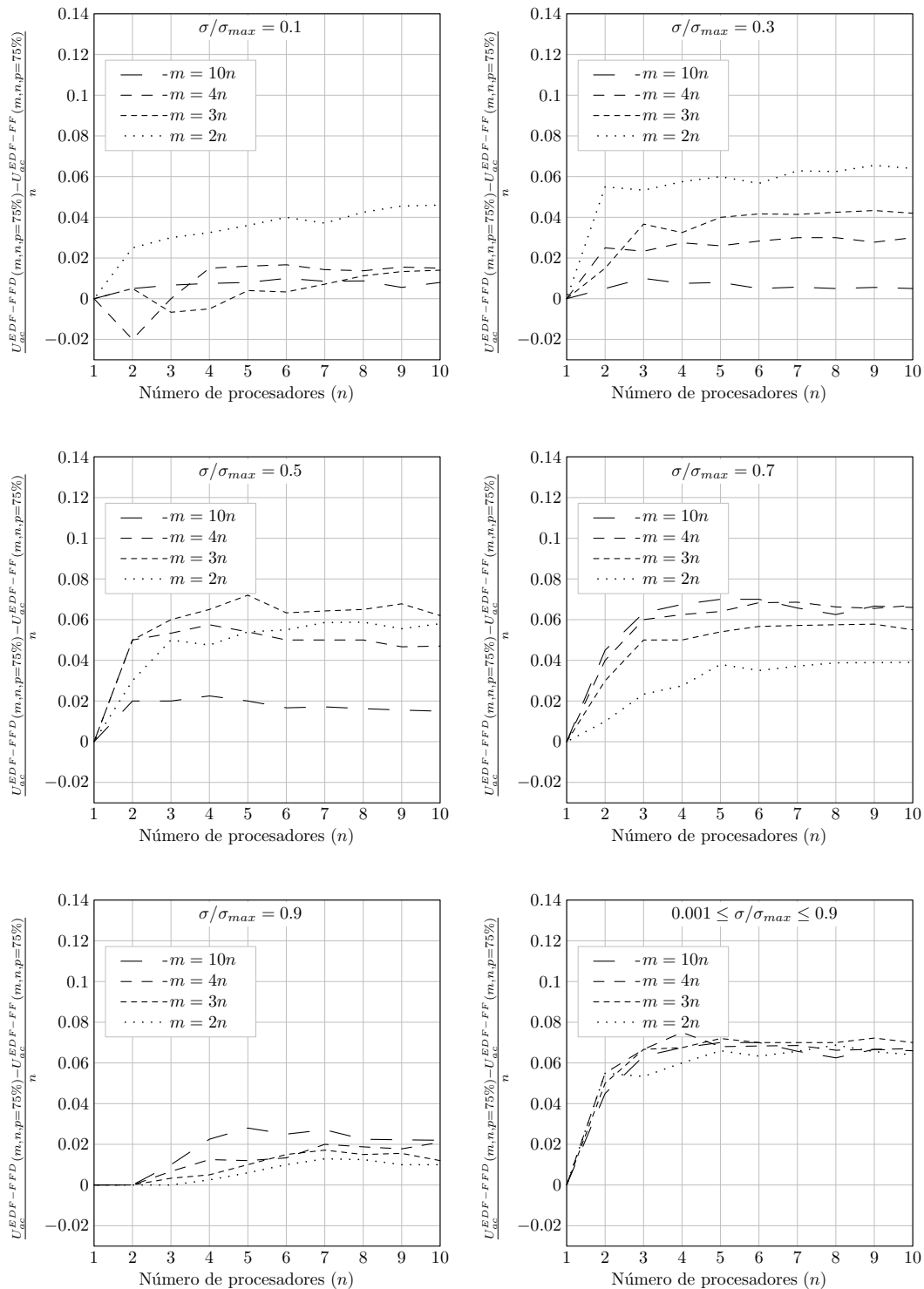


Figura 3.62: Comparación de los límites de utilización estadísticos para planificación EDF-FFD y EDF-FF para $p = 75\%$.

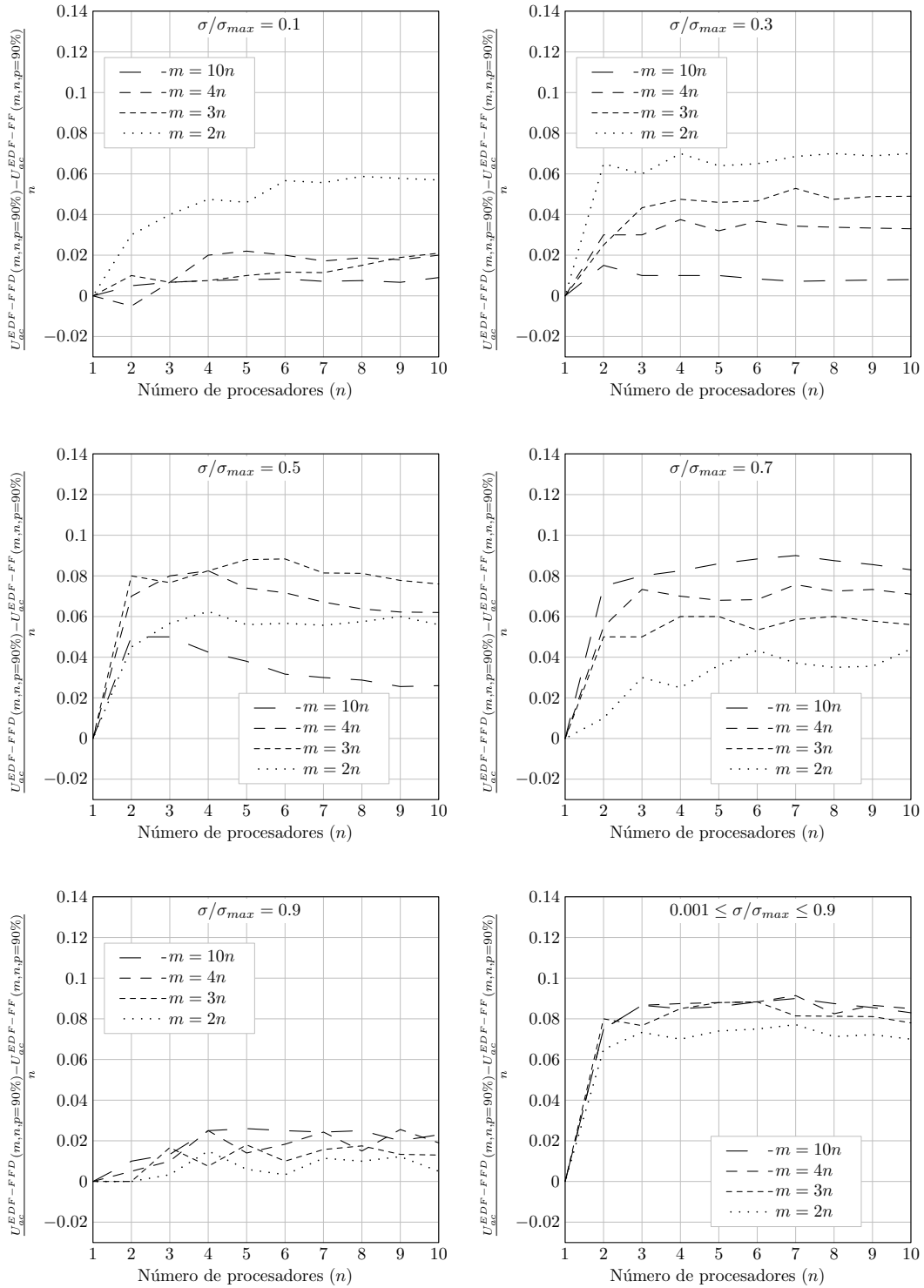


Figura 3.63: Comparación de los límites de utilización estadísticos para planificación EDF-FFD y EDF-FF para $p = 90\%$.

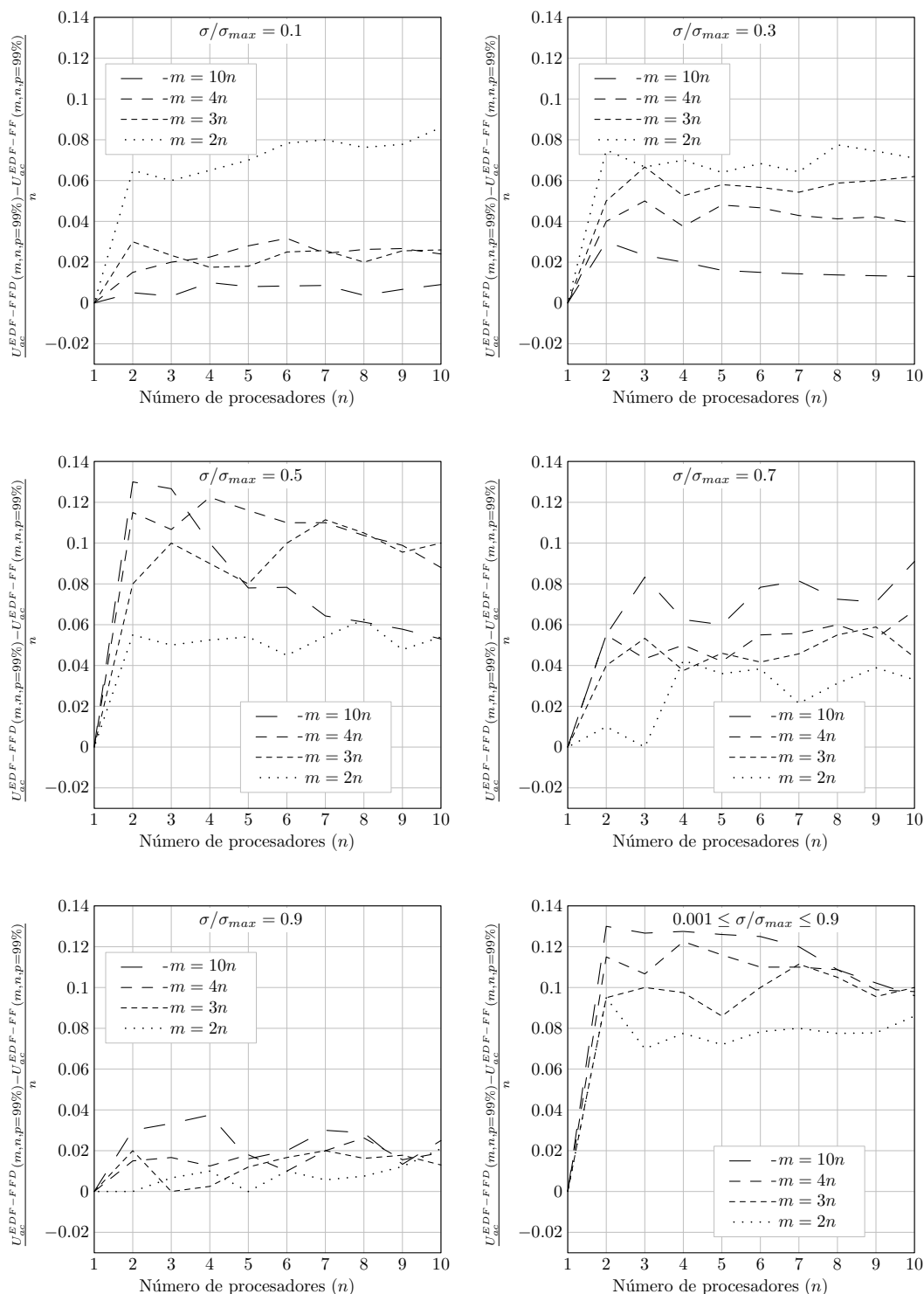


Figura 3.64: Comparación de los límites de utilización estadísticos para planificación EDF-FFD y EDF-FF para $p = 99\%$.

Capítulo 4

Límites de utilización empleando planificación RM

En este capítulo se presentan los límites de utilización para planificación Rate Monotonic (RM) sobre multiprocesadores homogéneos empleando técnicas de particionado. Las tareas se planifican con expulsión sobre cada monoprocesador empleando prioridades fijas asignadas de acuerdo con el algoritmo RM. Las tareas se asignan a los procesadores de acuerdo con los algoritmos de asignación presentados en el apartado 2.1.1. Las restricciones de tiempo real se modelan como plazos críticos de valor igual a los periodos de las tareas. Los límites de utilización para plazos menores que los periodos y planificación RM se presentan en el apartado 6.3.

La obtención de los límites de utilización empleando planificación RM es mucho más compleja que la obtención empleando planificación EDF. La razón está en que la capacidad del procesador ya no es fija, sino que depende del número de procesadores tal como indica el límite $m(2^{1/m} - 1)$, y por lo tanto puede variar entre $\ln 2$ y 1. Por lo demás se puede observar que este capítulo es análogo al capítulo 3.

En primer lugar, se define en el apartado 4.1 el concepto de límite de utilización para planificación RM y un algoritmo arbitrario de asignación de tareas AA dado. Al final del apartado 4.1 se demuestra la existencia de dicho límite para el caso de algoritmos de asignación razonables.

El límite de utilización asociado a cualquier algoritmo razonable de asignación de tareas y planificación RM se encuentra dentro de un intervalo $[L_{RM}, H_{RM}]$. El término L_{RM} del intervalo se corresponde con el mínimo de los límites de utilización U_{wc}^{RM-RA} , evaluado entre todos los algoritmos RA de asignación razonables. De forma análoga, el término H_{RM} del intervalo se corresponde con el máximo de los límites de utilización U_{wc}^{RM-RA} , evaluado entre todos los algoritmos RA de asignación razonables. El intervalo $[L_{RM}, H_{RM}]$ se calcula en el apartado 4.2.1.

El límite de utilización para los algoritmos de asignación FF, BF, WF, RF y OPT se calcula en los apartados 4.2.2 a 4.2.6.

La ordenación de las tareas por factores de utilización antes de realizar la asignación FF, BF, WF o RF, modifica los límites de utilización de la forma indicada en el apartado 4.2.7.

Las expresiones matemáticas de los límites de utilización obtenidas en los apartados 4.2.1 a 4.2.7 son analizadas en el apartado 4.3.

Con la intención de mostrar el pesimismo de los límites de utilización absolutos, en el apartado 4.4 se presentan los límites de utilización estadísticos, obtenidos mediante simulación exhaustiva. Los límites de utilización absolutos, denominados simplemente límites de utilización, se obtienen considerando el caso peor. Estos son el resultado de un enfoque puramente teórico del problema. Los límites de utilización estadísticos permiten un tanto por ciento de conjuntos de tareas generados aleatoriamente que no son planificables. Estos últimos son el resultado del enfoque de simulación. Por lo tanto, la información proporcionada por los enfoques teórico y de simulación es complementaria.

La comparación entre los diferentes algoritmos de asignación proporciona resultados análogos a los obtenidos en el apartado 3.5, con ciertos matices indicados al final del apartado 4.4

4.1 Definición y existencia de los límites de utilización

En este apartado se define el concepto de límite de utilización para planificación multiprocesador con particionado, empleando planificación monoprocesador RM, y un algoritmo arbitrario de asignación de tareas AA dado. Una vez definido el concepto de límite de utilización se demuestra su existencia para el caso de algoritmos de asignación razonables.

Por convenio, la planificación multiprocesador con particionado empleando RM como planificador monoprocesador y AA como algoritmo de asignación de tareas se representa por RM-AA. El límite de utilización asociado (si existe) se representa por U_{wc}^{RM-AA} . Cuando el algoritmo de asignación es razonable se sustituyen las iniciales AA por las iniciales RA.

Definición 5 *El límite de utilización para RM-AA se define como el número real U_{wc}^{RM-AA} que verifica las siguientes propiedades:*

- *Cualquier conjunto de tareas de utilización total $U \leq U_{wc}^{RM-AA}$ cabe en el multiprocesador, empleando planificación monoprocesador RM y asignación de tareas AA. Por lo tanto es planificable empleando planificación multiprocesador RM-AA.*
- *Para todo valor de utilización $U > U_{wc}^{RM-AA}$ existe al menos un conjunto de tareas con esa utilización que no cabe en el multiprocesador empleando planificación RM y asignación de tareas AA. Por lo tanto, el conjunto de tareas con esa utilización puede ser planificable o no empleando planificación multiprocesador RM-AA.*

Por consiguiente, el límite de utilización puede emplearse como una condición suficiente de planificabilidad.

Una tarea cabe en un procesador si la suma de los factores de utilización de la tarea en cuestión y de las tareas previamente asignadas al procesador acumulan una utilización total menor o igual que $m(2^{1/m} - 1)$. Este resultado es una consecuencia

del límite de utilización para planificación RM en monoprocesadores (Liu y Layland, 1973). El límite de utilización $m(2^{1/m} - 1)$ se obtiene cuando los factores de utilización de todas las tareas son iguales. Si estos no fuesen iguales la condición de planificabilidad ajustada en función de los factores de utilización vendría dada por el teorema siguiente.

Teorema 11 (Oh y Son, 1995). *Si un conjunto de $(m - 1)$ tareas es planificable en un monoprocesador empleando planificación RM y se añade una nueva tarea τ_m de factor de utilización u_m , el conjunto de m tareas resultante es planificable si al menos una de las m condiciones siguientes se cumple:*

$$u_m \leq \frac{2}{\lambda(k) \prod_{i=1}^k (1 + u_i)} - 1; \quad 0 \leq k \leq m - 1$$

donde

$$\lambda(k) = \begin{cases} 1 & \text{si } k = m - 1 \\ \left(1 + \frac{\sum_{i=k+1}^{m-1} u_i}{m-k-1}\right)^{(m-k-1)} & \text{en otro caso} \end{cases}$$

Además la condición de planificabilidad es ajustada.

El límite de utilización de Liu y Layland (1973) se obtiene particularizando esta condición para el caso de factores de utilización iguales.

La aplicación de esta condición de planificabilidad complica extraordinariamente la obtención de los límites de utilización multiprocesador empleando planificación monoprocesador RM. Además, la mejora de planificabilidad que se obtiene es muy reducida, de ahí que en este capítulo se considere siempre la condición de planificabilidad de Liu y Layland (1973). En el apartado 5.2 se cuantifica el incremento máximo de los límites de utilización que podría obtenerse a partir de la condición de planificabilidad de Oh y Son (1995).

A continuación se demuestra la existencia del límite de utilización U_{uc}^{RM-RA} empleando RM como algoritmo de planificación en cada procesador, para cualquier algoritmo razonable de asignación de tareas RA dado.

Lema 4 *Existe el límite de utilización para planificación RM y cada algoritmo razonable de asignación de tareas RA, el cual es una función (a lo sumo) del número de tareas, del número de procesadores y del máximo factor de utilización alcanzable.*

Demostración:

La demostración es idéntica a la dada en el lema 1, página 39, para planificación monoprocesador EDF. Deben sustituirse las iniciales “EDF” por las iniciales “RM”, y donde se hace referencia a la utilización uno, esta pasa a ser utilización $\ln 2$. ■

El lema 4 demuestra entre otras cosas que el límite de utilización para planificación RM en un multiprocesador, empleando un algoritmo razonable RA de asignación de tareas arbitrario, depende a lo sumo del número de tareas, m , del número

de procesadores, n , y del máximo factor de utilización alcanzable, α . Por esta razón el límite de utilización se representa como

$$U_{wc}^{RM-RA}(m, n, \alpha)$$

4.2 Cálculo de los límites de utilización

Una vez definido el concepto de límite de utilización y después de haber demostrado su existencia se procede al cálculo del mismo. Debe tenerse en cuenta que cada algoritmo de asignación de tareas tendrá asociado un límite de utilización en principio distinto.

En general, los límites de utilización son una función del número de tareas, m , del número de procesadores, n , y del máximo factor de utilización alcanzable, α , tal como se demostró en el lema 4.

Antes de calcular los límites de utilización se introducirá un nuevo parámetro denominado β_{RM} .

Definición 6 *El parámetro β_{RM} se define como el máximo número de tareas de factor de utilización α que caben en un procesador empleando planificación RM.*

De igual forma que sucedía con la planificación EDF, este parámetro es un concepto clave en la obtención de los límites de utilización. El parámetro β_{RM} es una función del máximo factor de utilización alcanzable, α , tal como demuestra el lema 5.

Lema 5

$$\beta_{RM} = \left\lfloor \frac{1}{\log_2(\alpha + 1)} \right\rfloor \quad (4.1)$$

Demostración:

A partir de la definición del parámetro β_{RM} se deduce que β_{RM} tareas de factor de utilización α caben en un procesador. De esta forma, a partir del límite de utilización para planificación RM en monoprocesadores se obtiene $\beta_{RM}\alpha \leq \beta_{RM}(2^{1/\beta_{RM}} - 1)$. Despejando β_{RM} resulta $\beta_{RM} \leq 1/\log_2(\alpha + 1)$. Se sabe que β_{RM} es un número entero, y por lo tanto se cumple

$$\beta_{RM} \leq \left\lfloor \frac{1}{\log_2(\alpha + 1)} \right\rfloor$$

Por otra parte, β_{RM} es el máximo número de tareas de factor de utilización α que caben en un procesador. Por lo tanto, $(\beta_{RM} + 1)$ tareas de factor de utilización α no caben en un procesador. Es decir, $(\beta_{RM} + 1)\alpha > (\beta_{RM} + 1)(2^{1/(\beta_{RM}+1)} - 1)$. Despejando β_{RM} se obtiene

$$\beta_{RM} > \frac{1}{\log_2(\alpha + 1)} - 1$$

Puesto que β_{RM} es un número entero resulta

$$\beta_{RM} \geq \left\lfloor \frac{1}{\log_2(\alpha + 1)} \right\rfloor$$

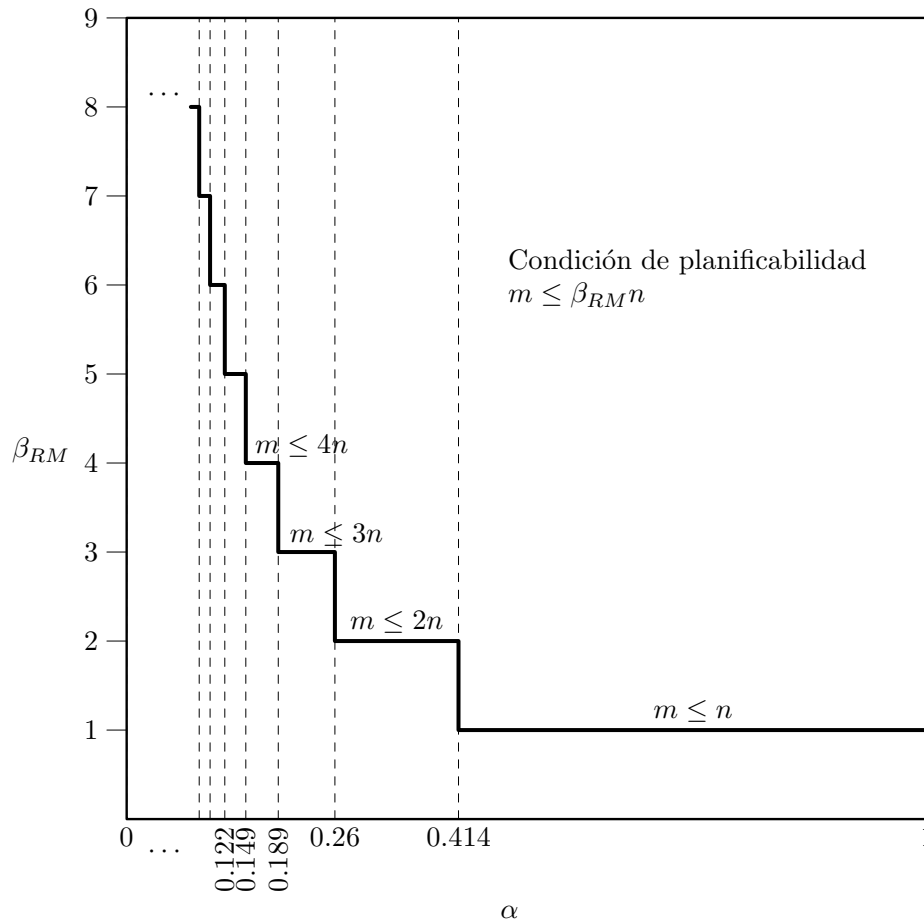


Figura 4.1: Representación de la función $\beta_{RM}(\alpha)$, y la condición de planificabilidad asociada.

En conclusión,

$$\beta_{RM} = \left\lfloor \frac{1}{\log_2(\alpha + 1)} \right\rfloor$$

■

El valor de β_{RM} puede emplearse para establecer la planificabilidad de ciertos conjuntos de tareas. A partir de la definición de β_{RM} se tiene que β_{RM} tareas de factor de utilización α caben en cada procesador. Puesto que todas las tareas tienen factores de utilización menores o iguales que α , al menos β_{RM} tareas de factores de utilización arbitrarios (menores o iguales que α) caben en cada procesador. Como consecuencia, un multiprocesador formado por n procesadores puede alojar al menos $\beta_{RM}n$ tareas. Por consiguiente, cualquier conjunto de m tareas de factores de utilización $u_i \leq \alpha$ con $m \leq \beta_{RM}n$ es planificable empleando RM como planificador monoprocesador, y empleando cualquier algoritmo razonable de asignación de tareas a procesadores.

La figura 4.1 muestra β_{RM} en función de α , mostrando asimismo la condición suficiente de planificabilidad $m \leq \beta_{RM}n$. Por ejemplo, si α se encuentra en el

intervalo $(2^{1/4} - 1, 2^{1/3} - 1] \approx (0'19, 0'26]$ entonces $\beta_{RM} = 3$. En este caso, si el conjunto de tareas está formado por $3n$ tareas o menos es planificable.

Otra consecuencia de la condición de planificabilidad $m \leq \beta_{RM}n$ es que merece la pena obtener el límite de utilización sólo si $m > \beta_{RM}n$. En caso contrario, el conjunto de tareas es directamente planificable.

A partir de ahora se supone que $m > \beta_{RM}n$, de tal forma que la condición de planificabilidad anterior no puede aplicarse.

Antes de calcular los límites de utilización asociados a los diferentes algoritmos de asignación de tareas se obtendrá el rango dentro del cual se encuentran. Este resultado es significativo desde tres puntos de vista. En primer lugar, establece los valores mínimo y máximo entre los que se encuentran los límites de utilización de todos los algoritmos de asignación razonables. En segundo lugar, permite obtener de forma sencilla el límite de utilización correspondiente a ciertos algoritmos de asignación razonables. En tercer y último lugar, servirá como comprobación de la validez de los límites de utilización obtenidos para los diferentes algoritmos de asignación de tareas razonables, pues deben encontrarse de forma forzosa dentro del rango definido.

4.2.1 Rango de los límites de utilización

El objetivo es obtener la pareja de valores L_{RM} y H_{RM} que verifican la condición

$$L_{RM} \leq U_{wc}^{RM-RA} \leq H_{RM} \quad \text{para todo } RA$$

donde U_{wc}^{RM-RA} es el límite de utilización para planificación RM empleando el algoritmo razonable RA de asignación de tareas a procesadores. El límite de utilización para cualquier algoritmo de asignación razonable es positivo, mayor o igual que $\ln 2$, y por lo tanto a partir de cualquier pareja (L_{RM}, H_{RM}) que cumpla la condición anterior se pueden obtener infinitas parejas que también la cumplen, simplemente decrementando el valor de L_{RM} o incrementando el valor de H_{RM} . Resulta conveniente entonces calcular la pareja de valores ajustada. Esto es, debe existir al menos un algoritmo razonable de asignación de tareas cuyo límite de utilización para planificación RM tome el valor L_{RM} . Por lo tanto

$$L_{RM} = \min_{RA} U_{wc}^{RM-RA} \quad (4.2)$$

De la misma forma, debe existir al menos un algoritmo razonable de asignación de tareas cuyo límite de utilización para planificación RM tome el valor H_{RM} . Por lo tanto

$$H_{RM} = \max_{RA} U_{wc}^{RM-RA} \quad (4.3)$$

A partir de ahora, cualquier referencia al rango de los límites de utilización en este capítulo se referirá a los valores L_{RM} y H_{RM} ajustados, definidos por las ecuaciones (4.2) y (4.3). El problema es que los límites de utilización obtenidos en este capítulo se basan en el empleo del límite de utilización $m(2^{1/m} - 1)$ para planificación RM. Este límite de utilización es ajustado para factores de utilización desconocidos, o

cuando los factores de utilización de todas las tareas son iguales (Liu y Layland, 1973). Sin embargo, cuando los factores de utilización de las tareas son diferentes este límite puede mejorarse, tal como indica la condición de planificabilidad de Oh y Son (1995), presentada en la página 143. La máxima mejora que puede obtenerse en los límites de utilización empleando esta condición con respecto a los límites de utilización proporcionados en este capítulo se indica en el apartado 5.2 y es muy escasa. No obstante, por simplicidad y por seguir una línea de desarrollo análoga a la del capítulo 3 se denominarán ajustados a los límites de utilización L_{RM} y H_{RM} obtenidos empleando el límite de utilización $m(2^{1/m} - 1)$, pues están muy cerca de serlo aunque estrictamente no lo sean.

El corolario 2 proporciona una cota inferior del límite de utilización, válida para todos los algoritmos de asignación razonables. En el apartado 4.2.4 se encuentra un algoritmo razonable de asignación de tareas cuyo límite de utilización coincide con esa cota. Por lo tanto, L_{RM} será igual a la cota dada por el corolario 2.

Por otra parte, el teorema 15 proporciona una cota superior del límite de utilización, válida para todos los algoritmos de asignación. En el apartado 4.2.7 se encuentran varios algoritmos razonables de asignación de tareas cuyo límite de utilización coincide con esa cota. Por lo tanto, H_{RM} será igual a la cota dada por el teorema 15.

La cota inferior se obtiene combinando tres cotas inferiores diferentes, eligiendo a intervalos aquella que proporciona el máximo valor de entre las tres. La obtención de estas tres cotas inferiores se realiza en los teoremas 12, 13 y 14. El teorema 12 proporcionará la primera de las tres cotas inferiores. No obstante, este teorema requiere el resultado proporcionado por el lema 6, el cual se enuncia y demuestra a continuación.

Lema 6 *Sea $\{r_1, \dots, r_n\}$ un conjunto cualquiera de n números enteros positivos verificando $\sum_{j=1}^n r_j = M$. Sea $g(r_1, \dots, r_n) = \sum_{j=1}^n r_j(2^{1/r_j} - 1)$. Se cumple*

$$g(r_1, \dots, r_n) \geq (M - \lfloor M/n \rfloor) \lceil M/n \rceil (2^{1/\lceil M/n \rceil} - 1) + \\ (n - M + \lfloor M/n \rfloor) \lfloor M/n \rfloor (2^{1/\lfloor M/n \rfloor} - 1)$$

Demostración:

En primer lugar se demostrará que el vector $\{\hat{r}_1, \dots, \hat{r}_n\}$ que produce el mínimo valor de la función g verifica $(\hat{r}_j - \hat{r}_k) \leq 1$, para $\hat{r}_j > \hat{r}_k$, con j, k en $1, \dots, n$. Esto es, el vector $\{\hat{r}_1, \dots, \hat{r}_n\}$ que produce el mínimo valor de la función g verifica que la máxima diferencia en valor absoluto entre dos valores \hat{r}_j y \hat{r}_k cualesquiera es uno.

Para demostrarlo considérense dos valores r_j y r_k tal que $(r_j - r_k) > 1$. En esta situación los términos j -ésimo y k -ésimo dentro del sumatorio que define la función g acumulan un valor

$$r_j(2^{1/r_j} - 1) + r_k(2^{1/r_k} - 1)$$

Si se decrementa r_j en una unidad y se incrementa r_k en esa unidad se sigue cumpliendo la restricción $\sum_{j=1}^n r_j = M$, y los términos j -ésimo y k -ésimo dentro del

sumatorio que define la función g pasan a acumular un valor

$$(r_j - 1)(2^{1/(r_j-1)} - 1) + (r_k + 1)(2^{1/(r_k+1)} - 1)$$

La primera derivada de la función $f(x) = x(2^{1/x} - 1)$ es negativa para $x > 0$. A medida que x crece la función decrece, alcanzando asintóticamente el valor $\ln 2$ cuando $x \rightarrow \infty$. La segunda derivada de la función $f(x) = x(2^{1/x} - 1)$ es positiva para $x > 0$, lo cual indica que la función decrece cada vez más lentamente. Esto llevado a nuestro problema significa que

$$r_k(2^{1/r_k} - 1) - (r_k + 1)(2^{1/(r_k+1)} - 1) > (r_j - 1)(2^{1/(r_j-1)} - 1) - r_j(2^{1/r_j} - 1)$$

y por lo tanto

$$(r_j - 1)(2^{1/(r_j-1)} - 1) + (r_k + 1)(2^{1/(r_k+1)} - 1) < r_j(2^{1/r_j} - 1) + r_k(2^{1/r_k} - 1)$$

Se deduce entonces que reduciendo r_j en una unidad e incrementando r_k en esa unidad se obtiene un valor menor de la función g . Este proceso podría repetirse hasta conseguir $(r_j - r_k) \leq 1$ para j, k en $1, \dots, n$.

Por lo tanto, para conseguir el mínimo valor de la función g el entero positivo M debe dividirse equitativamente (dentro de lo posible) entre $\{r_1, \dots, r_n\}$, de tal forma que haya $(M - \lfloor M/n \rfloor n)$ elementos de valor $\lceil M/n \rceil$, y $(n - M + \lfloor M/n \rfloor n)$ elementos de valor $\lfloor M/n \rfloor$. Por lo tanto se cumple

$$g(r_1, \dots, r_n) \geq (M - \lfloor M/n \rfloor n) \lceil M/n \rceil (2^{1/\lceil M/n \rceil} - 1) + \\ (n - M + \lfloor M/n \rfloor n) \lfloor M/n \rfloor (2^{1/\lfloor M/n \rfloor} - 1)$$

■

Teorema 12 *Sea RA un algoritmo razonable de asignación de tareas a procesadores arbitrario. Sea $m > \beta_{RM}n$. Se cumple*

$$U_{wc}^{RM-RA}(m, n, \alpha) \geq n_a U_a + n_b U_b - (n - 1)\alpha$$

donde

$$n_a = m + n - 1 - \left\lfloor \frac{m + n - 1}{n} \right\rfloor n \\ n_b = n - n_a \\ U_a = \left\lfloor \frac{m + n - 1}{n} \right\rfloor \left(2^{1/\lceil \frac{m+n-1}{n} \rceil} - 1 \right) \\ U_b = \left\lfloor \frac{m + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right)$$

Demostración:

Sea $\{\tau_1, \dots, \tau_m\}$ un conjunto de m tareas que no cabe en el multiprocesador. Sea τ_k la primera tarea del conjunto que no cabe en el multiprocesador. Puesto que el algoritmo de asignación es razonable, aplicando la ecuación (2.1) de la página 29 se

tiene

$$U_j > (m_j + 1)(2^{1/(m_j+1)} - 1) - u_k \quad (4.4)$$

para $j = 1, \dots, n$. Donde U_j es la utilización total de las tareas previamente asignadas al procesador P_j , m_j es el número de estas tareas y u_k es el factor de utilización de la tarea τ_k . Al ser τ_k una tarea del conjunto de m tareas resulta que la utilización total del conjunto de tareas denotada por U verifica

$$U = \sum_{i=1}^m u_i \geq \sum_{i=1}^k u_i = \sum_{j=1}^n U_j + u_k \quad (4.5)$$

A partir de la ecuación (4.4) se deduce

$$\begin{aligned} \sum_{j=1}^n U_j &> \sum_{j=1}^n ((m_j + 1)(2^{1/(m_j+1)} - 1) - u_k) = \\ &= \sum_{j=1}^n (m_j + 1)(2^{1/(m_j+1)} - 1) - nu_k \end{aligned}$$

Sustituyendo esta expresión en la ecuación (4.5) resulta

$$U > \sum_{j=1}^n (m_j + 1)(2^{1/(m_j+1)} - 1) - (n - 1)u_k$$

Como todos los factores de utilización son menores o iguales que α se tiene $u_k \leq \alpha$, y por lo tanto

$$U > \sum_{j=1}^n (m_j + 1)(2^{1/(m_j+1)} - 1) - (n - 1)\alpha$$

Una restricción de los valores m_j es que $\sum_{j=1}^n m_j = (k - 1)$. Esta restricción es totalmente equivalente a la restricción $\sum_{j=1}^n (m_j + 1) = k + n - 1$. Teniendo en cuenta esta restricción, definiendo $r_j = (m_j + 1)$, $M = (k + n - 1)$, $g(r_1, \dots, r_n) = \sum_{j=1}^n r_j(2^{1/r_j} - 1)$ y aplicando el lema 6 se deduce

$$\begin{aligned} \sum_{j=1}^n (m_j + 1)(2^{1/(m_j+1)} - 1) &\geq \\ &\left(k + n - 1 - \left\lfloor \frac{k + n - 1}{n} \right\rfloor n \right) \left\lceil \frac{k + n - 1}{n} \right\rceil \left(2^{1/\lceil \frac{k+n-1}{n} \rceil} - 1 \right) + \\ &\left(n - (k + n - 1) + \left\lfloor \frac{k + n - 1}{n} \right\rfloor n \right) \left\lfloor \frac{k + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{k+n-1}{n} \rfloor} - 1 \right) = \\ &\left(k + n - 1 - \left\lfloor \frac{k + n - 1}{n} \right\rfloor n \right) \left\lceil \frac{k + n - 1}{n} \right\rceil \left(2^{1/\lceil \frac{k+n-1}{n} \rceil} - 1 \right) + \\ &\left(1 - k + \left\lfloor \frac{k + n - 1}{n} \right\rfloor n \right) \left\lfloor \frac{k + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{k+n-1}{n} \rfloor} - 1 \right) \end{aligned}$$

y por lo tanto

$$U > \left(k + n - 1 - \left\lfloor \frac{k + n - 1}{n} \right\rfloor n \right) \left\lceil \frac{k + n - 1}{n} \right\rceil \left(2^{1/\lceil \frac{k+n-1}{n} \rceil} - 1 \right) + \left(1 - k + \left\lfloor \frac{k + n - 1}{n} \right\rfloor n \right) \left\lfloor \frac{k + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{k+n-1}{n} \rfloor} - 1 \right) - (n - 1)\alpha \quad (4.6)$$

El segundo término de la expresión anterior es una función monótona decreciente de k , pues cualquier incremento de k hace que el número $(k + n - 1)$ a repartir equitativamente entre todos los procesadores sea mayor.

El índice k se encuentra dentro del rango $[1, m]$. Por lo tanto, para $k = m$ se obtiene el valor mínimo del sumatorio $\sum_{j=1}^n (m_j + 1)(2^{1/(m_j+1)} - 1)$.

A partir de la ecuación (4.6) se obtiene

$$U > \left(m + n - 1 - \left\lfloor \frac{m + n - 1}{n} \right\rfloor n \right) \left\lceil \frac{m + n - 1}{n} \right\rceil \left(2^{1/\lceil \frac{m+n-1}{n} \rceil} - 1 \right) + \left(1 - m + \left\lfloor \frac{m + n - 1}{n} \right\rfloor n \right) \left\lfloor \frac{m + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) - (n - 1)\alpha$$

Todo conjunto de tareas que no cabe en n procesadores cumple la condición anterior. Se deduce entonces que todo conjunto de tareas de utilización total menor o igual que el segundo término de la ecuación anterior cabe en n procesadores.

Debe tenerse en cuenta que no se ha hecho ninguna suposición con respecto al algoritmo de asignación de tareas a procesadores, salvo que este sea razonable. Por consiguiente se cumple

$$U_{wc}^{RM-RA}(m, n, \alpha) \geq \left(m + n - 1 - \left\lfloor \frac{m + n - 1}{n} \right\rfloor n \right) \left\lceil \frac{m + n - 1}{n} \right\rceil \left(2^{1/\lceil \frac{m+n-1}{n} \rceil} - 1 \right) + \left(1 - m + \left\lfloor \frac{m + n - 1}{n} \right\rfloor n \right) \left\lfloor \frac{m + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) - (n - 1)\alpha$$

donde RA es un algoritmo razonable de asignación de tareas arbitrario. Esta relación se puede expresar de forma más abreviada en función de n_a , n_b , U_a y U_b tal como sigue

$$U_{wc}^{RM-RA}(m, n, \alpha) \geq n_a U_a + n_b U_b - (n - 1)\alpha$$

■

A continuación se propone otra cota inferior del límite de utilización, válida para cualquier algoritmo razonable de asignación, RA.

Teorema 13 *Sea RA un algoritmo razonable de asignación de tareas a procesadores arbitrario. Sea $m > \beta_{RM}n$. Se cumple*

$$U_{wc}^{RM-RA}(m, n, \alpha) \geq U_b$$

donde

$$U_b = \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right)$$

Demostración:

Sea $\{\tau_1, \dots, \tau_m\}$ un conjunto de m tareas de utilización total menor o igual que

$$U_b = \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right)$$

Justo antes de intentar asignar la tarea τ_k del conjunto se han intentado asignar las $(k-1)$ tareas anteriores. En la asignación de la tarea τ_k el caso peor ocurre cuando las $(k-1)$ tareas anteriores se han asignado con éxito a alguno de los n procesadores. En este caso, el procesador que menos tareas ha recibido ha recibido a lo sumo $\lfloor (k-1)/n \rfloor$ tareas. Teniendo en cuenta este hecho, y las relaciones (ii) y (iv) de la página 245, este procesador tiene un límite de utilización mayor o igual que

$$\left\lfloor \frac{k+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{k+n-1}{n} \rfloor} - 1 \right)$$

Puesto que k puede variar entre 1 y m , y teniendo en cuenta de nuevo la relación (ii), resulta que el procesador anterior tiene un límite de utilización mayor o igual que

$$\left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right)$$

La utilización total de las k tareas es menor o igual que la utilización total del conjunto completo. Por lo tanto, la utilización total de las k tareas es menor o igual que el límite de utilización de dicho procesador. Por consiguiente, τ_k cabe en el multiprocesador. La demostración ha sido llevada a cabo considerando una tarea τ_k cualquiera del conjunto y un algoritmo de asignación de tareas razonable cualquiera.

Se concluye entonces que cualquier conjunto de m tareas y utilización total menor o igual que

$$U_b = \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right)$$

cabe en n procesadores empleando cualquier algoritmo de asignación razonable. ■

Finalmente se obtendrá la tercera cota inferior del límite de utilización, válida para cualquier algoritmo razonable de asignación, RA.

Teorema 14 *Sea RA un algoritmo razonable de asignación de tareas a procesadores arbitrario. Sea $m > \beta_{RM}n$. Se cumple*

$$U_{wc}^{RM-RA}(m, n, \alpha) \geq n_b U_b - (n_b - 1)\alpha$$

donde

$$\begin{aligned} n_a &= m + n - 1 - \left\lfloor \frac{m + n - 1}{n} \right\rfloor n \\ n_b &= n - n_a \\ U_b &= \left\lfloor \frac{m + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) \end{aligned}$$

Demostración:

Se restringirá la demostración a valores de α menores que U_b , pues en caso contrario

$$n_b U_b - (n_b - 1)\alpha \leq n_b U_b - (n_b - 1)U_b = U_b$$

y aplicando el teorema 13 cualquier conjunto de tareas de utilización menor o igual que U_b cabe en el multiprocesador.

Sea $\{\tau_1, \dots, \tau_m\}$ un conjunto de m tareas que no cabe en el multiprocesador. Sea τ_k la primera tarea del conjunto que no cabe en el multiprocesador. Obviamente $k \leq m$, y de esta forma hay dos casos posibles:

Caso 1: $\lfloor (k-1)/n \rfloor < \lfloor (m-1)/n \rfloor$

Como la tarea τ_k no cabe en el multiprocesador, aplicando el teorema 12 la utilización total de las k primeras tareas debe cumplir

$$\begin{aligned} \sum_{i=1}^k u_i &> \left(k + n - 1 - \left\lfloor \frac{k + n - 1}{n} \right\rfloor n \right) \left\lfloor \frac{k + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{k+n-1}{n} \rfloor} - 1 \right) + \\ &\quad \left(1 - k + \left\lfloor \frac{k + n - 1}{n} \right\rfloor n \right) \left\lfloor \frac{k + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{k+n-1}{n} \rfloor} - 1 \right) - (n-1)\alpha \end{aligned} \quad (4.7)$$

Defínase $(l-1) = \lfloor (m-1)/n \rfloor n$. Se cumple

$$\lfloor (l-1)/n \rfloor = \lfloor \lfloor (m-1)/n \rfloor \rfloor = \lfloor (m-1)/n \rfloor$$

Como en este caso

$$\lfloor (k-1)/n \rfloor < \lfloor (m-1)/n \rfloor = \lfloor (l-1)/n \rfloor$$

entonces se cumple $k < l$. El segundo término de la ecuación (4.7) es una función monótona decreciente en k , tal como se había indicado en el teorema 12. Por lo tanto se cumple

$$\begin{aligned} \sum_{i=1}^k u_i &> \left(l + n - 1 - \left\lfloor \frac{l + n - 1}{n} \right\rfloor n \right) \left\lfloor \frac{l + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{l+n-1}{n} \rfloor} - 1 \right) + \\ &\quad \left(1 - l + \left\lfloor \frac{l + n - 1}{n} \right\rfloor n \right) \left\lfloor \frac{l + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{l+n-1}{n} \rfloor} - 1 \right) - (n-1)\alpha \end{aligned}$$

Sustituyendo en esta ecuación $(l-1)$ y $\lfloor (l-1)/n \rfloor n$ por su valor, $\lfloor (m-1)/n \rfloor n$, y teniendo en cuenta además las relaciones (iv) y (v) de la página 245 resulta

$$\sum_{i=1}^k u_i > n \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) - (n-1)\alpha$$

Empleando la notación n_a , n_b y U_b propuesta en el enunciado del teorema, se tiene

$$\sum_{i=1}^k u_i > (n_a + n_b)U_b - (n_a + n_b - 1)\alpha = n_b U_b - (n_b - 1)\alpha + n_a(U_b - \alpha)$$

Una de las restricciones planteadas al principio de la demostración era $U_b > \alpha$, pues en caso contrario el conjunto de tareas cabía en el multiprocesador. Por lo tanto resulta

$$\sum_{i=1}^k u_i > n_b U_b - (n_b - 1)\alpha$$

Teniendo en cuenta que la utilización total de las k primeras tareas no puede ser mayor que la utilización total del conjunto completo

$$U = \sum_{i=1}^m u_i > \sum_{i=1}^k u_i > n_b U_b - (n_b - 1)\alpha$$

Una condición necesaria que debe cumplir un conjunto de tareas que no cabe en n procesadores es que su utilización total U cumpla la condición anterior. Se deduce entonces que todo conjunto de tareas de utilización total menor o igual que $n_b U_b - (n_b - 1)\alpha$ cabe en n procesadores.

Debe tenerse en cuenta que no se ha hecho ninguna suposición con respecto al algoritmo de asignación de tareas a procesadores, salvo que este sea razonable. Por consiguiente se cumple

$$U_{wc}^{RM-RA}(m, n, \alpha) \geq n_b U_b - (n_b - 1)\alpha$$

donde RA es un algoritmo razonable de asignación de tareas arbitrario.

Caso 2: $\lfloor (k-1)/n \rfloor = \lfloor (m-1)/n \rfloor$

Las $(k-1)$ primeras tareas se asignan con éxito a los procesadores y por lo tanto se cumple

$$\sum_{i=1}^k u_i = \sum_{j=1}^n U_j + u_k \quad (4.8)$$

Donde U_j es la utilización total de las tareas asignadas al procesador P_j . Puesto que el algoritmo de asignación es razonable, aplicando la ecuación (2.1) de la página 29 se tiene

$$U_j > (m_j + 1)(2^{1/(m_j+1)} - 1) - u_k \quad (4.9)$$

para $j = 1, \dots, n$. Donde m_j es el número de tareas previamente asignadas al

procesador P_j , y u_k es el factor de utilización de la tarea τ_k .

Antes de asignar la tarea τ_k se han asignado con éxito $(k - 1)$ tareas que se distribuyen entre los n procesadores. Si estas $(k - 1)$ tareas se distribuyesen equitativamente en número entre los n procesadores se llegaría a una situación en la que habría procesadores con $\lfloor (k - 1)/n \rfloor$ tareas y procesadores con $\lceil (k - 1)/n \rceil$ tareas. En total habría $(k - 1 - \lfloor (k - 1)/n \rfloor n)$ procesadores con $\lceil (k - 1)/n \rceil$ tareas, y $(n - k + 1 + \lfloor (k - 1)/n \rfloor n)$ procesadores con $\lfloor (k - 1)/n \rfloor n$ tareas.

El reparto equitativo en número de las $(k - 1)$ primeras tareas entre los n procesadores hace que el conjunto de los $(n - k + 1 + \lfloor (k - 1)/n \rfloor n)$ procesadores que menos tareas reciben reciba en total un número de tareas mayor que con cualquier otra asignación. Por lo tanto, el reparto equitativo maximiza el número de tareas a repartir entre dichos procesadores. Como además el reparto equitativo de las $(k - 1)$ tareas provoca que los $(n - k + 1 + \lfloor (k - 1)/n \rfloor n)$ procesadores que menos tareas reciben reciban el mismo número de tareas, resulta que aplicando el lema 6 el reparto equitativo minimiza la suma de los términos $(m_j + 1)(2^{1/(m_j+1)} - 1)$ asociada al conjunto de los $(n - k + 1 + \lfloor (k - 1)/n \rfloor n)$ que menos tareas reciben.

Los restantes $(k - 1 - \lfloor (k - 1)/n \rfloor n)$ procesadores cumplen $U_j > 0$, pues en caso contrario si existiese un procesador con $U_j = 0$ entonces la tarea τ_k cabría en ese procesador, lo cual representaría una contradicción.

Aplicando la ecuación (4.9) a los $(n - k + 1 + \lfloor (k - 1)/n \rfloor n)$ procesadores que menos tareas reciben, la ecuación $U_j > 0$ a los restantes y sustituyendo en la ecuación (4.8) resulta

$$\sum_{i=1}^k u_i > (n - k + 1 + \lfloor (k - 1)/n \rfloor n) \left\lfloor \frac{k + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{k+n-1}{n} \rfloor} - 1 \right) - (n - k + 1 + \lfloor (k - 1)/n \rfloor n - 1)u_k$$

Por definición de α se cumple $\alpha \geq u_k$. Por lo tanto se tiene

$$\sum_{i=1}^k u_i > (n - k + 1 + \lfloor (k - 1)/n \rfloor n) \left\lfloor \frac{k + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{k+n-1}{n} \rfloor} - 1 \right) - (n - k + 1 + \lfloor (k - 1)/n \rfloor n - 1)\alpha$$

El segundo término de la desigualdad es una función monótona decreciente en el valor de k bajo la restricción $\lfloor (k - 1)/n \rfloor = \lfloor (m - 1)/n \rfloor$ asociada al caso 2, y teniendo en cuenta que $k \leq m$

$$\sum_{i=1}^k u_i > (n - m + 1 + \lfloor (m - 1)/n \rfloor n) \left\lfloor \frac{m + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) - (n - m + 1 + \lfloor (m - 1)/n \rfloor n - 1)\alpha$$

La utilización total del conjunto de tareas es mayor o igual que la utilización total

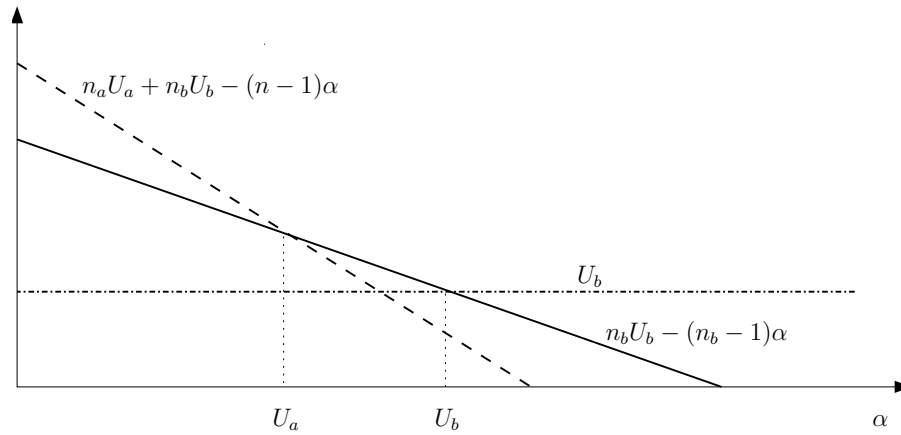


Figura 4.2: Representación de las cotas inferiores de los límites de utilización proporcionadas por los teoremas 12, 13, y 14.

de las k primeras tareas y por lo tanto se cumple

$$U \geq \sum_{i=1}^k u_i > (n - m + 1 + \lfloor (m - 1)/n \rfloor n) \left\lfloor \frac{m + n - 1}{n} \right\rfloor \left(2^{\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) - (n - m + 1 + \lfloor (m - 1)/n \rfloor n - 1)\alpha$$

Una condición necesaria que debe cumplir un conjunto de tareas que no cabe en n procesadores es que su utilización total U cumpla la condición anterior. Se deduce entonces que todo conjunto de tareas de utilización total menor o igual que el segundo término de la ecuación anterior cabe en n procesadores.

Debe tenerse en cuenta que no se ha hecho ninguna suposición con respecto al algoritmo de asignación de tareas a procesadores, salvo que este sea razonable. Por consiguiente, teniendo en cuenta las definiciones de n_b y U_b proporcionadas en el enunciado del teorema se obtiene

$$U_{wc}^{RM-RA}(m, n, \alpha) \geq n_b U_b - (n_b - 1)\alpha$$

donde RA es un algoritmo razonable de asignación de tareas arbitrario. ■

Una vez se ha llegado a este punto resulta conveniente comparar las cotas inferiores proporcionadas por los teoremas 12, 13, y 14. Para unos valores fijos de m y n se han representado en la figura 4.2 las tres cotas anteriores como una función de α . Puede observarse en la figura 4.2 que para $\alpha > U_b$ el teorema 13 proporciona la cota inferior más alta. Si $U_a \leq \alpha \leq U_b$ entonces es el teorema 14 el que proporciona la cota inferior más alta. Finalmente, si $\alpha < U_a$ el teorema 12 proporciona la cota inferior más alta. Teniendo en cuenta estas observaciones se procede a enunciar el corolario 2.

Corolario 2 Sea RA un algoritmo razonable de asignación de tareas a procesadores arbitrario. Sea $m > \beta_{RM}n$. Se cumple

$$U_{wc}^{RM-RA}(m, n, \alpha) \geq \begin{cases} n_a U_a + n_b U_b - (n-1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

donde

$$\begin{aligned} n_a &= m - \left\lfloor \frac{m-1}{n} \right\rfloor n - 1 \\ n_b &= n - n_a \\ U_a &= \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lceil \frac{m+n-1}{n} \rceil} - 1 \right) \\ U_b &= \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) \end{aligned}$$

Demostración:

La demostración es directa a partir de los teoremas 12, 13, y 14. ■

A continuación se obtiene una cota superior del límite de utilización asociado a cualquier algoritmo de asignación de tareas bajo planificación RM. Por lo tanto, dicha cota es aplicable a los algoritmos razonables de asignación.

Teorema 15 Sea AA un algoritmo de asignación de tareas a procesadores arbitrario (razonable o no). Si $m > \beta_{RM}n$ se cumple

$$U_{wc}^{RM-AA} \leq (\beta_{RM}n + 1)(2^{1/(\beta_{RM}+1)} - 1)$$

Demostración:

Se demostrará que existe un conjunto de m tareas $\{\tau_1, \dots, \tau_m\}$, con factores de utilización $0 < u_i \leq \alpha$ para $i = 1, \dots, m$, y utilización total

$$U = (\beta_{RM}n + 1)(2^{1/(\beta_{RM}+1)} - 1) + \epsilon$$

donde $\epsilon \rightarrow 0^+$, el cual no cabe en n procesadores empleando cualquier algoritmo de asignación de tareas a procesadores.

Se construirá este conjunto de m tareas a partir de dos subconjuntos de tareas: un primer subconjunto de $(m - \beta_{RM}n - 1)$ tareas, y un segundo subconjunto con $(\beta_{RM}n + 1)$ tareas.

Todas las tareas del primer subconjunto tienen el mismo factor de utilización de valor

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

donde $i = 1, \dots, (m - \beta_{RM}n - 1)$.

Todas las tareas del segundo subconjunto tienen el mismo factor de utilización de valor

$$u_i = (2^{1/(\beta_{RM}+1)} - 1) + \frac{\epsilon}{m}$$

donde $i = (m - \beta_{RM}n), \dots, m$.

Se puede comprobar que la utilización total del conjunto de tareas completo es

$$U = (\beta_{RM}n + 1)(2^{1/(\beta_{RM}+1)} - 1) + \epsilon$$

En primer lugar, es necesario comprobar que los factores de utilización de ambos subconjuntos son válidos, es decir, se cumple $0 < u_i \leq \alpha$ para $i = 1, \dots, m$.

Comprobación de los factores de utilización del primer subconjunto.

Eligiendo ϵ suficientemente pequeño se obtiene

$$0 < u_i = \frac{\epsilon}{m} \leq \alpha$$

Comprobación de los factores de utilización del segundo subconjunto.

A partir de la definición de β_{RM} , se tiene que $(\beta_{RM} + 1)$ tareas de factor de utilización α no caben en un procesador. Por lo tanto, $(\beta_{RM} + 1)\alpha > (\beta_{RM} + 1)(2^{1/(\beta_{RM}+1)} - 1)$, y se tiene

$$\alpha > (2^{1/(\beta_{RM}+1)} - 1) \quad (4.10)$$

Siempre es posible encontrar un número real entre dos números reales. Por consiguiente, existe un real positivo ϵ verificando

$$\alpha > (2^{1/(\beta_{RM}+1)} - 1) + \frac{\epsilon}{m} = u_i$$

lo cual demuestra que los factores de utilización del segundo subconjunto son menores o iguales que α cuando $\epsilon \rightarrow 0^+$. Además, los factores de utilización del segundo subconjunto son obviamente mayores que cero.

A partir de los resultados anteriores se concluye que el conjunto de tareas propuesto es válido. A continuación se demostrará que dicho conjunto de tareas no cabe en n procesadores empleando planificación RM en cada procesador, cualquiera que sea el algoritmo de asignación de tareas empleado.

El conjunto contiene $(\beta_{RM}n + 1)$ tareas que pertenecen al segundo subconjunto. Por consiguiente, al menos un procesador de los n disponibles debe alojar $(\beta_{RM} + 1)$ o más de estas tareas. Sin embargo, ningún procesador puede alojar $(\beta_{RM} + 1)$ o más tareas del segundo subconjunto, pues $(\beta_{RM} + 1)$ de estas tareas en total acumulan una utilización por encima del límite.

$$(\beta_{RM} + 1) \left((2^{1/(\beta_{RM}+1)} - 1) + \frac{\epsilon}{m} \right) > (\beta_{RM} + 1)(2^{1/(\beta_{RM}+1)} - 1)$$

De esta forma se deduce que el conjunto de tareas propuesto de utilización total

$$U = (\beta_{RM}n + 1)(2^{1/(\beta_{RM}+1)} - 1) + \epsilon$$

no cabe en n procesadores cualquiera que sea el algoritmo de asignación AA cuando

$\epsilon \rightarrow 0^+$, luego se cumple

$$U_{wc}^{RM-AA} \leq (\beta_{RM}n + 1)(2^{1/(\beta_{RM}+1)} - 1)$$

NOTA: las tareas del primer subconjunto son necesarias únicamente dentro de la demostración para cumplir la restricción de que el conjunto de tareas esté formado por m tareas. ■

En el apartado 4.2.4 se demostrará que el límite de utilización para RM-WF toma el valor

$$U_{wc}^{RM-WF} = \begin{cases} n_a U_a + n_b U_b - (n-1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

el cual coincide con la cota inferior dada en el corolario 2. Por lo tanto se tiene

$$L_{RM} = L_{RM}(m, n, \alpha) = \begin{cases} n_a U_a + n_b U_b - (n-1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases} \quad (4.11)$$

De forma análoga, en el apartado 4.2.7 se demostrará que el límite de utilización asociado a cualquier algoritmo razonable RAD con ordenación de tareas en sentido decreciente toma el valor

$$U_{wc}^{RM-RAD} = (\beta_{RM}n + 1)(2^{1/(\beta_{RM}+1)} - 1)$$

el cual coincide con la cota superior dada en el teorema 15. Por lo tanto se tiene

$$H_{RM} = H_{RM}(n, \beta_{RM}) = (\beta_{RM}n + 1)(2^{1/(\beta_{RM}+1)} - 1) \quad (4.12)$$

4.2.2 Límite de utilización para RM-FF

El algoritmo de asignación de tareas denominado FF asigna cada tarea al primer procesador que encuentra con capacidad residual suficiente para albergar la tarea. Los procesadores se recorren en el sentido $\{P_1, \dots, P_n\}$. La descripción detallada de este algoritmo de asignación de tareas puede encontrarse en el apartado 2.1.1.

La demostración del límite de utilización para RM-FF es bastante compleja, por lo que se indicará en primer lugar la estructura de la demostración. Los pasos que se seguirán son los siguientes:

1. El teorema 16 obtiene una cota superior del límite de utilización empleando planificación RM-FF.
2. Se demuestra el lema 7. Este lema relaciona los límites de utilización empleando el mismo algoritmo razonable de asignación, el mismo número de procesadores pero distinto número de tareas. Este lema se emplea para demostrar el teorema 17.

3. Se demuestra el teorema 17, el cual relaciona el límite de utilización asociado a m tareas y n procesadores con el límite de utilización asociado a $(m - \beta_{RM})$ tareas y $(n - 1)$ procesadores, para planificación RM-FF.
4. A partir del resultado dado en el paso 3 y el límite de utilización para planificación RM en un monoprocesador, el teorema 18 obtiene una cota inferior del límite de utilización para RM-FF.
5. La cota inferior dada en el paso 4 y la cota superior dada en el paso 1 coinciden. Por lo tanto, ambas cotas coinciden además con el límite de utilización para RM-FF, tal como muestra finalmente el teorema 18.

El teorema siguiente proporciona una cota superior del límite de utilización empleando planificación RM y asignación de tareas FF. La demostración está basada en encontrar un conjunto de tareas que no cabe en el multiprocesador.

Teorema 16 *Si $m > \beta_{RM}n$ se cumple*

$$U_{wc}^{RM-FF}(m, n, \alpha) \leq (n - 1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} \\ + (m - \beta_{RM}(n - 1))(2^{1/(m - \beta_{RM}(n - 1))} - 1)$$

Demostración:

Defínase

$$g(m, n, \beta_{RM}) = (n - 1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} \\ + (m - \beta_{RM}(n - 1))(2^{1/(m - \beta_{RM}(n - 1))} - 1)$$

Se probará que existe un conjunto de m tareas, $\{\tau_1, \dots, \tau_m\}$, con factores de utilización $0 < u_i \leq \alpha$ para $i = 1, \dots, m$ y utilización total $U = g(m, n, \beta_{RM}) + \epsilon$, con $\epsilon \rightarrow 0^+$, el cual no cabe en n procesadores empleando la asignación de tareas FF y la condición de planificabilidad de Liu y Layland (1973) para RM.

El conjunto de m tareas se divide en dos subconjuntos: un primer subconjunto de $(m - \beta_{RM}n)$ tareas y un segundo subconjunto de $\beta_{RM}n$ tareas.

Todas las tareas del primer subconjunto tienen el mismo factor de utilización de valor

$$u_i = \frac{(m - \beta_{RM}(n - 1))(2^{1/(m - \beta_{RM}(n - 1))} - 1) - (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}}{(m - \beta_{RM}n)} \quad (4.13)$$

donde $i = 1, \dots, (m - \beta_{RM}n)$.

Todas las tareas del segundo subconjunto tienen el mismo factor de utilización de valor

$$u_i = (2^{1/(\beta_{RM}+1)} - 1) + \frac{\epsilon}{\beta_{RM}n}$$

donde $i = (m - \beta_{RM}n + 1), \dots, m$.

Se puede comprobar que la utilización total del conjunto de tareas completo es $g(m, n, \beta_{RM}) + \epsilon$.

En primer lugar es necesario probar que los factores de utilización de ambos subconjuntos son válidos, es decir, $0 < u_i \leq \alpha$ para $i = 1, \dots, m$.

Comprobación de los factores de utilización del primer subconjunto.

Por hipótesis, $m > \beta_{RM}n$ y por lo tanto $m - \beta_{RM}(n - 1) > \beta_{RM}$. Aplicando la relación (i) de la página 245 se tiene $m - \beta_{RM}(n - 1) \geq \beta_{RM} + 1$. Aplicando ahora la relación (ii) se tiene $(m - \beta_{RM}(n - 1))(2^{1/(m - \beta_{RM}(n - 1))} - 1) \leq (\beta_{RM} + 1)(2^{1/(\beta_{RM} + 1)} - 1)$. Considerando esta expresión y la ecuación (4.13) se obtiene

$$u_i \leq \frac{(2^{1/(\beta_{RM} + 1)} - 1)}{(m - \beta_{RM}n)} \quad (4.14)$$

Por un lado, el lema 5 proporciona el valor $\beta_{RM} = \lfloor 1/\log_2(\alpha + 1) \rfloor$. De esta forma, $(\beta_{RM} + 1) > 1/\log_2(\alpha + 1)$ y despegando α

$$\alpha > (2^{1/(\beta_{RM} + 1)} - 1) \quad (4.15)$$

Por otro lado, $m > \beta_{RM}n$ por hipótesis, resultando $(m - \beta_{RM}n) > 0$ y aplicando la relación (i) se tiene

$$m - \beta_{RM}n \geq 1 \quad (4.16)$$

Sustituyendo (4.15) y (4.16) en (4.14) demuestra que $u_i < \alpha$ para todas las tareas del primer subconjunto.

A continuación se demostrará que los factores de utilización del primer subconjunto son mayores que cero. A partir de las relaciones (i), (ii), y a partir de la ecuación (4.16) se obtiene

$$\begin{aligned} (m - \beta_{RM}(n - 1))(2^{1/(m - \beta_{RM}(n - 1))} - 1) &> \ln 2 \\ (2^{1/(\beta_{RM} + 1)} - 1)\beta_{RM} &< \ln 2 \\ m - \beta_{RM}n &\geq 1 \end{aligned}$$

Sustituyendo las expresiones anteriores en la ecuación (4.13) proporciona $u_i > 0$ para todas las tareas del primer subconjunto.

Comprobación de los factores de utilización del segundo subconjunto.

Siempre es posible encontrar un número real entre dos números reales. Por consiguiente, a partir de la ecuación (4.15), debe existir un valor positivo $\epsilon/(\beta_{RM}n)$, tal que

$$\alpha > (2^{1/(\beta_{RM} + 1)} - 1) + \frac{\epsilon}{\beta_{RM}n} = u_i$$

lo cual demuestra que los factores de utilización del segundo subconjunto son menores que α cuando $\epsilon \rightarrow 0^+$. Además, los factores de utilización del segundo subconjunto son obviamente mayores que cero.

A partir de los resultados anterior se concluye que el conjunto de tareas propuesto es válido. A continuación se demostrará que no cabe en n procesadores, empleando el límite de utilización de Liu y Layland (1973) para RM y la asignación FF.

El primer subconjunto de tareas, $\{\tau_1, \dots, \tau_{m-\beta_{RM}n}\}$, y las primeras β_{RM} tareas del segundo subconjunto, $\{\tau_{m-\beta_{RM}n+1}, \dots, \tau_{m-\beta_{RM}n+\beta_{RM}}\}$, no caben en el procesador P_1 , puesto que la utilización total de estas tareas está por encima del límite

$$\begin{aligned} \sum_{i=1}^{m-\beta_{RM}n+\beta_{RM}} u_i &= \sum_{i=1}^{m-\beta_{RM}n} u_i + \sum_{i=m-\beta_{RM}n+1}^{m-\beta_{RM}n+\beta_{RM}} u_i \\ &= (m - \beta_{RM}(n - 1))(2^{1/(m-\beta_{RM}(n-1))} - 1) + \frac{\epsilon}{n} \\ &> (m - \beta_{RM}(n - 1))(2^{1/(m-\beta_{RM}(n-1))} - 1) \end{aligned}$$

Sin embargo, a partir de la expresión anterior se puede demostrar que si se elimina la tarea $\tau_{m-\beta_{RM}n+\beta_{RM}}$, entonces el primer subconjunto de tareas y las primeras $(\beta_{RM} - 1)$ tareas del segundo subconjunto caben en el procesador P_1 .

Por lo tanto quedan $(\beta_{RM}(n - 1) + 1)$ tareas de factor de utilización

$$(2^{1/(\beta_{RM}+1)} - 1) + \frac{\epsilon}{\beta_{RM}n}$$

la cuales FF intenta asignar a los últimos $(n - 1)$ procesadores, $\{P_2, \dots, P_n\}$.

Ningún procesador del conjunto $\{P_2, \dots, P_n\}$ puede albergar $(\beta_{RM} + 1)$ o más tareas del segundo subconjunto, pues $(\beta_{RM} + 1)$ de estas tareas tienen una utilización total por encima del límite.

$$(\beta_{RM} + 1) \left((2^{1/(\beta_{RM}+1)} - 1) + \frac{\epsilon}{\beta_{RM}n} \right) > (\beta_{RM} + 1)(2^{1/(\beta_{RM}+1)} - 1)$$

Sin embargo, por definición de β_{RM} , cada procesador en $\{P_2, \dots, P_n\}$ puede contener β_{RM} tareas. De tal forma que β_{RM} tareas pueden asignarse a cada uno de esos procesadores.

Como consecuencia, las tareas $\{\tau_{m-\beta_{RM}n+\beta_{RM}}, \dots, \tau_{m-1}\}$ se asignan a procesadores, pero la última, τ_m , no puede asignarse a ningún procesador.

De esta forma se concluye que el conjunto de tareas propuesto, de utilización total $g(m, n, \beta_{RM}) + \epsilon$, no cabe en n procesadores cuando $\epsilon \rightarrow 0^+$, y por lo tanto el límite de utilización $U_{wc}^{RM-FF}(m, n, \beta_{RM})$, debe ser menor o igual que $g(m, n, \beta_{RM})$. ■

La demostración del teorema 17 requiere el lema 7 que se demuestra a continuación. Este lema relaciona los límites de utilización asociados a un mismo número de procesadores, pero distinto número de tareas, empleando una algoritmo de asignación razonable, representado por RA. El resultado es válido para la asignación de tareas FF, pues es válido para cualquier algoritmo razonable de asignación que no realice ordenación de tareas.

Lema 7 *Sea RA un algoritmo razonable de asignación de tareas a procesadores que no realiza ordenación previa de las tareas antes de realizar la asignación. Se cumple*

$$U_{wc}^{RM-RA}(q, n, \alpha) \geq U_{wc}^{RM-RA}(m, n, \alpha) \quad \text{para } q < m$$

Demostración:

La demostración es totalmente análoga a la dada en el lema 3, página 47, para planificación EDF-RA. Es suficiente sustituir las iniciales EDF por las iniciales RM. ■

A continuación se demostrará una expresión que relaciona los límites de utilización asociados a multiprocesadores con n y $(n - 1)$ procesadores, empleando asignación de tareas FF. Esto permitirá obtener una cota inferior del límite de utilización para asignación de tareas FF, yendo del caso $n = 1$ (caso monoprocesador) hasta un multiprocesador general con un número arbitrario n de procesadores.

Teorema 17 *Si $m > \beta_{RM}n$ se cumple*

$$U_{wc}^{RM-FF}(m, n, \alpha) \geq (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + U_{wc}^{RM-FF}(m - \beta_{RM}, n - 1, \alpha)$$

Demostración:

Se probará que cualquier conjunto de m tareas $\{\tau_1, \dots, \tau_m\}$, con factores de utilización $0 < u_i \leq \alpha$ para $i = 1, \dots, m$, y utilización total menor o igual que

$$(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + U_{wc}^{RM-FF}(m - \beta_{RM}, n - 1, \alpha)$$

cabe en n procesadores empleando planificación RM en cada procesador y asignación de tareas FF.

Hay dos casos posibles:

Caso 1: Las primeras $(m - \beta_{RM})$ tareas tienen utilización total menor o igual que $U_{wc}^{RM-FF}(m - \beta_{RM}, n - 1, \alpha)$, esto es

$$\sum_{i=1}^{m-\beta_{RM}} u_i \leq U_{wc}^{RM-FF}(m - \beta_{RM}, n - 1, \alpha)$$

Las primeras $(m - \beta_{RM})$ tareas caben en los primeros $(n - 1)$ procesadores, pues su utilización total está por debajo del límite de utilización. Por lo tanto, empleando el algoritmo FF se asignan a los $(n - 1)$ primeros procesadores. Las restantes β_{RM} tareas caben en el último procesador, pues la definición de β_{RM} implica que al menos β_{RM} tareas siempre caben en un procesador. El hecho de que quepan en el último procesador no significa que se asignen al último procesador. De todas formas, el caso en el cual alguna de estas β_{RM} se asigna a alguno de los primeros $(n - 1)$ procesadores es más favorable, pues habría menos tareas que asignar al último procesador.

Caso 2: Las primeras $(m - \beta_{RM})$ tareas tienen una utilización total mayor que $U_{wc}^{RM-FF}(m - \beta_{RM}, n - 1, \alpha)$, esto es,

$$\sum_{i=1}^{m-\beta_{RM}} u_i > U_{wc}^{RM-FF}(m - \beta_{RM}, n - 1, \alpha)$$

En este caso se demostrará que el conjunto de las m tareas aún cabe en n procesa-

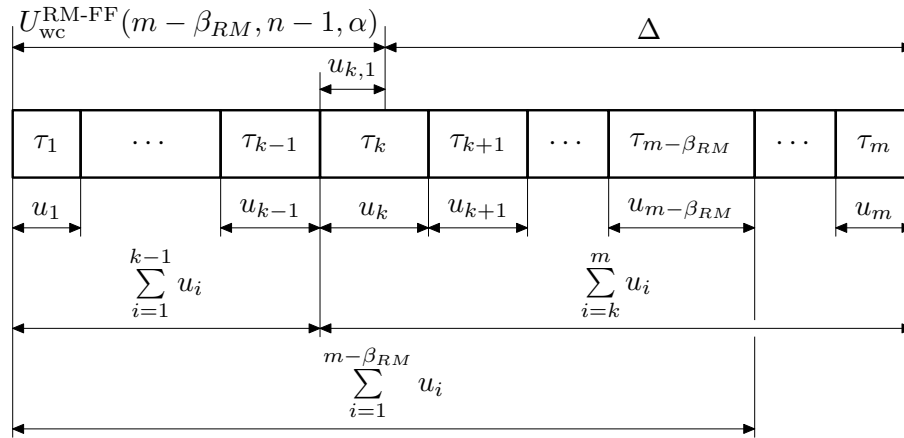


Figura 4.3: Situación general en el *caso 2* del teorema 17.

dores si su utilización total es igual a

$$U_{wc}^{RM-FF}(m - \beta_{RM}, n - 1, \alpha) + \Delta$$

donde $\Delta \in \mathbb{R}$, y

$$\Delta \leq (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}$$

Debe existir una tarea τ_k cuyo factor de utilización u_k , sumado a los previos factores de utilización u_i , hace que el límite $U_{wc}^{RM-FF}(m - \beta_{RM}, n - 1, \alpha)$ se supere. Esta situación se muestra en la figura 4.3, la cual representa gráficamente los factores de utilización de todas las tareas y las relaciones que existen entre diferentes cantidades y sumatorios empleados a lo largo de la demostración. El valor del subíndice k se obtiene como el entero que cumple la siguiente condición

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

Debe notarse que $k \leq m - \beta_{RM}$, pues de otra forma se estaría en el *caso 1*.

Se puede demostrar que las primeras $(k - 1)$ tareas caben en los primeros $(n - 1)$ procesadores. La utilización total de las primeras $(k - 1)$ tareas cumple

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

Aplicando el lema 7 con $(m - \beta_{RM}) > (k - 1)$ y con RA=FF, se obtiene

$$U_{wc}^{RM-FF}(m - \beta_{RM}, n - 1, \alpha) \leq U_{wc}^{RM-FF}(k - 1, n - 1, \alpha)$$

y por lo tanto

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

De esta forma, las primeras $(k-1)$ tareas caben en los $(n-1)$ primeros procesadores. Por lo tanto, empleando el algoritmo de asignación FF estas primeras $(k-1)$ tareas se asignan a los $(n-1)$ primeros procesadores. Falta por demostrar que las restantes $(m-k+1)$ tareas se pueden asignar a alguno de los huecos que quedan en los primeros $(n-1)$ procesadores, o bien al último procesador que de momento está vacío.

Analicemos la situación actual, representada en la figura 4.3. Se tienen $(m-k+1)$ tareas de factores de utilización comprendidos entre cero y α . Se tiene que $(m-k+1) > \beta_{RM}$, pues en caso contrario se estaría en el *caso 1*. La utilización total de estas $(m-k+1)$ tareas es $u_{k,1} + \Delta$, y por lo tanto es menor o igual que $u_{k,1} + (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}$. El valor de $u_{k,1}$ es menor que α , pues $u_{k,1} < u_k \leq \alpha$. Se dispone de huecos en los primeros $(n-1)$ procesadores suficientes para albergar al menos $(m-k+1 - \beta_{RM})$ tareas de utilización total menor o igual que $u_{k,1}$, además de un procesador, el último, completamente vacío.

Bajo estas restricciones debe demostrarse que los huecos disponibles en los primeros $(n-1)$ procesadores, y el último procesador proporcionan espacio suficiente para albergar las últimas $(m-k+1)$ tareas.

Para realizar la demostración el conjunto de todas las posibilidades definido por las restricciones anteriores se reducirá, introduciendo una restricción adicional que llevará al caso más desfavorable. La restricción adicional consiste en considerar que ninguna de las últimas $(m-k+1)$ tareas se asigna a alguno de los $(n-1)$ primeros procesadores. Este resultado es intuitivamente claro, pero se puede demostrar de una manera formal. Supóngase que alguna de las $(m-k+1)$ tareas se asignase a uno de los $(n-1)$ primeros procesadores, habría dos posibilidades. En la primera posibilidad, el nuevo número de tareas sin asignar, esto es $(m-k)$, es igual a β_{RM} por lo que las $(m-k)$ tareas caben en el último procesador y finaliza la demostración. La segunda posibilidad proporciona el caso más desfavorable, en el cual $(m-k)$ es mayor que β_{RM} . En este último caso la situación es análoga a la representada en la figura 4.4, pero con valores de $u_{k,1}$ y Δ aún más restringidos. Este razonamiento se puede repetir con todas las tareas en $\{\tau_k, \dots, \tau_m\}$ que se asignasen a los $(n-1)$ primeros procesadores obteniéndose que el conjunto de tareas cabe en los procesadores, o bien un nuevo conjunto $\{\tau_k, \dots, \tau_m\}$ análogo al de la figura 4.3 en el que ninguna de estas tareas se asigna a alguno de los $(n-1)$ primeros procesadores.

La consideración del caso más desfavorable descrito por las restricciones anteriores tiene efecto sobre los factores de utilización de las tareas $\{\tau_k, \dots, \tau_m\}$. Estos factores de utilización deben ser mayores que $u_{k,1}$. La demostración es simple, en caso contrario el algoritmo FF asignaría alguna de estas tareas a alguno de los primeros $(n-1)$ procesadores incumpliendo la restricción del caso peor impuesta.

Para probar que las últimas $(m-k+1)$ tareas caben en el último procesador se debe probar que la utilización total de esas tareas es menor o igual que $(m-k+1)(2^{1/(m-k+1)} - 1)$, esto es,

$$\sum_{i=k}^m u_i \leq (m-k+1)(2^{1/(m-k+1)} - 1)$$

La figura 4.3 muestra que

$$\sum_{i=k}^m u_i = u_{k,1} + \Delta \quad (4.17)$$

Como se comentó anteriormente, todos los factores de utilización en ese sumatorio son mayores que o iguales a $u_{k,1}$, y por lo tanto

$$(m - k + 1)u_{k,1} < u_{k,1} + \Delta < u_{k,1} + (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}$$

Despejando $u_{k,1}$

$$u_{k,1} < \frac{(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}}{m - k} \quad (4.18)$$

Sustituyendo el valor de $u_{k,1}$ dado por la ecuación (4.18) en la ecuación (4.17) se obtiene

$$\begin{aligned} \sum_{i=k}^m u_i &< \frac{(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}}{m - k} + \Delta \\ &< \frac{(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}}{m - k} + (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} \quad \text{por def. de } \Delta \\ &= \frac{(m - k + 1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}}{m - k} \end{aligned}$$

En el caso 2 se tiene $(m - k) \geq \beta_{RM}$. Por lo tanto, aplicando la relación (iii) de la página 245 se obtiene

$$(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} \leq (2^{1/(m-k+1)} - 1)(m - k)$$

y por lo tanto

$$\sum_{i=k}^m u_i \leq (m - k + 1)(2^{1/(m-k+1)} - 1)$$

Esta ecuación muestra que las $(m - k + 1)$ tareas cumplen la condición de planificabilidad de Liu y Layland (1973) para RM, de tal forma que son planificables sobre el último procesador.

Se ha probado que cualquier conjunto de m tareas y utilización total

$$U_{wc}^{RM-FF}(m - \beta_{RM}, n - 1, \beta_{RM}) + \Delta \leq U_{wc}^{RM-FF}(m - \beta_{RM}, n - 1, \beta_{RM}) + (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}$$

cabe en n procesadores, de tal forma que el límite de utilización $U_{wc}^{RM-FF}(m, n, \alpha)$ debe cumplir

$$U_{wc}^{RM-FF}(m, n, \alpha) \geq (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + U_{wc}^{RM-FF}(m - \beta_{RM}, n - 1, \alpha)$$

■

El límite de utilización para planificación RM y asignación de tareas FF se obtiene

a partir de los teoremas 16 y 17, tal como muestra el teorema 18.

Teorema 18 *Si $m > \beta_{RM}n$ se cumple que el límite de utilización para RM-FF es una función única y exclusivamente de m , n y β_{RM} , y viene dada por*

$$U_{wc}^{RM-FF}(m, n, \beta_{RM}) = (n-1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + (m - \beta_{RM}(n-1))(2^{1/(m-\beta_{RM}(n-1))} - 1) \quad (4.19)$$

Demostración:

En primer lugar se obtendrá una cota inferior del límite de utilización para un conjunto de m tareas sobre un multiprocesador con n procesadores.

El teorema 17 relaciona el límite de utilización de conjuntos de m tareas en multiprocesadores de n procesadores, con el límite de utilización de conjuntos de $(m - \beta_{RM})$ tareas en multiprocesadores de $(n - 1)$ procesadores.

$$U_{wc}^{RM-FF}(m, n, \alpha) \geq (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + U_{wc}^{RM-FF}(m - \beta_{RM}, n - 1, \alpha) \quad (4.20)$$

Pero el teorema 17 también relaciona el límite de utilización de conjuntos de $(m - \beta_{RM})$ tareas en multiprocesadores de $(n - 1)$ procesadores, con el límite de utilización de conjuntos de $(m - 2\beta_{RM})$ tareas en multiprocesadores de $(n - 2)$ procesadores.

$$U_{wc}^{RM-FF}(m - \beta_{RM}, n - 1, \alpha) \geq (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + U_{wc}^{RM-FF}(m - 2\beta_{RM}, n - 2, \alpha) \quad (4.21)$$

Sustituyendo la ecuación (4.21) en la ecuación (4.20) se obtiene

$$U_{wc}^{RM-FF}(m, n, \alpha) \geq 2(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + U_{wc}^{RM-FF}(m - 2\beta_{RM}, n - 2, \alpha)$$

Este procedimiento puede repetirse, hasta finalmente relacionar el límite de utilización de conjuntos de m tareas en multiprocesadores de n procesadores, con el límite de utilización de conjuntos de $(m - (n - 1)\beta_{RM})$ tareas en un monoprocesador.

$$U_{wc}^{RM-FF}(m, n, \alpha) \geq (n - 1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + U_{wc}^{RM-FF}(m - (n - 1)\beta_{RM}, 1, \alpha) \quad (4.22)$$

El límite de utilización para $(m - (n - 1)\beta_{RM})$ tareas y un procesador coincide con el límite de Liu y Layland (1973) para RM, el cual es independiente del valor de α .

$$U_{wc}^{RM-FF}(m - (n - 1)\beta_{RM}, 1, \alpha) = (m - (n - 1)\beta_{RM})(2^{1/(m - (n - 1)\beta_{RM})} - 1) \quad (4.23)$$

Sustituyendo la ecuación (4.23) en la ecuación (4.22) se obtiene una cota inferior del

límite de utilización de m tareas en n procesadores.

$$U_{wc}^{RM-FF}(m, n, \alpha) \geq (n-1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + (m - (n-1)\beta_{RM})(2^{1/(m-(n-1)\beta_{RM})} - 1) \quad (4.24)$$

El teorema 16 demostró que

$$U_{wc}^{RM-FF}(m, n, \alpha) \leq (n-1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + (m - (n-1)\beta_{RM})(2^{1/(m-(n-1)\beta_{RM})} - 1) \quad (4.25)$$

A partir de la ecuación (4.24) y de la ecuación (4.25) se tiene

$$U_{wc}^{RM-FF}(m, n, \alpha) = (n-1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + (m - (n-1)\beta_{RM})(2^{1/(m-(n-1)\beta_{RM})} - 1)$$

Se observa que U_{wc}^{RM-FF} sólo depende del número de tareas m , del número de procesadores n , y del valor de β_{RM} , luego finalmente se concluye con el resultado

$$U_{wc}^{RM-FF} = U_{wc}^{RM-FF}(m, n, \beta_{RM}) = (n-1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + (m - \beta_{RM}(n-1))(2^{1/(m-\beta_{RM}(n-1))} - 1)$$

■

4.2.3 Límite de utilización para RM-BF

El algoritmo de asignación *Best Fit* asigna cada tarea al procesador con menor capacidad residual, pero suficiente para albergar la tarea. Si dos o más procesadores tienen la misma capacidad residual se elige entre estos empleando el algoritmo FF, recorriendo los procesadores en el sentido $\{P_1, \dots, P_n\}$. La descripción detallada de este algoritmo de asignación de tareas puede encontrarse en el apartado 2.1.1.

La demostración del límite de utilización para planificación RM-BF es análoga a la vista en el apartado 4.2.2 para la planificación RM-FF.

Teorema 19 *Si $m > \beta_{RM}n$ se cumple*

$$U_{wc}^{RM-BF}(m, n, \alpha) \leq (n-1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + (m - \beta_{RM}(n-1))(2^{1/(m-\beta_{RM}(n-1))} - 1)$$

Demostración:

La demostración es idéntica a la del teorema 19. Es suficiente cambiar las iniciales FF por las iniciales BF. ■

A continuación se demostrará una expresión que relaciona los límites de utilización asociados a multiprocesadores con n y $(n-1)$ procesadores, empleando

asignación de tareas BF. Esto permitirá obtener una cota inferior del límite de utilización para asignación de tareas BF, yendo del caso $n = 1$ (caso monoprocesador) hasta un multiprocesador general con un número arbitrario n de procesadores. Una gran parte de la demostración es idéntica a la demostración del teorema 17. No obstante, la demostración se proporciona completa por motivos de claridad.

Teorema 20 *Si $m > \beta_{RM}n$ se cumple*

$$U_{wc}^{RM-BF}(m, n, \alpha) \geq (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + U_{wc}^{RM-BF}(m - \beta_{RM}, n - 1, \alpha)$$

Demostración:

Se probará que cualquier conjunto de m tareas $\{\tau_1, \dots, \tau_m\}$, con factores de utilización $0 < u_i \leq \alpha$ para $i = 1, \dots, m$, y utilización total menor o igual que

$$(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + U_{wc}^{RM-BF}(m - \beta_{RM}, n - 1, \alpha)$$

cabe en n procesadores empleando planificación RM en cada procesador y asignación de tareas BF.

Hay dos casos posibles:

Caso 1: Las primeras $(m - \beta_{RM})$ tareas tienen utilización total menor o igual que $U_{wc}^{RM-BF}(m - \beta_{RM}, n - 1, \alpha)$, esto es

$$\sum_{i=1}^{m-\beta_{RM}} u_i \leq U_{wc}^{RM-BF}(m - \beta_{RM}, n - 1, \alpha)$$

Las primeras $(m - \beta_{RM})$ tareas caben en los primeros $(n - 1)$ procesadores, pues su utilización total está por debajo del límite de utilización. Por lo tanto, empleando el algoritmo BF se asignan a los $(n - 1)$ primeros procesadores. Las restantes β_{RM} tareas caben en el último procesador, pues la definición de β_{RM} implica que al menos β_{RM} tareas siempre caben en un procesador. El hecho de que quepan en el último procesador no significa que se asignen al último procesador. De todas formas, el caso en el cual alguna de estas β_{RM} se asigna a alguno de los primeros $(n - 1)$ procesadores es más favorable, pues habría menos tareas que asignar al último procesador.

Caso 2: Las primeras $(m - \beta_{RM})$ tareas tienen una utilización total mayor que $U_{wc}^{RM-BF}(m - \beta_{RM}, n - 1, \alpha)$, esto es,

$$\sum_{i=1}^{m-\beta_{RM}} u_i > U_{wc}^{RM-BF}(m - \beta_{RM}, n - 1, \alpha)$$

En este caso se demostrará que el conjunto de las m tareas aún cabe en n procesadores si su utilización total es igual a

$$U_{wc}^{RM-BF}(m - \beta_{RM}, n - 1, \alpha) + \Delta$$

donde $\Delta \in \mathbb{R}$, y

$$\Delta \leq (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}$$

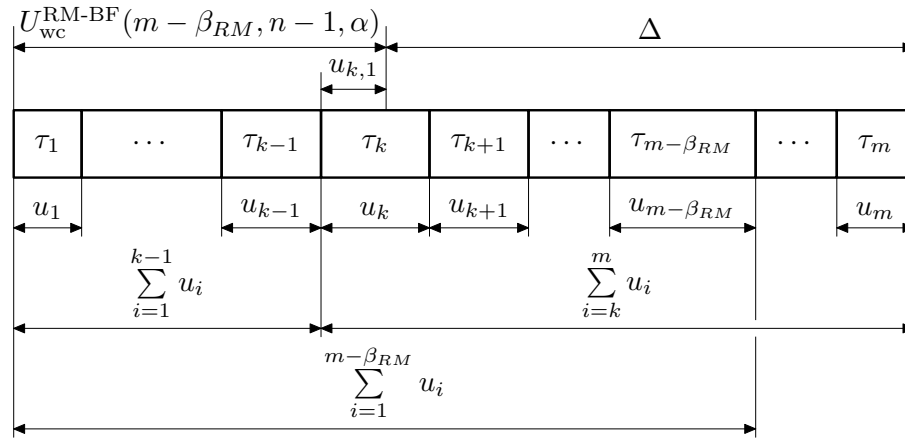


Figura 4.4: Situación general en el *caso 2* del teorema 20.

Debe existir una tarea τ_k , cuyo factor de utilización u_k , sumado a los previos factores de utilización u_i , hace que el límite $U_{wc}^{RM-BF}(m - \beta_{RM}, n - 1, \alpha)$ se supere. Esta situación se muestra en la figura 4.4, la cual representa gráficamente los factores de utilización de todas las tareas y las relaciones que existen entre diferentes cantidades y sumatorios empleados a lo largo de la demostración. El valor del subíndice k se obtiene como el entero que cumple la siguiente condición:

$$\sum_{i=1}^{k-1} u_i \leq U_{wc}^{RM-BF}(m - \beta_{RM}, n - 1, \alpha) < \sum_{i=1}^k u_i$$

Debe notarse que $k \leq m - \beta_{RM}$, pues de otra forma se estaría en el *caso 1*.

Se puede demostrar que las primeras $(k - 1)$ tareas caben en los primeros $(n - 1)$ procesadores. La utilización total de las primeras $(k - 1)$ tareas cumple

$$\sum_{i=1}^{k-1} u_i \leq U_{wc}^{RM-BF}(m - \beta_{RM}, n - 1, \alpha)$$

Aplicando el lema 7 con $(m - \beta_{RM}) > (k - 1)$ y con RA=BF, se obtiene

$$U_{wc}^{RM-BF}(m - \beta_{RM}, n - 1, \alpha) \leq U_{wc}^{RM-BF}(k - 1, n - 1, \alpha)$$

y por lo tanto

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

De esta forma, las primeras $(k - 1)$ tareas caben en los $(n - 1)$ primeros procesadores. Por lo tanto, empleando el algoritmo de asignación BF estas primeras $(k - 1)$ tareas se asignan a los $(n - 1)$ primeros procesadores. Falta por demostrar que las restantes $(m - k + 1)$ tareas se pueden asignar a alguno de los huecos que quedan en los primeros $(n - 1)$ procesadores, o bien al último procesador que de momento está vacío.

Analicemos la situación actual, representada en la figura 4.4. Se tienen $(m - k + 1)$

tareas de factores de utilización comprendidos entre cero y α . Se tiene que $(m - k + 1) > \beta_{RM}$, pues en caso contrario se estaría en el *caso 1*. La utilización total de estas $(m - k + 1)$ tareas es $u_{k,1} + \Delta$, y por lo tanto es menor que $u_{k,1} + (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}$. El valor de $u_{k,1}$ es menor o igual que α , pues $u_{k,1} < u_k \leq \alpha$. Se dispone de huecos en los primeros $(n - 1)$ procesadores suficientes para albergar al menos $(m - k + 1 - \beta_{RM})$ tareas de utilización total menor o igual que $u_{k,1}$, además de un procesador, el último, completamente vacío.

Bajo estas restricciones debe demostrarse que los huecos disponibles en los primeros $(n - 1)$ procesadores, y el último procesador proporcionan espacio suficiente para albergar las últimas $(m - k + 1)$ tareas.

Para realizar la demostración el conjunto de todas las posibilidades definido por las restricciones anteriores se reducirá, introduciendo restricciones adicionales que llevarán al caso más desfavorable. Estas restricciones adicionales se indican a continuación:

- Se tiene un único procesador entre los $(n - 1)$ primeros con capacidad residual $u_{k,1}$ capaz de albergar hasta $(m - k + 1 - \beta_{RM})$ tareas. Los demás procesadores de entre los $(n - 1)$ primeros están completamente llenos, es decir, su capacidad residual es cero.
- Ninguna de las últimas $(m - k + 1)$ tareas se asigna a alguno de los $(n - 1)$ primeros procesadores. Este resultado es intuitivamente claro, pero se puede demostrar de una manera formal. Supóngase que alguna de las $(m - k + 1)$ tareas se asignase a uno de los $(n - 1)$ primeros procesadores, habría dos posibilidades. En la primera posibilidad, el nuevo número de tareas sin asignar, esto es $(m - k)$, es igual a β_{RM} por lo que las $(m - k)$ tareas caben en el último procesador y finaliza la demostración. La segunda posibilidad proporciona el caso más desfavorable, en el cual $(m - k)$ es mayor que β_{RM} . En este último caso la situación es análoga a la representada en la figura 4.4, pero con valores de $u_{k,1}$ y Δ aún más restringidos. Este razonamiento se puede repetir con todas las tareas en $\{\tau_k, \dots, \tau_m\}$ que se asignasen a los $(n - 1)$ primeros procesadores obteniéndose que el conjunto de tareas cabe en los procesadores, o bien un nuevo conjunto $\{\tau_k, \dots, \tau_m\}$ análogo al de la figura 4.4 en el que ninguna de estas tareas se asigna a alguno de los $(n - 1)$ primeros procesadores.

La consideración del caso más desfavorable descrito por las restricciones anteriores tiene efecto sobre los factores de utilización de las tareas $\{\tau_k, \dots, \tau_m\}$. Estos factores de utilización deben ser mayores que $u_{k,1}$. No obstante, la demostración de este hecho es más compleja que la vista en el teorema 17 para RM-FF.

En primer lugar, las hipotéticas tareas en $\{\tau_k, \dots, \tau_m\}$ que no caben en ninguno de los n procesadores se mueven a la derecha de τ_m en la figura 4.4. La denominación de hipotéticas responde a que en realidad como resultado de la demostración estas no existirán. Este movimiento no modifica la asignación de tareas a procesadores. A continuación se modifican los subíndices de las tareas teniendo en cuenta la nueva ordenación, de tal forma que la figura 4.4 sigue siendo válida. En primer lugar debe notarse que $u_m > u_{k,1}$ pues de acuerdo con una de las restricciones τ_m no se asigna a ninguno de los $(n - 1)$ primeros procesadores. Si u_m fuese menor o igual que $u_{k,1}$

la única posibilidad sería que $\{\tau_k, \dots, \tau_m\}$ cupiesen en el último procesador, en cuyo caso el conjunto de tareas sería planificable y terminaría la demostración.

La demostración de que el factor de utilización de todas las tareas $\{\tau_k, \dots, \tau_m\}$ es mayor que $u_{k,1}$ se realizará por reducción al absurdo.

Sea τ_l la primera tarea en $\{\tau_k, \dots, \tau_m\}$ de factor de utilización menor o igual que $u_{k,1}$. Debe cumplirse que la capacidad residual del procesador una vez asignadas las tareas $\{\tau_k, \dots, \tau_{l-1}\}$ debe ser menor que $u_{k,1}$, pues en caso contrario τ_l se asignaría a uno de los primeros $(n-1)$ procesadores. Esto es,

$$(l-k+1)(2^{1/(l-k+1)} - 1) - \sum_{i=k}^{l-1} u_i < u_{k,1} \quad (4.26)$$

La figura 4.4 muestra que

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

Como $u_m > u_{k,1}$ y $m \geq l$ resulta

$$\sum_{i=k}^{l-1} u_i < \Delta \leq (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} \quad (4.27)$$

Llevando este resultado a la ecuación (4.26) se tiene

$$u_{k,1} > (l-k+1)(2^{1/(l-k+1)} - 1) - (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} \quad (4.28)$$

Por otra parte, al ser τ_l la primera tarea en $\{\tau_k, \dots, \tau_m\}$ de factor de utilización menor o igual que $u_{k,1}$, resulta que las tareas $\{\tau_k, \dots, \tau_{l-1}\}$ deben tener factor de utilización mayor que $u_{k,1}$. Por lo tanto teniendo en cuenta la ecuación (4.27) se deduce

$$u_{k,1} < \frac{(2^{1/(\beta_{RM}+1)-1} - 1)\beta_{RM}}{l-k} \quad (4.29)$$

Juntando las ecuaciones (4.28) y (4.29) se obtiene

$$(l-k+1)(2^{1/(l-k+1)} - 1) - (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} < \frac{(2^{1/(\beta_{RM}+1)-1} - 1)\beta_{RM}}{l-k} \quad (4.30)$$

Simplificando esta expresión resulta

$$(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} > (2^{1/(l-k+1)} - 1)(l-k)$$

Aplicando las relaciones (iii) y (i) de la página 245 en este orden

$$\beta_{RM} \geq (l-k+1)$$

A partir de esta expresión, la relación (ii) y que los factores de utilización de todas

las tareas son menores o iguales que α resulta

$$(l - k + 1)(2^{1/(l-k+1)} - 1) - \sum_{i=k}^{l-1} u_i \geq \beta_{RM}(2^{1/\beta_{RM}} - 1) - (\beta_{RM} - 1)\alpha \quad (4.31)$$

Por definición de β_{RM} se tiene

$$\beta_{RM} = \left\lfloor \frac{1}{\log_2(\alpha + 1)} \right\rfloor$$

por lo tanto $\beta_{RM} \leq 1/\log_2(\alpha + 1)$ y $\alpha \leq (2^{1/\beta_{RM}} - 1)$. Sustituyendo esta desigualdad en la ecuación (4.31) resulta

$$(l - k + 1)(2^{1/(l-k+1)} - 1) - \sum_{i=k}^{l-1} u_i \geq (2^{1/\beta_{RM}} - 1) \geq \alpha$$

Llevando este resultado a la ecuación (4.26) resulta $u_{k,1} > \alpha$ lo cual representa una contradicción pues $u_{k,1}$ debe ser menor que α .

Se acaba de demostrar que en el caso más desfavorable las tareas $\{\tau_k, \dots, \tau_m\}$ tienen un factor de utilización mayor que $u_{k,1}$. A partir de este punto la demostración es idéntica a la vista en el teorema 18 para RM-FF.

Para probar que las últimas $(m - k + 1)$ tareas caben en el último procesador se debe probar que la utilización total de esas tareas es menor o igual que $(m - k + 1)(2^{1/(m-k+1)} - 1)$, esto es,

$$\sum_{i=k}^m u_i \leq (m - k + 1)(2^{1/(m-k+1)} - 1)$$

La figura 4.4 muestra que

$$\sum_{i=k}^m u_i = u_{k,1} + \Delta \quad (4.32)$$

Como se comentó anteriormente, todos los factores de utilización en ese sumatorio son mayores que o iguales a $u_{k,1}$, y por lo tanto

$$(m - k + 1)u_{k,1} < u_{k,1} + \Delta < u_{k,1} + (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}$$

Despejando $u_{k,1}$

$$u_{k,1} < \frac{(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}}{m - k} \quad (4.33)$$

Sustituyendo el valor de $u_{k,1}$ dado por la ecuación (4.33) en la ecuación (4.32) se

obtiene

$$\begin{aligned} \sum_{i=k}^m u_i &< \frac{(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}}{m-k} + \Delta \\ &< \frac{(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}}{m-k} + (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} \quad \text{por def. de } \Delta \\ &= \frac{(m-k+1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}}{m-k} \end{aligned}$$

En el caso 2 se tiene $(m-k) \geq \beta_{RM}$. Por lo tanto, aplicando la relación (iii) se obtiene

$$(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} \leq (2^{1/(m-k+1)} - 1)(m-k)$$

y por lo tanto

$$\sum_{i=k}^m u_i \leq (m-k+1)(2^{1/(m-k+1)} - 1)$$

Esta ecuación muestra que las $(m-k+1)$ tareas cumplen la condición de planificabilidad de Liu y Layland (1973) para RM, de tal forma que son planificables sobre el último procesador.

Se ha probado que cualquier conjunto de m tareas y utilización total

$$U_{wc}^{RM-BF}(m-\beta_{RM}, n-1, \beta_{RM}) + \Delta \leq U_{wc}^{RM-BF}(m-\beta_{RM}, n-1, \beta_{RM}) + (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}$$

cabe en n procesadores, de tal forma que el límite de utilización $U_{wc}^{RM-BF}(m, n, \alpha)$ debe cumplir

$$U_{wc}^{RM-BF}(m, n, \alpha) \geq (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + U_{wc}^{RM-BF}(m - \beta_{RM}, n - 1, \alpha)$$

■

El límite de utilización para planificación RM y asignación de tareas BF se obtiene a partir de los teoremas 19 y 20, tal como muestra el teorema 21.

Teorema 21 Si $m > \beta_{RM}n$ se cumple que el límite de utilización para RM-BF es una función única y exclusivamente de m , n y β_{RM} , y viene dada por

$$\begin{aligned} U_{wc}^{RM-BF}(m, n, \beta_{RM}) &= (n-1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + \\ &\quad (m - \beta_{RM}(n-1))(2^{1/(m-\beta_{RM}(n-1))} - 1) \end{aligned}$$

Demostración:

La demostración es idéntica a la realizada en el teorema 18, sólo hay que sustituir las iniciales FF por las iniciales BF. ■

4.2.4 Límite de utilización para RM-WF

El algoritmo de asignación *Worst Fit* (WF) asigna cada tarea al procesador con mayor capacidad residual, suficiente para albergar la tarea. Si dos o más procesadores tienen la misma capacidad residual se elige entre estos empleando el algoritmo FF, recorriendo los procesadores en el sentido $\{P_1, \dots, P_n\}$. La descripción detallada de este algoritmo de asignación de tareas puede encontrarse en el capítulo 2.1.1.

Para obtener el límite de utilización U_{wc}^{RM-WF} se obtendrá en primer lugar una cota superior de dicho límite de utilización.

Esta cota superior se verá que coincide con la cota inferior proporcionada por el corolario 2 para cualquier algoritmo razonable de asignación de tareas a procesadores. Como el algoritmo de asignación WF es razonable resulta que el límite de utilización asociado al mismo coincide con la cota inferior dada por el corolario 2, tal como indica el corolario 3.

Teorema 22 *Si $m > \beta_{RM}n$ se cumple*

$$U_{wc}^{RM-WF}(m, n, \alpha) \leq \begin{cases} n_a U_a + n_b U_b - (n-1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

donde

$$\begin{aligned} n_a &= m - \left\lfloor \frac{m-1}{n} \right\rfloor n - 1 \\ n_b &= n - n_a \\ U_a &= \left\lceil \frac{m+n-1}{n} \right\rceil \left(2^{\lceil \frac{m+n-1}{n} \rceil} - 1 \right) \\ U_b &= \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) \end{aligned}$$

Demostración:

La demostración se dividirá en tres casos, cada caso correspondiente a un intervalo de valores de α .

Caso 1:

Este caso se corresponde con $\alpha \leq U_a$. Se demostrará que existe un conjunto de m tareas $\{\tau_1, \dots, \tau_m\}$, de factores de utilización menores o iguales que α , y utilización total

$$n_a U_a + n_b U_b - (n-1)\alpha + \epsilon$$

con $\epsilon \rightarrow 0^+$, el cual no cabe en n procesadores empleando el algoritmo WF de asignación de tareas a procesadores.

El conjunto de m tareas se construye tal como sigue, estrictamente en el orden indicado. Se tienen $\lfloor (m-1)/n \rfloor$ subconjuntos de n tareas cada uno. Cada uno de

estos subconjuntos esta formado por n_a tareas de factor de utilización

$$u_a = \frac{U_a - \alpha}{\lceil (m-1)/n \rceil} + \frac{\epsilon}{m-1}$$

y por n_b tareas de factor de utilización

$$u_b = \frac{U_b - \alpha}{\lfloor (m-1)/n \rfloor} + \frac{\epsilon}{m-1}$$

A continuación de los $\lfloor (m-1)/n \rfloor$ subconjuntos anteriores vienen n_a tareas de factor de utilización u_a . Finalmente viene la última tarea, τ_m , de factor de utilización α .

Se puede comprobar que el conjunto tiene m tareas y que la utilización total del conjunto es

$$n_a U_a + n_b U_b - (n-1)\alpha + \epsilon$$

En primer lugar hay que demostrar que el conjunto de tareas es válido, es decir, hay que demostrar que los factores de utilización cumplen $0 < u_i \leq \alpha$ para $i = 1, \dots, n$.

El factor de utilización de todas las tareas es obviamente mayor que cero. Además el factor de utilización de la última tarea es menor o igual que α pues es igual a α . Hay que demostrar entonces que u_a y u_b son menores o iguales que α . A partir de la definición de u_a y u_b se deduce que $u_b \geq u_a$, pues $U_b \geq U_a$ y $\lfloor (m-1)/n \rfloor \leq \lceil (m-1)/n \rceil$. Por lo tanto es suficiente demostrar $u_b \leq \alpha$.

Sustituyendo el valor de U_b en la definición de u_b

$$u_b = \frac{\left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) - \alpha}{\lfloor (m-1)/n \rfloor} + \frac{\epsilon}{m-1} \quad (4.34)$$

Por hipótesis del teorema se tiene $m > \beta_{RM}n$, y por lo tanto aplicando la relación (i) de la página 245 se tiene $m \geq (\beta_{RM}n + 1)$. Como consecuencia, $\lfloor (m-1)/n \rfloor \geq \beta_{RM}$, y $\lfloor (m+n-1)/n \rfloor \geq \beta_{RM} + 1$.

Por otra parte en la página 157 se demostró

$$\alpha > (2^{1/(\beta_{RM}+1)} - 1)$$

Se deduce entonces empleando la relación (ii)

$$\left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) \leq (\beta_{RM} + 1)(2^{1/(\beta_{RM}+1)} - 1)$$

Llevando todas estas relaciones a la ecuación (4.34) se obtiene

$$u_b < (2^{1/(\beta_{RM}+1)} - 1) + \frac{\epsilon}{m-1}$$

Tal como se indicó anteriormente $\alpha > (2^{1/(\beta_{RM}+1)} - 1)$. Entre dos números reales es siempre posible encontrar otro número real, y por lo tanto eligiendo un ϵ

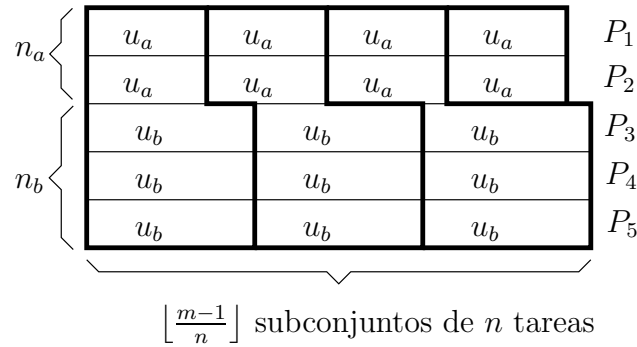


Figura 4.5: Asignación de las $(m - 1)$ primeras tareas en el *caso 1* del teorema 22, para $m = 18$ y $n = 5$.

suficientemente pequeño se tiene $u_b < \alpha$.

A continuación se demostrará que el conjunto de tareas propuesto no cabe en el multiprocesador. Las $(m - 1)$ primeras tareas se asignan tal como se indica en la figura 4.5. Como resultado de la asignación de las $(m - 1)$ primeras tareas, los n_a primeros procesadores reciben $\lceil (m - 1)/n \rceil$ tareas de factor de utilización u_a . Estos procesadores tienen una capacidad residual de valor

$$U_a - \lceil (m - 1)/n \rceil u_a = \alpha - \epsilon/(m - 1)$$

Por lo tanto, la última tarea de factor de utilización α no cabe en estos procesadores.

Con respecto a los restantes n_b procesadores sucede algo análogo. Estos procesadores reciben $\lfloor (m - 1)/n \rfloor$ tareas de factor de utilización u_b . Estos procesadores tienen una capacidad residual de valor

$$U_b - \lfloor (m - 1)/n \rfloor u_b = \alpha - \epsilon/(m - 1)$$

Por lo tanto, la última tarea de factor de utilización α no cabe en estos procesadores.

Caso 2:

Este caso se corresponde con $U_a \leq \alpha < U_b$. Se demostrará que existe un conjunto de m tareas $\{\tau_1, \dots, \tau_m\}$, de factores de utilización menores o iguales que α , y utilización total

$$n_b U_b - (n_b - 1)\alpha + \epsilon$$

con $\epsilon \rightarrow 0^+$, el cual no cabe en n procesadores empleando el algoritmo WF de asignación de tareas a procesadores.

El conjunto de m tareas se divide en cuatro subconjuntos. Un primer subconjunto formado por $(m - n - 1)$ tareas de factor de utilización

$$u_i = \frac{\epsilon}{(m - n_b - 1)}$$

para $i = 1, \dots, (m - n - 1)$.

Un segundo subconjunto formado por n_b tareas de factor de utilización

$$u_i = U_b - \alpha$$

para $i = (m - n), \dots, (m - n_a - 1)$.

Un tercer subconjunto formado por n_a tareas de factor de utilización

$$u_i = \frac{\epsilon}{(m - n_b - 1)}$$

para $i = (m - n_a), \dots, (m - 1)$.

Un cuarto subconjunto formado por una única tarea, τ_m , de factor de utilización $u_m = \alpha$

Se puede comprobar que la utilización total del conjunto de m tareas es igual a

$$n_b U_b - (n_b - 1)\alpha + \epsilon$$

Antes de demostrar que el conjunto de tareas no cabe en el multiprocesador es necesario demostrar que dicho conjunto es válido, es decir, debe demostrarse que se cumple $0 < u_i \leq \alpha$ para $i = 1, \dots, n$.

Resulta obvio que los factores de utilización de todas las tareas son mayores que cero. Por otra parte, los factores de utilización del primer y tercer subconjunto son menores o iguales que α eligiendo un ϵ suficientemente pequeño. El factor de utilización de la única tarea que compone el cuarto subconjunto es obviamente menor o igual que α . Resta por demostrar entonces que el factor de utilización de las tareas del segundo subconjunto son menores o iguales que α . Para ello debe tenerse en cuenta que para valores de x mayores que cero la función $x(2^{1/x} - 1)$ es monótona decreciente y decrece cada vez más lentamente, pues su primera derivada es negativa, y su segunda derivada positiva. Llevado esto a nuestro problema significa que

$$\begin{aligned} U_b - U_a &= \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) - \left\lceil \frac{m+n-1}{n} \right\rceil \left(2^{1/\lceil \frac{m+n-1}{n} \rceil} - 1 \right) \\ &\leq 1(2^1 - 1) - 2(2^{1/2} - 1) = 1 - 2(\sqrt{2} - 1) \end{aligned}$$

Como α se encuentra entre U_a y U_b , resulta

$$U_b - \alpha \leq U_b - U_a \leq 1 - 2(\sqrt{2} - 1) \approx 0'172$$

Al encontrarse α entre U_a y U_b entonces $\alpha > U_a$. Se sabe que $U_a > \ln 2$ por ser la función $x(2^{1/x} - 1)$ monótona decreciente para valores de x positivos, y por ser $\lim_{x \rightarrow \infty} x(2^{1/x} - 1) = \ln 2$. Como $0'172 < \ln 2 \approx 0'69$ los factores de utilización de las tareas del segundo subconjunto son menores que α .

A continuación se probará que el conjunto de tareas anterior no cabe en el multiprocesador. Las $(m-1)$ primeras tareas se asignan tal como se indica en la figura 4.6. Como resultado de la asignación de las $(m-1)$ primeras tareas, los n_a primeros procesadores reciben $\lceil (m-1)/n \rceil$ tareas de factor de utilización $u_i = \epsilon/(m - n_b - 1)$. La tarea τ_m de factor de utilización α no puede asignarse a ninguno de estos n_a pri-

$$u_\epsilon = \frac{\epsilon}{(m-n_b-1)}$$

n_a	u_ϵ	u_ϵ	u_ϵ	u_ϵ	P_1
	u_ϵ	u_ϵ	u_ϵ	u_ϵ	P_2
n_b	u_ϵ	u_ϵ	$U_b - \alpha$		P_3
	u_ϵ	u_ϵ	$U_b - \alpha$		P_4
	u_ϵ	u_ϵ	$U_b - \alpha$		P_5

Figura 4.6: Asignación de las $(m - 1)$ primeras tareas en el *caso 2* del teorema 22, para $m = 18$ y $n = 5$.

meros procesadores, pues el límite de utilización asociado a $\lceil (m - 1)/n \rceil + 1$ tareas es U_a y por lo tanto la capacidad residual de estos procesadores es

$$U_a - \left\lceil \frac{m - 1}{n} \right\rceil \epsilon < \alpha$$

Como resultado de la asignación de las $(m - 1)$ primeras tareas, los restantes n_b procesadores reciben $\lfloor (m - 1)/n \rfloor$ tareas, las cuales acumulan una utilización total por encima de $U_b - \alpha$. La tarea τ_m de factor de utilización α no puede asignarse a ninguno de estos n_b procesadores, pues el límite de utilización asociado a $\lfloor (m - 1)/n \rfloor + 1$ tareas es U_b .

Caso 3:

Este caso se corresponde con $U_b \leq \alpha$. Se demostrará que existe un conjunto de m tareas $\{\tau_1, \dots, \tau_m\}$, de factores de utilización menores o iguales que α , y utilización total

$$U = U_b + \epsilon$$

con $\epsilon \rightarrow 0^+$, el cual no cabe en n procesadores empleando el algoritmo WF de asignación de tareas a procesadores.

El conjunto de m tareas se construye a partir de dos subconjuntos. Un primer subconjunto formado por $(m - 1)$ tareas de factor de utilización $u_i = \epsilon/(m - 1)$, y una tarea τ_m de factor de utilización $u_m = U_b$. Puede comprobarse que la utilización total de este conjunto de tareas es $(U_b + \epsilon)$, y que además el factor de utilización de todas las tareas cumple $0 < u_i \leq \alpha$.

De acuerdo con el algoritmo de asignación WF las $(m - 1)$ primeras tareas se dividen equitativamente en número (en lo posible) entre los n procesadores. Como resultado, el procesador o procesadores que menos tareas recibirán, recibirán $\lfloor (m - 1)/n \rfloor$ tareas. Los demás recibirán $\lceil (m - 1)/n \rceil$ tareas. Si la tarea τ_m cupiese en alguno de los procesadores, cabría en uno de los procesadores con menos tareas, pues para estos el límite de utilización es más alto. Si embargo, la utilización total de $\lfloor (m - 1)/n \rfloor$ tareas de factor de utilización $\epsilon/(m - 1)$ y una tarea de factor de

utilización U_b es igual a

$$\lfloor (m-1)/n \rfloor \frac{\epsilon}{m-1} + U_b = \lfloor (m-1)/n \rfloor \frac{\epsilon}{m-1} + \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right)$$

y por lo tanto mayor que

$$\left(\left\lfloor \frac{m-1}{n} \right\rfloor + 1 \right) \left(2^{1/(\lfloor \frac{m-1}{n} \rfloor + 1)} - 1 \right) = \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right)$$

que es el límite de utilización para $\lfloor (m-1)/n \rfloor + 1$ tareas. En conclusión, la tarea τ_m no cabe en ninguno de los procesadores.

NOTA: El rango asociado a cada uno de los casos de la demostración no coincide exactamente con los rangos que aparecen en el enunciado del teorema. No obstante, debe tenerse en cuenta que la cota proporcionada en el enunciado del teorema es una función continua. ■

Corolario 3 Si $m > \beta_{RM}n$ se cumple

$$U_{wc}^{RM-WF}(m, n, \alpha) = \begin{cases} n_a U_a + n_b U_b - (n-1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

donde

$$\begin{aligned} n_a &= m - \left\lfloor \frac{m-1}{n} \right\rfloor n - 1 \\ n_b &= n - n_a \\ U_a &= \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) \\ U_b &= \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) \end{aligned}$$

Demostración

A partir del teorema 22 se tiene

$$U_{wc}^{RM-WF}(m, n, \alpha) \leq \begin{cases} n_a U_a + n_b U_b - (n-1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

Por otra parte, como WF es un algoritmo razonable de asignación de tareas, aplicando el corolario 2 se deduce

$$U_{wc}^{RM-WF}(m, n, \alpha) \geq \begin{cases} n_a U_a + n_b U_b - (n-1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

y por lo tanto

$$U_{wc}^{RM-FF}(m, n, \alpha) = \begin{cases} n_a U_a + n_b U_b - (n-1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

■

El límite de utilización empleando planificación RM-WF es el más bajo posible de entre todos los algoritmos razonables de asignación de tareas, pues coincide con la cota inferior proporcionada en el apartado 4.2.1.

4.2.5 Límite de utilización para RM-RF

El algoritmo de asignación *Random Fit* (RF) asigna de forma aleatoria cada tarea entre todos los procesadores con capacidad al residual suficiente para albergar la tarea. La descripción detallada de este algoritmo de asignación de tareas puede encontrarse en el apartado 2.1.1.

Este algoritmo es el peor de los algoritmos de asignación razonables en términos de límite utilización. La explicación es simple, la asignación de tareas a procesadores realizada por cualquier algoritmo razonable de asignación podría ser el resultado también de la asignación de tareas empleando el algoritmo RF. La probabilidad de que RF realice la misma asignación puede ser mayor o menor, pero en cualquier caso esa posibilidad siempre existe y debe ser tenida en cuenta.

A continuación, el teorema 23 proporciona el límite de utilización empleando planificación RM-RF.

Teorema 23 *Si $m > \beta_{RM}n$ se cumple*

$$U_{wc}^{RM-RF}(m, n, \alpha) = \begin{cases} n_a U_a + n_b U_b - (n-1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

donde

$$\begin{aligned} n_a &= m - \left\lfloor \frac{m-1}{n} \right\rfloor n - 1 \\ n_b &= n - n_a \\ U_a &= \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{\lceil \frac{m+n-1}{n} \rceil} - 1 \right) \\ U_b &= \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) \end{aligned}$$

Demostración

En el apartado 4.2.4 se obtuvo el límite de utilización empleando el algoritmo (razonable) de asignación de tareas WF. Como el límite de utilización empleando asignación RF no puede ser mayor que el límite de utilización empleando asignación WF

resulta

$$U_{wc}^{RM-RA}(m, n, \alpha) \leq \begin{cases} n_a U_a + n_b U_b - (n-1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

Por otra parte, a partir del corolario 2 se sabe que el límite de utilización asociado a todos los algoritmos razonables de asignación de tareas debe ser mayor o igual que

$$\begin{cases} n_a U_a + n_b U_b - (n-1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

y como el algoritmo RF es razonable se deduce

$$U_{wc}^{RM-RF}(m, n, \alpha) = \begin{cases} n_a U_a + n_b U_b - (n-1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

El algoritmo de asignación RF puede emplearse para obtener los límites de utilización en el caso de la creación y destrucción dinámica de tareas que ocurre en los cambios de modo, tal como se indica en el apartado 6.5.

4.2.6 Límite de utilización para RM-OPT

El algoritmo de asignación de tareas a procesadores denominado *Optimal* (OPT) es el algoritmo óptimo de asignación de tareas a procesadores. La obtención de la asignación óptima de tareas, es análoga al problema denominado *bin-packing*. Este problema pertenece a la clase de problemas *NP-hard* en el sentido estricto, por lo que no puede resolverse en general en tiempo pseudopolinomial. Aunque la búsqueda de la asignación óptima de tareas a procesadores no resulta práctica, sirve como referencia para conocer lo lejos que está un algoritmo heurístico de asignación del algoritmo óptimo de asignación. Por ejemplo, si los resultados obtenidos para un determinado algoritmo de asignación son cercanos a los obtenidos por el algoritmo óptimo, entonces no merece la pena la búsqueda de algoritmos heurísticos más complejos pues la mejora que puede obtenerse es pequeña.

La descripción detallada de este algoritmo de asignación de tareas puede encontrarse en el apartado 2.1.1.

El algoritmo de asignación OPT asigna de forma factible todos los conjuntos de tareas que pueden asignarse con otros algoritmos. Por lo tanto se deduce que el límite de utilización para RM-OPT es mayor o igual que el límite de utilización empleando otros algoritmos de asignación y planificación RM en cada procesador. En particular, los algoritmos razonables tipo RAD que incluyen una etapa previa de ordenación de las tareas en sentido de factores de utilización decrecientes proporcionan el máximo límite de utilización de entre todos los algoritmos razonables de asignación. La

demostración puede verse en el teorema 4 de la página 185. Como el algoritmo OPT es también razonable, resulta que el límite de utilización asociado al mismo debe coincidir con el límite de utilización asociado a los algoritmos RAD. Es decir,

$$U_{wc}^{RM-OPT} = U_{wc}^{RM-RAD}(m, n, \alpha) = (\beta_{RM}n + 1)(2^{1/(\beta_{RM}+1)} - 1)$$

El algoritmo FFD es del tipo RAD. Sin embargo, esto no significa que los algoritmos FFD y OPT sean capaces de asignar los mismos conjuntos de tareas, sino que comparten el caso peor. En realidad, hay conjuntos de tareas que pueden ser asignados empleando el algoritmo OPT pero no el FFD. El resultado inverso no es cierto por definición del algoritmo óptimo de asignación.

4.2.7 Límites de utilización empleando ordenación

Por motivos de claridad, en este apartado se repite la introducción a los límites de utilización empleando ordenación proporcionada en el apartado 3.2.7.

Los algoritmos de asignación FF, BF, WF y RF no son más que distintas formas de elegir un procesador de entre todos los procesadores en los que cabe una tarea. Estos algoritmos no suponen una determinada ordenación de las tareas antes de hacer la asignación. Las tareas vienen definidas únicamente por sus factores de utilización, por lo que la ordenación de tareas consiste en la ordenación basada en factores de utilización. Se consideran dos tipos de ordenación:

- Factores de utilización crecientes. Este tipo de ordenación se denota por I (*Increasing*). Después de realizar la ordenación los factores de utilización de las tareas cumplen la propiedad:

$$u_1 \leq u_2 \leq \dots \leq u_{m-1} \leq u_m$$

donde τ_1 es la primera tarea que se asigna, y τ_n la última.

- Factores de utilización decrecientes. Este tipo de ordenación se denota por D (*Decreasing*). Después de realizar la ordenación los factores de utilización de las tareas cumplen la propiedad:

$$u_1 \geq u_2 \geq \dots \geq u_{m-1} \geq u_m$$

donde τ_1 es la primera tarea que se asigna, y τ_n la última.

Si se considera ningún tipo de ordenación los algoritmos de asignación conservan los nombres anteriores, esto es, FF, BF, WF y RF. No obstante si llevan a cabo un determinado tipo de ordenación antes de hacer la asignación llevan una letra adicional que hace referencia a la misma. Para ordenación en sentido de factores de utilización se tienen los algoritmos FFI, BFI, WFI y RFI. Para ordenación en sentido decreciente de factores de utilización crecientes se tienen los algoritmos FFD, BFD, WFD y RFD.

En general, la ordenación de tareas tiene efecto sobre los límites de utilización absolutos y estadísticos, tal como se muestra en este apartado y en el apartado 4.4.

Sin embargo hay un precio a pagar. En primer lugar, el algoritmo de ordenación tiene una complejidad $O(m \log m)$, donde m es el número de tareas. Este algoritmo tiene en general un coste computacional no despreciable, aunque asumible en muchos casos. En segundo lugar, el proceso de ordenación es incompatible con una asignación en línea de tareas a procesadores. Con la asignación en línea no se conoce a priori el conjunto completo de tareas a asignar, por lo que cada vez que llega una nueva tarea, el proceso de ordenación puede dar lugar a que una tarea previamente asignada a un procesador deba reasignarse a otro procesador. Esto va en contra de la filosofía de la planificación multiprocesador con particionado, donde se trata de evitar la migración de tareas entre procesadores. La asignación en línea de tareas plantea además otros problemas que se tratan en el capítulo 6.5.

A continuación se muestra el efecto de la ordenación de tareas sobre los límites de utilización. Deliberadamente se ha excluido el análisis de los algoritmos que incluyen ordenación antes de realizar la asignación OPT (óptima). La razón es simple, la ordenación de tareas no influye sobre la asignación realizada empleando el algoritmo OPT.

Algoritmos de ordenación en sentido decreciente

De igual forma que sucedía empleando planificación local EDF, empleando planificación local RM la ordenación previa de las tareas en sentido de factores de utilización decrecientes hace que el límite de utilización asociado a todos los algoritmos de asignación razonables sea el mismo, y además de idéntico valor a la cota superior proporcionada por el teorema 15.

Teorema 24 *Sea RA un algoritmo razonable de asignación de tareas a procesadores que no asume ninguna ordenación de los factores de utilización. Sea RAD el algoritmo de asignación de tareas a procesadores obtenido ordenando previamente las tareas en sentido decreciente de factores de utilización y aplicando posteriormente el algoritmo de asignación RA. Si $m > \beta_{RM}n$ y $n > 1$, se cumple*

$$U_{wc}^{RM-RAD} \geq (\beta_{RM}n + 1)(2^{1/(\beta_{RM}n+1)} - 1)$$

Demostración:

Sea $\{\tau_1, \dots, \tau_m\}$ un conjunto de m tareas que no cabe en el multiprocesador. Sea τ_k la primera tarea del conjunto que no cabe en el multiprocesador. Puesto que el algoritmo de asignación es razonable, aplicando la ecuación (2.1) de la página 29 se tiene

$$U_j > (m_j + 1)(2^{1/(m_j+1)} - 1) - u_k \quad (4.35)$$

para $j = 1, \dots, n$. Donde U_j es la utilización total de las tareas previamente asignadas al procesador P_j , m_j es el número de estas tareas y u_k es el factor de utilización de la tarea τ_k .

La utilización total de las k primeras tareas se puede obtener como

$$\sum_{i=1}^k u_i = \sum_{j=1}^n U_j + u_k \quad (4.36)$$

A partir de la ecuación (4.35) y de la ecuación (4.36) se deduce

$$\sum_{i=1}^k u_i > \sum_{j=1}^n (m_j + 1)(2^{1/(m_j+1)} - 1) - (n - 1)u_k \quad (4.37)$$

Una consecuencia de la ordenación de tareas en sentido de factores de utilización decrecientes es que el factor de utilización de la tarea τ_k es menor o igual que el factor de utilización de las tareas $\{\tau_1, \dots, \tau_{k-1}\}$. Por lo tanto se cumple

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

Sustituyendo esta expresión en la ecuación (4.37) y despejando $\sum_{i=1}^k u_i$ se obtiene

$$\sum_{i=1}^k u_i > \frac{k}{k + n - 1} \sum_{j=1}^n (m_j + 1)(2^{1/(m_j+1)} - 1)$$

La suma de los factores de utilización de las k primeras tareas es siempre menor o igual que U , la suma de los factores de utilización de todas las tareas. Por lo tanto también se cumple

$$U > \frac{k}{k + n - 1} \sum_{j=1}^n (m_j + 1)(2^{1/(m_j+1)} - 1) \quad (4.38)$$

Las variables k y $\{m_1, \dots, m_n\}$ no pueden tomar cualquier valor. En particular $m_j \geq \beta_{RM}$, pues en caso contrario la tarea τ_k cabría en el procesador P_j , lo cual va en contra de la hipótesis. Debe tenerse en cuenta que todo procesador puede albergar al menos β_{RM} tareas. Por otra parte, se sabe que

$$k = 1 + \sum_{j=1}^n m_j$$

y por lo tanto $k \geq (\beta_{RM}n + 1)$. Bajo estas restricciones se puede demostrar que para $n > 1$ la función

$$\frac{k}{k + n - 1} \sum_{j=1}^n (m_j + 1)(2^{1/(m_j+1)} - 1)$$

se minimiza¹ para los valores $k = (\beta_{RM}n + 1)$ y $m_j = \beta_{RM}$ para $j = 1, \dots, n$. A partir de la ecuación (4.38) se deduce entonces

$$U > \frac{\beta_{RM}n + 1}{\beta_{RM}n + 1 + n - 1} \sum_{j=1}^n (\beta_{RM} + 1)(2^{1/(\beta_{RM}+1)} - 1) = (\beta_{RM}n + 1)(2^{1/(\beta_{RM}n+1)} - 1)$$

Una condición necesaria para que un conjunto de tareas no quepa en n procesadores es que su utilización total sea mayor que

$$(\beta_{RM}n + 1)(2^{1/(\beta_{RM}n+1)} - 1)$$

o lo que es equivalente, cualquier conjunto de tareas de utilización menor o igual que

$$(\beta_{RM}n + 1)(2^{1/(\beta_{RM}n+1)} - 1)$$

es planificable sobre n procesadores. Como no se ha realizado ninguna hipótesis sobre el algoritmo de asignación de tareas salvo que este sea razonable, y que todas las tareas a asignar estén ordenadas en sentido de factores de utilización decrecientes resulta que

$$U_{wc}^{RM-RAD} \geq (\beta_{RM}n + 1)(2^{1/(\beta_{RM}n+1)} - 1)$$

donde RAD es un algoritmo de asignación de tareas obtenido obtenido mediante previa ordenación de las tareas en sentido de factores de utilización decrecientes, y posterior asignación razonable de tareas. ■

Corolario 4 *Sea RA un algoritmo razonable de asignación de tareas a procesadores que no asume ninguna ordenación de los factores de utilización. Sea RAD el algoritmo de asignación de tareas a procesadores obtenido ordenando previamente las tareas en sentido decreciente de factores de utilización y aplicando posteriormente el algoritmo de asignación RA. Si $m > \beta_{RM}n$ se cumple*

$$U_{wc}^{RM-RAD}(n > 1, \beta_{RM}) = (\beta_{RM}n + 1)(2^{1/(\beta_{RM}n+1)} - 1) \quad (4.39)$$

Demostración:

Por una parte el teorema 24 proporciona la cota inferior

$$U_{wc}^{RM-RAD} \geq (\beta_{RM}n + 1)(2^{1/(\beta_{RM}n+1)} - 1)$$

para el caso $n > 1$.

Por otra parte el teorema 15 proporciona una cota superior del límite de utilización de cualquier algoritmo de asignación, y por lo tanto es aplicable a los algoritmos RAD.

$$U_{wc}^{RM-RAD} \leq (\beta_{RM}n + 1)(2^{1/(\beta_{RM}n+1)} - 1)$$

¹Para $n = 1$ la minimización ocurriría para $m_1 = m$, independientemente del valor de k .

Se obtiene entonces

$$U_{wc}^{RM-RAD}(n > 1, \beta_{RM}) = (\beta_{RM}n + 1)(2^{1/(\beta_{RM}n+1)} - 1)$$

■

En particular los algoritmos FFD, BFD, WFD y RFD tienen el mismo límite de utilización de valor

$$U_{wc}^{RM-FFD}(n > 1, \beta_{RM}) = U_{wc}^{RM-BFD}(n > 1, \beta_{RM}) = U_{wc}^{RM-WFD}(n > 1, \beta_{RM}) = \\ U_{wc}^{RM-RFD}(n > 1, \beta_{RM}) = (\beta_{RM}n + 1)(2^{1/(\beta_{RM}n+1)} - 1)$$

que coincide con el máximo posible. Por consiguiente todos estos algoritmos son óptimos desde el punto de vista del límite de utilización. Ni siquiera el algoritmo de asignación óptimo puede tener un límite de utilización mayor (de hecho tiene el mismo).

Para el caso $n = 1$, el límite de utilización coincide obviamente con el de Liu y Layland (1973) para planificación RM.

Algoritmos de ordenación en sentido creciente

De igual forma que se demostraba en el apartado 3.2.7 para planificación local EDF que la ordenación previa de las tareas no puede reducir el límite de utilización, se puede demostrar lo mismo de forma análoga para el caso de planificación local RM.

Como consecuencia $U_{wc}^{RM-FFI} \geq U_{wc}^{RM-FF}$, $U_{wc}^{RM-BFI} \geq U_{wc}^{RM-BF}$, $U_{wc}^{RM-WFI} \geq U_{wc}^{RM-WF}$, y $U_{wc}^{RM-RFI} \geq U_{wc}^{RM-RF}$.

En el caso de factores de utilización decrecientes el límite de utilización no es el mismo para todos los algoritmos razonables, al contrario de lo que sucede cuando los factores de utilización siguen una secuencia decreciente.

Algoritmo de asignación FFI

El teorema 16, en la página 159 demuestra que

$$U_{wc}^{RM-FFI}(m, n, \alpha) \leq (n - 1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} \\ + (m - \beta_{RM}(n - 1))(2^{1/(m-\beta_{RM}(n-1))} - 1)$$

para ello se empleó un conjunto de m tareas, dividido en dos subconjuntos: un primer subconjunto de $(m - \beta_{RM}n)$ tareas, y un segundo subconjunto de $\beta_{RM}n$ tareas, y se comprobó que no cabían en n procesadores empleando asignación FF.

Todas las tareas del primer subconjunto tienen el mismo factor de utilización de valor

$$u_i = \frac{(m - \beta_{RM}(n - 1))(2^{1/(m-\beta_{RM}(n-1))} - 1) - (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}}{(m - \beta_{RM}n)}$$

donde $i = 1, \dots, (m - \beta_{RM}n)$.

Todas las tareas del segundo subconjunto tienen el mismo factor de utilización

de valor

$$u_i = (2^{1/(\beta_{RM}+1)} - 1) + \frac{\epsilon}{\beta_{RM}n}$$

donde $i = (m - \beta_{RM}n + 1), \dots, m$.

Este conjunto de m tareas está ordenado en sentido de factores de utilización crecientes, pues como se demostrará a continuación los factores de utilización de las tareas del segundo subconjunto son mayores o iguales que los factores de utilización de las tareas del primer subconjunto. En la ecuación (4.14) del teorema 16 se muestra la siguiente restricción de los factores de utilización de las tareas del primer subconjunto

$$u_i \leq \frac{(2^{1/(\beta_{RM}+1)} - 1)}{(m - \beta_{RM}n)}$$

Como $m > \beta_{RM}n$, y m , β_{RM} y n son enteros resulta $m \geq (\beta_{RM} + 1)$. Se obtiene entonces

$$u_i \leq (2^{1/(\beta_{RM}+1)} - 1)$$

Por lo tanto, los factores de utilización de las tareas del primer subconjunto son menores o iguales que los factores de utilización de las tareas del segundo subconjunto.

El razonamiento empleado para obtener la cota superior del límite de utilización para planificación RM-FF podría emplearse para planificación RM-FFI, obteniéndose exactamente la misma cota. Teniendo en cuenta que el límite de utilización empleando asignación FFI no puede ser menor que el obtenido con asignación FF resulta que ambos límites de utilización son iguales.

$$\begin{aligned} U_{wc}^{RM-FFI}(m, n, \beta_{RM}) &= U_{wc}^{RM-FF}(m, n, \beta_{RM}) = \\ &= (n - 1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + (m - \beta_{RM}(n - 1))(2^{1/(m - \beta_{RM}(n - 1))} - 1) \end{aligned} \quad (4.40)$$

Algoritmo de asignación BFI

Siguiendo una línea de razonamiento idéntica a la mostrada en el caso de asignación FFI, se deduce que la cota superior proporcionada por el teorema 19 de la página 167 es válida para asignación BFI. Como por otra parte el límite de utilización para asignación BFI no puede ser menor que el límite de utilización para asignación BF resulta que ambos límites de utilización son iguales.

$$\begin{aligned} U_{wc}^{RM-BFI}(m, n, \beta_{RM}) &= U_{wc}^{RM-BF}(m, n, \beta_{RM}) = \\ &= (n - 1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + (m - \beta_{RM}(n - 1))(2^{1/(m - \beta_{RM}(n - 1))} - 1) \end{aligned} \quad (4.41)$$

Algoritmo de asignación WFI

Cuando las tareas se ordenan en sentido de factores de utilización crecientes el algoritmo de asignación de tareas WF asigna las tareas a los procesadores de forma estrictamente secuencial. Es decir, si se modifican los índices de las tareas para ser

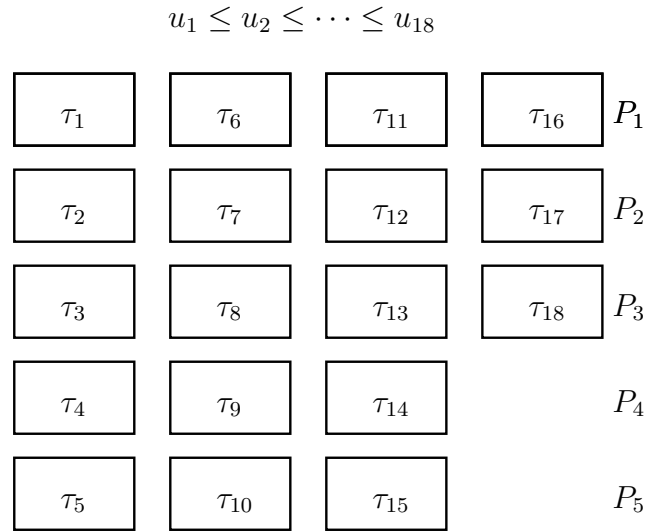


Figura 4.7: Ejemplo de asignación de tareas empleando el algoritmo WFI, para $m = 18$ y $n = 5$.

consecuentes con la ordenación, la tarea τ_k se asigna al procesador de índice

$$k - \left\lfloor \frac{k-1}{n} \right\rfloor n$$

Si no cabe en dicho procesador, entonces no cabe en ningún otro procesador. En la Figura 4.7 puede verse un ejemplo de asignación de tareas empleando el algoritmo WFI. La asignación WFI podría por lo tanto implementarse siguiendo el algoritmo simple 4.2.1. No se proporciona la demostración formal de que este algoritmo es equivalente al WFI pues es un resultado intuitivamente claro.

A continuación se obtiene el límite de utilización para asignación WFI. Para ello, en primer lugar se obtiene una cota inferior de dicho límite de utilización.

Teorema 25 *Si $m > \beta_{RM}n$ se cumple*

$$U_{wc}^{RM-WFI}(m, n, \alpha) \leq \begin{cases} nU_b - (n-1)\alpha & \text{si } \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

donde

$$U_b = \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right)$$

Demostración:

La demostración se dividirá en dos casos, cada caso correspondiente a un intervalo de valores de α .

Caso 1:

Este caso se corresponde con $\alpha < U_b$. Se demostrará que existe un conjunto de m

tareas $\{\tau_1, \dots, \tau_m\}$, de factores de utilización menores o iguales que α , y utilización total

$$nU_b - (n - 1)\alpha + \epsilon$$

con $\epsilon \rightarrow 0^+$, el cual no cabe en n procesadores empleando el algoritmo WF de asignación de tareas a procesadores.

El conjunto de tareas se divide en tres subconjuntos. El primer subconjunto está formado por n_a tareas de factor de utilización

$$u_i = \frac{\epsilon}{m - 1}$$

para $i = 1, \dots, n_a$. Donde

$$n_a = m + n - 1 - \left\lfloor \frac{m + n - 1}{n} \right\rfloor n$$

El segundo subconjunto está formado por $(m - n_a - 1)$ tareas de factor de utilización

$$u_i = \frac{U_b - \alpha}{\left\lfloor \frac{m-1}{n} \right\rfloor} + \frac{\epsilon}{m - 1}$$

para $i = (n_a + 1), \dots, (m - 1)$.

El tercer subconjunto esta formado por una única tarea, τ_m , de factor de utilización α .

Se puede comprobar que la utilización total del conjunto de tareas es

$$nU_b - (n - 1)\alpha + \epsilon$$

Otra asignación WFI

ordenar las tareas en sentido creciente de utilizations

$j := 1$

desde $i := 1$ hasta m

 si $CR(j) \geq u_i$

 asignar τ_i a P_j

 si no

 retornar NO PLANIFICABLE

 fin de si

$j := j + 1$

 si $j > n$

$j := 1$

 fin de si

fin de desde

retornar PLANIFICABLE

fin de otra asignación WFI

Algoritmo 4.2.1: Un algoritmo simple de asignación de tareas WFI.

Hay que comprobar que el conjunto de tareas es válido, esto es, debe verificarse que $0 < u_i \leq \alpha$ para $i = 1, \dots, n$. Además debe cumplirse que los factores de utilización siguen una secuencia creciente.

Los factores de utilización son obviamente mayores que cero. El factor de utilización de las tareas del primer subconjunto es menor que α eligiendo un ϵ suficientemente pequeño. El factor de utilización de la única tarea del tercer subconjunto es menor o igual que α , pues de hecho toma el valor α . Se demostrará a continuación que los factores de utilización de las tareas del segundo subconjunto son menores o iguales que α . Sustituyendo el valor de U_b en la definición de u_i de las tareas del segundo subconjunto se tiene

$$u_i = \frac{\lfloor \frac{m+n-1}{n} \rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) - \alpha}{\lfloor \frac{m-1}{n} \rfloor} + \frac{\epsilon}{m-1} \quad (4.42)$$

Por hipótesis del teorema se tiene $m > \beta_{RM}n$, y por lo tanto $m \geq (\beta_{RM}n + 1)$. Como consecuencia, $\lfloor (m-1)/n \rfloor \geq \beta_{RM}$, y $\lfloor (m+n-1)/n \rfloor \geq \beta_{RM} + 1$. Por otra parte de acuerdo con la ecuación (4.10) de la página 157 se cumple

$$\alpha > (2^{1/(\beta_{RM}+1)} - 1)$$

Llevando todas estas relaciones a la ecuación (4.42) se obtiene

$$u_i < (2^{1/(\beta_{RM}+1)} - 1) + \frac{\epsilon}{m-1}$$

Tal como se indico anteriormente $\alpha > (2^{1/(\beta_{RM}+1)} - 1)$. Entre dos números reales es siempre posible encontrar otro número real, y por lo tanto eligiendo un ϵ suficientemente pequeño se tiene $u_i \leq \alpha$.

Los factores de utilización siguen una secuencia creciente pues eligiendo un ϵ suficientemente pequeño se tiene

$$\frac{\epsilon}{m-1} < \frac{U_b - \alpha}{\lfloor \frac{m-1}{n} \rfloor} + \frac{\epsilon}{m-1} < \alpha$$

A continuación se probará que el conjunto de tareas anterior no cabe en el multiprocesador.

Las $(m-1)$ primeras tareas se asignan de forma secuencial a los procesadores, de acuerdo con el algoritmo 4.2.1 de la página 189. La situación después de la asignación puede verse en la figura 4.8.

Si la tarea τ_m cupiese en algún procesador cabría en uno de los $n_b = (n - n_a)$ últimos procesadores. La utilización total de uno cualquiera de estos n_b procesadores es

$$\left\lfloor \frac{m-1}{n} \right\rfloor \left(\frac{U_b - \alpha}{\lfloor \frac{m-1}{n} \rfloor} + \frac{\epsilon}{m-1} \right)$$

y por lo tanto mayor que $U_b - \alpha$. El límite de utilización asociado a $\lfloor (m-1)/n \rfloor + 1$ tareas es U_b , y por lo tanto la última tarea de factor de utilización α no cabe en

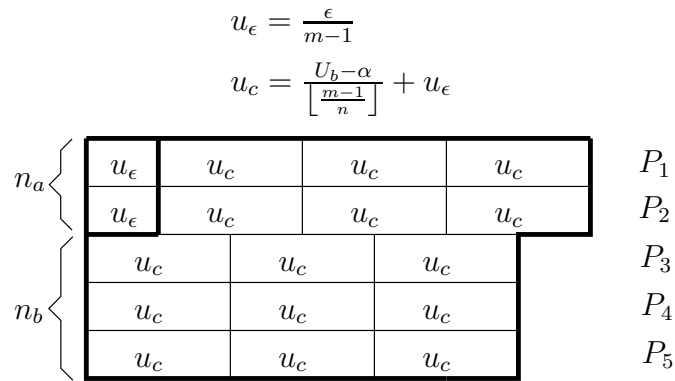


Figura 4.8: Asignación de las $(m - 1)$ primeras tareas en el *caso* 1 del teorema 25, para $m = 18$ y $n = 5$.

ninguno de los procesadores.

Caso 2:

Este caso se corresponde con $\alpha \geq U_b$. Se demostrará que existe un conjunto de m tareas $\{\tau_1, \dots, \tau_m\}$, de factores de utilización menores o iguales que α , y utilización total $U_b + \epsilon$, con $\epsilon \rightarrow 0^+$, el cual no cabe en n procesadores empleando el algoritmo WFI de asignación de tareas a procesadores.

Se considera el mismo conjunto de tareas que el *caso* 3 del teorema 22, en la página 174. Este teorema obtenía una cota superior para la asignación WF. El conjunto de m tareas se construye a partir de dos subconjuntos. Un primer subconjunto formado por $(m - 1)$ tareas de factor de utilización $u_i = \epsilon / (m - 1)$, y una tarea τ_m de factor de utilización $u_m = U_b$. Puede comprobarse que los factores de utilización de las tareas están en sentido creciente. Por lo tanto si no cabían en los procesadores empleando asignación WF, tampoco caben empleando asignación WFI.

NOTA: El rango asociado a cada uno de los casos de la demostración no coincide exactamente con los rangos que aparecen en el enunciado del teorema. No obstante, debe tenerse en cuenta que la cota proporcionada en el enunciado del teorema es una función continua. ■

El teorema 26 cuyo enunciado y demostración se proporciona a continuación presenta una cota inferior del límite de utilización empleando asignación de tareas WFI.

Teorema 26 *Si $m > \beta_{RM}n$ se cumple*

$$U_{wc}^{RM-WFI}(m, n, \alpha) \geq \begin{cases} nU_b - (n - 1)\alpha & \text{si } \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

donde

$$U_b = \left\lfloor \frac{m + n - 1}{n} \right\rfloor \left(2^{\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right)$$

Demostración:

La demostración se dividirá en dos casos, cada caso correspondiente a un intervalo de valores de α .

Caso 1:

Este caso se corresponde con $\alpha < U_b$. Se demostrará que cualquier conjunto de m tareas $\{\tau_1, \dots, \tau_m\}$, de factores de utilización menores o iguales que α , y utilización total menor o igual que

$$nU_b - (n - 1)\alpha$$

cabe en n procesadores empleando el algoritmo WFI de asignación de tareas a procesadores. Para ello se demostrará que la utilización total de cualquier conjunto de m tareas que no cabe en el multiprocesador debe ser mayor que la expresión anterior.

Sea $\{\tau_1, \dots, \tau_m\}$ un conjunto de m tareas que no cabe en el multiprocesador. Sea τ_k la primera tarea del conjunto que no cabe en el multiprocesador. La asignación WFI consiste en ordenar previamente las tareas en sentido de factores de utilización crecientes, y hacer a continuación una la asignación WF. De acuerdo con el algoritmo 4.2.1 y la figura 4.7, las tareas se asignan de manera secuencial a los procesadores. Por lo tanto, si la tarea τ_k no cabe, esto significa

$$U_l > \left\lfloor \frac{k + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{k+n-1}{n} \rfloor} - 1 \right) - u_k \quad (4.43)$$

donde $l = k - \lfloor (k - 1)/n \rfloor$, y donde U_l es la utilización total de las tareas asignadas al procesador P_l justo antes de intentar asignar la tarea τ_k .

La utilización total del resto de procesadores es mayor que U_l . Los $(l-1)$ primeros procesadores tienen una tarea más que P_l . Además para toda tarea asignada a P_l existe otra asignada a cada uno de estos procesadores de factor de utilización mayor. Los últimos $(m-l)$ procesadores tienen el mismo número de tareas que P_l justo antes de intentar asignar τ_k , y además para toda tarea asignada a P_l existe otra asignada a cada uno de estos procesadores de factor de utilización mayor. Como consecuencia, $U_j \geq U_l$ para $j = 1, \dots, n$. A partir de la ecuación (4.43) se obtiene

$$\sum_{j=1}^n U_j > n \left\lfloor \frac{k + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{k+n-1}{n} \rfloor} - 1 \right) - nu_k$$

donde U_j es la utilización total del procesador P_j justo antes de intentar asignar la tarea τ_k .

Se sabe que

$$U = \sum_{i=1}^m u_i \geq \sum_{i=1}^k u_i = \sum_{j=1}^n U_j + u_k$$

y por lo tanto se cumple

$$U > n \left\lfloor \frac{k + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{k+n-1}{n} \rfloor} - 1 \right) - (n - 1)u_k$$

Teniendo en cuenta que $u_k \leq \alpha$, que el segundo término de la ecuación es una función monótona decreciente en k , que k pertenece al rango $[1, m]$ y la definición de U_b resulta

$$U > nU_b - (n - 1)\alpha$$

Por consiguiente, cualquier conjunto de m tareas de utilización total menor o igual que $nU_b - (n - 1)\alpha$ cabe en n procesadores. Es decir,

$$U_{wc}^{RM-WFI}(m, n, \beta_{RM}) \geq nU_b - (n - 1)\alpha$$

Caso 2:

Este caso se corresponde con $\alpha \geq U_b$. El límite de utilización empleando asignación WF toma el valor U_b para $\alpha \geq U_b$, tal como se indica en el corolario 3 de la página 179. Como el límite de utilización para asignación WFI no puede ser menor que para asignación WF resulta

$$U_{wc}^{RM-WFI}(m, n, \beta_{RM}) \geq U_b$$

NOTA: El rango asociado a cada uno de los casos de la demostración no coincide exactamente con los rangos que aparecen en el enunciado del teorema. No obstante, debe tenerse en cuenta que la cota proporcionada en el enunciado del teorema es una función continua. ■

Finalmente, el corolario 5 proporciona el límite de utilización para asignación WFI.

Corolario 5 *Sea RA un algoritmo razonable de asignación de tareas a procesadores arbitrario. Sea $m > \beta_{RM}n$. Se cumple*

$$U_{wc}^{RM-WFI}(m, n, \alpha) = \begin{cases} nU_b - (n - 1)\alpha & \text{si } \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

donde

$$U_b = \left\lfloor \frac{m + n - 1}{n} \right\rfloor \left(2^{\lceil \frac{m+n-1}{n} \rceil} - 1 \right)$$

Demostración:

La demostración es directa a partir de los teoremas 25, y 26. ■

Algoritmo de asignación RFI

El límite de utilización empleando asignación RFI no puede ser menor que el límite de utilización empleando asignación RF. Por lo tanto,

$$U_{wc}^{RM-RFI}(m, n, \alpha) \geq \begin{cases} n_a U_a + n_b U_b - (n - 1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases} \quad (4.44)$$

donde

$$\begin{aligned}
 n_a &= m - \left\lfloor \frac{m-1}{n} \right\rfloor n - 1 \\
 n_b &= n - n_a \\
 U_a &= \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lceil \frac{m+n-1}{n} \rceil} - 1 \right) \\
 U_b &= \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right)
 \end{aligned}$$

El teorema 22 de la página 174 proporciona una cota superior del límite de utilización empleando asignación WF. Para ello se consideran tres casos diferentes, en cada uno de los cuales se construye un conjunto de tareas diferente. Si cada uno de los conjuntos de tareas propuesto se ordenarse en sentido de factores de utilización crecientes se obtendría un nuevo conjunto de tareas. Este nuevo conjunto de tareas podría asignarse a los procesadores empleando el algoritmo RFI de forma idéntica a como lo hace el algoritmo WF en la demostración del teorema 22. El resultado es que la cota superior propuesta por el teorema 22 para asignación WF es válida para asignación RFI. Como dicha cota superior coincide con la cota inferior proporcionada por la ecuación (4.44) resulta

$$U_{wc}^{RM-RFI}(m, n, \alpha) = \begin{cases} n_a U_a + n_b U_b - (n-1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha < U_b \\ U_b & \text{si } U_b < \alpha \end{cases} \quad (4.45)$$

4.3 Análisis de los resultados teóricos

En este apartado se analizan los resultados teóricos obtenidos en el apartado 4.2. Estos resultados son los siguientes:

- El límite de utilización U_{wc}^{RM-RA} , asociado a la planificación RM y a una asignación de tareas razonable cualquiera, RA, se encuentra dentro del rango

$$L_{RM}(m, n, \alpha) \leq U_{wc}^{RM-RA} \leq H_{RM}(n, \beta_{RM})$$

donde

$$L_{RM}(m, n, \alpha) = \begin{cases} n_a U_a + n_b U_b - (n-1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

$$H_{RM}(n > 1, \beta_{RM}) = (\beta_{RM}n + 1)(2^{1/(\beta_{RM}+1)} - 1)$$

$$\beta_{RM} = \left\lfloor \frac{1}{\log_2(\alpha + 1)} \right\rfloor$$

$$n_a = m + n - 1 - \left\lfloor \frac{m + n - 1}{n} \right\rfloor n$$

$$n_b = n - n_a$$

$$U_a = \left\lfloor \frac{m + n - 1}{n} \right\rfloor \left(2^{1/\lceil \frac{m+n-1}{n} \rceil} - 1 \right)$$

$$U_b = \left\lfloor \frac{m + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right)$$

El parámetro α es el máximo factor de utilización alcanzable, m es el número de tareas ($m > \beta_{RM}n$), y n es el número de procesadores que constituyen el multiprocesador.

- El límite de utilización U_{wc}^{RM-RAD} , asociado a la planificación RM y a una asignación de tareas RAD toma el valor

$$U_{wc}^{RM-RAD} = H_{RM}(n, \beta_{RM})$$

donde RAD representa un algoritmo de asignación de tareas obtenido mediante ordenación previa de los factores de utilización en sentido decreciente, y posterior asignación empleando un algoritmo razonable cualquiera RA que no realice internamente ningún tipo de ordenación de tareas. En particular el límite de utilización asociado a los algoritmos FFD, BFD, WFD y RFD toma el valor $H_{RM}(n, \beta_{RM})$.

- El límite de utilización asociado a la planificación RM, y a las asignaciones de

tareas FF, FFI, BF y BFI toma el valor

$$U_{wc}^{RM-FF}(m, n, \beta_{RM}) = U_{wc}^{RM-FFI}(m, n, \beta_{RM}) = U_{wc}^{RM-BF}(m, n, \beta_{RM}) = U_{wc}^{RM-BFI}(m, n, \beta_{RM}) = K_{RM}(m, n, \beta_{RM})$$

donde

$$K_{RM}(m, n, \beta_{RM}) = (n-1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + (m - \beta_{RM}(n-1))(2^{1/(m-\beta_{RM}(n-1))} - 1)$$

- El límite de utilización asociado a la planificación RM y a las asignaciones de tareas WF, RF y RFI toma el valor

$$U_{wc}^{RM-WF}(m, n, \alpha) = U_{wc}^{RM-RF}(m, n, \alpha) = U_{wc}^{RM-RFI}(m, n, \alpha) = L_{RM}(m, n, \alpha)$$

- El límite de utilización asociado a la planificación RM y a la asignación de tareas WFI toma el valor

$$U_{wc}^{RM-WFI}(m, n, \alpha) = I_{RM}(m, n, \alpha)$$

donde

$$I_{RM}(m, n, \alpha) = \begin{cases} nU_b - (n-1)\alpha & \text{si } \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

Se puede ver que todos los resultados teóricos giran alrededor de cuatro expresiones matemáticas

$$L_{RM}(m, n, \alpha) = \begin{cases} n_a U_a + n_b U_b - (n-1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

$$I_{RM}(m, n, \alpha) = \begin{cases} nU_b - (n-1)\alpha & \text{si } \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

$$K_{RM}(m, n, \beta_{RM}) = (n-1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + (m - \beta_{RM}(n-1))(2^{1/(m-\beta_{RM}(n-1))} - 1)$$

$$H_{RM}(n > 1, \beta_{RM}) = (\beta_{RM}n + 1)(2^{1/(\beta_{RM}+1)} - 1)$$

que se analizarán en los apartados 4.3.1 a 4.3.4. Asimismo se puede comprobar que

$$L_{RM}(m, n, \alpha) \leq I_{RM}(m, n, \alpha) \leq K_{RM}(m, n, \beta_{RM}) \leq H_{RM}(n, \beta_{RM})$$

4.3.1 Análisis del límite $L_{RM}(m, n, \alpha)$

En este apartado se analiza la función

$$L_{RM}(m, n, \alpha) = \begin{cases} n_a U_a + n_b U_b - (n-1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

donde

$$\begin{aligned} n_a &= m + n - 1 - \left\lfloor \frac{m + n - 1}{n} \right\rfloor n \\ n_b &= n - n_a \\ U_a &= \left\lfloor \frac{m + n - 1}{n} \right\rfloor \left(2^{1/\lceil \frac{m+n-1}{n} \rceil} - 1 \right) \\ U_b &= \left\lfloor \frac{m + n - 1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) \end{aligned}$$

la cual se corresponde con el límite de utilización empleando planificación RM y asignación de tareas WF, RF y RFI. Además también coincide con el límite de utilización más bajo de entre todos los algoritmos de asignación razonables bajo planificación RM.

Para el caso $n = 1$ se puede comprobar que se obtiene el límite de Liu y Layland (1973).

$$m(2^{1/m} - 1)$$

La figura 4.9 representa la función $L_{RM}(m, n, \alpha)$ para diferentes valores de α . La representación es continua a pesar de ser n una variable discreta, con el objetivo de mejorar la visualización. Además la representación ha sido normalizada dividiendo $L_{RM}(m, n, \alpha)$ por el número de procesadores, mostrando así la utilización total promedio de los procesadores. Para cada valor de α se han dibujado dos curvas, una inferior y otra superior. La curva superior es el límite de utilización para una número de tareas $m = (\beta_{RM}n + 1)$. La curva inferior es el límite de utilización para una número de tareas infinito. Entre cada pareja de estas curvas se encuentra un área sombreada dentro de la cual se encuentran todas las curvas de límite de utilización para distintos valores del número de tareas. Debe tenerse en cuenta que la función $L_{RM}(m, n, \alpha)$ es monótona decreciente en m .

El caso más general se corresponde con $\alpha = 1$ pues este valor de α no supone ninguna restricción práctica sobre los factores de utilización de las tareas². En este caso, el límite de utilización apenas se ve afectado por la adición de nuevos procesadores. En particular, para un número infinito de procesadores el valor $L_{RM}(m, n, \alpha)/n$ tiende asintóticamente a cero. Esta desafortunada situación puede ilustrarse con un ejemplo. Consideremos una asignación de tareas WF, para la cual se tiene un

²Cualquier tarea de factor de utilización $u_i > 1$ no puede planificarse con ningún planificador, ni siquiera disponiendo de un número infinito de procesadores.

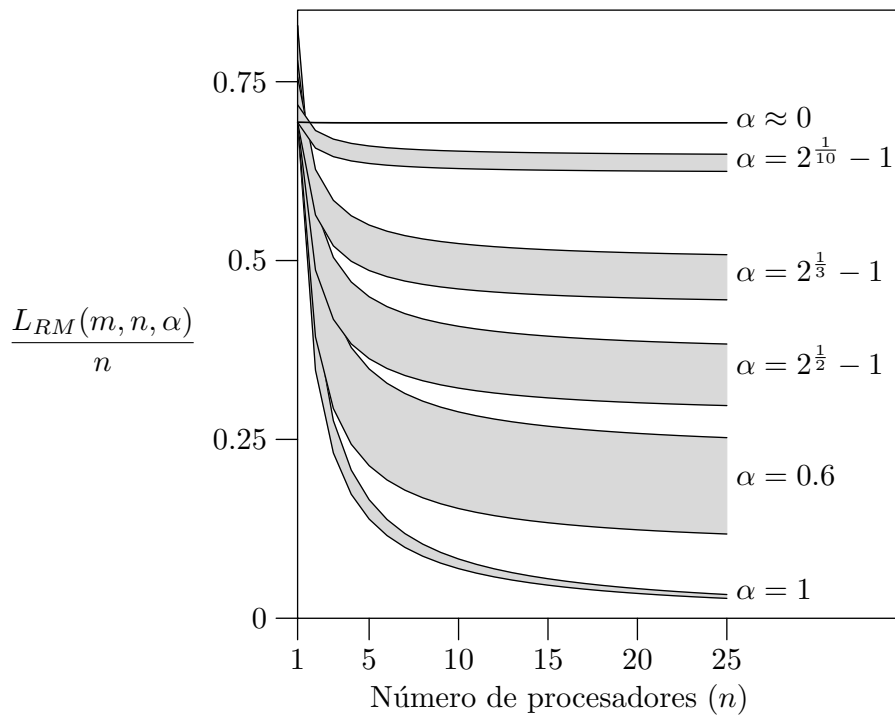


Figura 4.9: Representación del límite $L_{RM}(m, n, \alpha)$.

límite de utilización empleando planificación RM igual a $L_{RM}(m, n, \alpha)$. Considérese un conjunto de 20 tareas, de las cuales las 19 primeras tienen un factor de utilización 0'000001, y la última un factor de utilización $2(2^{1/2} - 1)$. Este conjunto de tareas no cabe en ningún multiprocesador verificando $m > \beta n$, es decir, $n < 20$. La utilización de este conjunto de tareas es aproximadamente 0'83, y sin embargo no cabe en 19 procesadores. El resultado es una utilización total promedio menor que $0'83/19 \approx 0'04$.

Si bien para $\alpha = 1$ se tiene una situación muy desafortunada, para $\alpha \rightarrow 0$ se tiene el mejor caso posible. Cualquier procesador que se añada al sistema incrementa el valor de $L_{RM}(m, n, \alpha)$ en $\ln 2$. La explicación es clara, cuando los factores de utilización son muy pequeños se pueden llenar los procesadores casi hasta su capacidad. En este caso un multiprocesador de n procesadores se comporta como un monoprocesador n veces más rápido.

Para el resto de valores de α a medida que aumenta el número de procesadores la utilización total promedio de los procesadores disminuye de forma asintótica hacia un valor comprendido entre cero y $\ln 2$.

Con respecto al número de tareas, m , se observa una gran influencia de este parámetro sobre el límite de utilización especialmente para valores intermedios de α .

La función $L_{RM}(m, n, \alpha)$ proporciona el límite de utilización empleando asignación WF, RF y RFI. Sin embargo, hay casos prácticos en los cuales se está interesado en el número mínimo de procesadores n_{wc}^{RM} que aseguran la planificabilidad de un conjunto de m tareas de utilización total U y máximo factor de utilización alcanzable α . En el caso de los algoritmos de asignación cuyo límite de utilización asociado viene dado por $L_{RM}(m, n, \alpha)$, el número mínimo de procesadores se representa por

n_{wc}^{RM-L} y se obtiene tal como se explica a continuación.

En la tesis se han presentado dos condiciones de planificabilidad complementarias empleando planificación RM. La primera de ellas proviene de la definición de β_{RM} dada en el apartado 4.1 (véase la figura 4.1)

$$m \leq \beta_{RM} n_{wc}^{RM-L}$$

Despejando n y eligiendo el mínimo valor se obtiene

$$n_{wc}^{RM-L} = \left\lceil \frac{m}{\beta_{RM}} \right\rceil \quad (4.46)$$

La segunda condición de planificabilidad se deduce del empleo del límite de utilización

$$U \leq L_{RM}(m, n_{wc}^{RM-L}, \alpha) \quad (4.47)$$

A partir de esta condición se puede obtener el mínimo valor de n_{wc}^{RM-L} que la cumple. El valor de n_{wc}^{RM-L} se obtiene como el mínimo entre el valor mínimo de n_{wc}^{RM-L} que cumple la condición (4.47), y el valor de n_{wc}^{RM-L} proporcionado por la ecuación (4.46)

Por ejemplo, veamos cuantos procesadores son necesarios para garantizar la planificabilidad de 100 tareas de utilización total 15, y factores de utilización menores o iguales que 0'25 (es decir $\beta_{RM} = 3$) empleando asignación de tareas WF, RF o RFI, y planificación RM. A partir de la condición de planificabilidad dada por la ecuación (4.46) se necesitan $\lceil 100/3 \rceil = 34$ procesadores. Aplicando la condición de planificabilidad dada por la ecuación (4.47) con $n = 33$ procesadores el límite de utilización es 14'9. Si se emplean $n = 34$ procesadores se obtiene un límite de utilización de valor 15,34. Por lo tanto son necesarios 34 procesadores para asegurar la planificabilidad del conjunto de tareas.

4.3.2 Análisis del límite $I_{RM}(m, n, \alpha)$

En este apartado se analiza la función

$$I_{RM}(m, n, \alpha) = \begin{cases} nU_b - (n-1)\alpha & \text{si } \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

donde

$$U_b = \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right)$$

la cual se corresponde con el límite de utilización empleando planificación RM y asignación de tareas WFI.

Para el caso $n = 1$ se puede comprobar que se obtiene el límite de Liu y Layland (1973).

$$m(2^{1/m} - 1)$$

A simple vista la representación gráfica de la función $I_{RM}(m, n, \alpha)$, dada en la figura 4.10, es idéntica a la representación gráfica de la función $L_{RM}(m, n, \alpha)$,

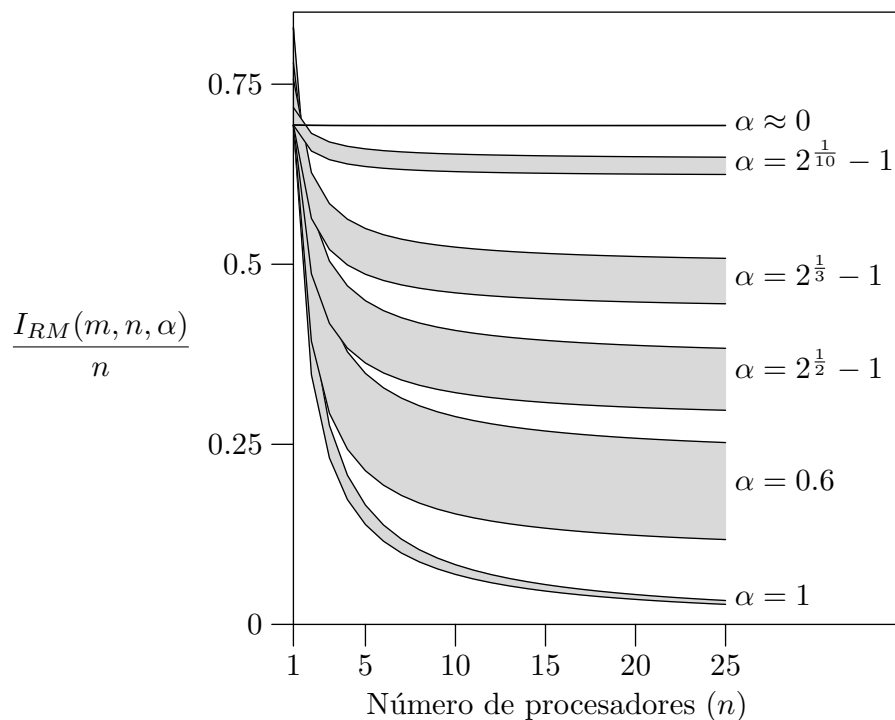


Figura 4.10: Representación del límite $I_{RM}(m, n, \alpha)$.

dada en la figura 4.9, y en realidad lo son. Ambas representaciones son idénticas cuando $n_a = 0$, es decir, cuando $(m - 1)$ es múltiplo de n , o para un número de tareas infinito, pues en ese caso $U_a = U_b$. Para valores de m no muy grandes que no sean múltiplos de $(\beta_{RM} + 1)$, las curvas de límite de utilización estarán en la zona sombreada pero serán diferentes. Dentro de la zona sombreada se cumple $I_{RM}(m, n, \alpha) > L_{RM}(m, n, \alpha)$. No obstante, la diferencia entre ambas límites de utilización es pequeña.

Todos los comentarios acerca de la evolución del límite de utilización en función de α , n y m realizados en el apartado 4.3.1 para la función $L_{RM}(m, n, \alpha)$ son aplicables a la función $I_{RM}(m, n, \alpha)$.

Si se está interesado en el mínimo número de procesadores necesarios para asegurar la planificabilidad de un conjunto de tareas, este número se puede obtener de forma análoga a como se explico en el apartado 4.3.1

Por ejemplo, para garantizar la planificabilidad de 100 tareas de utilización total 15, y factores de utilización menores o iguales que 0'25 (es decir $\beta_{RM} = 3$) empleando asignación de tareas WFI, y planificación RM se necesitan $n = 34$ procesadores, el mismo número que empleando el límite de utilización $L_{RM}(m, n, \alpha)$.

4.3.3 Análisis del límite $K_{RM}(m, n, \alpha)$

En este apartado se analiza la función

$$K_{RM}(m, n, \beta_{RM}) = (n - 1)(2^{1/(\beta_{RM} + 1)} - 1)\beta_{RM} + (m - \beta_{RM}(n - 1))(2^{1/(m - \beta_{RM}(n - 1))} - 1)$$

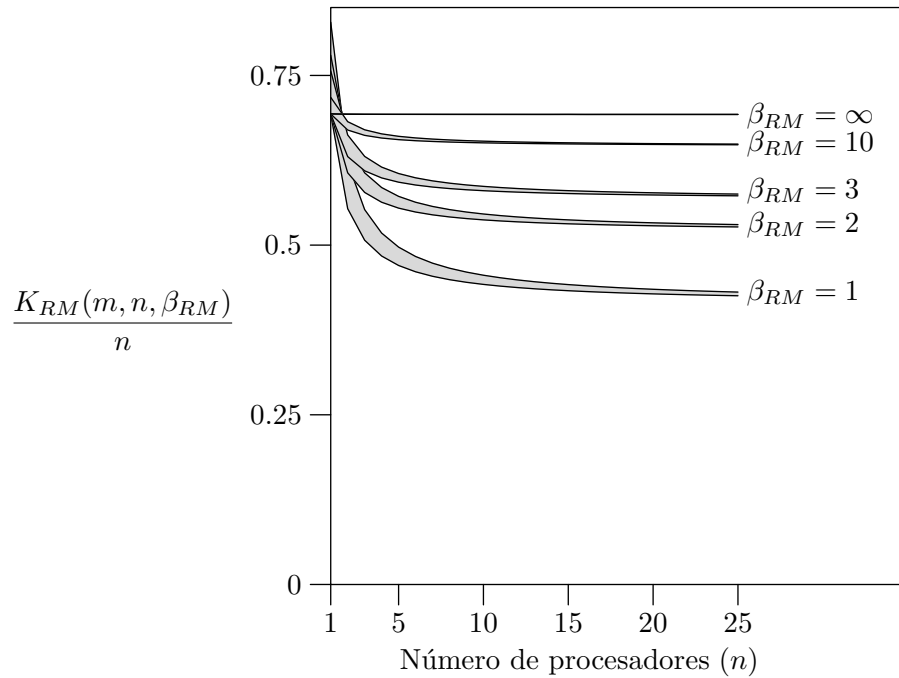


Figura 4.11: Representación del límite $K_{RM}(m, n, \beta_{RM})$.

la cual se corresponde con el límite de utilización empleando planificación RM y asignación de tareas FF, FFI, BF o BFI.

Para el caso $n = 1$ se puede comprobar que se obtiene el límite de Liu y Layland (1973).

$$m(2^{1/m} - 1)$$

La figura 4.11 representa la función $K_{RM}(m, n, \beta_{RM})$ para diferentes valores de β_{RM} , y por lo tanto de α . La representación es continua a pesar de ser n una variable discreta, con el objetivo de mejorar la visualización. Además la representación ha sido normalizada dividiendo $K_{RM}(m, n, \beta_{RM})$ por el número de procesadores, mostrando así la utilización total promedio de los procesadores. Para cada valor de β_{RM} se han dibujado dos curvas, una inferior y otra superior. La curva superior es el límite de utilización para un número de tareas $m = (\beta_{RM}n + 1)$. La curva inferior es el límite de utilización para un número de tareas infinito. Entre cada pareja de estas curvas se encuentra un área sombreada dentro de la cual se encuentran todas las curvas de límite de utilización para distintos valores del número de tareas. Debe tenerse en cuenta que la función $K_{RM}(m, n, \beta_{RM})$ es monótona decreciente en m . Puede observarse que el número de tareas tiene una influencia menor sobre $K_{RM}(m, n, \beta_{RM})$ que la que tenía sobre las funciones $L_{RM}(m, n, \alpha)$, y $I_{RM}(m, n, \alpha)$, descritas en los apartados 4.3.1 y 4.3.2.

A medida que crece el número de procesadores la utilización media de los procesadores tiende asintóticamente a un valor no nulo.

A simple vista puede observarse que el límite de utilización para la asignación FF, FFI, BF o BFI, dado por $K_{RM}(m, n, \alpha)$ es mucho mayor que el límite de utilización asociado a la asignación WF, WFI, RF o RFI, especialmente para valores elevados

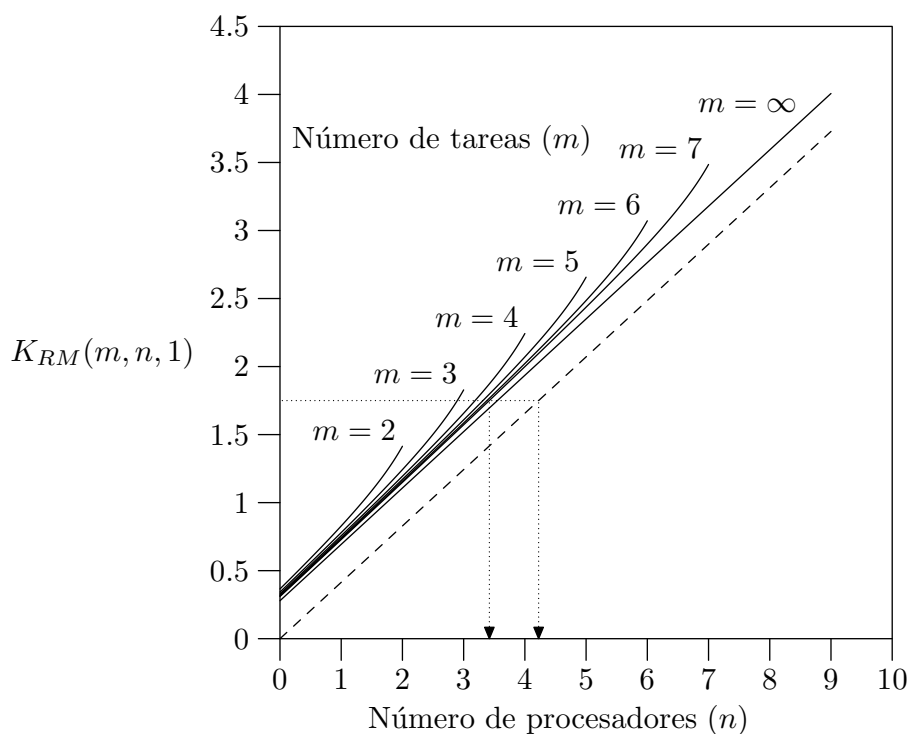


Figura 4.12: Representación del límite $K_{RM}(m, n, 1)$.

de α (valores bajos de β_{RM}).

La figura 4.12 muestra en detalle la influencia del número de tareas sobre el valor de la función $K_{RM}(m, n, \beta_{RM})$ para el caso general $\beta_{RM} = 1$, es decir, $\alpha = 1$. Además representa con una línea de puntos el límite de utilización (no ajustado) proporcionado por Oh y Baker (1998) para la asignación FF. La figura 4.13 es análoga a la figura 4.12, pero muestra la situación en la cual los factores de utilización de todas las tareas son menores que $(2^{1/3} - 1) \approx 0.26$, lo cual implica $\beta_{RM} = 3$. Se observa una mejora sustancial con respecto al caso $\beta_{RM} = 1$. Además se observa que en este caso el número de tareas tiene una influencia menor sobre el valor de la función $K_{RM}(m, n, \beta_{RM})$. Esta figura también representa con una línea de puntos el límite de utilización (no ajustado) proporcionado por Oh y Baker (1998) para asignación FF.

En general, cualquier incremento de α produce un incremento de β_{RM} , y un incremento de la función $K_{RM}(m, n, \beta_{RM})$.

$$\frac{\partial K_{RM}(m, n, \beta_{RM})}{\partial \beta_{RM}} > 0$$

En el límite, si $\alpha \rightarrow 0$ entonces $\beta_{RM} \rightarrow \infty$, y se cumple

$$\lim_{\alpha \rightarrow 0} K_{RM}(m, n, \beta_{RM}) = n \ln 2$$

Por lo tanto, si el factor de utilización de todas las tareas es bajo, el multiprocesador se aproxima a un monoprocesador ideal n veces más rápido que cada uno de sus

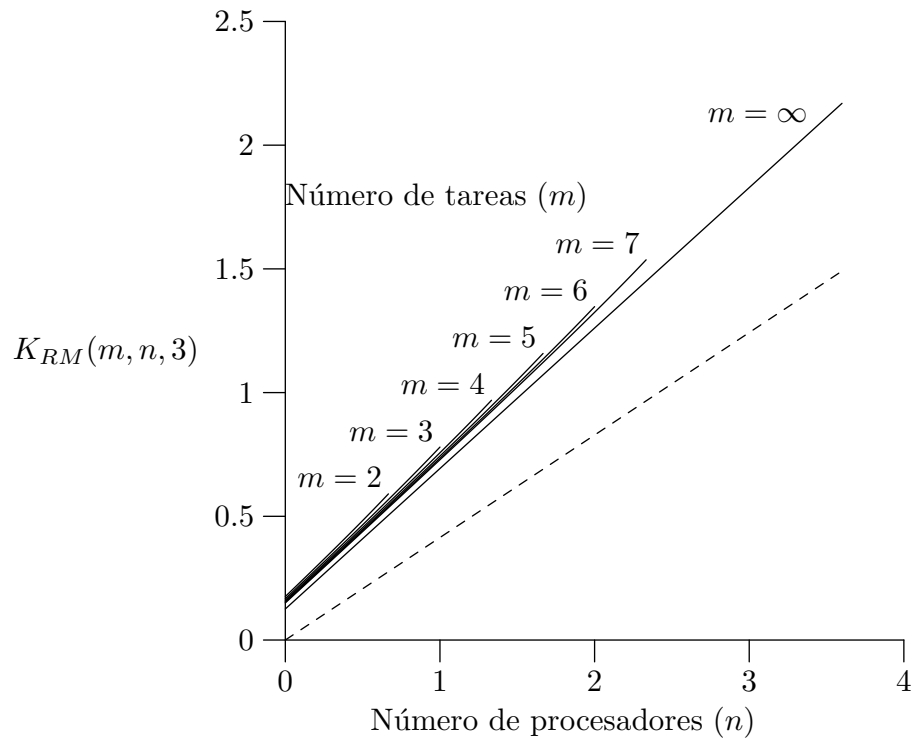


Figura 4.13: Representación del límite $K_{RM}(m, n, 3)$.

procesadores.

Se puede obtener un límite de utilización independiente del número de tareas haciendo $m = \infty$ en la expresión de $K_{RM}(m, n, \beta_{RM})$.

$$K_{RM}(m, n, \beta_{RM}) > K_{RM}(\infty, n, \beta_{RM}) = \ln 2 + (n - 1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}$$

La obtención del número mínimo de procesadores requeridos para asegurar la planificabilidad de un conjunto de tareas dado se puede obtener de forma análoga a como se hizo en el apartado 4.3.1. Por ejemplo, veamos cuantos procesadores son necesarios para garantizar la planificabilidad de 100 tareas de utilización total 15, y factores de utilización menores o iguales que 0'25 (es decir $\beta_{RM} = 3$) empleando asignación de tareas FF, FFI, BF o BFI y planificación RM. A partir de la condición de planificabilidad dada por la ecuación (4.46) se necesitan $\lceil 100/3 \rceil = 34$ procesadores. Aplicando la expresión del límite de utilización, con $n = 26$ procesadores el límite de utilización es 14'9. Si se emplean $n = 27$ procesadores se obtiene un límite de utilización de valor 15,5. Por lo tanto son necesarios 27 procesadores para asegurar la planificabilidad del conjunto de tareas.

Otra posibilidad es emplear gráficas como las mostradas en las figuras 4.12 y 4.13. Por ejemplo, considérese un conjunto de $m = 7$ tareas, de utilización total $U = 1'75$, y factores de utilización arbitrarios. A partir de la figura 4.12, aplicando el límite de utilización propuesto por Oh y Baker (1998), se necesitan cinco procesadores. Aplicando la expresión ajustada dada por la función $K_{RM}(m, n, \beta_{RM})$ se necesitan sólo cuatro procesadores. Si se empleasen tres procesadores sería posible encontrar

un conjunto de tareas no planificable con asignación FF por ejemplo. Uno de estos conjuntos de tareas es el formado por siete tareas de factores de utilización $\{u_1 = 0'01, u_2 = 0'01, u_3 = 0'01, u_4 = 0'43, u_5 = 0'43, u_6 = 0'43, u_7 = 0'43\}$, el cual no cabe en tres procesadores.

Las figuras 4.12 y 4.13 son válidas únicamente para $m > \beta_{RM}n$. Por lo tanto, si no hay punto en estas figuras para alguna pareja de valores (m, U) , el número de procesadores se obtiene como

$$n_{wc}^{RM-K} = \left\lceil \frac{m}{\beta_{RM}} \right\rceil$$

Por ejemplo, para $m = 3$, $U = 2'5$, y $\beta_{RM} = 1$, no hay punto en la figura 4.12, y así se necesitan $\lceil 3/1 \rceil = 3$ procesadores.

4.3.4 Análisis del límite $H_{RM}(n > 1, \beta_{RM})$

En este apartado se analiza la función

$$H_{RM}(n > 1, \beta_{RM}) = (\beta_{RM}n + 1)(2^{1/(\beta_{RM}+1)} - 1)$$

la cual se corresponde con el límite de utilización empleando planificación RM y asignación de tareas FFD, BFD, WFD, RFD, y en general cualquier algoritmo de asignación de tareas razonable con ordenación previa de las tareas en sentido de factores de utilización decrecientes. Además se corresponde con el máximo límite de utilización que puede tener cualquier algoritmo de asignación razonable.

Para el caso $n = 1$ el límite de utilización no viene dado por la expresión anterior. Sin embargo, este coincide con el proporcionado por Liu y Layland (1973) para planificación RM.

$$m(2^{1/m} - 1)$$

La figura 4.14 representa la función $H_{RM}(n > 1, \beta_{RM})$ para diferentes valores de β_{RM} , y por lo tanto de α . La representación es continua a pesar de ser n una variable discreta, con el objetivo de mejorar la visualización. Además la representación ha sido normalizada dividiendo $H_{RM}(n > 1, \beta_{RM})$ por el número de procesadores, mostrando así la utilización total promedio de los procesadores. Puede observarse que el límite de utilización proporcionado por la función $H_{RM}(n > 1, \beta_{RM})$ no depende del número de tareas. En particular, coincide con el límite de utilización proporcionado por la función $K_{RM}(m, n, \beta_{RM})$ para el caso $m = (\beta_{RM}n + 1)$. Por lo tanto, la evolución es similar a la de la función $K_{RM}(m, n, \beta_{RM})$, descrita en el apartado 4.3.3.

En el caso de la función $H_{RM}(n, \beta_{RM})$, el cálculo del número mínimo de procesadores que aseguran la planificabilidad de un conjunto de tareas puede expresarse fácilmente de forma analítica.

En la tesis se han presentado dos condiciones de planificabilidad complementarias empleando planificación RM. La primera de ellas proviene de la definición de β_{RM}

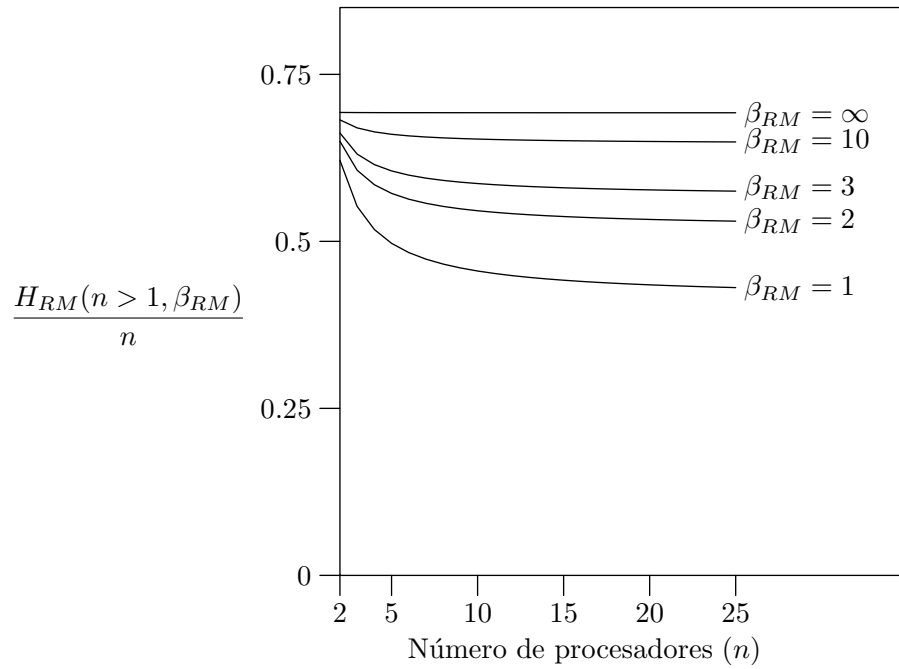


Figura 4.14: Representación del límite $H_{RM}(n > 1, \beta_{RM})$.

dada en el apartado 4.1 (véase la figura 4.1)

$$m \leq \beta_{RM} n_{wc}^{RM-H}$$

Despejando n_{wc}^{RM-H} y eligiendo el mínimo valor se obtiene

$$n_{wc}^{RM-H} = \left\lceil \frac{m}{\beta_{RM}} \right\rceil \quad (4.48)$$

La segunda condición de planificabilidad se deduce del empleo del límite de utilización

$$U \leq H_{RM}(n_{wc}^{RM-H} > 1, \beta_{RM}) = (\beta_{RM} n_{wc}^{RM-H} + 1)(2^{1/(\beta_{RM}+1)} - 1)$$

Despejando n_{wc}^{RM-L} y eligiendo el mínimo valor se obtiene

$$n_{wc}^{RM-H} = \left\lceil \frac{\frac{U}{2^{1/(\beta_{RM}+1)} - 1} - 1}{\beta_{RM}} \right\rceil \quad (4.49)$$

La ecuación (4.49) no es válida para $U \leq (2^{1/(\beta_{RM}+1)})$. No obstante, como se cumple $(2^{1/(\beta_{RM}+1)}) < m(2^{1/m} - 1)$ resulta que en este caso un único procesador es suficiente para planificar de forma factible el conjunto de tareas. Teniendo en cuenta esta

observación, la ecuación (4.48) y la ecuación (4.49) se obtiene

$$n_{wc}^{RM-H}(m, U, \beta_{RM}) = \begin{cases} 1 & \text{si } U \leq m(2^{1/m} - 1) \\ \min \left\{ \left\lceil \frac{m}{\beta_{RM}} \right\rceil, \left\lceil \frac{U}{\frac{2^{1/(\beta_{RM}+1)} - 1}{\beta_{RM}}} - 1 \right\rceil \right\} & \text{en otro caso} \end{cases} \quad (4.50)$$

Por ejemplo, para garantizar la planificabilidad de 100 tareas de utilización total 15, y factores de utilización menores o iguales que 0'25 (es decir $\beta_{RM} = 3$) empleando asignación de tareas FFD, BFD, WFD o RFD y planificación RM se necesitan 27 procesadores

$$n_{wc}^{RM-H}(15, 0'25) = \min \left\{ \left\lceil \frac{100}{3} \right\rceil, \left\lceil \frac{15}{\frac{2^{1/(3+1)} - 1}{3}} - 1 \right\rceil \right\} = 27$$

4.4 Obtención de los límites de utilización estadísticos

El límite de utilización (absoluto) se obtiene para conjuntos de tareas que pueden ser infrecuentes en la práctica. Por ejemplo, considerando sólo el caso peor, los algoritmos de asignación FF y OPT (el algoritmo óptimo) serían equivalentes bajo planificación RM, pues ambos llevan asociado el mismo límite de utilización. Sin embargo, está claro que el algoritmo OPT es superior al FF en términos de planificabilidad. Esto hace ver que el caso peor es más pesimista para asignación OPT que para asignación FF.

Con la intención de percibir el pesimismo asociado al límite de utilización absoluto definiremos el concepto de límite de utilización estadístico, denotado por $U = U_{ac}^{RM-AA}(m, n, p)$, de forma análoga a como se hizo en el apartado 4.4 para planificación EDF.

Definición 7 *Un conjunto de m tareas, de utilización total $U = U_{ac}^{RM-AA}(m, n, p)$ es planificable sobre n procesadores con probabilidad p empleando planificación RM-AA.*

Si se generasen aleatoriamente conjuntos de m tareas (sin ninguna restricción sobre su tamaño), de utilización total $U = U_{ac}^{RM-AA}(m, n, p)$, el $p\%$ de los conjuntos de tareas serían planificables sobre n procesadores empleando planificación RM-AA. El $(100\% - p)$ restante se correspondería con conjuntos de tareas que no caben en los n procesadores.

Para el caso $p = 100\%$, si el límite de utilización absoluto existe se tiene que el límite de utilización estadístico y absoluto coinciden. Como el límite de utilización absoluto existe para los algoritmos de asignación razonables resulta

$$U_{wc}^{RM-RA}(n, \alpha = 1) = U_{ac}^{RM-RA}(m, n, p = 100\%)$$

donde RA es un algoritmo de asignación de tareas razonable cualquiera.

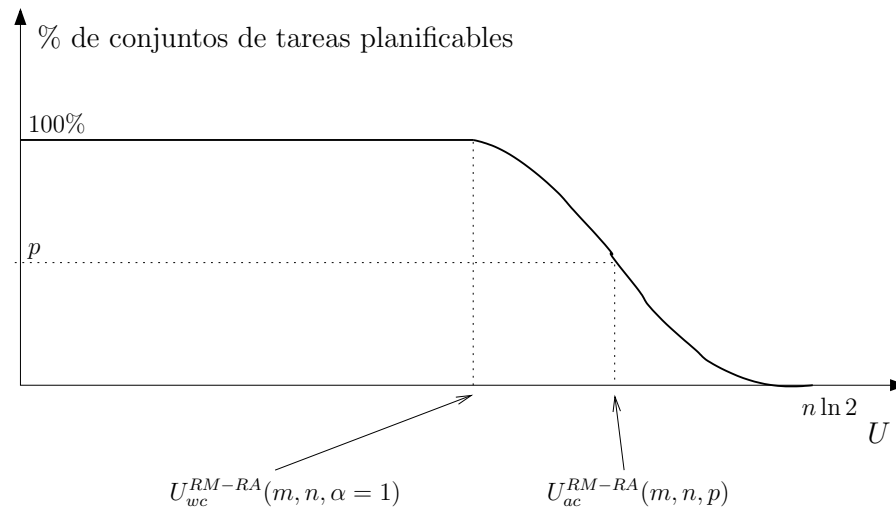


Figura 4.15: Relación entre los conceptos de límite de utilización (absoluto) y estadístico para unos valores de m y n fijos, y para una distribución estadística dada.

La distribución estadística que se ha elegido para generar los conjuntos de tareas es la *Distribución Beta*. Esta función de distribución ha sido descrita en el apartado 3.4.

La figura 4.15, análoga a la figura 3.7, muestra de forma intuitiva los conceptos de límite de utilización (absoluto) y estadístico para unos valores de m y n fijos. En realidad para cada pareja de valores m y n se obtendría una curva diferente, suponiendo que los parámetros de la distribución estadística son fijos.

Cuando la desviación típica de la distribución tiende a cero los factores de utilización de todas las tareas toman el mismo valor. En este caso el límite de utilización proporciona una condición de planificabilidad suficiente independiente del algoritmo razonable de asignación de tareas, tal como demuestra el teorema 27.

Teorema 27 *Sea RA un algoritmo razonable de asignación de tareas a procesadores arbitrario. Sea $\{\tau_1, \dots, \tau_m\}$ un conjunto de m tareas de idénticos factores de utilización y utilización total U .*

El conjunto de tareas es planificable empleando planificación RM-RA sobre n procesadores si

$$U \leq m \left(2^{\left\lceil \frac{1}{\frac{m+n-1}{n}} \right\rceil} - 1 \right) \quad (4.51)$$

Además el límite de utilización propuesto es ajustado.

Demostración:

En primer lugar se demostrará que cualquier conjunto de m tareas de factores de utilización idénticos y utilización total menor o igual que

$$m \left(2^{\left\lceil \frac{1}{\frac{m+n-1}{n}} \right\rceil} - 1 \right)$$

es planificable empleando planificación RM en cada procesador, y cualquier algoritmo de asignación de tareas razonable, RA. El factor de utilización de cada una de las m tareas será a lo sumo

$$\left(2^{\lceil \frac{1}{\lfloor \frac{m+n-1}{n} \rfloor} \rceil} - 1\right)$$

y por lo tanto en cada procesador caben al menos $\lfloor \frac{m+n-1}{n} \rfloor$ tareas. En n procesadores caben al menos $\lfloor \frac{m+n-1}{n} \rfloor n$ tareas. A partir de la relación (vi) se tiene

$$\left\lfloor \frac{m+n-1}{n} \right\rfloor \geq \frac{m}{n}$$

y por lo tanto

$$\left\lfloor \frac{m+n-1}{n} \right\rfloor n \geq m$$

Se tiene entonces que al menos m tareas caben en el multiprocesador, por lo que el conjunto de m tareas es planificable.

A continuación se demuestra que existe un conjunto de m tareas de utilización total

$$m \left(2^{\lceil \frac{1}{\lfloor \frac{m+n-1}{n} \rfloor} \rceil} - 1\right) + \epsilon$$

con $\epsilon \rightarrow 0$, el cual no cabe en los n procesadores. Si el conjunto de tareas fuese planificable, el procesador que más tareas recibiese recibiría como mínimo $\lceil m/n \rceil$ tareas. Aplicando la relación (vii) se deduce

$$\lceil \frac{m}{n} \rceil = \left\lfloor \frac{m-1}{n} \right\rfloor + 1$$

Aplicando la relación (iv)

$$\lceil \frac{m}{n} \rceil = \left\lfloor \frac{m+n-1}{n} \right\rfloor$$

La utilización asociada a este procesador sería

$$\left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{\lceil \frac{1}{\lfloor \frac{m+n-1}{n} \rfloor} \rceil} - 1\right) + \left\lfloor \frac{m+n-1}{n} \right\rfloor \epsilon/m$$

mayor que

$$\left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{\lceil \frac{1}{\lfloor \frac{m+n-1}{n} \rfloor} \rceil} - 1\right)$$

que es el límite de utilización para $\lfloor \frac{m+n-1}{n} \rfloor$ tareas. Por lo tanto se llega a una contradicción, y el conjunto de tareas no cabe en el multiprocesador. ■

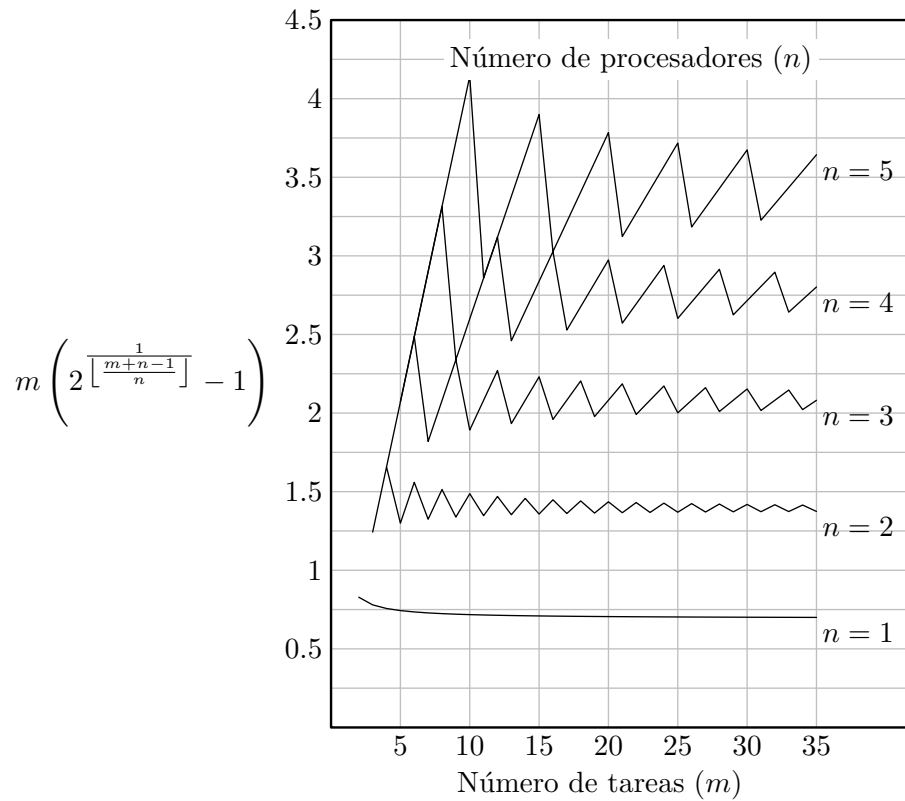


Figura 4.16: Representación del límite de utilización para el caso de factores de utilización iguales y planificación RM-RA.

La figura 4.16 muestra la representación gráfica de la función

$$m \left(2 \left[\frac{1}{m+n-1} \right] - 1 \right)$$

Cabe destacar las siguientes características de la ecuación (4.51):

- Aparecen unos ciclos de amplitud proporcional al número de procesadores al variar el número de tareas.
- La amplitud de los ciclos decrece a medida que aumenta el número de tareas.
- Para el caso $n = 1$ el límite de utilización coincide con el proporcionado por Liu y Layland (1973) para planificación RM.

Con la intención de establecer el pesimismo del límite de utilización absoluto empleando planificación RM-FF se han llevado a cabo experimentos de simulación análogos a los descritos en el apartado 3.4 para planificación EDF.

Los resultados obtenidos son similares a los obtenidos para el caso de planificación EDF. De hecho, independientemente de si se emplea planificación EDF o RM, la comparativa entre unos algoritmos de asignación y otros presenta los mismos resultados. Cabe destacar no obstante tres diferencias básicas:

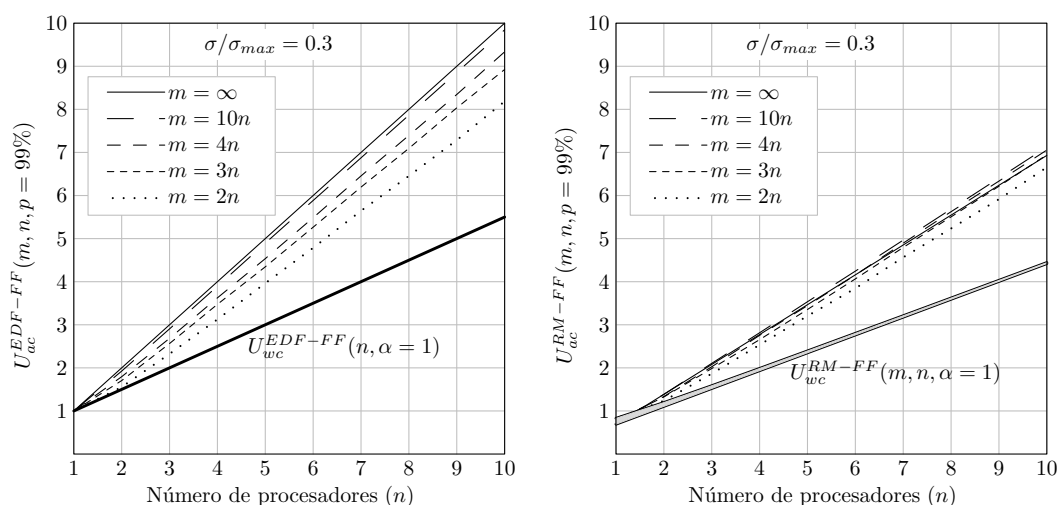


Figura 4.17: Comparación de los límites de utilización estadísticos para planificación EDF-FF y RM-FF.

- Los límites de utilización estadísticos para planificación local RM son entre un 20% y un 30% por ciento menores que los mismos para planificación EDF. Esto es debido al menor límite de utilización monoprocesador asociado a la planificación RM. En la figura 4.17 puede verse una comparación del mismo límite de utilización estadístico empleando planificación EDF o RM.
- La dependencia que tienen los límites de utilización estadísticos para planificación local RM es menos monótona que la dependencia de los mismos límites de utilización estadísticos para planificación local EDF. Un ejemplo de esto puede verse en la figura 4.17. Para planificación EDF un aumento del número de tareas repercutía en un incremento de los límites de utilización estadísticos. La explicación de esta falta de monotonía está en que al aumentar el número de tareas los límites de utilización monoprocesador empleando planificación RM tienden a bajar, lo que repercute sobre los límites de utilización multiprocesador. Sin embargo, por otra parte al aumentar el número de tareas la probabilidad de que los factores de utilización sean mayores es más elevada, lo cual repercute en un incremento de los límites de utilización multiprocesador. Por lo tanto, empleando planificación RM hay dos efectos contrapuestos provocados por el incremento de tareas. De ahí esa falta de monotonía.
- Los límites de utilización estadísticos asociados a los algoritmos WF, RF, WFI y RFI tienden a un valor fijo entre $\ln 2$ y 1 cuando la desviación típica tiende a su valor máximo. La explicación es clara, con desviaciones típicas elevadas los factores de utilización son todos o casi cero o casi uno. Por lo tanto se está cerca del peor caso descrito por los teoremas 22 y 25. La figura 4.18 muestra un ejemplo de la evolución del límite de utilización estadístico para planificación RM-WF cuando se incrementa la desviación típica de la distribución.

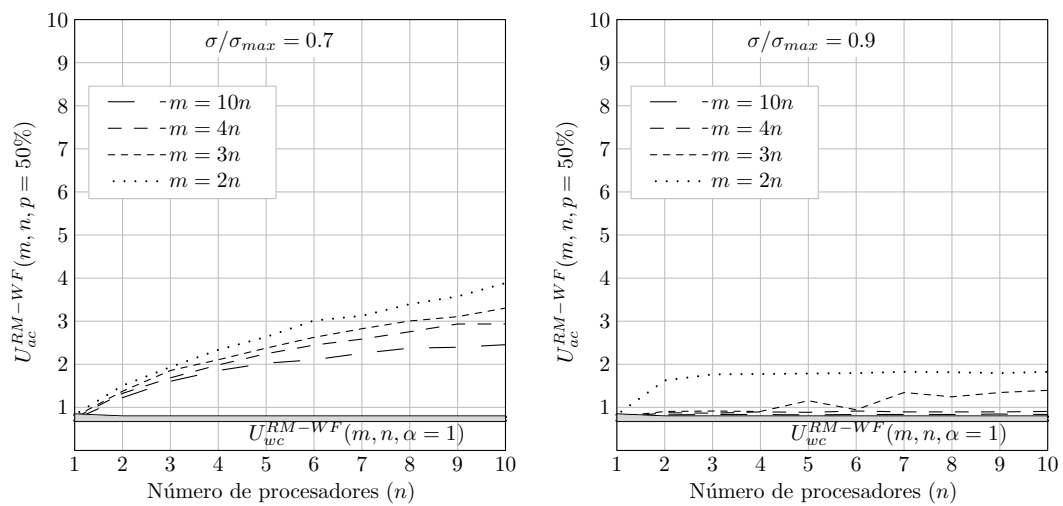


Figura 4.18: Evolución del límite de utilización estadístico para planificación RM-WF cuando se incrementa la desviación típica de la distribución.

Capítulo 5

Mejora de los límites de utilización en multiprocesadores

Los límites de utilización multiprocesador obtenidos en los capítulos anteriores pueden mejorarse empleando toda la información disponible acerca de los factores de utilización de las tareas.

Una de las mejoras proviene de la definición del parámetro α . Este parámetro había sido definido como un valor que verifica $\alpha \geq u_i$ para toda tarea τ_i del sistema. Esta definición no obliga a que exista al menos una tarea con factor de utilización α , pues todas las tareas pueden tener un factor de utilización menor. No obstante, desde un punto de vista práctico α debe elegirse tan bajo como sea posible, pues de esta forma se maximiza el límite de utilización. Este valor más bajo posible se corresponde con el máximo factor de utilización del conjunto de tareas. En las demostraciones de los límites de utilización se toma la definición “al pie de la letra” en el sentido de que no se impone que al menos una de las tareas tenga un factor de utilización igual a α . Por ejemplo, en la demostración de la cota superior del límite de utilización para RM-FF, en la página 159, ninguna de las tareas del conjunto alcanza el valor de α . Debe tenerse en cuenta que si se redefiniere α de tal forma que fuese igual al máximo factor de utilización del conjunto, el límite de utilización multiprocesador no podría disminuir, pues se restringirían los conjuntos de tareas posibles con respecto a la definición original de α . En lugar de redefinir α se define el parámetro u_{max} como el máximo factor de utilización de las tareas del conjunto.

$$u_{max} = \max_{i=1}^m \{u_i\} \quad (5.1)$$

En el apartado 5.1 se proporcionan límites de utilización mejorados empleando el parámetro u_{max} . Una pregunta que puede surgir es por que no se obtuvieron desde un principio los límites de utilización empleando el parámetro u_{max} en lugar del parámetro α , pues en ese caso se habrían obtenido límites más altos, y así toda esta explicación y el siguiente apartado podrían eliminarse. La razón está en la mayor complejidad de las demostraciones empleando el parámetro u_{max} . Además en muchos casos los límites de utilización empleando el parámetro u_{max} pueden obtenerse de forma exacta o aproximarse fácilmente a partir de los límites de utilización empleando el parámetro α .

Otra de las mejoras de los límites de utilización proviene de utilizar los valores de los factores de utilización de las tareas asignadas a un procesador para obtener un límite de utilización monoprocesador más alto, y por lo tanto obtener un límite de utilización multiprocesador también más alto. En el caso de planificación RM se empleaba la condición de planificabilidad

$$\sum_{i=1}^n u_i \leq m(2^{1/m} - 1)$$

Esta condición recibe como entradas la utilización total del conjunto de tareas y el número de tareas, pero no los factores de utilización de las tareas. En particular este límite se obtiene para la peor combinación posible de factores de utilización que es aquella en la cual todos los factores de utilización de las tareas son iguales (Liu y Layland, 1973). Cuando los factores de utilización son diferentes, este límite se puede incrementar. Por ejemplo, en el teorema 16, en la página 159, para el caso $m = 4$, $n = 2$ y $\beta_{RM} = 1$ se afirmaba que las tareas τ_1 , τ_2 y τ_3 de factores de utilización

$$u_1 = \frac{3(2^{1/3} - 1) - (2^{1/2} - 1)}{2} \approx 0'183$$

$$u_2 = u_1$$

$$u_3 = (2^{1/2} - 1) + \epsilon \approx 0'414 + \epsilon, \quad \text{con } \epsilon \rightarrow 0$$

no cabían en un procesador. Sin embargo, a partir de la condición de planificabilidad proporcionada por el teorema 11, en la página 143, a un procesador con dos tareas de factores de utilización

$$\frac{3(2^{1/3} - 1) - (2^{1/2} - 1)}{2}$$

se puede añadir una tarea más, de factor de utilización

$$\frac{2}{\left(1 + \frac{3(2^{1/3}-1)-(2^{1/2}-1)}{2}\right)^2} - 1 \approx 0'429 > u_3$$

y el sistema sigue siendo planificable. Esto indica que en el ejemplo el límite de utilización multiprocesador podría mejorarse empleando la condición de planificabilidad para RM que tiene en cuenta los factores de utilización de las tareas.

La aplicación de esta condición para las obtención de los límites de utilización multiprocesador con planificación local RM resultaría extraordinariamente compleja. Además, tal como se demostrará en el apartado 5.2 la mejora es mínima, y por lo tanto los límites de utilización multiprocesador proporcionados en el capítulo 4 son prácticamente ajustados.

El conocimiento de los factores de utilización de las tareas en el caso de planificación EDF monoprocesador no reporta ninguna mejora. La explicación es que sin considerarlos el límite de utilización es el máximo, y por lo tanto imposible de mejorar.

5.1 Consideración del máximo factor de utilización

El límite de utilización se puede obtener como la mínima utilización total de los conjuntos de tareas que no son planificables, menos un valor ϵ arbitrariamente pequeño. Cuando se restringen los conjuntos sobre los cuales se calcula dicho mínimo, el límite de utilización no puede decrecer. Esto es lo que sucede cuando se considera el parámetro u_{max} en la obtención de los límites de utilización, en lugar del parámetro α . De todos los conjuntos cuyos factores de utilización cumplen $u_i \leq \alpha$, sólo se consideran aquellos que además tienen una o más tareas de factor de utilización u_{max} .

En este apartado se proporcionan los límites de utilización multiprocesador considerando u_{max} , o en su defecto, se proporcionan rangos estrechos dentro de los cuales se encuentran.

Para distinguir los límites de utilización que emplean α de los que emplean u_{max} se sustituye α o el parámetro β asociado por u_{max} . Por ejemplo, el límite $U_{wc}^{RM-FF}(m, n, \beta_{RM})$ pasa a ser $U_{wc}^{RM-FF}(m, n, u_{max})$, y el límite $U_{wc}^{EDF-RF}(n, \alpha)$ pasa a ser $U_{wc}^{EDF-RF}(m, n, u_{max})$. La inclusión del parámetro m en este último límite es debido a que a priori no se sabe si el límite en función de u_{max} va a depender del número de tareas o no.

En el capítulo 2 se presentaron los algoritmos de asignación de tareas denominados FF, FFI, FFD, BF, BFD, BFI, WF, WFD, WFI, RF, RFD, RFI y OPT. De acuerdo con los resultados obtenidos en los capítulos 3 y 4, de de todos ellos los más destacados en términos de límite de utilización y simplicidad son:

- De entre los que no realizan ordenación de tareas, el algoritmo FF.
- De entre los que realizan ordenación de tareas, el algoritmo FFD.

Además, el límite de utilización para el algoritmo RF es útil como condición de planificabilidad cuando hay cambios de modo, tal como se indica en el apartado 6.5.

Por lo tanto, los algoritmos de asignación FF, FFD y RF son los únicos que se consideran a la hora de obtener los límites de utilización en función de

5.1.1 Planificación EDF

En este apartado se obtienen los límites de utilización en función del máximo factor de utilización, u_{max} , para planificación monoprocesador EDF, empleando asignación FF, FFD o RF.

En primer lugar se obtiene el límite de utilización $U_{wc}^{EDF-RF}(m, n, u_{max})$. El teorema 7, en la página 58, proporcionaba una cota superior del límite de utilización para asignación WF, en función de α . Para ello se presentaba un conjunto de tareas que no cabía en los procesadores. Este conjunto tenía una tarea de factor de utilización α . Eligiendo $\alpha = u_{max}$ se tiene que una de las tareas tiene el factor de utilización u_{max} , y por lo tanto se obtiene para asignación WF la cota superior

$$U_{wc}^{EDF-WF}(m, n, u_{max}) \leq n - (n - 1)u_{max}$$

Por otra parte, la cota inferior proporcionada por el teorema 1, en la página 43, es válida independientemente de si el factor de utilización de alguna de las tareas alcanza el valor α o no. Esta cota inferior es válida para todos los algoritmos razonables de asignación. Como esta cota inferior coincide con la cota superior para planificación EDF-WF, resulta

$$U_{wc}^{EDF-WF}(m, n, u_{max}) = U_{wc}^{EDF-WF}(n, u_{max}) = n - (n - 1)u_{max}$$

y además el límite de utilización para asignación RF, el cual es el mínimo de entre todos los algoritmos de asignación razonables, toma el mismo valor

$$U_{wc}^{EDF-RF}(n, u_{max}) = n - (n - 1)u_{max}$$

A continuación se acota el rango dentro del cual se encuentran los límites de utilización asociados a la planificación EDF-FF y EDF-FFD en función de u_{max} . La cota superior de los límites de utilización para planificación EDF-FF y EDF-FFD en función de α viene dada por el teorema 2, en la página 44. Este teorema presenta un conjunto de tareas que no cabe en el multiprocesador. Este conjunto se divide en dos subconjuntos, un primer subconjunto formado por tareas de factor de utilización

$$\frac{1}{\beta_{EDF} + 1} + \epsilon/m$$

y otro formado por tareas de factor de utilización ϵ/m . Para valores de ϵ arbitrariamente pequeños se tiene que ninguno de los factores de utilización del conjunto alcanza el valor α . Sin embargo, si se incrementase el factor de utilización de la primera tarea para pasar de

$$\frac{1}{\beta_{EDF} + 1} + \epsilon/m$$

a α , el conjunto de tareas resultante seguiría sin caber en los procesadores. Eligiendo $\alpha = u_{max}$, y teniendo en cuenta que la cota superior dada por el teorema 2 coincide con los límites de utilización para planificación EDF-FF y EDF-FFD en función de α resulta

$$\begin{aligned} U_{wc}^{EDF-FF}(m, n, u_{max}) &\leq U_{wc}^{EDF-FF}(n, \alpha = u_{max}) + u_{max} - \frac{1}{\beta_{EDF} + 1} \\ U_{wc}^{EDF-FFD}(m, n, u_{max}) &\leq U_{wc}^{EDF-FFD}(n, \alpha = u_{max}) + u_{max} - \frac{1}{\beta_{EDF} + 1} \end{aligned} \quad (5.2)$$

Sustituyendo las expresiones de los límites de utilización en función de α y operando se obtiene

$$\begin{aligned} U_{wc}^{EDF-FF}(m, n, u_{max}) &\leq u_{max} + \frac{\beta_{EDF}n}{\beta_{EDF} + 1} \\ U_{wc}^{EDF-FFD}(m, n, u_{max}) &\leq u_{max} + \frac{\beta_{EDF}n}{\beta_{EDF} + 1} \end{aligned} \quad (5.3)$$

Al haber elegido $\alpha = u_{max}$, el parámetro β_{EDF} se puede obtener como

$$\beta_{EDF} = \left\lfloor \frac{1}{u_{max}} \right\rfloor$$

Comparando estas expresiones con los límites de utilización en función de α se obtienen los siguientes incrementos máximos. Para $\beta_{EDF} = 1$ el máximo incremento es $(1 - 0'5) = 0'5$, para $\beta_{EDF} = 2$ el máximo incremento es $(0'5 - 0'33) = 0'17$, para $\beta_{EDF} = 3$ el máximo incremento es $(0'33 - 0'25) = 0'08$, y así sucesivamente.

Para el caso particular en que $\beta_{EDF} = 1$ se pueden obtener de forma simple los límites de utilización multiprocesador para planificación EDF-FF y EDF-FFD en función de u_{max} .

Considérese un conjunto de m tareas, y un multiprocesador formado por $n > 1$ procesadores. Se tiene una tarea τ_k de factor de utilización u_{max} , y la utilización total del conjunto es menor o igual que

$$U_{wc}^{EDF-FF}(n-1, \alpha = u_{max}) + u_{max}$$

Se demostrará que dicho conjunto cabe en n procesadores, y por lo tanto

$$U_{wc}^{EDF-FF}(m, n, u_{max}) \geq U_{wc}^{EDF-FF}(n-1, \alpha = u_{max}) + u_{max}$$

Las tareas $\{\tau_1, \dots, \tau_{k-1}\}$ se asignan a los primeros $(n-1)$ procesadores, pues la utilización total de las mismas es menor o igual que

$$U_{wc}^{EDF-FF}(n-1, \alpha = u_{max})$$

y el algoritmo FF tiende a llenar los primeros procesadores. La tarea τ_k se puede asignar también a los procesadores, pues si no cupiese en los $(n-1)$ primeros procesadores se asignaría al último procesador. En el peor de los casos el procesador que recibe la tarea τ_k no recibe ninguna otra tarea. En este caso, las restantes $(m-k)$ tareas se pueden asignar a los huecos que hay en los otros $(n-1)$ procesadores, puesto que conjuntamente con las $(k-1)$ primeras tareas acumulan una utilización menor o igual que

$$U_{wc}^{EDF-FF}(n-1, \alpha = u_{max})$$

Por lo tanto se concluye

$$U_{wc}^{EDF-FF}(m, n, u_{max}) \geq U_{wc}^{EDF-FF}(n-1, \alpha = u_{max}) + u_{max}$$

Para $u_{max} > 0'5$ se obtiene $\beta_{EDF} = 1$, y considerando el caso $n > 1$ se deduce

$$U_{wc}^{EDF-FF}(m, n > 1, u_{max} > 0'5) \geq n/2 + u_{max}$$

El razonamiento empleado es válido porque la asignación FF tiende a llenar los primeros procesadores vacantes antes de acudir a los últimos. Otros algoritmos como el RF o el WF no cumplen esta condición, y por lo tanto este tipo de resultado no es válido.

Empleando la misma línea de razonamiento con el algoritmo FFD se llega a la conclusión

$$U_{wc}^{EDF-FFD}(m, n > 1, u_{max} > 0'5) \geq n/2 + u_{max}$$

Para el caso $\alpha > 0'5$, a partir de las ecuaciones (5.3) se deduce

$$U_{wc}^{EDF-FF}(m, n, u_{max} > 0'5) \leq n/2 + u_{max}$$

$$U_{wc}^{EDF-FFD}(m, n, u_{max} > 0'5) \leq n/2 + u_{max}$$

y por lo tanto

$$U_{wc}^{EDF-FF}(m, n > 1, u_{max} > 0'5) = n/2 + u_{max}$$

$$U_{wc}^{EDF-FFD}(m, n > 1, u_{max} > 0'5) = n/2 + u_{max}$$

El incremento en el límite de utilización empleando u_{max} en lugar de α es significativo para valores de u_{max} mucho mayores que $0'5$. Por ejemplo, para $n = 2$ procesadores y $u_{max} = 1$, el nuevo límite de utilización es 2, y por lo tanto $0'5$ unidades mayor que el obtenido empleando α . La idea es simple, si se sabe que una de las tareas tiene factor de utilización 1 esta tarea cabe en un procesador, y en el otro procesador queda espacio para una utilización adicional de valor 1.

Para el caso de valores de α menores que $0'5$ y asignación FF el límite de utilización empleando u_{max} es el mismo que empleando α . Para demostrar este punto, basta con redistribuir la utilización de las primeras β_{EDF} tareas del segundo subconjunto en el teorema 2, página 44, entre las cuales acumulan una utilización de valor

$$\frac{\beta_{EDF}}{\beta_{EDF} + 1} + \frac{\beta_{EDF}\epsilon}{m}$$

Esta utilización es siempre mayor que u_{max} para $\beta_{EDF} \geq 2$. La primera tarea del segundo subconjunto pasa a tener un factor de utilización u_{max} , y las siguientes $(\beta_{EDF} - 1)$ tareas del segundo subconjunto se reparten la utilización

$$\frac{\beta_{EDF}}{\beta_{EDF} + 1} + \frac{\beta_{EDF}\epsilon}{m} - u_{max}$$

Tras la redistribución, la asignación de tareas a procesadores empleando el algoritmo FF no es posible, y por lo tanto la cota superior dada por el teorema 2 es válida para asignación FF cuando se elige $\alpha = u_{max}$, y $\beta_{EDF} \geq 2$. Como esta cota coincide con el límite de utilización en función de α , y el límite de utilización en función de u_{max} no puede ser menor resulta

$$U_{wc}^{EDF-FF}(n, u_{max}) = \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \quad \text{para } \beta_{EDF} \geq 2$$

La modificación de las tareas del conjunto presentado en el teorema 2 hace que estas no estén ordenadas en sentido de factores de utilización decrecientes, y por lo tanto la cota obtenida no es válida para asignación FFD.

Como resumen, en este apartado se han obtenido los límites de utilización

$$U_{wc}^{EDF-FF}(n > 1, u_{max}) = \begin{cases} u_{max} + n/2 & \text{si } \beta_{EDF} = 1 \\ \frac{\beta_{EDF}n+1}{\beta_{EDF}+1} & \text{si } \beta_{EDF} > 1 \end{cases}$$

$$U_{wc}^{EDF-FFD}(n > 1, u_{max}) = u_{max} + n/2 \quad \text{si } \beta_{EDF} = 1$$

$$\frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1} \leq U_{wc}^{EDF-FFD}(m, n, u_{max}) \leq u_{max} + \frac{\beta_{EDF}n}{\beta_{EDF} + 1}$$

$$U_{wc}^{EDF-RF}(n, u_{max}) = n - (n - 1)u_{max}$$

donde

$$\beta_{EDF} = \left\lfloor \frac{1}{u_{max}} \right\rfloor$$

La obtención del límite de utilización para planificación EDF-FFD en función de u_{max} para $\beta_{EDF} > 1$ no se lleva a cabo en la tesis. Esta forma parte del trabajo futuro.

5.1.2 Planificación RM

Este apartado es análogo al anterior pero empleando planificación local RM en lugar de EDF. El objetivo es obtener los límites de utilización multiprocesador en función del máximo factor de utilización, u_{max} , para planificación RM y empleando los algoritmos de asignación FF, FFD y RF.

En primer lugar se obtiene el límite de utilización $U_{wc}^{RM-RF}(m, n, u_{max})$. El teorema 22, en la página 174, proporcionaba la siguiente cota superior del límite de utilización para asignación WF en función de α .

$$U_{wc}^{RM-WF}(m, n, \alpha) \leq \begin{cases} n_a U_a + n_b U_b - (n - 1)\alpha & \text{si } \alpha < U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a \leq \alpha \leq U_b \\ U_b & \text{si } U_b < \alpha \end{cases}$$

donde

$$\begin{aligned} n_a &= m - \left\lfloor \frac{m-1}{n} \right\rfloor n - 1 \\ n_b &= n - n_a \\ U_a &= \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{\lceil \frac{m+n-1}{n} \rceil} - 1 \right) \\ U_b &= \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{\lceil \frac{m+n-1}{n} \rceil} - 1 \right) \end{aligned}$$

Aparecen tres casos dependiendo del valor de α . Sin embargo, en la práctica se elige $\alpha = u_{max}$ por lo que el último caso se puede eliminar. Téngase en cuenta que para valores de u_{max} mayores que U_b el conjunto de tareas no es planificable, por lo que no tiene sentido definir un límite de utilización en ese caso. En los dos primeros casos de la demostración de la cota superior, el conjunto de tareas empleado tenía una tarea de factor de utilización α . Por lo tanto, la cota propuesta por el teorema 22 es válida en función de u_{max} para los dos primeros casos, los únicos en los que tiene sentido el límite de utilización.

$$U_{wc}^{RM-WF}(m, n, u_{max}) \leq \begin{cases} n_a U_a + n_b U_b - (n-1)u_{max} & \text{si } u_{max} < U_a \\ n_b U_b - (n_b - 1)u_{max} & \text{si } U_a \leq u_{max} \leq U_b \end{cases}$$

La demostración de la cota inferior del límite de utilización asociada a planificación local RM con cualquier algoritmo de asignación razonable, proporcionada por el corolario 2, es totalmente válida en los dos primeros casos, independientemente de si alguna tarea alcanza el valor α . Esto es, para cualquier algoritmo razonable de asignación, RA, se cumple

$$U_{wc}^{RM-RA}(m, n, u_{max}) \geq \begin{cases} n_a U_a + n_b U_b - (n-1)u_{max} & \text{si } u_{max} < U_a \\ n_b U_b - (n_b - 1)u_{max} & \text{si } U_a \leq u_{max} \leq U_b \end{cases}$$

y por consiguiente

$$U_{wc}^{RM-WF}(m, n, u_{max}) = \begin{cases} n_a U_a + n_b U_b - (n-1)u_{max} & \text{si } u_{max} < U_a \\ n_b U_b - (n_b - 1)u_{max} & \text{si } U_a \leq u_{max} \leq U_b \end{cases}$$

Como RF es el peor algoritmo de asignación en términos de límite de utilización se obtiene

$$U_{wc}^{RM-RF}(m, n, u_{max}) = \begin{cases} n_a U_a + n_b U_b - (n-1)u_{max} & \text{si } u_{max} < U_a \\ n_b U_b - (n_b - 1)u_{max} & \text{si } U_a \leq u_{max} \leq U_b \end{cases}$$

A continuación se acota el rango dentro del cual se encuentran los límites de utilización asociados a la planificación RM-FF y RM-FFD en función de u_{max} .

La cota superior del límite de utilización para planificación RM-FF en función de α viene dada por el teorema 16, en la página 159. En este teorema se presenta un conjunto de tareas que no cabe en el procesador. Este conjunto se divide en dos subconjuntos, un primer subconjunto formado por tareas de factor de utilización

$$u_i = \frac{(m - \beta_{RM}(n-1))(2^{1/(m-\beta_{RM}(n-1))} - 1) - (2^{1/(\beta_{RM}+1)} - 1)\beta_{RM}}{(m - \beta_{RM}n)}$$

y un segundo subconjunto formado por tareas de factor de utilización

$$u_i = (2^{1/(\beta_{RM}+1)} - 1) + \frac{\epsilon}{\beta_{RM}n}$$

Para valores de ϵ arbitrariamente pequeños se tiene que ninguno de los factores de utilización del conjunto alcanza el valor α . Sin embargo, si se incrementase el factor de utilización de una tarea del segundo subconjunto, para pasar de

$$(2^{1/(\beta_{RM}+1)} - 1) + \frac{\epsilon}{\beta_{RM}n}$$

a tomar el valor α , el conjunto de tareas resultante seguiría sin caber en los procesadores. Eligiendo $\alpha = u_{max}$ se tiene

$$U_{wc}^{RM-FF}(m, n, u_{max}) \leq U_{wc}^{RM-FF}(m, n, \alpha = u_{max}) + u_{max} - (2^{1/(\beta_{RM}+1)} - 1) \quad (5.5)$$

La cota superior del límite de utilización para planificación RM-FFD en función de α viene dada por el teorema 15, en la página 156. En este teorema se propone un conjunto de tareas que no cabe en el procesador. Este conjunto se divide en dos subconjuntos, un primer subconjunto formado por tareas de factor de utilización ϵ/m , y un segundo subconjunto formado por tareas de factor de utilización

$$u_i = (2^{1/(\beta_{RM}+1)} - 1) + \epsilon/m$$

Para valores de ϵ arbitrariamente pequeños se tiene que ninguno de los factores de utilización del conjunto alcanza el valor α . Sin embargo, si se incrementase el factor de utilización de una tarea del segundo subconjunto, para pasar de

$$(2^{1/(\beta_{RM}+1)} - 1) + \epsilon/m$$

a tomar el valor α , el conjunto de tareas resultante seguiría sin caber en los procesadores. Eligiendo $\alpha = u_{max}$ se tiene

$$U_{wc}^{RM-FFD}(m, n, u_{max}) \leq U_{wc}^{RM-FFD}(m, n, \alpha = u_{max}) + u_{max} - (2^{1/(\beta_{RM}+1)} - 1) \quad (5.6)$$

Sustituyendo las expresiones de los límites de utilización en función de α en las ecuaciones (5.5) y (5.6) se obtiene

$$\begin{aligned} U_{wc}^{RM-FF}(m, n, u_{max}) &\leq (m - \beta_{RM}(n - 1))(2^{1/(m - \beta_{RM}(n - 1))} - 1) + \\ &\quad (n - 2)(2^{1/(\beta_{RM}+1)} - 1) + u_{max} \end{aligned} \quad (5.7)$$

$$U_{wc}^{RM-FFD}(m, n > 1, u_{max}) \leq \beta_{RM}n(2^{1/(\beta_{RM}+1)} - 1) + u_{max}$$

Al haber elegido $\alpha = u_{max}$, el parámetro β_{RM} se puede obtener como

$$\beta_{RM} = \left\lfloor \frac{1}{1 + \log_2(u_{max})} \right\rfloor$$

Comparando estas expresiones con los límites de utilización en función de α se obtienen los siguientes incrementos máximos. Para $\beta_{RM} = 1$ el máximo incremento es $(1 - (2^{1/2} - 1)) \approx 0'586$, para $\beta_{RM} = 2$ el máximo incremento es $((2^{1/2} - 1) - (2^{1/3} - 1)) \approx 0'154$, para $\beta_{RM} = 3$ el máximo incremento es $((2^{1/3} - 1) - (2^{1/4} - 1)) \approx 0'071$,

y así sucesivamente.

Para el caso particular en que $\beta_{RM} = 1$ y $n > 1$, se pueden obtener de forma simple los límites de utilización multiprocesador para planificación RM-FF y RM-FFD en función de u_{max} . Se puede demostrar de forma análoga a como se hizo en el apartado anterior para planificación local EDF que

$$\begin{aligned} U_{wc}^{RM-FF}(m, n > 1, u_{max}) &\geq U_{wc}^{RM-FF}(m-1, n-1, \alpha = u_{max}) + u_{max} \\ U_{wc}^{RM-FFD}(m, n > 1, u_{max}) &\geq U_{wc}^{RM-FFD}(m-1, n-1, \alpha = u_{max}) + u_{max} \end{aligned}$$

Haciendo $u_{max} > 0'414$ resulta $\beta_{RM} = 1$ y

$$\begin{aligned} U_{wc}^{RM-FF}(m, n > 1, u_{max} > 0'414) &\geq (m-n+1)(2^{1/(m-n+1)} - 1) + \\ &\quad (n-2)(2^{1/2} - 1) + u_{max} \\ U_{wc}^{RM-FFD}(m, n > 1, u_{max} > 0'414) &\geq \begin{cases} (m-1)(2^{1/(m-1)} - 1) + u_{max} & \text{si } n = 2 \\ n(2^{1/2} - 1) + u_{max} & \text{si } n > 2 \end{cases} \end{aligned} \quad (5.8)$$

Para el caso $\alpha > 0'414$, a partir de las ecuaciones (5.7) se deduce

$$\begin{aligned} U_{wc}^{RM-FF}(m, n > 1, u_{max} > 0'414) &\leq (m-n+1)(2^{1/(m-n+1)} - 1) + \\ &\quad (n-2)(2^{1/2} - 1) + u_{max} \\ U_{wc}^{RM-FFD}(m, n > 1, u_{max} > 0'414) &\leq n(2^{1/2} - 1) + u_{max} \end{aligned}$$

y por lo tanto

$$\begin{aligned} U_{wc}^{RM-FF}(m, n > 1, u_{max} > 0'414) &= (m-n+1)(2^{1/(m-n+1)} - 1) + \\ &\quad (n-2)(2^{1/2} - 1) + u_{max} \\ U_{wc}^{RM-FFD}(n > 2, u_{max} > 0'414) &= n(2^{1/2} - 1) + u_{max} \end{aligned}$$

Queda pendiente el caso en el cual $n = 2$, $u_{max} > 0.414$ y se tiene asignación FFD. En este caso, el límite de utilización se obtiene directamente teniendo en cuenta que no puede ser menor que el asociado a la asignación FF. Esto es,

$$\begin{aligned} U_{wc}^{RM-FFD}(m, n = 2, u_{max} > 0'414) &\geq U_{wc}^{RM-FF}(m, n = 2, u_{max} > 0'414) = \\ &\quad (m-1)(2^{1/(m-1)} - 1) + u_{max} \end{aligned}$$

Por lo tanto, a partir de esta cota inferior y de la cota superior dada por la ecuación 5.8 para el caso $n = 2$, se deduce

$$U_{wc}^{RM-FFD}(m, n = 2, u_{max} > 0'414) = (m-1)(2^{1/(m-1)} - 1) + u_{max}$$

El incremento en el límite de utilización empleando u_{max} en lugar de α es significativo para valores de u_{max} mucho mayores que 0'414. Por ejemplo, para asignación

FF, $n = 2$ procesadores y $u_{max} = 1/0$, el nuevo límite de utilización es 0'586 unidades mayor que el obtenido empleando α . La idea es simple, si se sabe que una de las tareas tiene factor de utilización 1 esta tarea cabe en un procesador, y en el otro procesador queda espacio para una utilización adicional de al menos $\ln(2)$.

Para el caso de valores de α menores que 0'414 y asignación FF el límite de utilización empleando u_{max} es el mismo que empleando α . Para demostrar este punto, basta con redistribuir la utilización de las primeras $(m - \beta_{RM}n + 1)$ tareas del conjunto presentado en el teorema 15, página 15 entre las cuales acumulan una utilización de valor

$$(m - \beta_{RM}(n - 1))(2^{1/(m-\beta_{RM}(n-1))} - 1) - (\beta_{RM} - 1)(2^{1/(\beta_{RM}+1)} - 1)$$

Esta utilización es siempre mayor que u_{max} para $\beta_{RM} \geq 2$. De las primeras $(m - \beta_{RM}n + 1)$ tareas, la última pasa a tener un factor de utilización u_{max} , y las $(m - \beta_{RM}n)$ restantes se reparten la utilización

$$(m - \beta_{RM}(n - 1))(2^{1/(m-\beta_{RM}(n-1))} - 1) - (\beta_{RM} - 1)(2^{1/(\beta_{RM}+1)} - 1) - u_{max}$$

Tras la redistribución, la asignación de tareas a procesadores empleando el algoritmo FF no es posible, y por lo tanto la cota superior dada por el teorema 16 es válida para asignación FF cuando al menos una de las tareas tiene un factor de utilización de valor u_{max} , y $\beta_{RM} \geq 2$. Como esta cota coincide con el límite de utilización para $\alpha = u_{max}$, y el límite de utilización en función de u_{max} no puede ser menor resulta

$$U_{wc}^{RM-FF}(m, n, u_{max}) = (m - \beta_{RM}(n - 1))(2^{1/(m-\beta_{RM}(n-1))} - 1) + (n - 1)(2^{1/(\beta_{RM}+1)} - 1)$$

para $\beta_{RM} \geq 2$.

Como resumen, en este apartado se han obtenido los límites de utilización

$$U_{wc}^{RM-FF}(m, n > 1, u_{max}) = \begin{cases} (m - n + 1)(2^{1/(m-n+1)} - 1) + (n - 2)(2^{1/2} - 1) + u_{max} & \text{si } \beta_{RM} = 1 \\ (m - (\beta_{RM} - 1))(2^{1/(m-\beta_{RM}-1)} - 1) + (n - 1)(2^{1/(\beta_{RM}+1)} - 1) & \text{si } \beta_{RM} > 1 \end{cases}$$

$$U_{wc}^{RM-FFD}(m, n > 1, u_{max}) = \begin{cases} (m - 1)(2^{1/(m-1)} - 1) + u_{max} & \text{si } \beta_{RM} = 1 \text{ y } n = 2 \\ n(2^{1/2} - 1) + u_{max} & \text{si } \beta_{RM} = 1 \text{ y } n > 2 \end{cases}$$

$$(\beta_{RM}n + 1)(2^{1/(\beta_{RM}+1)} - 1) \leq U_{wc}^{RM-FFD}(m, n > 2, u_{max}) \leq \beta_{RM}n(2^{1/(\beta_{RM}+1)} - 1) + u_{max}$$

$$(\beta_{RM}n + 1)(2^{1/(\beta_{RM}+1)} - 1) \leq U_{wc}^{RM-FFD}(m, n = 2, u_{max}) \leq (m - 1)(2^{1/(m-1)} - 1) + u_{max}$$

$$U_{wc}^{RM-RF}(m, n, u_{max}) = \begin{cases} n_a U_a + n_b U_b - (n-1)u_{max} & \text{si } u_{max} < U_a \\ n_b U_b - (n_b - 1)u_{max} & \text{si } U_a \leq u_{max} \leq U_b \end{cases}$$

donde

$$\begin{aligned} \beta_{RM} &= \left\lfloor \frac{1}{1 + \log_2(u_{max})} \right\rfloor \\ n_a &= m - \left\lfloor \frac{m-1}{n} \right\rfloor n - 1 \\ n_b &= n - n_a \\ U_a &= \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lceil \frac{m+n-1}{n} \rceil} - 1 \right) \\ U_b &= \left\lfloor \frac{m+n-1}{n} \right\rfloor \left(2^{1/\lfloor \frac{m+n-1}{n} \rfloor} - 1 \right) \end{aligned}$$

La obtención del límite de utilización para planificación RM-FFD en función de u_{max} para $\beta_{RM} > 1$ no se lleva a cabo en la tesis. Esta forma parte del trabajo futuro.

5.2 Consideración de los factores de utilización de las tareas

En este apartado se establece cual es la máxima mejora de los límites de utilización multiprocesador para planificación local RM derivada del incremento del límite de utilización monoprocesador, obtenido a partir del conocimiento de los factores de utilización de las tareas.

Para ello se aplicará la condición de planificabilidad del teorema 11, la cual tiene en cuenta los factores de utilización de las tareas, y permite obtener un límite de utilización más alto.

Una parte fundamental en la obtención de los límites de utilización multiprocesador era la obtención de cotas superiores de utilización. Es decir, valores por debajo de los cuales se encuentran los límites de utilización. Para el caso de planificación local RM estas cotas vienen dadas por los teoremas 15, 16, 22, y 25, en las páginas 156, 159, 174, y 188 respectivamente.

La idea es obtener el incremento de las cotas anteriores que hace que el conjunto de tareas no quepa en el multiprocesador, incluso aplicando la condición de planificabilidad del teorema 11.

Se han realizado experimentos con dicha condición a partir de los cuales se ha obtenido las siguientes conclusiones:

- La máxima mejora con respecto al límite de Liu y Layland (1973) ocurre cuando se tiene una tarea de factor de utilización conocido y un número infinito de tareas de factores de utilización desconocidos, pero idénticos.

α	ΔL_{RM}
0'414	0'0674n
0'200	0'0176n
0'100	0'0047n
0'010	0'0001n

Tabla 5.1: Máximos incrementos del límite de utilización multiprocesador L_{RM} , para distintos valores de α .

- La mejora obtenida en ese caso es tanto mayor cuanto mayor es el factor de utilización conocido.

Empleando planificación local RM se habían encontrado cuatro límites de utilización multiprocesador diferentes: H_{RM} , K_{RM} , L_{RM} , e I_{RM} . A continuación verá cual es el máximo incremento de los mismos que puede obtenerse empleando el teorema 11.

La cota superior proporcionada por el teorema 15 establecía un valor que no podía ser superado por el límite de utilización H_{RM} . El factor de utilización era el mismo para todas las tareas del conjunto empleado en la demostración. Por lo tanto, el límite de utilización obtenido a partir del teorema 11 coincide con el clásico de Liu y Layland (1973), y no se obtiene ninguna mejora en el límite de utilización multiprocesador H_{RM} .

La cota superior proporcionada por el teorema 16 establecía un valor que no podía ser superado por el límite de utilización multiprocesador K_{RM} . Este límite se obtenía empleando la condición de planificabilidad de Liu y Layland (1973), aún cuando los factores de utilización eran diferentes, lo que introdujo cierto pesimismo. Como se deduce del conjunto de tareas empleado en la demostración, sólo puede mejorarse el límite de utilización del primer procesador. Considerando $\epsilon = 0$, este primer procesador recibe tareas de factor de utilización $(2^{1/(\beta_{RM}+1)} - 1)$, y el hueco que queda en el primer procesador se reparte equitativamente entre un conjunto de $(m - \beta_{RM}n)$ tareas. En el caso más favorable, la máxima mejora con respecto al límite de utilización de Liu y Layland (1973) se obtiene cuando $(m - \beta_{RM}n) \rightarrow \infty$, y se tiene una tarea adicional de factor de utilización $(2^{1/2} - 1)$. La mejora obtenida en este caso es de 0'067 unidades. Por lo tanto, en el límite de utilización multiprocesador K_{RM} puede aumentar a lo sumo en 0'067 unidades.

En cuanto al límite de utilización multiprocesador L_{RM} , el teorema 22 proporcionaba una cota superior del mismo. Aparecían tres casos en la demostración dependiendo del valor de α . La situación más favorable en cualquiera de los tres casos consiste en tener infinitas tareas en un procesador, de las cuales una tiene un factor de utilización conocido de valor $\ln(2)$. En este caso se obtiene una mejora de valor 0'166n. Sin embargo, el límite de utilización multiprocesador L_{RM} tiene utilidad práctica, únicamente para valores de α menores o iguales que $(2^{1/2} - 1) \approx 0'414$, pues en caso contrario el límite de utilización multiprocesador es similar al límite monoprocesador. La tabla 5.1 muestra la máxima mejora en el límite de utilización multiprocesador L_{RM} para distintos valores de α menores o iguales que 0'414.

En cuanto al límite de utilización multiprocesador I_{RM} , el teorema 25 proporció-

naba una cota superior del mismo. La situación más favorable en cuanto a mejora es la misma que la descrita para el límite L_{RM} . Por lo tanto, los máximos incrementos del límite L_{RM} son aplicables al límite I_{RM} .

Como se ha mostrado, los incrementos de los límites de utilización multiprocesador son escasos. Por lo tanto los límites de utilización multiprocesador proporcionados en los capítulos 4 son prácticamente ajustados. Además, en el caso promedio la mejora en el límite de utilización monoprocesador proporcionada por el conocimiento de los factores de utilización es muy bajo (J.M. López y García, 2000a).

En cualquier caso, los máximos incrementos proporcionados no tienen por que alcanzarse. Para que se alcanzasen debería ser posible demostrar que cualquier conjunto de tareas de utilización total menor o igual que el límite incrementado es planificable.

Los límites de utilización proporcionados en el capítulo 3 para planificación monoprocesador EDF son estrictamente ajustados, ya que la condición de planificabilidad empleada era ajustada, incluso conociendo los factores de utilización de algunas tareas.

Capítulo 6

Extensiones del modelo de tareas

En este capítulo se obtienen límites de utilización multiprocesador para planificación local EDF o RM, y para un modelo de tareas más general que el considerado en los capítulos 3 y 4.

El nuevo modelo de tareas extiende el modelo de tareas de Liu y Layland (1973) para tener en cuenta:

- Acceso a recursos compartidos.
- *Jitter* de activación.
- Plazos menores que los periodos.
- Tareas aperiódicas.
- Cambios de modo.

Desde el punto de vista del análisis, las cuatro primeras extensiones transforman el conjunto de tareas periódicas extendido en un nuevo conjunto de tareas periódicas que sigue el modelo de Liu y Layland (1973), y es tanto o más difícil de planificar que el original. La idea es aplicar sobre este nuevo conjunto de tareas se los límites de utilización de los capítulos 3 y 4.

Los cambios de modo implican la modificación del conjunto de tareas a lo largo del tiempo. Independientemente del algoritmo de asignación razonable empleado, el caso peor desde el punto de vista del análisis es equivalente al caso de un conjunto de tareas estático asignado empleando el algoritmo RF.

6.1 Acceso a recursos compartidos

Una de las restricciones empleadas en la obtención de los límites de utilización, propuestos en los capítulos 3 y 4, era que las tareas debían ser independientes.

Sin embargo, en la práctica es necesario en ciertos casos comunicar las tareas entre sí. Una forma de realizar esta comunicación es mediante el acceso a recursos compartidos, como puede ser un área de memoria. Cuando dos o más tareas acceden a un recurso compartido pueden aparecer problemas de coherencia debido a efectos

de carrera, por lo que es habitual emplear mecanismos que arbitren el acceso a los mismos como pueden ser los semáforos (Tanenbaum, 1991). En este caso se dice que el recurso compartido está protegido. El objetivo de la protección es puramente funcional, evitando resultados incorrectos debidos a efectos de carrera.

Sin embargo, al evitar los problemas de coherencia mediante mecanismos de protección se elimina toda predecibilidad del sistema de tiempo real. La explicación es simple. La protección consiste en evitar el acceso simultáneo al recurso compartido por dos o más tareas simultáneamente. Por lo tanto una activación de alta prioridad puede bloquearse esperando a que una activación de baja prioridad libere el recurso compartido, lo que penaliza su tiempo de respuesta. Con la simple protección del recurso sin incluir ningún mecanismo adicional, una activación de alta prioridad puede bloquearse durante un tiempo ilimitado en el peor de los casos. Lógicamente esto es inadmisibile en un sistema de tiempo real estricto. Una solución al problema de funcionalidad y predecibilidad cuando se accede a recursos compartidos consiste en el empleo de protocolos de acceso. Con estos protocolos se limita el tiempo máximo que pueden bloquearse las activaciones de una tarea τ_i debido a activaciones de menor prioridad (Buttazzo, 1997; J.A. Stankovic y Buttazzo, 1998). Este máximo tiempo de bloqueo se denota por B_i .

En el contexto de la tesis se supone el empleo de los protocolos monoprocesador PCP y SRP para planificación RM, y SRP para planificación EDF (Buttazzo, 1997; J.A. Stankovic y Buttazzo, 1998).

Empleando el protocolo PCP o SRP con planificación basada en las prioridades fijas RM, el límite de utilización de un procesador P_j pasa a ser

$$m_j(2^{1/m_j} - 1) - \max_{\tau_i \in P_j}(B_i/T_i) \quad (6.1)$$

Empleando el protocolo SRP con planificación EDF el límite de utilización de un procesador P_j pasa a ser

$$1 - \max_{\tau_i \in P_j}(B_i/T_i)$$

En las expresiones anteriores m_j es el número de tareas asignadas al procesador P_j , y $\tau_i \in P_j$ hace referencia a las tareas que se asignan al procesador P_j .

El problema es como generalizar estos límites de utilización al caso multiprocesador con particionado. Puesto que los protocolos de acceso a recursos compartidos empleados son protocolos monoprocesador es necesario que todo subconjunto formado por tareas que dependen entre si se asigne al mismo procesador. Las tareas que dependen entre si constituyen lo que se denominará una *macrotarea* en el contexto de la tesis.

Para ilustrar los conceptos de dependencia y macrotarea se hará referencia a la figura 6.1. Las tareas τ_1 , τ_2 y τ_3 tienen un recurso compartido, representado por las líneas que las unen. Además la tarea τ_4 comparte un recurso con la tarea τ_3 . Por lo tanto, las tareas τ_1 , τ_2 , τ_3 y τ_4 son tareas dependientes (directa o indirectamente) y forman la macrotarea γ_1 . La tarea τ_5 es independiente y forma la macrotarea γ_2 . Finalmente las tareas τ_6 y τ_7 dependen sólo una de otra, y forman la macrotarea γ_3 . El número de macrotareas del sistema se representa por \hat{m} . En el caso de la

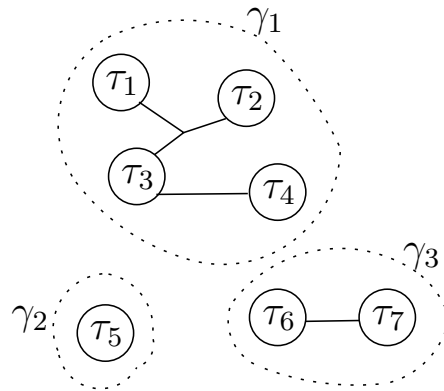


Figura 6.1: Concepto de independencia y definición de macrotareas.

figura 6.1 se tiene $\hat{m} = 3$.

Desde el punto de vista de la asignación cada macrotarea funciona como una tarea monolítica para cumplir con la restricción de que las tareas dependientes entre sí se asignen al mismo procesador. Sin embargo para poder realizar la asignación es necesario asignar un factor de utilización válido a la macrotarea. El factor de utilización de la macrotarea γ_k se define como

$$\hat{u}_k = \sum_{\tau_i \in \gamma_k} u_i + \max_{\tau_i \in \gamma_k} (B'_i/T_i) \quad (6.2)$$

B'_i representa el máximo bloqueo de la tarea τ_i en la situación hipotética de que todas las tareas del sistema se encontrasen en un mismo procesador. Por el contrario B_i representa el máximo bloqueo de la tarea τ_i considerando únicamente las tareas del sistema que se encuentran en su mismo procesador después de la asignación. Cuantas más tareas hay en un procesador tantos más casos posibles de bloqueo hay, y por lo tanto se cumple $B'_i \geq B_i$.

La utilización total de las macrotareas del sistema, \hat{U} , se obtiene como la suma de los factores de utilización de las macrotareas.

$$\hat{U} = \sum_{k=1}^{\hat{m}} \hat{u}_i \quad (6.3)$$

Como se puede observar, a partir de un conjunto de m tareas dependientes se ha obtenido un conjunto de \hat{m} macrotareas independientes, de factores de utilización proporcionados por la ecuación (6.2).

6.1.1 Planificación EDF

Para el caso de planificación local EDF se puede demostrar que si el conjunto de macrotareas obtenido a partir del conjunto original es planificable en el multiprocesador, también lo es el conjunto de tareas original. Para ello, considérese un conjunto

de \hat{m} macrotareas que es planificable en el multiprocesador. A uno cualquiera de los procesadores P_j se han asignado un conjunto de \hat{m}_j macrotareas, planificable en ese monoprocesador. Esto es, $\sum_{k=1}^{\hat{m}_j} \hat{u}_k \leq 1$. Por lo tanto, aplicando la ecuación (6.2) se cumple

$$\sum_{k=1}^{\hat{m}_j} \sum_{\tau_i \in \gamma_k} u_i + \sum_{k=1}^{\hat{m}_j} \max_{\tau_i \in \gamma_k} (B'_i/T_i) \leq 1$$

Sea m_j el número de tareas asignadas a P_j . Se cumple entonces

$$\sum_{i=1}^{m_j} u_i + \sum_{k=1}^{\hat{m}_j} \max_{\tau_i \in \gamma_k} (B'_i/T_i) \leq 1$$

Por una parte, $B'_i \geq B_i$, y por otra parte

$$\max_{\tau_i \in P_j} (B_i/T_i) \leq \sum_{k=1}^{\hat{m}_j} \max_{\tau_i \in \gamma_k} (B_i/T_i)$$

Se deduce entonces

$$\sum_{i=1}^{m_j} u_i + \max_{\tau_i \in P_j} (B_i/T_i) \leq 1$$

Lo cual demuestra que el conjunto de tareas asignado a P_j es planificable.

Por lo tanto, para obtener los límites de utilización de un conjunto de macrotareas, lo único que hay que hacer es sustituir en los límites de utilización del capítulo 3 las variables m , α , β_{EDF} y U por sus homólogas \hat{m} , $\hat{\alpha}$, $\hat{\beta}_{EDF}$ y \hat{U} .

Los límites de utilización multiprocesador con recursos compartidos dependen de los parámetros $\hat{\alpha}$ y $\hat{\beta}_{EDF}$, los cuales se obtienen de forma análoga a los parámetros α y β_{EDF} .

$$\begin{aligned} \hat{\alpha} &\geq \hat{u}_i \\ \hat{\beta}_{EDF} &= \left\lfloor \frac{1}{\hat{\alpha}} \right\rfloor \end{aligned} \tag{6.4}$$

En el caso de no existir recursos compartidos $m \leq \beta_{EDF}n$ y era una condición suficiente de planificabilidad multiprocesador para planificación local EDF. De igual forma, bajo la presencia de recursos compartidos $\hat{m} \leq \hat{\beta}_{EDF}n$ es una condición suficiente de planificabilidad multiprocesador para planificación local EDF.

Debe tenerse en cuenta que los límites de utilización multiprocesador propuestos para el caso de recursos compartidos son válidos únicamente cuando se cumple $\hat{m} \geq \hat{\beta}_{EDF}n$ y $\hat{\alpha} \leq 1$. Un valor $\alpha > 1$ hace no planificable el conjunto de tareas, independientemente de la utilización total del mismo.

Por ejemplo, considérese el conjunto de tareas dependientes de la figura 6.1. Supóngase vienen definidas por los parámetros de la tabla 6.1. Se recuerda que los valores B' se corresponden con el máximo bloqueo de la tarea suponiendo que las siete tareas se encontrasen en el mismo procesador. Se desea saber si el conjunto

	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7
u	0'1	0'2	0'2	0'1	0'4	0'1	0'2
B'/T	0'05	0'03	0'02	0	0	0'1	0'03

Tabla 6.1: Ejemplo de análisis de planificabilidad de tareas dependientes en multi-procesadores.

de tareas anterior es planificable sobre dos procesadores empleando asignación de tareas FF y planificación EDF. En el ejemplo se tienen tres macrotareas: γ_1 , γ_2 y γ_3 de factores de utilización

$$\hat{u}_1 = 0'1 + 0'2 + 0'2 + 0'1 + \max\{0'05 + 0'03 + 0'02\} = 0'65$$

$$\hat{u}_2 = 0'4 + \max\{0\} = 0'4$$

$$\hat{u}_3 = 0'1 + 0'2 + \max\{0'1 + 0'03\} = 0'4$$

El máximo factor de utilización alcanzable se elige como $\hat{\alpha} = 0'65$. Por lo tanto $\hat{\beta}_{EDF} = \lceil 1/0'65 \rceil = 1$. El límite de utilización para asignación FF y planificación EDF es

$$\hat{U}_{wc}^{EDF-FF}(n, \hat{\beta}_{EDF}) = \frac{\hat{\beta}_{EDF}n + 1}{\hat{\beta}_{EDF} + 1}$$

Para $\hat{\beta}_{EDF} = 1$ y $n = 2$ toma el valor 1'5. La utilización total del conjunto de las tres macrotareas es $(0'65 + 0'4 + 0'4) = 1'45$. Por lo tanto, el conjunto de tareas del ejemplo es planificable.

Una posible mejora de los límites de utilización consiste en redefinir el valor de B' , considerando para su obtención únicamente las tareas que pertenecen a la misma macrotarea, en lugar de considerar todas las tareas del sistema como se hizo hasta ahora. Sin embargo, no se dispone de una demostración definitiva que avale la validez de este resultado. Su obtención forma parte del trabajo futuro.

6.1.2 Planificación RM

Desgraciadamente con planificación RM el proceso de obtención de los límites de utilización no es tan simple como en el caso de planificación EDF. La razón estriba en que la planificabilidad de las macrotareas asignadas a un procesador no depende del número de macrotareas sino del número total de tareas asignadas al procesador. Una solución muy conservadora consiste en utilizar un análisis idéntico al de EDF y multiplicar los límites de utilización para EDF por el término $\ln(2)$. Es decir, considerar un caso muy pesimista en el cual todos los procesadores reciben un número infinito de tareas, y por lo tanto el límite de utilización de cada procesador es $\ln(2)$. Pueden verse otros ejemplos de esta técnica en el apartado 6.3.

Otra solución menos conservadora sería obtener los límites de utilización sobre n procesadores para un conjunto de \hat{m} macrotareas, tal como se hizo en el capítulo 4

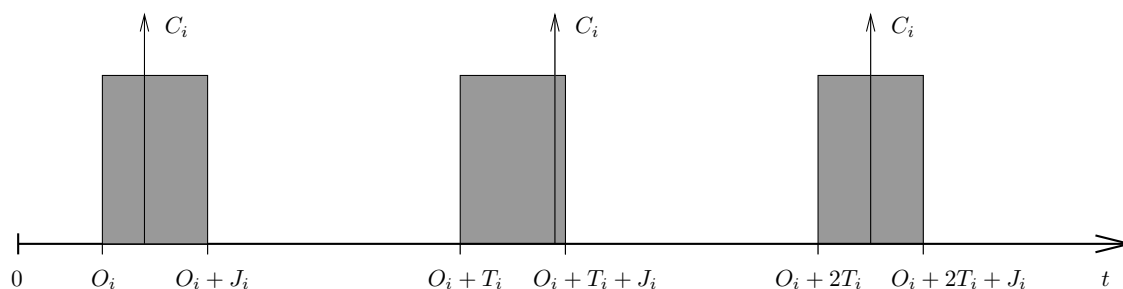


Figura 6.2: Ejemplo de tarea periódica con *jitter*.

para un conjunto de m tareas. No obstante las demostraciones en este caso serían mucho más complejas que las vistas en el capítulo 4.

6.2 *Jitter* de activación

Una de las restricciones del modelo de tarea periódica ideal es que el tiempo transcurrido entre dos activaciones consecutivas de una misma tarea es constante e igual al periodo.

Sin embargo en la práctica esto no es así y se suele observar un retraso de la activación denominado *jitter* de activación en la literatura. Las causas de este retraso son debidas fundamentalmente a la gestión interna de las tareas llevada a cabo por el planificador, o a problemas de granularidad temporal (K.W. Tindell y Wellings, 1994).

El *jitter* es una fuente de impredecibilidad que debe acotarse de alguna forma. Lo habitual es proporcionar un valor J_i que indica el *jitter* máximo de una tarea periódica τ_i .

En la figura 6.2 puede verse un ejemplo de una tarea periódica con *jitter*. El área sombreada indica instantes posibles de activación. El ancho es igual al *jitter*. Puede verse como las tres primeras activaciones de la tarea periódica ocurren en los intervalos $[O_i, O_i + J_i]$, $[O_i + T_i, O_i + T_i + J_i]$ y $[O_i + 2T_i, O_i + 2T_i + J_i]$.

Sin embargo, independientemente del *jitter* el plazo absoluto sigue siendo el mismo que el de la tarea sin *jitter*.

Se podría obtener el límite de utilización monoprocesador ajustado para RM y EDF teniendo en cuenta el *jitter*. El problema del límite así obtenido es que requeriría un proceso completo de demostración para generalizarlo al caso multiprocesador. Este proceso de demostración sería a buen seguro mucho más complejo que el ya de por sí complejo proceso de demostración de los capítulos 3 y 4. Como solución de compromiso se emplearán los límites de utilización monoprocesador habituales sobre un conjunto de tareas sin *jitter* obtenido mediante transformación del original.

El efecto del *jitter* sobre los límites de utilización se puede incluir admitiendo cierto grado de pesimismo. Si el tiempo de computación de cada tarea se incrementa en el valor de su *jitter* se obtiene un nuevo conjunto de tareas de tiempos de computación $C'_i = (C_i + J_i)$ y periodos $T'_i = T_i$. Los nuevos factores de utiliza-

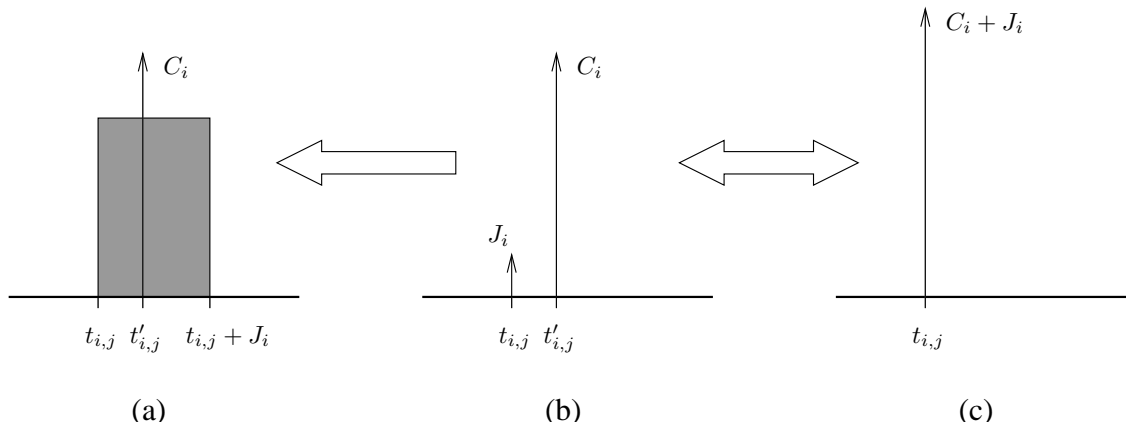


Figura 6.3: Transformación de un conjunto de tareas con *jitter* a un conjunto de tareas sin *jitter*.

ción de las tareas pasarían a ser $u'_i = C'_i/T'_i$. Si el conjunto obtenido a partir de la transformación es planificable también lo será el conjunto original.

Para demostrar este hecho considérese una activación cualquiera de una tarea τ_i de instante de activación ideal $t_{i,j}$, instante de activación real $t'_{i,j}$ y tiempo de computación C_i . La diferencia entre el instante de activación real e ideal verifica $t'_{i,j} - t_{i,j} \leq J_i$, donde J_i es el *jitter* de la tarea. Esta situación se corresponde con la subfigura (a) de la figura 6.3.

Si artificialmente se añade al sistema otra activación de igual prioridad, igual plazo absoluto, instante de activación $t_{i,j}$ y tiempo de computación J_i , la planificabilidad empleando RM o EDF no puede mejorar. Esto es, si se sustituyese la activación con *jitter* de la subfigura (a) por las dos activaciones sin *jitter* de la subfigura (b) y el sistema resultante fuese planificable, también lo sería el sistema original que incluía la activación con *jitter*. Esta relación de planificabilidad se representa en la figura 6.3 mediante una flecha que va de la subfigura (b) a la subfigura (a).

Como las dos activaciones de la subfigura (b) tienen el mismo tiempo plazo absoluto, misma prioridad, y no puede haber periodos de inactividad entre los instantes de activación de ambas al ser $t'_{i,j} \leq t_{i,j} + J_i$, resulta que ambas activaciones se pueden sustituir por una activación equivalente de instante de activación $t_{i,j}$, tiempo de computación $(C_i + J_i)$, e idéntica prioridad y plazo absoluto que las anteriores. Esta activación equivalente se representa en la subfigura (c) de la figura 6.3. La equivalencia entre la planificabilidad en las situaciones (b) y (c) se ha representado en la figura 6.3 con una flecha doble.

Resumiendo, una forma sencilla de tener en cuenta el efecto del *jitter* en los límites de utilización multiprocesador consiste en incrementar el tiempo de computación de cada tarea en el valor de su *jitter* antes de aplicar los límites de utilización dados en los capítulos 3 y 4. El inconveniente es el elevado grado de pesimismo para valores elevados del *jitter*.

6.3 Plazos menores que los periodos

Los límites de utilización de Liu y Layland (1973) para planificación RM y EDF son válidos para el caso de plazos iguales a los periodos.

En el caso de planificación RM y plazos menores que los periodos Lehoczky (1990) generalizó el modelo de las tareas, permitiendo plazos de valor $D_i = \Delta T_i$. Cuando $\Delta < 1$ se tienen plazos menores que los periodos y para $\Delta = 1$ plazos iguales a los periodos. La expresión del límite de utilización obtenida para un número de tareas arbitrario es demasiado compleja para ser generalizado al caso multiprocesador. Además la aplicabilidad de estos límites de utilización es muy reducida pues habitualmente el plazo de cada tarea cumple $D_i = \Delta_i T_i$, y Δ_i es diferente para cada tarea. En estas condiciones la planificación no es RM sino DM (basada en prioridades fijas asignadas de acuerdo con el criterio DM) y se elige de manera pesimista $\Delta = \min_{1 \leq i \leq m} \Delta_i$.

Se considera el caso más simple caracterizado por un número de tareas infinito, el cual se corresponde además con el número de tareas más desfavorable. El límite de utilización viene dado por la siguiente ecuación.

$$U_{wc}^{RM}(\Delta) = \begin{cases} \Delta & \text{si } 0 < \Delta \leq 1/2 \\ \ln(2\Delta) + 1 - \Delta & \text{si } 1/2 \leq \Delta \leq 1 \end{cases}$$

En el caso de planificación EDF y plazos de valor $D_i = \Delta T_i$, con $\Delta \leq 1$ se había demostrado en el apartado 1.4.2 el límite de utilización

$$U_{wc}^{EDF}(\Delta) = \Delta$$

Los límites de utilización monoprocador mostrados tanto para planificación RM como EDF y plazos menores que los periodos se obtienen multiplicando por un valor constante menor que uno el límite de utilización para planificación monoprocador EDF con plazos iguales a los periodos.

La consecuencia es inmediata, los límites de utilización multiprocador se obtienen multiplicando por la misma constante los límites de utilización multiprocador para EDF con plazos iguales a los periodos. La diferencia entre la planificación EDF y RM radica en el valor de dicha constante.

Así por ejemplo, para planificación EDF o RM, asignación FFD y plazos menores o iguales que los periodos se tiene

$$U_{wc}^{EDF-FFD}(n, \gamma_{EDF}, \Delta) = \Delta \frac{\gamma_{EDF} n + 1}{\gamma_{EDF} + 1}$$

$$U_{wc}^{RM-FFD}(n, \gamma_{RM}, \Delta) = \begin{cases} \Delta \frac{\gamma_{RM} n + 1}{\gamma_{RM} + 1} & \text{si } 0 < \Delta \leq 1/2 \\ (\ln(2\Delta) + 1 - \Delta) \frac{\gamma_{RM} n + 1}{\gamma_{RM} + 1} & \text{si } 1/2 \leq \Delta \leq 1 \end{cases}$$

Los parámetros γ_{EDF} y γ_{RM} se obtienen de forma análoga al parámetro β_{EDF} ,

teniendo en cuenta que ahora la capacidad del procesador no es uno.

$$\gamma_{EDF}(\Delta) = \lfloor \Delta/\alpha \rfloor$$

$$\gamma_{RM}(\Delta) = \begin{cases} \lfloor \Delta/\alpha \rfloor & \text{si } 0 < \Delta \leq 1/2 \\ \lfloor \frac{\ln(2\Delta)+1-\Delta}{\alpha} \rfloor & \text{si } 1/2 \leq \Delta \leq 1 \end{cases}$$

La demostración de estos valores es análoga a la proporcionada en el lema 2 para la obtención de β_{EDF} .

Los límites de utilización propuestos son válidos única y exclusivamente cuando $\gamma_{EDF}(\Delta) > 0$ para planificación EDF multiprocesador y cuando $\gamma_{RM}(\Delta) > 0$ para planificación RM multiprocesador. En caso contrario, habría tareas que no cabrían en un procesador ni siquiera siendo las únicas asignadas al procesador. Bajo estas circunstancias el conjunto de tareas no es planificable independientemente de la utilización total del mismo.

De igual forma que sucedía en el caso de plazos iguales a los periodos el conjunto de tareas es planificable si cumple $m \leq \gamma_{EDF}(\Delta)n$ en el caso de planificación EDF multiprocesador, o si cumple $m \leq \gamma_{RM}(\Delta)n$ en el caso de planificación RM multiprocesador.

Los límites de utilización propuestos para el caso de plazos menores que los periodos son más pesimistas que los proporcionados en los capítulos 3 y 4. La razón está en la elección $\Delta = \min_{1 \leq i \leq m} u_i$ tanto para planificación EDF como RM, y la elección de un número infinito de tareas para el caso de planificación RM.

El pesimismo se puede reducir considerablemente para el caso de planificación EDF si se utiliza la condición de planificabilidad

$$\sum_{i=1}^m \frac{C_i}{D_i} \leq 1$$

descrita en el apartado 1.4.2. Si se redefiniere el concepto de factor de utilización de una tarea τ_i para que este fuese igual a (C_i/D_i) , todos los resultados proporcionados en el capítulo 3 serían totalmente válidos para el caso de plazos menores que los periodos. Por ejemplo, para el caso de planificación EDF-FF se tiene la condición de planificabilidad

$$\sum_{i=1}^m \frac{C_i}{D_i} \leq \frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1}$$

6.4 Tareas aperiódicas

Los sistemas de tiempo real además de contener tareas periódicas contienen habitualmente tareas aperiódicas, las cuales se caracterizan por plazos no estrictos, e instantes de llegada impredecibles.

El objetivo en este tipo de sistemas que contienen tareas periódicas y aperiódicas es garantizar los plazos de las tareas periódicas y hacer lo más bajo posible el tiempo medio de respuesta de las tareas aperiódicas.

En la literatura se plantean diferentes esquemas de planificación conjunta de tareas periódicas y aperiódicas. Uno de los más habituales es el empleo de servidores de tareas aperiódicas (Buttazzo, 1997; Bernat y Burns, 1999). Dependiendo del tipo de servidor empleado, estos interfieren con la planificación de las tareas periódicas de diferente manera.

Sólo se consideran aquellos servidores que interfieren de manera estrictamente periódica sobre las tareas periódicas. Es decir, a efectos de estudio de planificabilidad cada servidor y las tareas aperiódicas que sirve se pueden sustituir por una tarea periódica. Ejemplos de estos servidores son el Sporadic Server (SS) y Priority Exchange (PE) para planificación RM, y sus versiones dinámicas para planificación EDF (Buttazzo, 1997).

Empleando servidores con interferencia estrictamente periódica los límites de utilización propuestos en los capítulos 3 y 4 siguen siendo válidos.

Con la consideración de tareas aperiódicas surgen nuevos problemas como son la elección de los servidores, el ajuste de sus parámetros y el algoritmo de asignación de tareas aperiódicas a servidores. No obstante, estos problemas están fuera del contexto de esta tesis.

6.5 Cambios de modo

En determinadas aplicaciones de tiempo real el conjunto de tareas a planificar no es fijo, sino que varía a lo largo del tiempo para adaptarse a un entorno cambiante. Hay situaciones en las que el desconocimiento del entorno hace que el conjunto de tareas evolucione de una manera desconocida a priori. Sin embargo, hay situaciones en las que el entorno puede encontrarse en un conjunto de estados claramente definidos y conocidos a priori, cada uno de los cuales requiere un conjunto de tareas también definido y conocido a priori. En este apartado se supone que el sistema se encuentra en esta última situación, asociada a lo que se denomina cambios de modo en la literatura (Pedro y Burns, 1998).

La variación del conjunto de tareas puede suponer la modificación de parámetros de ciertas tareas como pueden ser los periodos de las mismas, o incluso la destrucción de tareas y la creación de nuevas tareas. En general, puede considerarse que el conjunto de tareas cambia única y exclusivamente mediante la creación y destrucción de tareas. Aquellas tareas que simplemente cambian de parámetros se pueden modelar como una tarea con los parámetros antiguos que se destruye, y otra nueva que se crea a continuación con los nuevos parámetros.

En el caso más simple de cambio de modo se destruyen todas las tareas del sistema antes de crear nuevas tareas. De esta forma antes de asignar las nuevas tareas se tiene un multiprocesador vacío. Por lo tanto, aplicando los límites de utilización proporcionados se puede estudiar de manera simple la planificabilidad del sistema. Sin embargo en la práctica las cosas no son tan simples. Por ejemplo, el sistema puede contener tareas de control críticas que no pueden ser abortadas y son necesarias en todos los modos.

Cada vez que se crea una tarea esta debe asignarse a un procesador. Los algoritmos de asignación de tareas que implican la ordenación del conjunto de tareas no

pueden aplicarse, pues esto puede suponer la migración de tareas entre procesadores. Ejemplos de estos algoritmos son el FFD y el BFD. Por el contrario, algoritmos en línea como el FF o BF pueden aplicarse en estas situaciones dinámicas.

Una vez fijado el algoritmo en línea de asignación de tareas y el algoritmo de planificación monoprocesador, podría pensarse que el análisis de planificabilidad puede realizarse aplicando simplemente a cada modo el límite de utilización asociado. Sin embargo esto no es válido por dos razones:

- En determinados casos puede aparecer un periodo transitorio en el cual se produce una sobrecarga del sistema. El análisis de planificabilidad en dicho periodo debe realizarse con técnicas diferentes a las habituales (Pedro y Burns, 1998; Real y Wellings, 1999).
- La destrucción de tareas hace que la asignación en línea de nuevas tareas a procesadores sea diferente a la que resultaría aplicando el mismo algoritmo fuera de línea. Puede llegarse al caso incluso de que la asignación en línea realizada sea una asignación inválida fuera de línea. Como consecuencia, los límites de utilización multiprocesador proporcionados en los capítulos anteriores pueden no ser válidos incluso en régimen permanente (una vez finalizado el periodo transitorio).

El análisis de planificabilidad del periodo transitorio en monoprocesadores ha sido llevado a cabo en otros trabajos (Pedro y Burns, 1998; Real y Wellings, 1999). Sin embargo en el caso multiprocesador aparecen nuevos problemas que se indican al final de este apartado.

La tesis considera únicamente el análisis de planificabilidad en el régimen permanente. En el caso monoprocesador el análisis de planificabilidad del régimen permanente con cambios de modo se realiza simplemente aplicando las condiciones de planificabilidad habituales, ya sean basadas en límites de utilización o no. Sin embargo en el caso multiprocesador las cosas se complican tal como ilustra la figura 6.4. En la figura 6.4 se muestra la transición del sistema entre modos $1 \rightarrow 2$, $2 \rightarrow 3$ y $3 \rightarrow 1$. Se tienen tres procesadores con un planificador EDF y las tareas se asignan empleando el algoritmo FF. Cada tarea se representa mediante una caja, de ancho igual al factor de utilización de la misma. El área sombreada indica la capacidad residual de cada uno de los procesadores.

La transición entre los modos 1 y 2 consiste en la destrucción de las tareas τ_2 y τ_6 , y la creación de las tareas τ_7 y τ_8 . A pesar de asignar las tareas τ_7 y τ_8 de acuerdo con el algoritmo FF, sucede que en el modo 2 la tarea τ_3 se encuentra asignada al segundo procesador cuando hay un hueco suficiente para ella en el primer procesador. Esto es consecuencia de la destrucción de la tarea τ_2 , y hace inaplicable el límite de utilización propuesto en el capítulo 3 para planificación EDF-FF. Independientemente del algoritmo de asignación de tareas empleado puede suceder que la asignación en línea de tareas sea equivalente a la peor asignación posible fuera de línea.

Por ejemplo, considérese un multiprocesador formado por dos procesadores con planificador EDF. Inicialmente se tiene un conjunto de cuatro tareas de factores de utilización ϵ , $(1 - \epsilon)$, ϵ , y $(1 - \epsilon)$. Estas tareas se asignan empleando el algoritmo

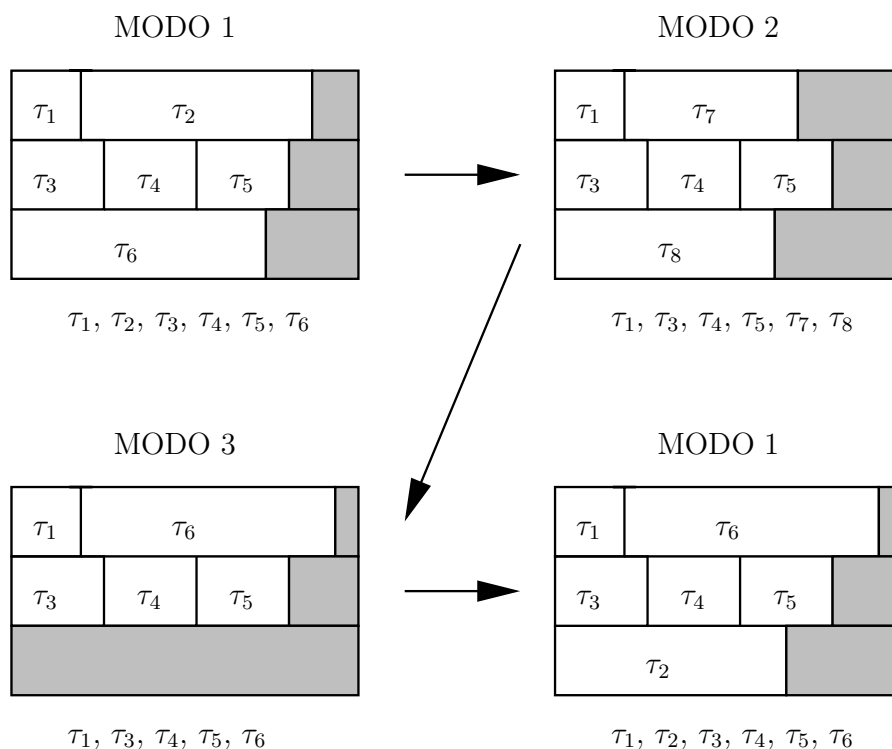


Figura 6.4: Cambios de modo en multiprocesadores.

FF y son planificables. Cada uno de los procesadores tendrá una tarea de factor de utilización ϵ , y una tarea de factor de utilización $(1 - \epsilon)$. Supóngase que se produce un cambio de modo que consiste en la destrucción de las tareas de factor de utilización $(1 - \epsilon)$ y en la creación de una nueva tarea de factor de utilización $(1 - 0'5\epsilon)$. Esta nueva tarea no puede asignarse a ninguno de los procesadores y resulta que el sistema no es planificable en el nuevo modo. Sin embargo, la utilización total del nuevo conjunto de tareas es aproximadamente 1, menor que el límite de utilización $1'5$ para dos procesadores con planificación EDF-FF. Además puede observarse que la situación justo antes de asignar la tarea de factor de utilización $(1 - 0'5\epsilon)$ es la peor posible.

Puesto que los límites de utilización para asignación de tareas fuera de línea no son válidos para asignación de tareas en línea la pregunta que se plantea es:

¿Cómo se puede analizar a priori la planificabilidad del sistema en los distintos modos?

El problema principal reside en que puede haber diferentes asignaciones de tareas a procesadores, incluso para un mismo modo. Por ejemplo, la figura 6.4 muestra dos asignaciones diferentes para el modo 1. Una forma de abordar el problema sería comprobando la planificabilidad de cada procesador, para cada modo y con cada asignación diferente correspondiente al mismo modo. Si la secuencia de cambios de modo fuese conocida a priori (por ejemplo modo 1, modo 2, modo 3, modo 1, ...), y tuviese una periodicidad de periodo bajo (por ejemplo, un periodo de tres modos), el número de asignaciones diferentes para un mismo modo podría limitarse, y analizar

la planificabilidad en cada procesador para cada una de dichas asignaciones. Sin embargo, a medida que crece el número de tareas que se crean y destruyen al pasar de un modo a otro, el número de procesadores, y el número de modos diferentes, incluso en ese caso el número de asignaciones diferentes para un mismo modo puede ser muy elevado. Además, si el periodo de la secuencia de cambios de modo es elevado, o la secuencia de cambios de modo desconocida a priori, el problema resulta intratable.

Una solución al problema consiste en analizar la planificabilidad del sistema de forma global, empleando un límite de utilización multiprocesador. Sin embargo, los límites de utilización propuestos en los capítulos 3 y 4 son válidos únicamente para asignación estática de tareas, es decir, sin creación y destrucción de las mismas. No obstante, independientemente de la creación o destrucción de tareas, el algoritmo de asignación es razonable, pues sólo se fracasa en la asignación cuando no hay espacio suficiente en ningún procesador para albergar una tarea. Por lo tanto, en el caso peor, el límite de utilización no puede ser menor que el límite de utilización fuera de línea empleando asignación RF. Debe tenerse en cuenta que el algoritmo RF proporciona el límite de utilización más bajo de entre todos los algoritmos de asignación razonables.

Así, para planificación EDF y asignación de tareas en línea de acuerdo con un algoritmo razonable cualquiera, el límite de utilización

$$n - (n - 1)\alpha \quad (6.5)$$

es válido, incluso con destrucción y creación de tareas.

De igual forma, para planificación RM y asignación de tareas en línea de acuerdo con un algoritmo razonable cualquiera, el límite de utilización

$$\begin{cases} n_a U_a + n_b U_b - (n - 1)\alpha & \text{si } \alpha \leq U_a \\ n_b U_b - (n_b - 1)\alpha & \text{si } U_a < \alpha < U_b \\ U_b & \text{si } U_b \leq \alpha \end{cases} \quad (6.6)$$

donde

$$\begin{aligned} n_a &= m - \left\lfloor \frac{m-1}{n} \right\rfloor n - 1 \\ n_b &= n - n_a \\ U_a &= \left\lceil \frac{m+n-1}{n} \right\rceil \left(2^{1/\lceil \frac{m+n-1}{n} \rceil} - 1 \right) \\ U_b &= \left\lceil \frac{m+n-1}{n} \right\rceil \left(2^{1/\lceil \frac{m+n-1}{n} \rceil} - 1 \right) \end{aligned}$$

es válido, incluso con destrucción y creación de tareas.

Estos límites de utilización tienen utilidad práctica únicamente cuando β_{EDF} y β_{RM} son mayores que uno, esto es, cuando $\alpha \leq 0'5$ para planificación local EDF, y cuando $\alpha \leq 0'414$ para planificación local RM. En caso contrario los límites de

utilización son similares o idénticos a los de un monoprocesador. Además para el caso $\alpha \leq 0'414$ y planificación local RM la expresión del límite de utilización se simplifica en gran medida y se obtiene el límite de utilización:

$$n_a U_a + n_b U_b - (n - 1)\alpha \approx (m + n - 1)(2^{n/(m+n-1)} - 1) - (n - 1)\alpha$$

Mediante simulación con valores de n comprendidos entre 1 y 1000, y valores de m comprendidos entre $(n + 1)$ y $1000n$ se ha encontrado la siguiente relación.

$$0 \leq \frac{n_a U_a + n_b U_b - (m + n - 1)(2^{n/(m+n-1)} - 1)}{n} \leq 0'0054$$

Por lo tanto, a efectos prácticos el límite de utilización dado por la ecuación (6.6) se puede sustituir por el límite de utilización

$$(m + n - 1)(2^{n/(m+n-1)} - 1) - (n - 1)\alpha$$

cuando $\alpha \leq 0,414$. Es interesante comparar este límite con el proporcionado por la ecuación (6.5) para planificación EDF.

Los límites de utilización propuestos permiten analizar de forma suficiente la planificabilidad en el régimen permanente. Sin embargo no son válidos para el periodo transitorio asociado a los cambios de modo. La dificultad de analizar la planificabilidad del sistema en los periodos de tiempo transitorios mientras se produce el cambio de modo es muy elevada. Si el número de asignaciones posibles diferentes de tareas a procesadores es elevado, en muchos casos un estudio procesador a procesador del cambio de modo como el que se aplica en los sistemas monoprocesador (Pedro y Burns, 1998) es inviable. Este aspecto del problema permanece abierto y será motivo de investigación futura.

Capítulo 7

Conclusiones

El análisis de planificabilidad de sistemas de tiempo real multiprocesador está en una fase inicial. Al contrario de los que sucede con sistemas de tiempo real monoprocesador se conocen muy pocos resultados prácticos. Una de las causas principales es la extraordinaria complejidad del análisis de planificabilidad cuando se pasa de uno a más procesadores. Partiendo de la inviabilidad del tratamiento exacto del problema de planificabilidad multiprocesador, en la tesis se ha atacado el problema de manera aproximada, intentando aprovechar los resultados conocidos para el caso monoprocesador.

Para ello se ha seguido la estrategia de particionado, la cual permite tratar el multiprocesador como un conjunto de monoprocesadores independientes. Esta estrategia requiere el uso de un planificador local para cada procesador y un algoritmo de asignación de tareas a procesadores. Como planificadores locales se han elegido los más representativos en el mundo monoprocesador. Esto es, el planificador basado en prioridades dinámicas EDF, y el basado en prioridades estáticas RM.

Ante la imposibilidad práctica de encontrar un algoritmo de asignación óptimo en la mayor parte de los casos, se ha optado por la elección de algoritmos de asignación heurísticos.

Se han obtenido los límites de utilización multiprocesador para planificación local RM y EDF empleando los principales algoritmos heurísticos de asignación disponibles en la literatura. Una de las características más destacadas de estos límites es la inclusión de un parámetro que tiene en cuenta el tamaño¹ de las tareas. Este parámetro permite en un gran número de casos la obtención de límites de utilización multiprocesador cercanos al ideal.

Por ejemplo, en el caso de asignación de tareas FF y planificación local FF el límite de utilización es $n \ln 2$ para el caso de tareas de factores de utilización muy pequeños. Este límite pasa a ser n en el caso de planificación local EDF, asignación FF y tareas de factores de utilización muy pequeños.

Todos los límites de utilización obtenidos empleando planificación RM dependen del número de tareas, número de procesadores y de un parámetro que tiene en cuenta el tamaño de las tareas. Tanto el número de procesadores como el parámetro

¹El término tamaño se corresponde con el factor de utilización de las tareas en el contexto de la tesis.

de tamaño tienen una gran influencia sobre los límites de utilización. El número de tareas tiene una influencia menor sobre los límites de utilización. Por ejemplo, para la asignación FF se ha obtenido el límite de utilización

$$(n-1)(2^{1/(\beta_{RM}+1)} - 1)\beta_{RM} + (m - \beta_{RM}(n-1))(2^{1/(m-\beta_{RM}(n-1))} - 1)$$

y para asignación FFD el límite

$$(\beta_{RM}n + 1)(2^{1/(\beta_{RM}+1)} - 1)$$

Los límites de utilización para planificación local EDF no dependen del número de tareas, lo mismo que sucede con el límite de utilización para planificación EDF monoprocesador. El número de procesadores y el parámetro de tamaño de las tareas tienen una influencia notable sobre los límites de utilización en este caso. Por ejemplo, para la asignación FF y FFD se ha obtenido el límite de utilización

$$\frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1}$$

Pero la tesis no sólo ha proporcionado los límites de utilización asociados a diferentes algoritmos de asignación. También ha proporcionado el intervalo ajustado en el cual se encuentra el límite de utilización asociado a cualquier algoritmo razonable. Es decir, ha planteado a las limitaciones teóricas de la planificación local en multiprocesadores. Por ejemplo, para el caso de planificación local RM, se tiene que el límite de utilización asociado a cualquier algoritmo razonable de asignación es mayor o igual que

$$(m+n-1)(2^{n/(m+n-1)} - 1) - (n-1)\alpha$$

para valores de α mayores que 0.41, y menor o igual que

$$(\beta_{RM}n + 1)(2^{1/(\beta_{RM}+1)} - 1)$$

Para el caso de planificación local EDF, se tiene que el límite de utilización asociado a cualquier algoritmo razonable de asignación es mayor o igual que

$$n - (n-1)\alpha$$

y menor o igual que

$$\frac{\beta_{EDF}n + 1}{\beta_{EDF} + 1}$$

Se ha observado tanto en el caso de planificación local EDF como RM que la ordenación de tareas en sentido de factores de utilización decrecientes aumenta los límites de utilización hasta el máximo valor posible. Es más, todos los algoritmos de asignación razonables con ordenación previa de tareas en sentido de factores de utilización decrecientes tienen el mismo límite de utilización, igual al máximo

posible.

De entre todos los algoritmos de asignación considerados, los mejores en términos de coste computacional y límite de utilización son el FF y el FFD. El problema del algoritmo FFD es que no puede emplearse en condiciones dinámicas como son los cambios de modo.

El problema de los límites de utilización es que proporcionan condiciones de planificabilidad no exactas. Por ello, resulta interesante subir dichos límites tanto como sea posible, tratando de disminuir dicho pesimismo. Eso se ha realizado en la tesis de acuerdo con dos criterios diferentes. El primer criterio consistió en emplear un parámetro más realista para tener en cuenta el tamaño de las tareas. La mejora obtenida resultó ser importante para tareas con factores de utilización elevados. El segundo criterio consistió en emplear el factor de utilización asociado a todas las tareas en el caso de planificación local RM. No obstante, la mejora obtenida en este caso fue mínima.

Puesto que los límites de utilización se obtienen considerando el peor caso, se han realizado extensos experimentos de simulación, en los cuales se han variado los parámetros más relevantes del sistema, con la intención de obtener una percepción del grado de pesimismo de los límites de utilización obtenidos teóricamente. Estos experimentos han mostrado la elevada planificabilidad de los sistemas multiprocesador que emplean técnicas de particionado. Los resultados de los experimentos han permitido además comparar los diferentes algoritmos de asignación. El resultado fue que los algoritmos FF y FFD son los mejores en términos de planificabilidad y sencillez.

El modelo de tareas básico empleado a lo largo de la tesis ha sido el de tareas periódicas independientes. No obstante, al haber considerado el caso peor, los resultados son también aplicables a tareas esporádicas.

La inclusión de tareas aperiódicas es simple suponiendo que estas se ejecutan a través de algún servidor periódico.

Hay situaciones prácticas en las que el modelo de tareas perfectamente periódicas e independientes es poco realista. Esto a llevado a incluir mejoras del modelo de tareas como la inclusión del *jitter* de activación, el acceso a recursos compartidos y los cambios de modo.

El acceso a recursos compartidos con planificación local EDF fue resuelto desde el punto de vista del análisis de planificabilidad basado en utilidades. No obstante, este problema no ha sido resuelto de forma práctica para planificación RM. Uno de los inconvenientes de la solución proviene de la restricción de que todas las tareas que comparten algún recurso compartido se encuentren en el mismo procesador.

En resumen, el análisis de planificabilidad basado en utilidades de sistemas de tiempo real implementados sobre multiprocesadores ha sido poco estudiado. Salvo el límite de utilización aproximado proporcionado por Oh y Baker (1998), no se ha encontrado ningún trabajo al respecto en la literatura. La tesis ha explorado dicho campo mejorando el resultado de Oh y Baker (1998), tal como se muestra en la figura 7.1, y proporcionando límites de utilización para otros algoritmos de planificación y asignación.

Una parte del trabajo realizado en la tesis ha sido publicado o esta a punto

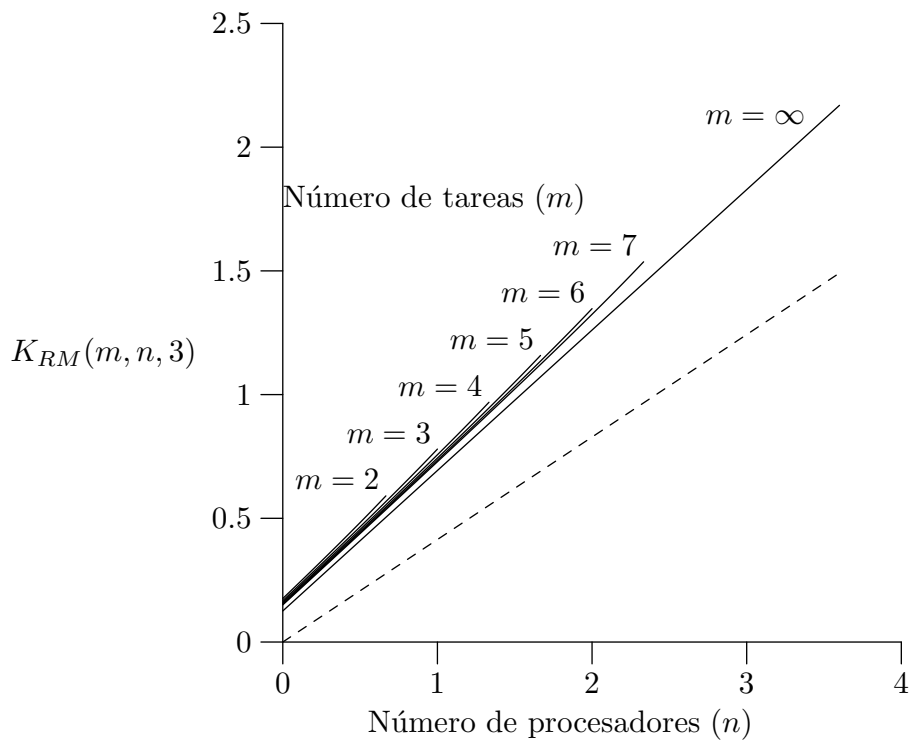


Figura 7.1: Representación del límite de utilización para planificación RM-FF, y tareas de factor de utilización menor o igual que 0'26. La línea discontinua muestra el límite de utilización propuesto por Oh y Baker.

de publicarse. En particular, parte de los resultados obtenidos para planificación local EDF han sido publicados en el Euromicro Workshop on Real-Time Systems 2000, celebrado el pasado mes de Junio de 2000 en Estocolmo (J.M. López y García, 2000b). Asimismo, parte de los resultados obtenidos para planificación local RM serán publicados en la revista Real Time Systems Journal, de la editorial Kluwer en el año 2001.

Apéndice A

Relaciones matemáticas entre números enteros

A continuación se presentan sin demostración algunas relaciones de los números enteros positivos, \mathbb{Z}^+ .

- (i) $x + 1 \leq y \quad \forall x, y \in \mathbb{Z}^+ \mid x < y$
- (ii) $(2^{1/x} - 1)x > (2^{1/y} - 1)y > \ln 2 \quad \forall x, y \in \mathbb{Z}^+ \mid x < y$
- (iii) $(2^{1/(x+1)} - 1)x < (2^{1/(y+1)} - 1)y < \ln 2 \quad \forall x, y \in \mathbb{Z}^+ \mid x < y$
- (iv) $\lfloor x/y + z \rfloor = \lfloor x/y \rfloor + z \quad \forall x, y, z \in \mathbb{Z}^+$
- (v) $\lceil x/y + z \rceil = \lceil x/y \rceil + z \quad \forall x, y, z \in \mathbb{Z}^+$
- (vi) $\left\lfloor \frac{x+y-1}{y} \right\rfloor \geq \frac{x}{y} \quad \forall x, y \in \mathbb{Z}^+$
- (vii) $\lceil x/y \rceil = \left\lfloor \frac{x-1}{y} \right\rfloor + 1 \quad \forall x, y \in \mathbb{Z}^+$

El documento de tesis se refiere a ellas como relación (i) a relación (vii).

Bibliografía

- A. Burchard, J. Liebeherr, Y. O. y Son, S. (1995). New strategies for assigning real-time tasks to multiprocessor systems, *IEEE Transactions on Computers* **44**(12).
- A. Ermedahl, H. H. y Sjodin, M. (1997). Response-time guarantees in atm networks, *Proceedings of the Real-Time Systems Symposium*, pp. 274–284.
- A. Molano, K. J. y Rajkumar, R. (1997). Real-time filesystems. guaranteeing timing constraints for disk accesses in rt-mach, *Proceedings of the Real-Time Systems Symposium*, pp. 155–165.
- B. Sprunt, L. S. y Lehoczky, J. (1989). Aperiodic task scheduling for hard real-time systems, *Real-Time Systems* **1**(1): 27–60.
- Baker, T. (1991). Stack-based scheduling of real-time processes, *Real-Time Systems* **3**(1): 301–324.
- Baruah, S. y Rosier, L. (1990). Algorithms and complexity concerning the preemptive scheduling of periodic, real-time tasks on one processor, *Real-Time Systems* **2**: 301–324.
- Baruah, S. y Wang, F. (1992). On the competitiveness of on-line real-time task scheduling, *Real-Time Systems* **4**: 124–144.
- Bernat, G. y Burns, A. (1999). New results on fixed priority aperiodic servers, *Proceedings of the Real-Time Systems Symposium*.
- Buttazzo, G. (1997). *Hard Real-Time Computing Systems. Predictable Scheduling Algorithms and Applications*, Kluwer Academic Publishers, Boston/Dordrecht/London, chapter 7.
- Canavos, G. (1988). *Probabilidad y Estadística. Aplicaciones y métodos*, Mc Graw Hill, México, chapter 5.
- D. Peng, K. S. y Abdelzaher, T. (1997). Assignment and scheduling communicating periodic tasks in distributed real-time systems, *Transactions on Software Engineering* **23**(12): 745–758.
- Dall, S. y Liu, C. (1978). On a real-time scheduling problem, *Operations Research* **6**(1): 127–140.

- Davari, S. y Dhall, S. (1986a). On a periodic real-time task allocation problem, *Annual international Conference on Systems Sciences*, pp. 133–141.
- Davari, S. y Dhall, S. (1986b). An on line algorithm for real time tasks allocation, *Proceedings of the IEEE Real-Time Systems Symposium*, pp. 194–200.
- Dertouzos, M. (1974). Control robotics: The procedural control of physical processes, *Proceedings of IFIP Congress*, pp. 807–813.
- Dertouzos, M. y Mok, A. (1989). Multiprocessor on-line scheduling of hard-real-time tasks, *Transactions on Software Engineering* **15**(12): 1497–1506.
- F. Wang, K. R. y Stankovich, J. A. (1992). Bound on the performance of heuristic algorithms for multiprocessor scheduling of hard real-time tasks, *Proceedings of the Real-Time Systems Symposium*, pp. 136–145.
- G. Buttazzo, M. S. y Sensini, F. (1995). Value vs. deadline scheduling in overload conditions, *Proceedings of the Real-Time Systems Symposium*, pp. 90–99.
- Gallmeister, B. (1995). *Programming for the Real World: POSIX.4*, O'Really & Associates.
- Garey, M. y Johnson, D. (1979a). *Computers and Intractability*, W.H. Freeman, New York, chapter 6.
- Garey, M. y Johnson, D. (1979b). *Computers and Intractability*, W.H. Freeman, New York, chapter 2.
- Han, C. y Lin, K. (1989). Scheduling parallelizable jobs on multiprocessors, *Proceedings of the Real-Time Systems Symposium*, pp. 59–67.
- I. Ripoll, A. C. y Mok, A. (1996). Improvement in feasibility testing for real-time tasks, *Real-Time Systems* **11**: 19–39.
- J. Lehoczky, L. S. y Ding, Y. (1989). The rate monotonic scheduling algorithm: Exact characterization and average case behavior, *Proceedings of the Real-Time Systems Symposium*, pp. 166–171.
- J.A. Stankovic, M. Spuri, K. R. y Buttazzo, G. (1998). *Deadline Scheduling for Real-Time Systems. EDF and Related Algorithms*, Kluwer Academic Publishers, Boston/Dordrecht/London, pp. 43–43.
- J.M. López, J.L. Díaz, M. G. y García, D. (2000a). Some practical results about fixed priority scheduling and offsets, *Proceedings of the IFAC Workshop on Real-Time Programming*.
- J.M. López, J.L. Díaz, M. G. y García, D. (2000b). Worst-case utilization bound for edf scheduling on real-time multiprocessor systems, *Proceedings of the Euro-micro Workshop on Real-Time Systems*, pp. 25–33.

- K.W. Tindell, A. B. y Wellings, A. (1994). An extendible approach for analyzing fixed priority hard real-time tasks, *Real-Time Systems* **6**(2): 133–151.
- L. Sha, R. R. y Lehoczky, J. (1990). Priority inheritance protocols: An approach to real-time synchronization, *IEEE Transactions on Computers* **39**(9): 1175–1185.
- Lehoczky, J. (1990). Fixed priority scheduling of periodic task sets with arbitrary deadlines, *Proceedings of the Real-Time Systems Symposium*, pp. 201–209.
- Lehoczky, J. (1999). Exploring precedence relations in the schedulability analysis of distributed real-time systems, *Proceedings of the Real-Time Systems Symposium*.
- Leung, J. y Whitehead, J. (1982). On the complexity of fixed-priority scheduling of periodic, real-time tasks, *Performance Evaluation* **2**: 237–250.
- Liu, C. L. y Layland, J. (1973). Scheduling algorithms for multiprogramming in a hard-real-time environment, *Journal of the ACM* **20**(1): 46–61.
- Manabe, J. y Aoyagi, S. (1998). A feasibility decision algorithm for rate monotonic and deadline monotonic scheduling, *Real-Time Systems* **14**: 171–181.
- N. Audsley, A. Burns, M. R. y Burns, A. (1991). Hard real-time scheduling: the deadline-monotonic approach, *Proceedings of the IEEE Workshop on Real-Time Systems and Software*, pp. 133–137.
- Natele, M. y Stankovic, J. (1994). Dynamic end-to-end guarantees in distributed real time systems, *Proceedings of the Real-Time Systems Symposium*, pp. 216–227.
- Oh, D. y Baker, T. (1998). Utilization bounds for n-processor rate monotone scheduling with static processor assignment, *Real-Time Systems* **15**(2): 183–193.
- Oh, Y. y Son, S. (1995). Allocating fixed-priority periodic tasks on multiprocessor systems, *Real-Time Systems* **9**(3): 207–239.
- Pedro, P. y Burns, A. (1998). Schedulability changes for mode changes in flexible real-time systems, *Proceedings of the Euromicro Workshop on Real Time Systems*, pp. 172–179.
- Real, J. y Wellings, A. (1999). Implementing mode changes with shared resources in ada, *Proceedings of the Euromicro Workshop on Real Time Systems*, pp. 86–93.
- S. Lauzac, R. M. y Mossé, D. (1998). An efficient rms admission control and its application to multiprocessor scheduling, *Proceedings of the International Parallel Processing Symposium*, pp. 511–518.
- S. Sáez, J. V. y Crespo, A. (1998). Using exact feasibility tests for allocating real-time tasks in multiprocessor systems, *Proceedings of the 10th Euromicro Workshop on Real-Time Systems*, pp. 53–60.

- Spuri, M. y Buttazzo, G. (1996). Scheduling aperiodic tasks in dynamic priority systems, *Real-Time Systems* **10**(2).
- Stankovic, J. (1988). Misconceptions about real-time computing, *IEEE Computer* **21**(10).
- Tanenbaum, A. (1991). *Sistemas operativos: Diseño e implementación*, Prentice Hall, Londres, chapter 2.
- Tindell, K. y Clark, J. (1994). Holistic schedulability analysis for distributed hard real-time systems, *Microprocessing and Microprogramming* **50**(2): 117–134.
- Xu, J. y Parnas, D. (1990). Scheduling processes with release times, deadlines, precedences, and exclusion, *Transactions on Software Engineering* **16**(3): 360–369.