

TESIS DEFENDIDA POR
JOSÉ LUIS GONZÁLEZ GARCÍA
Y APROBADA POR EL SIGUIENTE COMITÉ

Dr. Andrei Tchernykh

Director del Comité

Dr. José Alberto Fernández Zepeda

Miembro del Comité

Dr. Carlos Alberto Brizuela Rodríguez

Miembro del Comité

Dr. Raúl Rangel Rojo

Miembro del Comité

Dra. Ana Isabel Martínez García

*Coordinadora del programa de posgrado en
Ciencias de la Computación*

Dr. David Hilario Covarrubias Rosales

Director de Estudios de Posgrado

7 de septiembre de 2009

**CENTRO DE INVESTIGACIÓN CIENTÍFICA Y DE EDUCACIÓN SUPERIOR
DE ENSENADA**



**PROGRAMA DE POSGRADO EN CIENCIAS
DE LA COMPUTACIÓN**

**CALENDARIZACIÓN EN LÍNEA PARA GRID JERÁRQUICO DE DOS NIVELES CON
UN ESQUEMA DE ASIGNACIÓN ADMISIBLE**

TESIS

que para cubrir parcialmente los requisitos necesarios para obtener el grado de
MAESTRO EN CIENCIAS

Presenta:

JOSÉ LUIS GONZÁLEZ GARCÍA

Ensenada, Baja California, México, septiembre de 2009.

RESUMEN de la tesis de **José Luis González García**, presentada como requisito parcial para la obtención del grado de MAESTRO EN CIENCIAS en Ciencias de la Computación. Ensenada, Baja California. Septiembre de 2009.

CALENDARIZACIÓN EN LÍNEA PARA GRID JERÁRQUICO DE DOS NIVELES CON UN ESQUEMA DE ASIGNACIÓN ADMISIBLE

Resumen aprobado por:

Dr. Andrei Tchernykh

Director de Tesis

El paradigma de computación en Grid introduce nuevos y difíciles problemas en calendarización y manejo de recursos. Los métodos de calendarización tradicional no satisfacen las necesidades actuales para la administración de recursos en un Grid computacional.

En esta investigación analizamos estrategias de calendarización en Grid computacional de dos niveles utilizando un esquema de admisibilidad. En el primer nivel, el metacalendarizador selecciona una máquina del conjunto de máquinas admisibles para cada tarea según algún criterio. En el segundo nivel, el calendarizador local crea el calendario de ejecución para las tareas asignadas.

El esquema de admisibilidad es un concepto muy simple y fácil de aplicar, basta con evitar la asignación de las tareas pequeñas a máquinas grandes. La consideración original es de restringir el 50% de los procesadores disponibles para una tarea. Mostramos que este porcentaje es el que más se acerca al óptimo si lo generalizamos para todas las estrategias de asignación de recursos. Sin embargo, algunas estrategias trabajan mejor con otro porcentaje. Un mejor ajuste, independiente para cada estrategia, ofrece mejores resultados.

Palabras Clave: Calendarización en Grid, Admisibilidad, Calendarización de dos niveles, Calendarización en línea.

ABSTRACT of the thesis presented by **José Luis González García** as a partial requirement to obtain the MASTER OF SCIENCE degree in Computer Science. Ensenada, Baja California, México. September 2009.

ONLINE SCHEDULING FOR TWO LEVEL HIERARCHICAL GRID WITH AN ADMISSIBLE SELECTION SCHEME

The Grid computing paradigm introduces new and hard problems in scheduling and resource management. Traditional scheduling does not longer meet today's requirements.

We analyze two level hierarchical Grid scheduling strategies using an admissible scheme. At the first level, the meta-scheduler allocates jobs to a suitable machine from the set of admissible machines. At the second level, a local scheduler is applied to each machine independently.

The admissible scheme is a simple concept and easy to implement. This scheme excludes certain machines with many processors from the set of machines available to execute jobs with little parallelism. The original scheme restricts 50% of the processors. We show that this percent (admissible degree) is the most accurate if it is generalized to all scheduling strategies. But, some strategies show better results with a different percent; an adapted degree offers better results.

Dedicatorias

A mis amados padres

Rebeca y José Luis

por su amor, aliento y valioso apoyo incondicionales

A mis hermanos

Leticia, Sandra Patricia, María de los Ángeles,

Rebeca Lourdes, Joel Benjamín y Ernesto Eduardo

por estar siempre presentes y ayudarme cuando lo necesito

A mi novia

Stéphanie

por su amor, compañía y comprensión durante mis estudios

Agradecimientos

Al

Dr. Andrei Tchernykh

por su paciencia, aportaciones y valiosa ayuda

A los miembros del comité de tesis

Dr. José Alberto Fernández Zepeda

Dr. Carlos Alberto Brizuela Rodríguez

Dr. Raúl Rangel Rojo

A mis amigos

David Daniel (Ddaz), Susana, Julio César, Irvin Antero, Laura Emily, Oscar, José Martín,

Patricia, Dietmar

A los amigos de generación

Yair, Liz Ibeth, Daniel, Ricardo, José Santos, Juan Manuel, Nancy Daniela, Gustavo

Francisco, Pablo, Adolfo, Adán, José Luis, Luis Daniel, Ma. Guadalupe, Alma Lilia,

Fernando, Ismael, Raymundo, Raúl Gerardo, Jessica y Ariel Arturo

Al

Centro de Investigación Científica y de Educación Superior de Ensenada

Al

Consejo Nacional de Ciencia y Tecnología

por su valiosa ayuda económica

CONTENIDO

Página

Resumen	i
Resumen en inglés	ii
Dedicatorias	iii
Agradecimientos	iv
Contenido	v
Lista de Figuras	viii
Lista de Tablas	xxiii
Capítulo I. Introducción	1
<i>I. 1. Calendarización en Grid computacional</i>	<i>1</i>
<i>I. 2. Planteamiento del problema</i>	<i>3</i>
<i>I. 3. Justificación</i>	<i>4</i>
<i>I. 4. Objetivos de la investigación</i>	<i>5</i>
I. 4. 1 Objetivos generales.....	5
I. 4. 2 Objetivos específicos.....	6
<i>I. 5. Importancia de la investigación</i>	<i>6</i>
I. 5. 1 Limitaciones y suposiciones fundamentales.....	7
I. 5. 2 Contribución al conocimiento.....	7
<i>I. 6. Organización de la tesis</i>	<i>8</i>
Capítulo II. Marco teórico	9
<i>II. 1. Arquitecturas de Grid computacional</i>	<i>9</i>
II. 1. 1 Centralizada.....	10
II. 1. 2 Descentralizada.....	10
<i>II. 1. 2. 1. Comunicación directa</i>	<i>11</i>
<i>II. 1. 2. 2. Comunicación a través de una pila central de tareas</i>	<i>12</i>
II. 1. 3 Jerárquica.....	13
<i>II. 2. Retos de algoritmos en calendarización</i>	<i>14</i>
II. 2. 1 Heterogeneidad y autonomía.....	16
II. 2. 2 Desempeño dinámico.....	17
II. 2. 3 Separación entre la selección de recursos y el cálculo de datos.....	17
<i>II. 3. Calendarización basada en lista</i>	<i>18</i>
<i>II. 4. Calendarización en línea para Grid de un nivel</i>	<i>21</i>
<i>II. 5. Calendarización en línea en Grid jerárquico de dos niveles</i>	<i>22</i>
Capítulo III. Definición formal del problema	25
<i>III. 1. Definición del problema de calendarización</i>	<i>25</i>
<i>III. 2. Esquema de admisibilidad</i>	<i>27</i>
<i>III. 3. Modelo jerárquico de dos niveles con admisibilidad</i>	<i>28</i>

CONTENIDO (continuación)

	Página
<i>III. 4. Estrategias de calendarización</i>	29
III. 4. 1 Asignación de tareas a máquinas	30
III. 4. 2 Calendarización local	33
Capítulo IV. Carga de trabajo	35
<i>IV. 1. Cargas de trabajo reales y sintéticas</i>	35
<i>IV. 2. Formatos disponibles</i>	37
IV. 2. 1 Open Trace Format (OTF)	37
IV. 2. 2 Standard Workload Format (SWF)	38
IV. 2. 3 Grid Workload Format (GWF)	38
<i>IV. 3. Cargas de trabajo para Grid computacional</i>	39
IV. 3. 1 Carga de trabajo semi-sintética	39
IV. 3. 1. 1. Normalización de la zona horaria	41
IV. 3. 1. 2. Selección de tareas admisibles	42
IV. 3. 1. 3. Normalización de números de usuario	43
IV. 3. 1. 4. Filtrado de tareas no válidas	43
IV. 3. 1. 5. Generación de la carga de trabajo semi-sintética.....	44
IV. 3. 1. 6. Agregación de nuevas tareas	44
Capítulo V. Experimentación	45
<i>V. 1. Métricas para el análisis</i>	45
V. 1. 1 De acuerdo al criterio del sistema.....	45
V. 1. 2 De acuerdo al criterio del usuario	46
V. 1. 3 De acuerdo al criterio del algoritmo	48
<i>V. 2. Carga de trabajo utilizada</i>	48
V. 2. 1 Estadísticas de la carga de trabajo	52
<i>V. 3. Definición del experimento</i>	65
V. 3. 1 Configuración del Grid.....	65
V. 3. 2 Carga de trabajo.....	67
V. 3. 3 Parámetros de los experimentos	67
<i>V. 4. Análisis de resultados</i>	67
V. 4. 1 Estrategia <i>Random</i>	69
V. 4. 2 Estrategia <i>Min_Lp</i>	70
V. 4. 3 Estrategia <i>Min_PL</i>	71
V. 4. 4 Estrategia <i>Min_LBal</i>	72
V. 4. 5 Estrategia <i>Min_LB</i>	73
V. 4. 6 Estrategia <i>Min_CT</i>	74
V. 4. 7 Estrategia <i>Min_SWCT</i>	75
V. 4. 8 Estrategia <i>Min_WT</i>	76
V. 4. 9 Estrategia <i>Min_WWT</i>	77
V. 4. 10 Estrategia <i>Min_U</i>	79

CONTENIDO (continuación)

	Página
V. 4. 11 Estrategia <i>Min_ST</i>	80
V. 4. 12 Estrategia <i>Min_TA</i>	81
V. 4. 13 Estrategia <i>Min_WTA</i>	82
V. 4. 14 Estrategia <i>Min_WWoTA</i>	83
V. 4. 15 Conclusiones.....	84
Capítulo VI. Conclusiones	87
<i>VI. 1. Conclusiones finales</i>	87
<i>VI. 2. Trabajo futuro</i>	90
Referencias.....	92
Apéndice.....	97

LISTA DE FIGURAS

Figura	Página
1	Calendarizador centralizado.....10
2	Calendarización descentralizada con comunicación directa.....12
3	Calendarización descentralizada con comunicación a través de una pila central de tareas.....13
4	Calendarización de estructura jerárquica.14
5	Un problema de los algoritmos de calendarización basados en lista; a) calendario generado para una mala instancia del problema, b) calendario óptimo para la misma instancia.....23
6	Esquema de admisibilidad.24
7	Asignación admisible con factor a28
8	Modelo de calendarización jerárquica de dos niveles.....29
9	Algoritmo del proceso de generación de cargas de trabajo semi-sintéticas.....40
10	Número promedio de tareas sometidas por hora. Registro SDSC-SP2-1998-3.1-cln.swf, GMT-8.....41
11	Número promedio de tareas sometidas por hora. Registro KTH-SP2-1996-2.swf, GMT+1.....41
12	Número promedio de tareas sometidas por hora. Carga de trabajo semi-sintética resultante, GMT-8.42
13	Número promedio de tareas sometidas por hora del día.....54
14	Número promedio de tareas sometidas por día de la semana.54
15	Número promedio de tareas sometidas por día del mes.54
16	Número de tareas sometidas por hora de la carga de trabajo.....54
17	Cantidad promedio de recursos consumidos por hora del día.55
18	Cantidad promedio de recursos consumidos por día de la semana.....55
19	Cantidad promedio de recursos consumidos por día del mes.55
20	Número de tareas sometidas por usuario.56

LISTA DE FIGURAS (continuación)

Figura	Página
21	Número de tareas sometidas por usuario, vista detallada de los usuarios con menos de 5,000 tareas.56
22	Número ordenado de tareas sometidas por usuario.56
23	Número ordenado de tareas sometidas por usuario, vista detallada de los 100 usuarios con más tareas.56
24	Número de tareas sometidas por tamaño.57
25	Número de tareas sometidas por tamaño, vista detallada de los tamaños con menos de 50,000 tareas.57
26	Número de tareas sometidas por tiempo de ejecución, vista detallada de los tiempos de ejecución con menos de 50,000 tareas.58
27	Número de tareas sometidas por tiempo de ejecución, vista detallada de los tiempos de ejecución con menos de 10,000 tareas.58
28	Número de tareas sometidas por tiempo solicitado, vista detallada de los tiempos solicitados con menos de 50,000 tareas.59
29	Número de tareas sometidas por tiempo solicitado, vista detallada de los tiempos solicitados con menos de 10,000 tareas.59
30	Dispersión de tareas sometidas por tamaño y tiempo de ejecución.60
31	Dispersión de tareas sometidas por tamaño y tiempo solicitado.60
32	Dispersión de tareas sometidas por tamaño y relación de recursos (solicitados/consumidos).....61
33	Dispersión de tareas sometidas por tamaño y relación de recursos (solicitados/consumidos), vista detallada de relaciones menores a 500,000.61
34	Número de cambios de tarea sometida por usuario, variación del 30%.62
35	Número de cambios de tarea sometida por usuario, variación del 30%, vista detallada de usuarios con menos de 1,000 cambios de tarea.....62
36	Cantidad de recursos consumidos por tiempo de ejecución.63

LISTA DE FIGURAS (continuación)

Figura	Página
37	Cantidad de recursos consumidos por tiempo de ejecución, vista detallada de los tiempos de ejecución con recursos consumidos menores a 300,000.....63
38	Cantidad de recursos solicitados por tiempo solicitado.....63
39	Cantidad de recursos solicitados por tiempo solicitado, vista detallada de los tiempos solicitados con recursos solicitados menores a 300,000.....63
40	Cantidad de recursos consumidos por tamaño.....64
41	Cantidad de recursos solicitados por tamaño.....64
42	Número de tareas sometidas con estatus de error por tipo.....64
43	Número de tareas sometidas con estatus de error por usuario.....64
44	Cociente de competitividad de las estrategias de asignación sin el esquema de admisibilidad.....86
45	Cociente de competitividad de las estrategias de asignación con el esquema de admisibilidad.....86
46	Cociente de competitividad, estrategia <i>Random</i> con admisibilidad.....97
47	Longitud del calendario, estrategia <i>Random</i> con admisibilidad.....97
48	Tiempo promedio de permanencia ponderado, estrategia <i>Random</i> con admisibilidad.....97
49	Tiempo promedio de permanencia ponderado por trabajo, estrategia <i>Random</i> con admisibilidad.....97
50	Tiempo promedio de permanencia, estrategia <i>Random</i> con admisibilidad.....98
51	Tiempo promedio de espera, estrategia <i>Random</i> con admisibilidad.....98
52	Tiempo promedio de espera ponderado, estrategia <i>Random</i> con admisibilidad.....98
53	Cociente de respuesta promedio acotado, estrategia <i>Random</i> con admisibilidad.....98
54	Cociente de respuesta promedio, estrategia <i>Random</i> con admisibilidad.....98

LISTA DE FIGURAS (continuación)

Figura	Página
55	Tiempo de respuesta promedio, estrategia <i>Random</i> con admisibilidad.....98
56	Cociente de respuesta del sistema, estrategia <i>Random</i> con admisibilidad.....99
57	Rendimiento de procesamiento, estrategia <i>Random</i> con admisibilidad.....99
58	Recursos no utilizados, estrategia <i>Random</i> con admisibilidad.99
59	Trabajo total realizado, estrategia <i>Random</i> con admisibilidad.99
60	Utilización del sistema, estrategia <i>Random</i> con admisibilidad.....99
61	Cociente de competitividad, estrategia <i>Min_Lp</i> con admisibilidad.100
62	Longitud del calendario, estrategia <i>Min_Lp</i> con admisibilidad.....100
63	Tiempo promedio de permanencia ponderado, estrategia <i>Min_Lp</i> con admisibilidad.....100
64	Tiempo promedio de permanencia ponderado por trabajo, estrategia <i>Min_Lp</i> con admisibilidad.100
65	Tiempo promedio de permanencia, estrategia <i>Min_Lp</i> con admisibilidad.101
66	Tiempo promedio de espera, estrategia <i>Min_Lp</i> con admisibilidad.101
67	Tiempo promedio de espera ponderado, estrategia <i>Min_Lp</i> con admisibilidad.....101
68	Cociente de respuesta promedio acotado, estrategia <i>Min_Lp</i> con admisibilidad.....101
69	Cociente de respuesta promedio, estrategia <i>Min_Lp</i> con admisibilidad.....101
70	Tiempo de respuesta promedio, estrategia <i>Min_Lp</i> con admisibilidad.....101
71	Cociente de respuesta del sistema, estrategia <i>Min_Lp</i> con admisibilidad.102
72	Rendimiento de procesamiento, estrategia <i>Min_Lp</i> con admisibilidad.102
73	Recursos no utilizados, estrategia <i>Min_Lp</i> con admisibilidad.....102
74	Trabajo total realizado, estrategia <i>Min_Lp</i> con admisibilidad.....102
75	Utilización del sistema, estrategia <i>Min_Lp</i> con admisibilidad.102
76	Cociente de competitividad, estrategia <i>Min_PL</i> con admisibilidad.....103

LISTA DE FIGURAS (continuación)

Figura	Página
77	Longitud del calendario, estrategia <i>Min_PL</i> con admisibilidad.103
78	Tiempo promedio de permanencia ponderado, estrategia <i>Min_PL</i> con admisibilidad.....103
79	Tiempo promedio de permanencia ponderado por trabajo, estrategia <i>Min_PL</i> con admisibilidad.103
80	Tiempo promedio de permanencia, estrategia <i>Min_PL</i> con admisibilidad.....104
81	Tiempo promedio de espera, estrategia <i>Min_PL</i> con admisibilidad.....104
82	Tiempo promedio de espera ponderado, estrategia <i>Min_PL</i> con admisibilidad.....104
83	Cociente de respuesta promedio acotado, estrategia <i>Min_PL</i> con admisibilidad.....104
84	Cociente de respuesta promedio, estrategia <i>Min_PL</i> con admisibilidad.104
85	Tiempo de respuesta promedio, estrategia <i>Min_PL</i> con admisibilidad.104
86	Cociente de respuesta del sistema, estrategia <i>Min_PL</i> con admisibilidad.....105
87	Rendimiento de procesamiento, estrategia <i>Min_PL</i> con admisibilidad.....105
88	Recursos no utilizados, estrategia <i>Min_PL</i> con admisibilidad.105
89	Trabajo total realizado, estrategia <i>Min_PL</i> con admisibilidad.105
90	Utilización del sistema, estrategia <i>Min_PL</i> con admisibilidad.....105
91	Cociente de competitividad, estrategia <i>Min_LBal</i> con admisibilidad.106
92	Longitud del calendario, estrategia <i>Min_LBal</i> con admisibilidad.106
93	Tiempo promedio de permanencia ponderado, estrategia <i>Min_LBal</i> con admisibilidad.....106
94	Tiempo promedio de permanencia ponderado por trabajo, estrategia <i>Min_LBal</i> con admisibilidad.....106
95	Tiempo promedio de permanencia, estrategia <i>Min_LBal</i> con admisibilidad.107
96	Tiempo promedio de espera, estrategia <i>Min_LBal</i> con admisibilidad.....107

LISTA DE FIGURAS (continuación)

Figura	Página
97	Tiempo promedio de espera ponderado, estrategia <i>Min_LBal</i> con admisibilidad.....107
98	Cociente de respuesta promedio acotado, estrategia <i>Min_LBal</i> con admisibilidad.....107
99	Cociente de respuesta promedio, estrategia <i>Min_LBal</i> con admisibilidad.107
100	Tiempo de respuesta promedio, estrategia <i>Min_LBal</i> con admisibilidad.107
101	Cociente de respuesta del sistema, estrategia <i>Min_LBal</i> con admisibilidad.....108
102	Rendimiento de procesamiento, estrategia <i>Min_LBal</i> con admisibilidad.....108
103	Recursos no utilizados, estrategia <i>Min_LBal</i> con admisibilidad.108
104	Trabajo total realizado, estrategia <i>Min_LBal</i> con admisibilidad.108
105	Utilización del sistema, estrategia <i>Min_LBal</i> con admisibilidad.....108
106	Cociente de competitividad, estrategia <i>Min_LB</i> con admisibilidad.....109
107	Longitud del calendario, estrategia <i>Min_LB</i> con admisibilidad.109
108	Tiempo promedio de permanencia ponderado, estrategia <i>Min_LB</i> con admisibilidad.....109
109	Tiempo promedio de permanencia ponderado por trabajo, estrategia <i>Min_LB</i> con admisibilidad.....109
110	Tiempo promedio de permanencia, estrategia <i>Min_LB</i> con admisibilidad.....110
111	Tiempo promedio de espera, estrategia <i>Min_LB</i> con admisibilidad.....110
112	Tiempo promedio de espera ponderado, estrategia <i>Min_LB</i> con admisibilidad.....110
113	Cociente de respuesta promedio acotado, estrategia <i>Min_LB</i> con admisibilidad.....110
114	Cociente de respuesta promedio, estrategia <i>Min_LB</i> con admisibilidad.110
115	Tiempo de respuesta promedio, estrategia <i>Min_LB</i> con admisibilidad.110
116	Cociente de respuesta del sistema, estrategia <i>Min_LB</i> con admisibilidad.....111

LISTA DE FIGURAS (continuación)

Figura	Página
117 Rendimiento de procesamiento, estrategia <i>Min_LB</i> con admisibilidad.....	111
118 Recursos no utilizados, estrategia <i>Min_LB</i> con admisibilidad.	111
119 Trabajo total realizado, estrategia <i>Min_LB</i> con admisibilidad.	111
120 Utilización del sistema, estrategia <i>Min_LB</i> con admisibilidad.....	111
121 Cociente de competitividad, estrategia <i>Min_CT</i> con admisibilidad.	112
122 Longitud del calendario, estrategia <i>Min_CT</i> con admisibilidad.	112
123 Tiempo promedio de permanencia ponderado, estrategia <i>Min_CT</i> con admisibilidad.....	112
124 Tiempo promedio de permanencia ponderado por trabajo, estrategia <i>Min_CT</i> con admisibilidad.....	112
125 Tiempo promedio de permanencia, estrategia <i>Min_CT</i> con admisibilidad.	113
126 Tiempo promedio de espera, estrategia <i>Min_CT</i> con admisibilidad.....	113
127 Tiempo promedio de espera ponderado, estrategia <i>Min_CT</i> con admisibilidad.....	113
128 Cociente de respuesta promedio acotado, estrategia <i>Min_CT</i> con admisibilidad.....	113
129 Cociente de respuesta promedio, estrategia <i>Min_CT</i> con admisibilidad.	113
130 Tiempo de respuesta promedio, estrategia <i>Min_CT</i> con admisibilidad.	113
131 Cociente de respuesta del sistema, estrategia <i>Min_CT</i> con admisibilidad.....	114
132 Rendimiento de procesamiento, estrategia <i>Min_CT</i> con admisibilidad.....	114
133 Recursos no utilizados, estrategia <i>Min_CT</i> con admisibilidad.	114
134 Trabajo total realizado, estrategia <i>Min_CT</i> con admisibilidad.	114
135 Utilización del sistema, estrategia <i>Min_CT</i> con admisibilidad.....	114
136 Cociente de competitividad, estrategia <i>Min_SWCT</i> con admisibilidad.	115
137 Longitud del calendario, estrategia <i>Min_SWCT</i> con admisibilidad.	115

LISTA DE FIGURAS (continuación)

Figura	Página
138	Tiempo promedio de permanencia ponderado, estrategia <i>Min_SWCT</i> con admisibilidad.....115
139	Tiempo promedio de permanencia ponderado por trabajo, estrategia <i>Min_SWCT</i> con admisibilidad.115
140	Tiempo promedio de permanencia, estrategia <i>Min_SWCT</i> con admisibilidad.....116
141	Tiempo promedio de espera, estrategia <i>Min_SWCT</i> con admisibilidad.116
142	Tiempo promedio de espera ponderado, estrategia <i>Min_SWCT</i> con admisibilidad.....116
143	Cociente de respuesta promedio acotado, estrategia <i>Min_SWCT</i> con admisibilidad.....116
144	Cociente de respuesta promedio, estrategia <i>Min_SWCT</i> con admisibilidad.....116
145	Tiempo de respuesta promedio, estrategia <i>Min_SWCT</i> con admisibilidad.....116
146	Cociente de respuesta del sistema, estrategia <i>Min_SWCT</i> con admisibilidad.....117
147	Rendimiento de procesamiento, estrategia <i>Min_SWCT</i> con admisibilidad.117
148	Recursos no utilizados, estrategia <i>Min_SWCT</i> con admisibilidad.....117
149	Trabajo total realizado, estrategia <i>Min_SWCT</i> con admisibilidad.....117
150	Utilización del sistema, estrategia <i>Min_SWCT</i> con admisibilidad.117
151	Cociente de competitividad, estrategia <i>Min_WT</i> con admisibilidad.....118
152	Longitud del calendario, estrategia <i>Min_WT</i> con admisibilidad.118
153	Tiempo promedio de permanencia ponderado, estrategia <i>Min_WT</i> con admisibilidad.....118
154	Tiempo promedio de permanencia ponderado por trabajo, estrategia <i>Min_WT</i> con admisibilidad.....118
155	Tiempo promedio de permanencia, estrategia <i>Min_WT</i> con admisibilidad.....119

LISTA DE FIGURAS (continuación)

Figura	Página
156	Tiempo promedio de espera, estrategia <i>Min_WT</i> con admisibilidad.119
157	Tiempo promedio de espera ponderado, estrategia <i>Min_WT</i> con admisibilidad.119
158	Cociente de respuesta promedio acotado, estrategia <i>Min_WT</i> con admisibilidad.119
159	Cociente de respuesta promedio, estrategia <i>Min_WT</i> con admisibilidad.119
160	Tiempo de respuesta promedio, estrategia <i>Min_WT</i> con admisibilidad.119
161	Cociente de respuesta del sistema, estrategia <i>Min_WT</i> con admisibilidad.120
162	Rendimiento de procesamiento, estrategia <i>Min_WT</i> con admisibilidad.120
163	Recursos no utilizados, estrategia <i>Min_WT</i> con admisibilidad.120
164	Trabajo total realizado, estrategia <i>Min_WT</i> con admisibilidad.120
165	Utilización del sistema, estrategia <i>Min_WT</i> con admisibilidad.120
166	Cociente de competitividad, estrategia <i>Min_WWT</i> con admisibilidad.121
167	Longitud del calendario, estrategia <i>Min_WWT</i> con admisibilidad.121
168	Tiempo promedio de permanencia ponderado, estrategia <i>Min_WWT</i> con admisibilidad.121
169	Tiempo promedio de permanencia ponderado por trabajo, estrategia <i>Min_WWT</i> con admisibilidad.121
170	Tiempo promedio de permanencia, estrategia <i>Min_WWT</i> con admisibilidad.122
171	Tiempo promedio de espera, estrategia <i>Min_WWT</i> con admisibilidad.122
172	Tiempo promedio de espera ponderado, estrategia <i>Min_WWT</i> con admisibilidad.122
173	Cociente de respuesta promedio acotado, estrategia <i>Min_WWT</i> con admisibilidad.122
174	Cociente de respuesta promedio, estrategia <i>Min_WWT</i> con admisibilidad.122

LISTA DE FIGURAS (continuación)

Figura	Página
175	Tiempo de respuesta promedio, estrategia <i>Min_WWT</i> con admisibilidad.....122
176	Cociente de respuesta del sistema, estrategia <i>Min_WWT</i> con admisibilidad.....123
177	Rendimiento de procesamiento, estrategia <i>Min_WWT</i> con admisibilidad.....123
178	Recursos no utilizados, estrategia <i>Min_WWT</i> con admisibilidad.123
179	Trabajo total realizado, estrategia <i>Min_WWT</i> con admisibilidad.123
180	Utilización del sistema, estrategia <i>Min_WWT</i> con admisibilidad.....123
181	Cociente de competitividad, estrategia <i>Min_U</i> con admisibilidad.124
182	Longitud del calendario, estrategia <i>Min_U</i> con admisibilidad.124
183	Tiempo promedio de permanencia ponderado, estrategia <i>Min_U</i> con admisibilidad.....124
184	Tiempo promedio de permanencia ponderado por trabajo, estrategia <i>Min_U</i> con admisibilidad.....124
185	Tiempo promedio de permanencia, estrategia <i>Min_U</i> con admisibilidad.125
186	Tiempo promedio de espera, estrategia <i>Min_U</i> con admisibilidad.....125
187	Tiempo promedio de espera ponderado, estrategia <i>Min_U</i> con admisibilidad.....125
188	Cociente de respuesta promedio acotado, estrategia <i>Min_U</i> con admisibilidad.....125
189	Cociente de respuesta promedio, estrategia <i>Min_U</i> con admisibilidad.125
190	Tiempo de respuesta promedio, estrategia <i>Min_U</i> con admisibilidad.125
191	Cociente de respuesta del sistema, estrategia <i>Min_U</i> con admisibilidad.....126
192	Rendimiento de procesamiento, estrategia <i>Min_U</i> con admisibilidad.....126
193	Recursos no utilizados, estrategia <i>Min_U</i> con admisibilidad.126
194	Trabajo total realizado, estrategia <i>Min_U</i> con admisibilidad.126
195	Utilización del sistema, estrategia <i>Min_U</i> con admisibilidad.....126
196	Cociente de competitividad, estrategia <i>Min_ST</i> con admisibilidad.127

LISTA DE FIGURAS (continuación)

Figura	Página
197	Longitud del calendario, estrategia <i>Min_ST</i> con admisibilidad.....127
198	Tiempo promedio de permanencia ponderado, estrategia <i>Min_ST</i> con admisibilidad.....127
199	Tiempo promedio de permanencia ponderado por trabajo, estrategia <i>Min_ST</i> con admisibilidad.127
200	Tiempo promedio de permanencia, estrategia <i>Min_ST</i> con admisibilidad.128
201	Tiempo promedio de espera, estrategia <i>Min_ST</i> con admisibilidad.128
202	Tiempo promedio de espera ponderado, estrategia <i>Min_ST</i> con admisibilidad.....128
203	Cociente de respuesta promedio acotado, estrategia <i>Min_ST</i> con admisibilidad.....128
204	Cociente de respuesta promedio, estrategia <i>Min_ST</i> con admisibilidad.....128
205	Tiempo de respuesta promedio, estrategia <i>Min_ST</i> con admisibilidad.....128
206	Cociente de respuesta del sistema, estrategia <i>Min_ST</i> con admisibilidad.129
207	Rendimiento de procesamiento, estrategia <i>Min_ST</i> con admisibilidad.129
208	Recursos no utilizados, estrategia <i>Min_ST</i> con admisibilidad.....129
209	Trabajo total realizado, estrategia <i>Min_ST</i> con admisibilidad.....129
210	Utilización del sistema, estrategia <i>Min_ST</i> con admisibilidad.129
211	Cociente de competitividad, estrategia <i>Min_TA</i> con admisibilidad.....130
212	Longitud del calendario, estrategia <i>Min_TA</i> con admisibilidad.130
213	Tiempo promedio de permanencia ponderado, estrategia <i>Min_TA</i> con admisibilidad.....130
214	Tiempo promedio de permanencia ponderado por trabajo, estrategia <i>Min_TA</i> con admisibilidad.130
215	Tiempo promedio de permanencia, estrategia <i>Min_TA</i> con admisibilidad.....131
216	Tiempo promedio de espera, estrategia <i>Min_TA</i> con admisibilidad.....131

LISTA DE FIGURAS (continuación)

Figura	Página
217	Tiempo promedio de espera ponderado, estrategia <i>Min_TA</i> con admisibilidad.....131
218	Cociente de respuesta promedio acotado, estrategia <i>Min_TA</i> con admisibilidad.....131
219	Cociente de respuesta promedio, estrategia <i>Min_TA</i> con admisibilidad.131
220	Tiempo de respuesta promedio, estrategia <i>Min_TA</i> con admisibilidad.131
221	Cociente de respuesta del sistema, estrategia <i>Min_TA</i> con admisibilidad.....132
222	Rendimiento de procesamiento, estrategia <i>Min_TA</i> con admisibilidad.....132
223	Recursos no utilizados, estrategia <i>Min_TA</i> con admisibilidad.132
224	Trabajo total realizado, estrategia <i>Min_TA</i> con admisibilidad.132
225	Utilización del sistema, estrategia <i>Min_TA</i> con admisibilidad.....132
226	Cociente de competitividad, estrategia <i>Min_WTA</i> con admisibilidad.133
227	Longitud del calendario, estrategia <i>Min_WTA</i> con admisibilidad.133
228	Tiempo promedio de permanencia ponderado, estrategia <i>Min_WTA</i> con admisibilidad.....133
229	Tiempo promedio de permanencia ponderado por trabajo, estrategia <i>Min_WTA</i> con admisibilidad.....133
230	Tiempo promedio de permanencia, estrategia <i>Min_WTA</i> con admisibilidad.134
231	Tiempo promedio de espera, estrategia <i>Min_WTA</i> con admisibilidad.134
232	Tiempo promedio de espera ponderado, estrategia <i>Min_WTA</i> con admisibilidad.....134
233	Cociente de respuesta promedio acotado, estrategia <i>Min_WTA</i> con admisibilidad.....134
234	Cociente de respuesta promedio, estrategia <i>Min_WTA</i> con admisibilidad.134
235	Tiempo de respuesta promedio, estrategia <i>Min_WTA</i> con admisibilidad.....134
236	Cociente de respuesta del sistema, estrategia <i>Min_WTA</i> con admisibilidad.135

LISTA DE FIGURAS (continuación)

Figura	Página
237 Rendimiento de procesamiento, estrategia <i>Min_WTA</i> con admisibilidad.	135
238 Recursos no utilizados, estrategia <i>Min_WTA</i> con admisibilidad.	135
239 Trabajo total realizado, estrategia <i>Min_WTA</i> con admisibilidad.	135
240 Utilización del sistema, estrategia <i>Min_WTA</i> con admisibilidad.	135
241 Cociente de competitividad, estrategia <i>Min_WWoTA</i> con admisibilidad.	136
242 Longitud del calendario, estrategia <i>Min_WWoTA</i> con admisibilidad.	136
243 Tiempo promedio de permanencia ponderado, estrategia <i>Min_WWoTA</i> con admisibilidad.	136
244 Tiempo promedio de permanencia ponderado por trabajo, estrategia <i>Min_WWoTA</i> con admisibilidad.	136
245 Tiempo promedio de permanencia, estrategia <i>Min_WWoTA</i> con admisibilidad.	137
246 Tiempo promedio de espera, estrategia <i>Min_WWoTA</i> con admisibilidad.	137
247 Tiempo promedio de espera ponderado, estrategia <i>Min_WWoTA</i> con admisibilidad.	137
248 Cociente de respuesta promedio acotado, estrategia <i>Min_WWoTA</i> con admisibilidad.	137
249 Cociente de respuesta promedio, estrategia <i>Min_WWoTA</i> con admisibilidad.	137
250 Tiempo de respuesta promedio, estrategia <i>Min_WWoTA</i> con admisibilidad.	137
251 Cociente de respuesta del sistema, estrategia <i>Min_WWoTA</i> con admisibilidad.	138
252 Rendimiento de procesamiento, estrategia <i>Min_WWoTA</i> con admisibilidad.	138
253 Recursos no utilizados, estrategia <i>Min_WWoTA</i> con admisibilidad.	138
254 Trabajo total realizado, estrategia <i>Min_WWoTA</i> con admisibilidad.	138

LISTA DE FIGURAS (continuación)

Figura	Página
255	Utilización del sistema, estrategia <i>Min_WWoTA</i> con admisibilidad.138
256	Porcentaje de mejora de los algoritmos, en el cociente de competitividad, al utilizar el esquema de admisibilidad.....139
257	Porcentaje de mejora de los algoritmos, en la longitud del calendario, al utilizar el esquema de admisibilidad.....139
258	Porcentaje de mejora de los algoritmos, en la utilización del sistema, al utilizar el esquema de admisibilidad.....139
259	Porcentaje de mejora de los algoritmos, en los recursos no utilizados, al utilizar el esquema de admisibilidad.....139
260	Porcentaje de mejora de los algoritmos, en el rendimiento de procesamiento, al utilizar el esquema de admisibilidad.....140
261	Porcentaje de mejora de los algoritmos, en el tiempo promedio de permanencia, al utilizar el esquema de admisibilidad.140
262	Porcentaje de mejora de los algoritmos, en el tiempo promedio de permanencia ponderado, al utilizar el esquema de admisibilidad.....140
263	Porcentaje de mejora de los algoritmos, en el tiempo promedio de permanencia ponderado por trabajo, al utilizar el esquema de admisibilidad.....140
264	Porcentaje de mejora de los algoritmos, en el tiempo promedio de espera, al utilizar el esquema de admisibilidad.....141
265	Porcentaje de mejora de los algoritmos, en el tiempo promedio de espera ponderado, al utilizar el esquema de admisibilidad.141
266	Porcentaje de mejora de los algoritmos, en el tiempo de respuesta promedio, al utilizar el esquema de admisibilidad.....141
267	Porcentaje de mejora de los algoritmos, en el cociente de respuesta promedio, al utilizar el esquema de admisibilidad.....141

LISTA DE FIGURAS (continuación)

Figura		Página
268	Porcentaje de mejora de los algoritmos, en el cociente de respuesta promedio acotado, al utilizar el esquema de admisibilidad.	142
269	Porcentaje de mejora de los algoritmos, en el cociente de respuesta del sistema, al utilizar el esquema de admisibilidad.	142

LISTA DE TABLAS

Tabla	Página
I Evolución de los algoritmos de calendarización con los sistemas computacionales paralelos y distribuidos.	15
II Ventajas y desventajas de las cargas de trabajo reales y sintéticas.....	36
III Registro del Cornell Theory Center (CTC).	49
IV Registro de The Advanced School for Computing and Imaging (DAS-2 system).....	50
V Registro del High Performance Computing Center North, Sweden.	51
VI Registro del Swedish Royal Institute of Technology (KTH).....	51
VII Registro de Los Alamos National Lab (LANL).	52
VIII Configuración del Grid utilizado para los experimentos.	65
IX Parámetros de los experimentos.....	66
X Resultados experimentales al aplicar el esquema de admisibilidad a las diferentes estrategias de calendarización. En blanco se muestra la métrica del algoritmo, en gris claro las métricas del sistema y en gris oscuro las métricas del usuario.	68

Capítulo I

Introducción

I. 1. Calendarización en Grid computacional

Los problemas de calendarización se pueden ver, en general, como problemas de asignación de recursos sobre el tiempo para ejecutar un conjunto de tareas como parte de algún proceso, de los cuales los computacionales y de manufactura son los más importantes. Las tareas compiten individualmente por los recursos que pueden ser de diferente naturaleza, por ejemplo procesadores, dispositivos de entrada/salida, etc. Las tareas tienen sus propias características como tamaño (número de procesadores requeridos), tiempo de llegada, tiempo de procesamiento estimado, tiempo de procesamiento real, etc. De esta manera se generan calendarios que indican, para cada tarea, cuándo y qué conjunto de procesadores la ejecuta. Además existen diferentes criterios de optimización al generar los calendarios, como lo son la minimización de la longitud del calendario (medida en tiempo) en la que se termina la ejecución de todas las tareas, la minimización del tiempo de espera de cada tarea, etc.

El problema de calendarización no es nuevo. Se han hecho estudios para sistemas distribuidos y paralelos tales como máquinas con múltiples procesadores simétricos (SMP), máquinas con procesadores paralelos masivos (MPP) y *cluster* de estaciones de trabajo (COW). Los algoritmos de calendarización han evolucionado a la par de las arquitecturas de sistemas de cómputo.

El término Grid computacional fue acuñado a mitad de los años noventa para presentar una propuesta de infraestructura de cómputo que permite utilizar recursos

computacionales que están geográficamente distribuidos. El objetivo de esta propuesta es el crear un ambiente apto para ejecutar aplicaciones de alto rendimiento (o problemas de gran reto), para las cuales la infraestructura de cómputo actual no es suficiente, incluyendo las computadoras más poderosas instaladas hoy en día (Foster y Kesselman, 1998). Dentro de este tipo de problemas se encuentran los relacionados con la física de alta energía, meteorología, astronomía, biología, medicina, genética, farmacología y economía (EC-cofunded GridTalk project, 2008). De acuerdo a Foster y Kesselman (1998), el Grid computacional es una infraestructura que agrupa diversos recursos heterogéneos de múltiples dominios distribuidos geográficamente. Provee una plataforma para compartir recursos de manera eficiente, coordinada, segura y transparente, así como para la resolución de problemas. Entre estos recursos compartidos se encuentran supercomputadoras, estaciones de trabajo, bases de datos, dispositivos de almacenamiento, software, instrumentos especiales, dispositivos visuales avanzados, etc.

El paradigma de computación en Grid (Foster y Kesselman, 2004) introduce nuevos y difíciles problemas en calendarización y manejo de recursos (Pascual *et al.*, 2007). Un Grid se puede ver como un acuerdo para compartir recursos entre un número de organizaciones independientes (como laboratorios o universidades), con poco o ningún control administrativo central, forzándolos a interactuar (Foster, 2002). Una organización es una entidad administrativa que agrupa usuarios y recursos computacionales. Las organizaciones son libres de unirse o dejar el sistema, si la experiencia adquirida es menor que el costo de la participación. Sin embargo, para sostener el Grid, el sistema de administración de recursos debe lograr un desempeño aceptable no sólo al nivel de la comunidad de usuarios (como en la calendarización clásica de un solo criterio), sino también al nivel de las organizaciones.

Sin embargo, al intentar compartir una amplia variedad de recursos computacionales distribuidos geográficamente (tal como supercomputadoras, *clusters*, sistemas de almacenamiento, fuentes de datos, instrumentos) y presentarlos como un solo recurso unificado, surgen algunos problemas de investigación interesantes como: seguridad (ya sea

de la información o del propio sistema), descubrimiento y agregación dinámicos (la cantidad de recursos en el sistema es dinámica, en cualquier momento puede unirse o salir un recurso), y calidad de servicio entre otros (Foster y Kesselman, 2003).

I. 2. Planteamiento del problema

Los modelos de calendarización tradicional generalmente producen calendarios pobres en ambientes Grid. Esto se debe a las suposiciones utilizadas en la calendarización tradicional (Berman, 1998) como homogeneidad de los recursos, arquitectura de calendarización centralizada, interconexiones de red de alta velocidad, criterios únicos de optimización, etc. Estas suposiciones desafortunadamente no se pueden aplicar a un ambiente Grid por sus características, muchas de ellas únicas. En consecuencia, el diseño de algoritmos de calendarización para Grid computacional es más complejo (Zhu, 2004).

Se han propuesto e implementado varios sistemas de calendarización, entre ellos se encuentran Condor (The University of Wisconsin Madison, 1988), LSF (Platform Computing Inc., 2008), Gridbus (The University of Melbourne, 2004), Globus (The University of Chicago, 1997), Legion (The University of Virginia, 1993), entre otros.; sin embargo, estos sistemas se basan en modelos de calendarización tradicional. Estos proyectos desarrollan un software llamado *middleware* que administra los recursos del Grid computacional, permite a los usuarios no preocuparse de la infraestructura del Grid y solamente someter la tarea al sistema.

Recientemente se propuso el esquema de admisibilidad (Tchernykh *et al.*, 2006) que tiene como objetivo mejorar el desempeño del Grid computacional al tratar un problema en específico: el retraso en la ejecución de tareas grandes debido a la ocupación de las máquinas por tareas pequeñas. Los autores presentan un análisis teórico del esquema de admisibilidad y, muy importante, demuestran que en algunos casos provee una cota de competitividad constante para algunos algoritmos. Sin embargo, no proporcionan un análisis experimental del mismo. Si bien este esquema es muy eficiente según el análisis

teórico, se requiere de resultados experimentales para determinar el caso promedio y verificar su comportamiento al aplicarlo en un sistema real.

Surge otro problema al intentar hacer un análisis experimental del esquema de admisibilidad, se requieren cargas de trabajo para llevar a cabo las simulaciones. Una carga de trabajo es el conjunto de información que describe las tareas que se ejecutaron en un centro de cómputo, o en su caso, en un Grid computacional. Estas cargas de trabajo para Grid computacional son prácticamente inexistentes y, debido a esto, se debe tratar este problema adicional para realizar el análisis experimental. Es necesario crear un método de generación de cargas de trabajo semi-sintéticas para Grid computacional a partir de un conjunto de cargas de trabajo de centros de cómputo independientes. Estas nuevas cargas de trabajo contienen las características necesarias para simulaciones de calendarización en Grid computacional, además cuentan con la información proveniente de centros de cómputo reales.

I. 3. Justificación

Las características del Grid computacional definen los nuevos retos en calendarización. Una nueva generación de sistemas comienza a aparecer para cumplir con los nuevos requerimientos impuestos por una infraestructura como la del Grid computacional. Aún existen muchos problemas no resueltos en calendarización en Grid computacional como múltiples niveles de calendarización, dinamismo, escalabilidad, etc. El estudio de nuevas estrategias de calendarización es importante para aprovechar todo el potencial que ofrece un Grid computacional. Se requiere un proceso automático y eficiente para el manejo de recursos.

En el presente trabajo se realiza un análisis experimental de varias estrategias de calendarización en Grid computacional al aplicar un esquema de admisibilidad. Este es un paso importante en calendarización en Grid al mostrar que estrategias sencillas pueden

utilizarse para el manejo de recursos en un Grid computacional, además de obtener un buen desempeño que no se ve afectado por la complejidad ni escalabilidad del sistema.

I. 4. Objetivos de la investigación

I. 4. 1 Objetivos generales

El objetivo general de esta investigación es el estudio experimental de un esquema reciente de admisibilidad para calendarización de tareas en un Grid computacional jerárquico de dos niveles; así como encontrar una cota de competitividad promedio al aplicarlo a diferentes estrategias de selección de recursos durante la calendarización. Como primer paso, es necesaria una revisión bibliográfica exhaustiva para así conocer el estado del arte en relación a este problema. Además del estudio de las diferentes estrategias de selección de recursos, algoritmos de calendarización y el esquema de admisibilidad. Posteriormente, la implantación de las estrategias de calendarización y el esquema de admisibilidad en un simulador.

Como siguiente paso, el diseño y ejecución de los experimentos para el estudio del comportamiento del esquema de admisibilidad. Para ello, se ve la necesidad de crear cargas de trabajo semi-sintéticas debido a la escasez de cargas de trabajo para Grid computacional. Se propone un método de generación de cargas de trabajo semi-sintéticas que se pueden utilizar para simulaciones de calendarización en Grid computacional.

Finalmente, se requiere el análisis de la carga de trabajo generada, para así poder interpretar los resultados de las simulaciones. Se analizan los resultados obtenidos de las simulaciones y se hacen las conclusiones pertinentes.

I. 4. 2 Objetivos específicos

Los objetivos específicos del presente trabajo son:

- Definir formalmente el problema de calendarización en línea para Grid jerárquico de dos niveles con un esquema de asignación admisible.
- Formalizar el modelo utilizado para el estudio de las diferentes estrategias de Grid jerárquico de dos niveles.
- Analizar las estrategias de calendarización en línea para Grid jerárquico de dos niveles y del esquema de admisibilidad. Así mismo, su implantación con el propósito de entender mejor su funcionamiento.
- Analizar los diferentes simuladores para Grid y seleccionar el más adecuado para utilizarlo en el análisis experimental.
- Generar cargas de trabajo adecuadas para la simulación de calendarización en Grid.
- Analizar la carga de trabajo utilizada para las simulaciones.
- Proveer análisis experimental mediante simulaciones con las estrategias propuestas para determinar el caso promedio.
- Realizar estudios comparativos de las estrategias propuestas con estrategias existentes.

I. 5. Importancia de la investigación

Se requiere un proceso automático y eficiente para el manejo de recursos y desgraciadamente los métodos de calendarización tradicional no satisfacen las necesidades actuales para la administración de recursos en un Grid computacional. Es imperativo desarrollar nuevas estrategias de calendarización para aprovechar el potencial completo de un Grid computacional. Por tal motivo, en el presente trabajo se hace un análisis experimental de varias estrategias de calendarización al aplicar el esquema de admisibilidad. Se muestra que este esquema ayuda a reducir la longitud de los calendarios en grados diferentes dependiendo de la estrategia de calendarización utilizada. Estos resultados aportan un avance significativo en calendarización en Grid al demostrar que un

esquema simple y fácil de implementar ayuda a mejorar el desempeño de un Grid computacional.

I. 5. 1 Limitaciones y suposiciones fundamentales

Como primera limitación está el uso del modelo propuesto, calendarización en línea para Grid jerárquico de dos niveles con asignación admisible. Este trabajo se limita a estudios experimentales para analizar estadísticamente el desempeño de las estrategias propuestas. Se limita el uso de información para las simulaciones a los registros estándares de dominio público, que son una imagen fiel de las diferentes cargas de trabajo de centros de cómputo reales en el mundo. Estos registros son tratados antes de iniciar las simulaciones.

I. 5. 2 Contribución al conocimiento

La principal contribución de este trabajo es el análisis experimental de diferentes estrategias de calendarización en Grid al aplicar un esquema sencillo de admisibilidad. Los resultados experimentales muestran que este esquema de admisibilidad puede ayudar a mejorar el desempeño del Grid computacional al reducir la longitud de los calendarios generados por las estrategias de calendarización. Este es un aporte significativo en calendarización en Grid al mostrar que estrategias sencillas ofrecen buenos resultados, sobre todo en un ambiente tan complejo como lo es un Grid computacional.

A pesar que actualmente existen algunas implementaciones de Grid, sus limitaciones son claras debido a la complejidad del sistema. La contribución de este trabajo es proveer un análisis experimental de estrategias de calendarización para las cuales no existen resultados publicados.

Para efectuar el análisis experimental se requieren cargas de trabajo para Grid computacional, pero existe una escasez en este sentido. Debido a esto, se propone un método para generar una carga de trabajo semi-sintética que pueda representar un ambiente Grid. Este método combina diferentes cargas de trabajo independientes para simular una

carga de trabajo para Grid computacional. Esta es una nueva alternativa de generación de cargas de trabajo que se pueden utilizar para análisis experimental.

La publicación de resultados experimentales no siempre está completa, un análisis de la carga de trabajo es importante para proveer un sustento a esos resultados. En este trabajo se hace un análisis exhaustivo de la carga de trabajo para entender el comportamiento de las estrategias de calendarización al utilizar el esquema de admisibilidad. Este análisis de la carga de trabajo es de los más completos que registran en la literatura.

I. 6. Organización de la tesis

En el 0 se presenta un marco teórico relacionado con esta investigación, así como el estado del arte del problema tratado. El 0 presenta la definición formal del problema, el esquema de admisibilidad, el modelo y las estrategias de calendarización estudiadas. Posteriormente, en el 0 se habla de cargas de trabajo para los estudios experimentales, presentamos el algoritmo utilizado para generar las cargas de trabajo semi-sintéticas para Grid computacional. En el 0 se muestra el análisis experimental de las diferentes estrategias de calendarización, además del análisis de la carga de trabajo. Finalmente, en el 0 se presentan las conclusiones de este trabajo de investigación, así como el trabajo futuro.

Capítulo II

Marco teórico

En este capítulo se presentan las diferentes arquitecturas (modelos) y características del Grid computacional. Dichas arquitecturas se clasifican en tres tipos: centralizada, descentralizada y jerárquica. Cada una con características propias y comunes para el Grid computacional.

Los principales retos, que estas arquitecturas traen consigo, se relacionan con tres áreas principales como lo son la heterogeneidad y autonomía, el desempeño dinámico y la separación entre la selección de recursos y el cálculo de datos. Por lo que la calendarización en Grid es inherentemente más compleja.

Por último, en este capítulo se muestran los resultados previos en calendarización más importantes para esta investigación, además de las cotas de competitividad de las diferentes variantes de la calendarización basada en lista.

II. 1. Arquitecturas de Grid computacional

El Grid computacional está emergiendo como una nueva forma de compartir, seleccionar y agregar una amplia variedad de recursos incluyendo supercomputadoras y otros recursos de cómputo especializados. Estos recursos se encuentran distribuidos geográficamente. Pertenecientes a diversas organizaciones para la resolución de problemas computacionales de gran escala que requieren un gran poder de cómputo en la ciencia, la ingeniería y el comercio.

La estructura del Grid computacional tiene influencia en la arquitectura del calendarizador para sistemas manejadores de recursos (Foster y Kesselman, 2003). La estructura del calendarizador depende del número de recursos en los cuales las tareas se calendarizan y el dominio en el cual se localizan. Existen tres modelos o arquitecturas de calendarización: centralizada, descentralizada y jerárquica.

II. 1. 1 Centralizada

Este modelo se usa para manejar uno o múltiples recursos localizados en uno o múltiples dominios. En un ambiente centralizado, un dispositivo central calendariza todas las máquinas paralelas (Figura 1); la información del estado de todos los sistemas disponibles se recolecta aquí. Solamente puede dar soporte a políticas uniformes y es muy bueno para administrar sistemas de *clusters* (o colas por lotes) como Condor (The University of Wisconsin Madison, 1988), LSF (Platform Computing Inc., 2008), etc. No es adecuado para sistemas manejadores de recursos de Grid debido a que en ellos hay políticas impuestas por los propietarios de los recursos.

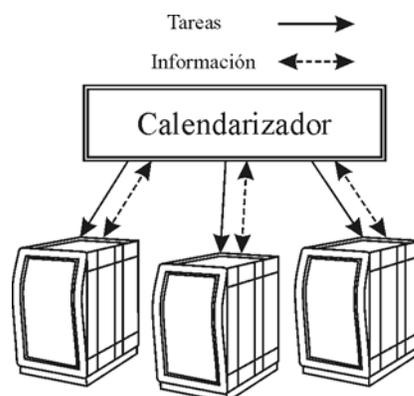


Figura 1. Calendarizador centralizado.

II. 1. 2 Descentralizada

En este modelo los calendarizadores interactúan entre ellos con el objetivo de decidir cuál recurso aplicar a las tareas que estén en espera. En este esquema, no hay ningún líder central responsable de la calendarización, por tanto este modelo es altamente escalable y tolerante a fallas. Este esquema es adecuado para sistemas de Grid debido a que

los propietarios de los recursos pueden definir sus propias políticas de calendarización. De cualquier manera, debido a que el estado de tareas y recursos remotos no está disponible en una sola ubicación, la generación de calendarios óptimos no es segura. Este modelo es difícil de implementar en un ambiente Grid de gran escala porque los dueños de los dominios de recursos no están de acuerdo en una política global para el manejo de los mismos.

II. 1. 2. 1. Comunicación directa

Los calendarizadores locales pueden enviar y recibir tareas hacia y desde otros calendarizadores de manera directa (par-a-par) (Figura 2).

Una arquitectura de red distribuida par-a-par está compuesta de participantes que hacen disponible una parte de sus recursos (como tiempo de procesamiento, almacenamiento, ancho de banda, etc.) a sus pares de manera directa sin intermediarios de red o servidores. Cada par es un proveedor y consumidor de recursos, en contraste con el modelo cliente-servidor tradicional, donde sólo los servidores son proveedores y los clientes consumidores.

Esta arquitectura par-a-par se popularizó por los sistemas de compartir archivos como Napster. Estas redes par-a-par inspiraron nuevas estructuras y filosofías en otras áreas de la interacción humana.

En el Grid, cada calendarizador tiene una lista de los calendarizadores remotos que puede contactar o hay un directorio que provee la información de los otros sistemas. Si no se puede iniciar una tarea en la máquina local, el calendarizador local busca una máquina alternativa. Si se encuentra una máquina adecuada, donde la tarea puede iniciarse inmediatamente, ésta y toda su información se transfiere a la otra máquina.

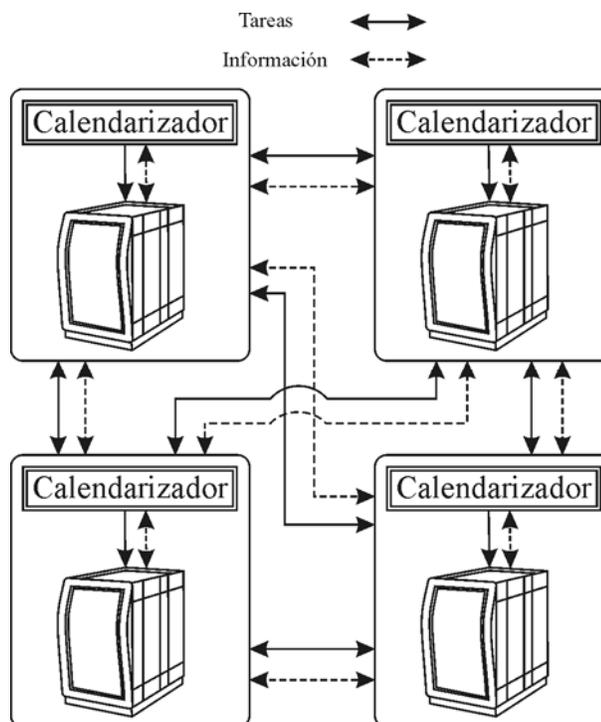


Figura 2. Calendarización descentralizada con comunicación directa.

II. 1. 2. 2. Comunicación a través de una pila central de tareas

En el esquema de comunicación a través de una pila, las tareas que no se pueden ejecutar inmediatamente se envían a una pila central de tareas en lugar de enviarse a una máquina remota (Figura 3). En contraste con la comunicación directa, el calendarizador local puede tomar tareas adecuadas para su calendario. En este escenario, las tareas se pueden poner o retirar de la pila. Se requiere una política para que todas las tareas de la pila se ejecuten en cierto tiempo y no queden esperando indefinidamente para ser procesadas. Este método se puede modificar para que todas las tareas se pongan en la pila una vez que llegan al sistema. De esta manera las tareas que requieren pocos recursos se pueden usar para asignarlas a recursos libres en todas las máquinas.

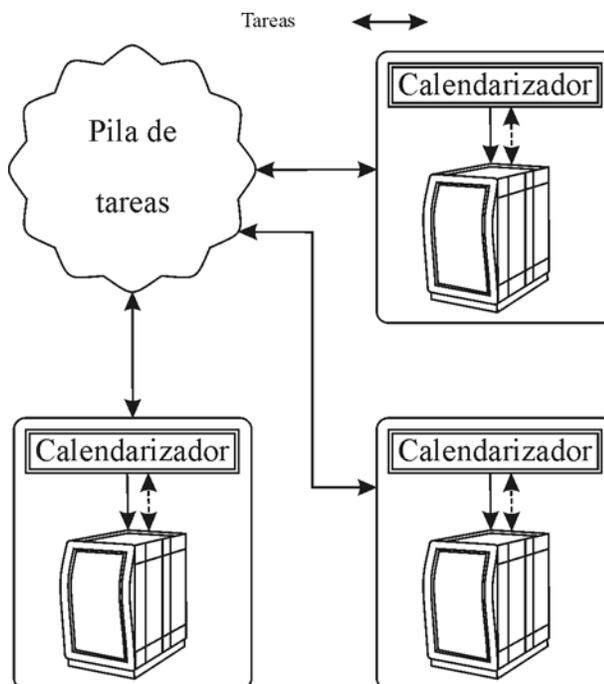


Figura 3. Calendarización descentralizada con comunicación a través de una pila central de tareas.

II. 1. 3 Jerárquica

El esquema jerárquico es adecuado para sistemas de Grid a gran escala porque permite a los propietarios de recursos remotos aplicar sus propias políticas sobre usuarios externos. Este modelo parece como si fuera un modelo híbrido (una combinación de los modelos centralizado y descentralizado), pero parece más un modelo centralizado y funciona bien para sistemas de Grid. En este esquema el metacalendarizador interactúa con los calendarizadores locales para, de esta manera, seleccionar los recursos adecuados (Figura 4).

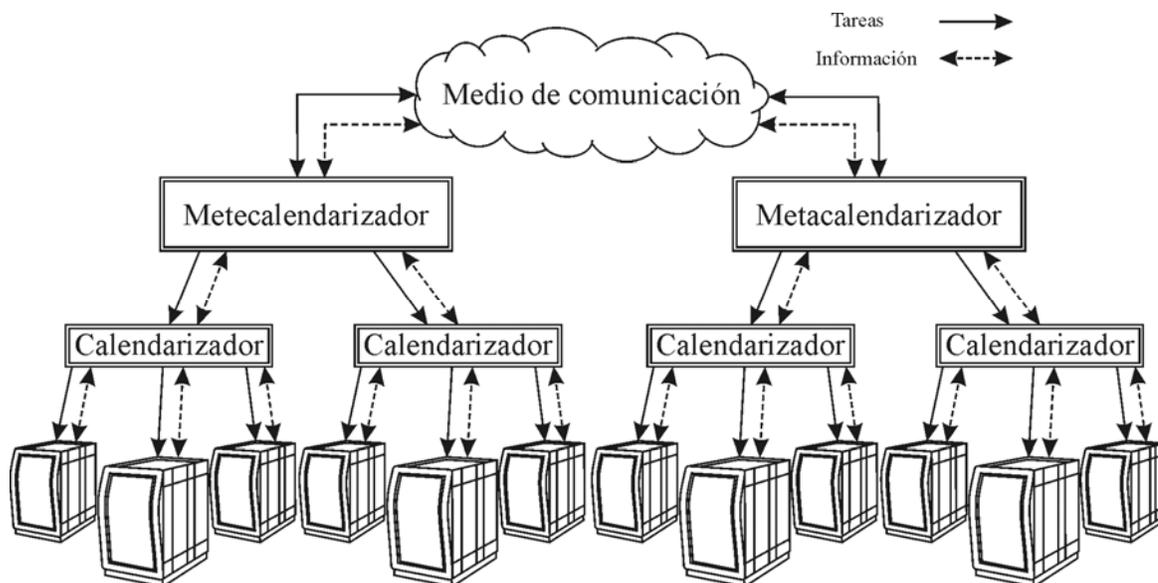


Figura 4. Calendarización de estructura jerárquica.

II. 2. Retos de algoritmos en calendarización

Se han estudiado ampliamente los algoritmos de calendarización como un problema básico de los sistemas distribuidos y paralelos, tales como SMP, MPP y COW. Los algoritmos de calendarización han evolucionado con las arquitecturas de sistemas paralelos y distribuidos. La Tabla I muestra algunas características importantes de sistemas paralelos y distribuidos, así como algunos algoritmos de calendarización típicos que estos sistemas han adoptado (Dong y Akl, 2006).

Los modelos de calendarización tradicional generalmente producen calendarios pobres en ambientes de Grid. Se puede encontrar una razón revisando las suposiciones en los sistemas de calendarización tradicional (Berman, 1998):

1. Todos los recursos residen en un solo dominio administrativo.
2. Para proveer una imagen única del sistema, el calendarizador controla todos los recursos.
3. El conjunto de recursos permanece invariante.

4. El sistema de calendarización puede administrar la competencia causada por las nuevas aplicaciones entrantes de acuerdo a algunas políticas, por lo que su impacto sobre el rendimiento que el sitio puede proporcionar a cada aplicación se puede predecir correctamente.
5. Los cálculos y datos residen en el mismo sitio o la organización de los datos es un proceso altamente predecible, usualmente desde una fuente hasta un destino predeterminados, los cuales se pueden ver como un costo constante.

Tabla I. Evolución de los algoritmos de calendarización con los sistemas computacionales paralelos y distribuidos.

Arquitectura típica	DSM, MPP	COW	Grid
Cronología	Fines de los años setenta	Fines de los años ochenta	Mediados de los años noventa
Interconexión típica de sistema	Bus, switch	LAN comercial, ATM	WAN/Internet
Costo de la interconexión	Muy baja, despreciable	Baja, usualmente no despreciable	Alta, no despreciable
Heterogeneidad de la interconexión	Ninguna	Baja	Alta
Heterogeneidad de los recursos	Ninguna	Baja	Alta
Imagen única del sistema	Sí	Sí	No
Pila de recursos estática/dinámica	Predeterminada y estática	Predeterminada y estática	No predeterminada y dinámica
Política de manejo de recursos	Monótona	Monótona	Diversa
Algoritmos típicos de calendarización	Algoritmos de calendarización homogéneos	Algoritmos de calendarización homogéneos	Algoritmos de calendarización en Grid

Desafortunadamente todas estas suposiciones no se pueden aplicar en el ambiente de un Grid. Existen muchas características únicas en un Grid computacional que hacen el diseño de algoritmos de calendarización más complejo (Zhu, 2004) como se explica más adelante.

II. 2. 1 Heterogeneidad y autonomía

Aunque la heterogeneidad no es nueva en los algoritmos tradicionales de calendarización, incluso antes del Grid computacional, éstos permanecen lejos de estar completamente dirigidos hacia el Grid. En el ambiente de un Grid, debido a que los recursos están distribuidos en múltiples dominios a través de Internet, no sólo los recursos computacionales y de almacenamiento son heterogéneos sino también las redes que los interconectan. La heterogeneidad resulta en capacidades diferentes para el procesamiento de tareas y el acceso a los datos.

En la calendarización tradicional, un solo punto de control maneja los recursos computacionales. El sistema de calendarización no sólo tiene la información completa acerca de la dependencia de las tareas y la utilización de los recursos, sino que también administra la cola de tareas. Éste puede fácilmente predecir el comportamiento de los recursos y es capaz de asignar tareas a los recursos de acuerdo a un cierto requerimiento de rendimiento.

En un Grid, sin embargo, los recursos son normalmente autónomos y el sistema de calendarización no tiene un control total de los recursos; éste no puede violar las políticas locales de los recursos, lo que hace más difícil para el sistema de calendarización en el Grid el estimar el costo exacto de ejecutar una tarea en diferentes sitios. La autonomía también resulta en la diversidad de las políticas en el manejo de los recursos locales y de control de acceso; por ejemplo, las configuraciones de prioridad para diferentes aplicaciones y los métodos de reservación de recursos. Por lo tanto, se requiere que el calendarizador del Grid se adapte a diferentes políticas locales. La heterogeneidad y autonomía del lado del usuario están representadas por varios parámetros, incluyendo tipos de aplicaciones, requerimientos de recursos, modelos de desempeño y objetivos de optimización. En esta situación, se proponen y aplican nuevos conceptos tal como calendarización al nivel de la aplicación y economía de Grid (Buyya *et al.*, 2005) para la calendarización en Grid.

II. 2. 2 Desempeño dinámico

Para hacer una calendarización factible se tiene que considerar la estimación del rendimiento que el recurso candidato pueda proporcionar, especialmente cuando los algoritmos son estáticos. Los calendarizadores de Grid trabajan en un ambiente dinámico donde los rendimientos de los recursos disponibles cambian constantemente. El cambio viene de la autonomía de los sitios y la competencia por los recursos entre las aplicaciones. Debido a la autonomía de los recursos, usualmente éstos no están dedicados a una aplicación específica. Por ejemplo, una tarea sometida internamente en un *cluster*, por su alta prioridad, podría interrumpir una tarea sometida remotamente en el mismo *cluster*; los nuevos recursos pueden unirse y proporcionar mejores servicios; o algunos recursos pueden no estar disponibles en cierto momento. El mismo problema ocurre con las redes que conectan los recursos en el Grid: el tráfico en Internet puede afectar el ancho de banda disponible, el cual no es relevante para las tareas en el Grid. Para las aplicaciones en el Grid, este tipo de competencia por los recursos resulta en una fluctuación en su rendimiento, lo cual hace más difícil el evaluar el desempeño de la calendarización del Grid bajo modelos clásicos de rendimiento. Desde el punto de vista de la calendarización de tareas, la fluctuación en el desempeño podría ser la característica más importante de un Grid computacional comparada con los sistemas tradicionales. Un algoritmo de calendarización factible debería ser capaz de adaptarse a tal comportamiento dinámico. Algunas otras medidas están también disponibles para mitigar el impacto de este problema, tales como la negociación de la calidad de servicio (QoS por sus siglas en inglés), reservación de recursos (proporcionada por el sistema administrador de recursos) y la recalendarización.

II. 2. 3 Separación entre la selección de recursos y el cálculo de datos

En los sistemas tradicionales, los códigos ejecutables de las aplicaciones y los datos de entradas y salida están localizados normalmente en el mismo sitio, o las fuentes de entrada y los destinos de las salidas se determinan antes de que la aplicación se someta. Así el costo por la organización de los datos se puede ignorar o el costo es una constante

determinada antes de la ejecución, y los algoritmos de calendarización no necesitan considerarlo. Pero en un Grid, el cual está formado por un gran número de sitios computacionales heterogéneos y sitios de almacenamiento conectados vía una red de área extensa, los sitios computacionales para las aplicaciones los selecciona normalmente el sistema de calendarización del Grid de acuerdo al estado del recurso y a ciertos modelos de rendimiento. Adicionalmente, en un Grid, el ancho de banda de la red está limitado y compartido por lo que la comunicación entre dominios no se puede ignorar. Además muchas aplicaciones incluyen gran cantidad de datos, por lo que el costo de la organización de datos es considerable. Esta situación lleva al problema de separación datos-computación: la ventaja es que al seleccionar un recurso que puede proporcionar bajo costo computacional podría neutralizarse por el alto costo de acceso al sitio de almacenamiento.

Estos retos definen características únicas del Grid computacional y ponen muchos obstáculos para el diseño e implementación de sistemas eficientes y efectivos de calendarización en tal arquitectura. Parece difícil el panorama para las metodologías tradicionales de calendarización frente a este gran problema, sin embargo, se cree que los logros obtenidos en los problemas tradicionales de calendarización puedan servir de base para una nueva generación de sistemas de calendarización (Andrieux *et al.*, 2003).

II. 3. Calendarización basada en lista

Los estudios teóricos de calendarización basada en lista datan desde hace más de 40 años. A pesar de todo este tiempo, los resultados aún siguen vigentes. La calendarización de tareas utilizando múltiples procesadores paralelos es compleja. Se sabe que al cambiar algunos parámetros como el orden de la lista de tareas, disminuir el tiempo de ejecución de las tareas, relajar las relaciones de precedencia de las tareas o cambiar el número de procesadores del sistema, se presentan ciertas anomalías como el aumento de la longitud del calendario (Graham, 1969). Uno de los resultados más importantes es la cota de competitividad para tareas secuenciales y calendarización determinística cuando todas las tareas están disponibles desde el inicio de la calendarización (Graham, 1966):

$$\frac{C_{max}(L)}{C_{max}^*} \leq 2 - \frac{1}{m} \quad (1)$$

donde $C_{max}(L)$ es la longitud del calendario generado por un algoritmo basado en lista, C_{max}^* es la longitud del calendario óptimo y m es el número de procesadores paralelos. Esta cota de desempeño también se aplica para tareas paralelas y calendarización determinística (Garey y Graham, 1975).

Es más difícil establecer una cota de desempeño cuando no todas las tareas están disponibles desde el inicio de la calendarización y, además, no se conocen los tiempos de ejecución de las tareas. A pesar de eso, algunos estudios recientes muestran que la cota anterior también se aplica para este tipo de problemas (Naroska y Schwiegelshohn, 2002).

Pero la calendarización en Grid es inherentemente más compleja al incluir múltiples máquinas paralelas. Este problema es similar al de *Partición* que es *NP*-difícil¹ (Garey y Johnson, 1979) de acuerdo a la teoría de la complejidad computacional. La cota de competitividad para algoritmos de calendarización basados en lista no se puede aplicar a un Grid computacional. Incluso, los algoritmos de calendarización basados en lista no pueden garantizar una cota de competitividad constante $\frac{C_{max}(L)}{C_{max}^*}$ (Schwiegelshohn *et al.*, 2008). Esto se debe a la complejidad de balancear la carga de trabajo, provocando que tareas paralelas grandes esperen su ejecución. No existe un algoritmo de tiempo polinomial que siempre produzca calendarios con $\frac{C_{max}(L)}{C_{max}^*} < 2$ para $GP_m | size_j | C_{max}^2$ a menos que $P = NP$ (Schwiegelshohn *et al.*, 2008).

¹ *NP*-difícil, en teoría de la complejidad computacional, es una clase de problemas que son al menos tan difíciles como los problemas más difíciles de la clase *NP*.

² Notación corta de tres campos utilizada para describir un problema de calendarización específico (Graham *et al.*, 1979). El primer campo describe las características del sistema, GP_m se refiere a la calendarización en Grid. El segundo campo describe las características de las tareas, $size_j$ indica que el tamaño de cada tarea es conocido. El tercer campo describe el criterio de optimización, C_{max} denota la longitud del calendario (medido en unidades de tiempo).

El algoritmo MOLBA (*Multi-Organization Load Balancing Algorithm*) (Pascual *et al.*, 2007) también tiene una cota de desempeño constante; $\frac{C_{max}}{C_{max}^*} < 3$ si la última tarea calendarizada es pequeña, $\frac{C_{max}}{C_{max}^*} < 4$ para el caso general. El modelo utilizado es más restringido, múltiples máquinas paralelas iguales. En un ambiente Grid esta suposición no es realista; por definición un Grid es una infraestructura distribuida heterogénea (Foster y Kesselman, 1998).

Se han considerado variantes de calendarización en línea para máquinas paralelas, donde parte de la información de trabajos futuros se conoce de antemano. El problema LL (Longest Last) es una versión semi en línea donde se sabe que la última tarea es la más larga, en otras palabras, la última tarea en la lista es la que tiene el tiempo de procesamiento mayor. Algunos algoritmos para este problema alcanzan cocientes de competitividad de $\sqrt{2}$ y $\frac{3}{2}$ para máquinas con 2 y 3 procesadores respectivamente y tareas secuenciales (Zhang y Ye, 2002). Sin embargo, a pesar de que estos algoritmos son tan eficientes, las suposiciones del modelo son tan restringidas que únicamente se pueden aplicar para casos muy específicos. Por tal razón resultan inadecuados para ambientes dinámicos y heterogéneos como los que presenta el Grid computacional.

Existen resultados experimentales para Grid jerárquico de dos niveles, sin embargo, algunos utilizan cargas de trabajo sintéticas, otros utilizan modelos restringidos o se enfocan en diferentes puntos del problema. En un trabajo previo (Ramírez *et al.*, 2007) se presenta una comparación de varias estrategias de calendarización, *Min_CT* (asigna las tareas a la máquina que ofrece el menor tiempo de terminación) y *Min_ST* (asigna las tareas a la máquina que ofrece el mejor tiempo de inicio) son las estrategias que presentan los mejores resultados. El estudio se enfoca primordialmente en estimar los tiempos de ejecución basándose en los tiempos solicitados por el usuario, de esta manera pretenden obtener mejores resultados. Un punto importante es el hecho que no se presenta un análisis de la carga de trabajo, por lo que los resultados no están bien sustentados, además la calendarización es determinística. Caso similar se presenta en (Galaviz *et al.*, 2006) donde

se realizan evaluaciones con más algoritmos de calendarización local, también para calendarización determinística. Y estudios teóricos de la estrategia *Min_CT* muestran cocientes de competitividad constantes al combinarlo con ciertos algoritmos de calendarización local como lo son *BL* (Bottom Left) y *BLD* (Bottom Left Decreasing) (Zhuk *et al.*, 2004).

Uno de los pocos estudios que incluyen un análisis teórico y experimental de estrategias de calendarización analiza dos de ellas, *FCFS* (First Come First Served) y *LS* (List Scheduling), con diferentes ordenamientos de la cola de tareas. Presentan una cota de desempeño de $\frac{1}{\varphi} \left(\frac{2^P - 2s + 1}{P - s + 1} \right)$ para el algoritmo *LS*, donde φ depende de un modelo de costos de comunicación, s es el tamaño mínimo de las tareas y P el total de procesadores en el Grid. Aunque este análisis también se enfoca en una calendarización determinística y los resultados experimentales utilizan configuraciones de Grid y cargas de trabajo sintéticas (Li, 2005).

Se han obtenido más resultados para calendarización determinística que para calendarización en línea. Refiérase al trabajo de Sgall (1998) y al de Iovanella (2002) para una revisión más completa de varios resultados y paradigmas en línea.

II. 4. Calendarización en línea para Grid de un nivel

La calendarización basada en lista es significativamente peor en ambientes Grid. Se han diseñado algoritmos más complejos para tratar de solucionar el problema. Una propuesta reciente es un algoritmo de robo de tareas; tiene una cota de desempeño $\frac{C_{max}}{C_{max}^*} < 5$ para una calendarización en línea y $\frac{C_{max}}{C_{max}^*} < 3$ para el caso determinístico (Schwiegelshohn *et al.*, 2008). Como la calendarización convencional basada en lista no es adecuada para Grid, este algoritmo utiliza múltiples listas (dos para cada máquina). Estas listas se definen a partir del tamaño de cada máquina en el Grid y las características de las tareas. De esta manera se puede alcanzar un mejor balance de la carga de trabajo de cada máquina.

Esta propuesta utiliza un modelo descentralizado de calendarización, donde las máquinas tienen una comunicación directa entre ellas. Además, la comunicación entre las máquinas es instantánea (no existen costos de comunicación). El algoritmo considera la existencia de dos colas de tareas por máquina, A y B , donde las tareas se encuentran exclusivamente en una sola cola. Estas colas se definen de la siguiente manera:

$$A_i = \left\{ J_j \mid \max \left\{ \frac{m_i}{2}, m_{i-1} \right\} < size_j \leq m_i \right\} \quad (2)$$

$$B_i = \left\{ J_j \mid m_{i-1} < size_j \leq \frac{m_i}{2} \right\} \quad (3)$$

donde J_j es la tarea j , m_i es el tamaño (número de procesadores) de la máquina i , $size_j$ es el tamaño de la tarea j .

La cola A_i contiene las tareas que ocupan más del 50% de los procesadores disponibles en la máquina i ; la cola B_i , las tareas que ocupan menos del 50% de los procesadores de la misma máquina. Las tareas en A_i y B_i sólo se pueden calendarizar en la máquina i o en una más grande.

Las tareas contenidas en la cola A_i se ejecutan en la máquina i . Una vez terminadas las tareas de esta cola, se ejecutan las tareas de la cola B_i . Una máquina puede “robar” tareas de máquinas más pequeñas cuando tiene procesadores disponibles y ha ejecutado las tareas de la cola local A .

II. 5. Calendarización en línea en Grid jerárquico de dos niveles

Otro modelo propuesto para calendarización en Grid es de dos niveles. En el primer nivel, el metacalendarizador asigna tareas a máquinas paralelas siguiendo algún criterio; en el segundo nivel, el calendarizador local de cada máquina genera su propio calendario con las tareas que le fueron asignadas.

Una razón de la ineficiencia de la calendarización en línea es la ocupación de máquinas grandes por tareas secuenciales, causando que las tareas altamente paralelas esperen su ejecución (ver Figura 5). Un estudio reciente muestra que al utilizar un esquema simple de admisibilidad ayuda a mejorar el desempeño de algunas estrategias de calendarización (Tchernykh *et al.*, 2006). Este esquema excluye ciertas máquinas con muchos procesadores del conjunto de máquinas disponibles para la ejecución de una tarea pequeña (ver Figura 6).

Las estrategias presentadas son *Tiempo mínimo de terminación (Min_CT)* y *Cota mínima de terminación (Min_LB)* para el primer nivel, *Tamaño máximo primero (LSF)* como calendarizador local para el segundo nivel. Estas estrategias por sí solas no ofrecen un cociente de competitividad constante. Sin embargo, al utilizar el esquema de admisibilidad, las mismas estrategias ofrecen un cociente de competitividad constante de 5.

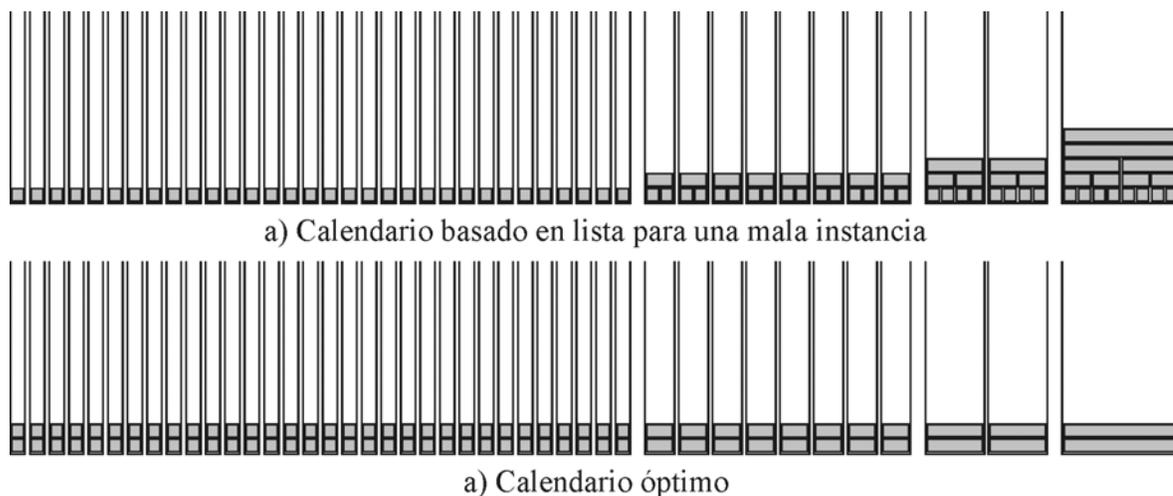


Figura 5. Un problema de los algoritmos de calendarización basados en lista; a) calendario generado para una mala instancia del problema, b) calendario óptimo para la misma instancia.

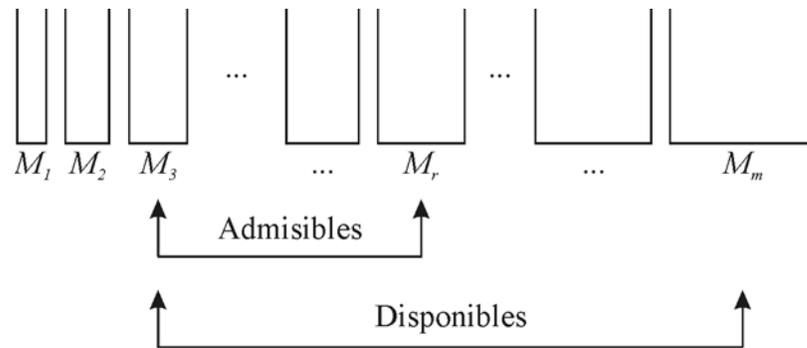


Figura 6. Esquema de admisibilidad.

La introducción de un nuevo parámetro, $0 \leq a \leq 1$, al esquema de admisibilidad modifica dinámicamente el grado de admisibilidad para la asignación de tareas tomando en cuenta las características de la carga de trabajo (Tchernykh *et al.*, 2008). La selección de $a = \frac{1}{2}$ produce la definición original (Tchernykh *et al.*, 2006).

Capítulo III

Definición formal del problema

En este capítulo se presenta la definición formal del problema de calendarización en Grid que consiste en asignar tareas a máquinas para completarlas en los límites impuestos. En la notación corta de tres campos se describe como $GP_m | r_j, size_j | C_{max}$.

También se presenta el esquema de admisibilidad que previene que tareas pequeñas ocupen máquinas grandes, evitando que las tareas grandes esperen por su ejecución.

Finalmente, se presentan el modelo jerárquico de dos niveles con el esquema de admisibilidad y las estrategias de calendarización utilizados en esta investigación.

III. 1. Definición del problema de calendarización

En esta investigación se trata el problema de calendarización en línea con el objetivo de minimizar la longitud del calendario: n tareas paralelas (J_1, J_2, \dots, J_n) se deben calendarizar en m máquinas paralelas (M_1, M_2, \dots, M_m) ; se denota al número de procesadores idénticos de la máquina M_i como m_i . Sin pérdida de generalidad, se supone que las máquinas paralelas se encuentran ordenadas en forma no decreciente por su tamaño $(m_1 \leq m_2 \leq \dots \leq m_m)$.

La tarea J_j está representada por una tripleta $(r_j, size_j, p_j)$: el tiempo de su llegada $r_j \geq 0$, su tamaño $1 \leq size_j \leq m_m$ que se refiere a su grado de paralelismo, y su tiempo de ejecución p_j . El tiempo de llegada de cada tarea no se conoce hasta que es sometida al

sistema (calendarización en línea), y su tiempo de ejecución es desconocido hasta que termina su procesamiento (calendarización no clarividente).

El presente trabajo supone que la tarea J_j sólo se puede ejecutar en la máquina M_i si $size_j \leq m_i$. También se supone una calendarización en modo de espacio compartido³, aplicado en muchas computadoras paralelas. Además, una tarea paralela J_j se ejecuta en exactamente $size_j$ procesadores disjuntos sin interrupción.

Finalmente, $g(J_j) = M_i$ denota que la tarea J_j se asigna a la máquina M_i . Sea n_i el número de tareas asignadas a la máquina M_i .

Todos los algoritmos se analizan de acuerdo al cociente de competitividad (cuántas veces se aleja del óptimo) para la longitud del calendario. Sea C_{max}^* y $C_{max}(A)$ las longitudes de los calendarios óptimo y el determinado por el algoritmo A , respectivamente. El cociente de competitividad del algoritmo A se define como $\rho_A = \max\left(\frac{C_{max}(A)}{C_{max}^*}\right)$ para todas las instancias del problema.

La notación GP_m describe el modelo de Grid utilizado. En la notación corta de tres campos (Graham *et al.*, 1979) se describe como $GP_m|r_j, size_j|C_{max}$ (Schwiegelshohn *et al.*, 2008). Para referirse a este problema se utiliza el término MPS , y PS para la calendarización de tareas paralelas en una sola máquina paralela ($P_m|r_j, size_j|C_{max}$) (Tchernykh *et al.*, 2008).

³ *Espacio compartido*: Las tareas son asignadas a un subconjunto de procesadores de una máquina paralela de manera exclusiva, pero la máquina (su "espacio") es compartida. Esto es, diferentes tareas paralelas pueden potencialmente ejecutarse al mismo tiempo si los requerimientos de las tareas permiten su asignación a diferentes subconjuntos de procesadores de esa máquina. El "espacio" se calcula al multiplicar el número de procesadores por el tiempo.

III. 2. Esquema de admisibilidad

Se define $f(j) = first(j)$ como el índice i más pequeño tal que $m_i \geq size_j$ para la tarea J_j . También se define $l(j) = last(j)$ como el índice i más grande tal que $m_i \geq size_j$ para la tarea J_j . Debido a la restricción $size_j \leq m_m \forall J_j$, se tiene que $l(j) \leq m$. El conjunto de máquinas $M_{available}(j)$ son las que están disponibles para ejecutar la tarea J_j y corresponde al conjunto de índices de las máquinas $s(f(j), m) = \{f(j), f(j) + 1, f(j) + 2, \dots, m\}$. El conjunto completo de máquinas M_{total} se representa por el conjunto de enteros $s(1, m) = 1, \dots, m$. El número total de procesadores que se encuentran en las máquinas m_f a m_l es $m(f, l) = \sum_{i=f}^l m_i$.

El conjunto $M_{admissible}(j)$ de máquinas admisibles para la tarea J_j es el conjunto de máquinas con índices en el conjunto $s(f(j), r(j))$, donde $r(j)$ es el número mínimo que cumple $m(f(j), r(j)) \geq \frac{1}{2} m(f(j), m)$ (Tchernykh *et al.*, 2006). La definición se generaliza introduciendo un parámetro nuevo $0 \leq a \leq 1$ que configura el grado de admisibilidad usado para la asignación de tareas (Tchernykh *et al.*, 2008). Por lo tanto, $s(f(j), r(j))$ es el conjunto de índices de máquinas admisibles si $r(j)$ es el índice mínimo tal que $m(f(j), r(j)) \geq a \cdot m(f(j), m)$. Cuando $a = \frac{1}{2}$ (o 50% en el grado de admisibilidad) se genera la definición original (Tchernykh *et al.*, 2006). Cuando $a = 1$ (o 100% en el grado de admisibilidad) equivale al algoritmo original sin utilizar el esquema de admisibilidad.

El análisis teórico de este esquema de admisibilidad utiliza Min_LB como estrategia MPS_{Alloc} (nivel de asignación de recursos) y define $Best_PS$ como el mejor algoritmo conocido de calendarización local con un cociente de competitividad de $2 - \frac{1}{m}$.

Teorema 1. (Tchernykh *et al.*, 2008) Supóngase un conjunto de máquinas con procesadores idénticos, un conjunto de tareas rígidas (aquellas tareas que solicitan un número de procesadores que no puede cambiar bajo ninguna circunstancia) y un rango de

asignación admisible $0 \leq a \leq 1$. Entonces el algoritmo *Min_LB + Best_PS* tiene un cociente de competitividad

$$\rho \leq \begin{cases} 1 + \frac{1}{a^2} - \frac{1}{m(1, m)} & \text{para } a \leq \frac{m(f(d), m)}{m(f(b), m)} \\ 1 + \frac{1}{a(1-a)} - \frac{1}{m(1, m)} & \text{para } a > \frac{m(f(d), m)}{m(f(b), m)} \end{cases} \quad (4)$$

con $1 \leq f(b) \leq f(d) \leq m$ siendo parámetros que dependen de la carga de trabajo (ver Figura 7). Donde J_d es la última tarea asignada a la máquina que define la longitud del calendario del Grid y J_b la tarea más pequeña ejecutada en el conjunto $M_{available}(d)$.

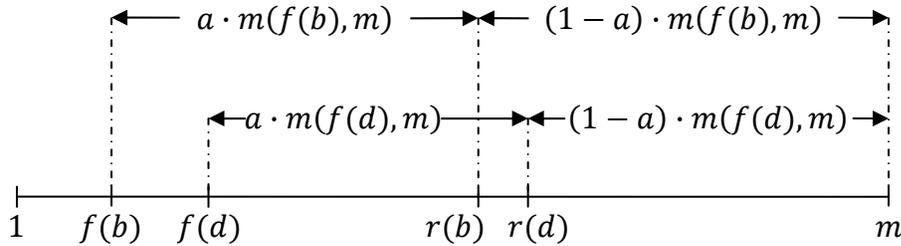


Figura 7. Asignación admisible con factor a .

III. 3. Modelo jerárquico de dos niveles con admisibilidad

El modelo jerárquico considera dos niveles de calendarización. En el primer nivel se lleva a cabo una asignación de tareas a máquinas (selección de recursos) utilizando un esquema de admisibilidad, en el segundo nivel se considera un calendarizador local por cada una de las máquinas que integran el Grid como se muestra en la Figura 8.

La asignación de tareas a máquinas consiste en aplicar alguna estrategia que selecciona la máquina más adecuada dentro de un conjunto de máquinas disponibles para ejecutar dicha tarea. Este conjunto de máquinas disponibles se reduce de acuerdo al esquema de admisibilidad utilizado. Esta reducción intenta evitar la asignación de tareas pequeñas a máquinas grandes, provocando que se retrasen las tareas grandes.

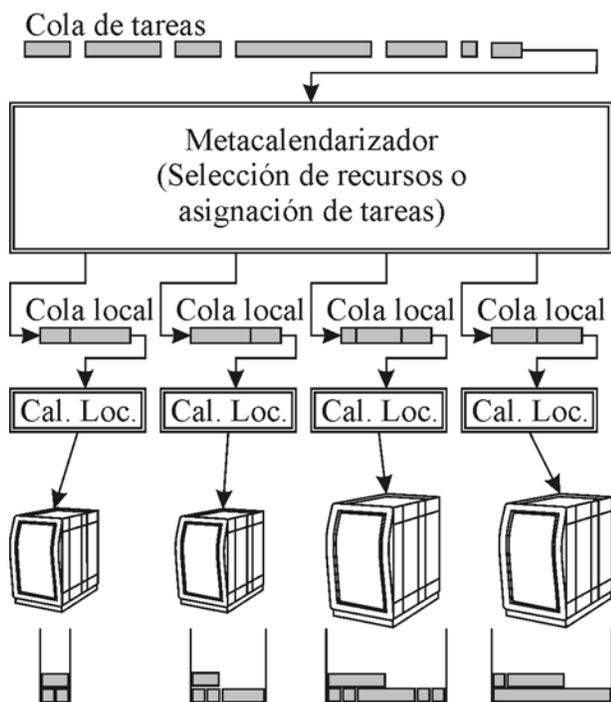


Figura 8. Modelo de calendarización jerárquica de dos niveles.

Durante la calendarización local, las tareas enviadas a cada máquina se asignan a los procesadores que habrán de ejecutarlas. En esta etapa es donde se crean los calendarios definitivos de ejecución.

III. 4. Estrategias de calendarización

La calendarización se basa en un modelo jerárquico de dos niveles. En el primer nivel, el metacalendarizador selecciona una máquina del conjunto de máquinas admisibles para cada tarea según algún criterio. En el segundo nivel, el calendarizador local crea el calendario de ejecución para las tareas asignadas.

III. 4. 1 Asignación de tareas a máquinas

Este estudio se enfoca en la asignación de tareas conociendo de antemano el número de máquinas paralelas y el tamaño de cada una. Se utilizan catorce estrategias de asignación de tareas:

1. *Random*. (Madrigal *et al.*, 2006) Selecciona la máquina de manera aleatoria.

$$f(j) \leq \text{random}(\) \leq r(j) \quad (5)$$

2. *Min_Lp*. (Madrigal *et al.*, 2006) Selecciona la máquina con la mínima carga por procesador.

$$\min_{f(j) \leq i \leq r(j)} \left(\frac{n_i}{m_i} \right) \quad (6)$$

3. *Min_PL*. (Madrigal *et al.*, 2006) Selecciona la máquina con la mínima carga paralela por procesador.

$$\min_{f(j) \leq i \leq r(j)} \left(\sum_{g(J_k)=M_i} \frac{\text{size}_k}{m_i} \right) \quad (7)$$

4. *Min_LBal*. (Kurowski *et al.*, 2008) Selecciona la máquina que genera la menor desviación estándar en la distribución de la carga de cada máquina del Grid.

$$\min_{f(j) \leq i \leq r(j)} \sqrt{\frac{\sum_{f(j) \leq i \leq r(j)} (PL_i - \overline{PL})^2}{r(j) - f(j) + 1}} \quad (8)$$

donde $PL_i = \sum_{g(J_k)=M_i} \frac{\text{size}_k}{m_i}$ es la carga paralela de la máquina M_i y $\overline{PL} = \sum_{f(j) \leq i \leq r(j)} \frac{PL_i}{r(j) - f(j) + 1}$ es la media de las cargas paralelas del conjunto de máquinas admisibles.

5. *Min_LB*. (Madrigal *et al.*, 2006) Selecciona la máquina con la menor cota inferior de tiempo de terminación de las tareas. En lugar del tiempo real de ejecución de una tarea, que no está disponible en calendarización no clarividente, se usa el valor proporcionado por el usuario al someter la tarea al sistema (tiempo solicitado).

$$\min_{f(j) \leq i \leq r(j)} \left(\sum_{g(J_k)=M_i} \frac{\text{size}_k \cdot p_k}{m_i} \right) \quad (9)$$

6. *Min_CT*. (Madrigal *et al.*, 2006) Se elige la máquina con el menor tiempo de terminación estimado de todas sus tareas asignadas.

$$\min_{f(j) \leq i \leq r(j)} (C_{max}^i) \quad (10)$$

donde $C_{max}^i = \max_{g(J_k)=M_i} (C_k^i)$ es el tiempo máximo de terminación de las tareas en la máquina M_i y C_k^i es el tiempo de terminación de la tarea J_k en la máquina M_i .

7. *Min_SWCT*. (Madrigal *et al.*, 2006) Se elige la máquina con la menor suma del producto del tamaño con el tiempo de terminación estimado de cada una de las tareas asignadas.

$$\min_{f(j) \leq i \leq r(j)} \left(\sum_{g(J_k)=M_i} size_k \cdot C_k^i \right) \quad (11)$$

8. *Min_WT*. (Madrigal *et al.*, 2006) Estrategia que selecciona la máquina con el promedio menor de tiempo de espera de las tareas asignadas.

$$\min_{f(j) \leq i \leq r(j)} \left(\sum_{g(J_k)=M_i} \frac{C_k^i - r_k - p_k}{n_i} \right) \quad (12)$$

9. *Min_WWT*. (Madrigal *et al.*, 2006) Se elige la máquina que tiene el menor promedio del producto del tamaño con el tiempo de espera de cada una de las tareas asignadas a la máquina.

$$\min_{f(j) \leq i \leq r(j)} \left(\sum_{g(J_k)=M_i} \frac{(C_k^i - r_k - p_k) \cdot size_k}{n_i} \right) \quad (13)$$

10. *Min_U*. (Madrigal *et al.*, 2006) Se elige la máquina con la menor utilización de sus procesadores.

$$\min_{f(j) \leq i \leq r(j)} \left(\frac{W_{total}^i}{C_{max}^i \cdot m_i} \right) \quad (14)$$

donde $W_{total}^i = \sum_{g(J_k)=M_i} size_k \cdot p_k$ es el trabajo realizado por la máquina M_i .

11. *Min_ST*. (Madrigal *et al.*, 2006) Se selecciona la máquina que ofrece el menor tiempo de inicio de ejecución para la tarea a asignar.

$$\min_{f(j) \leq i \leq r(j)} (C_j^i - r_j) \quad (15)$$

12. *Min_TA*. (Madrigal *et al.*, 2006) Se selecciona la máquina que tiene el promedio menor de tiempo de permanencia en el sistema de las tareas asignadas.

$$\min_{f(j) \leq i \leq r(j)} \left(\sum_{g(j_k)=M_i} \frac{C_k^i - r_k}{n_i} \right) \quad (16)$$

13. *Min_WTA*. (Błażewicz *et al.*, 2007) Se selecciona la máquina que tiene el promedio menor de tiempo de permanencia en el sistema de las tareas asignadas. Este tiempo lo pondera el tamaño de la tarea.

$$\min_{f(j) \leq i \leq r(j)} \left(\sum_{g(j_k)=M_i} \frac{(C_k^i - r_k) \cdot size_k}{n_i} \right) \quad (17)$$

14. *Min_WWoTA*. (Błażewicz *et al.*, 2007) Se selecciona la máquina que tiene el promedio menor de tiempo de permanencia en el sistema de las tareas asignadas. Este tiempo lo pondera el trabajo de la tarea.

$$\min_{f(j) \leq i \leq r(j)} \left(\sum_{g(j_k)=M_i} \frac{(C_k^i - r_k) \cdot (p_k \cdot size_k)}{n_i} \right) \quad (18)$$

Las fórmulas utilizadas en cada una de las estrategias toman en cuenta la información correspondiente a las tareas previamente asignadas más la que está por asignarse. Esto significa que el valor resultante es un estimado de cuál sería el valor de la métrica al asignar la nueva tarea a dicha máquina.

Las estrategias de asignación de tareas se dividen en 5 niveles de conocimiento dependiendo de la información que utilizan relacionada a la tarea y de las máquinas del Grid computacional. La información utilizada en cada nivel incluye la información disponible en el nivel anterior.

- Nivel 0: Se conoce el número de máquinas y el tamaño de cada una de ellas. El grado de paralelismo de cada tarea se utiliza únicamente para conocer el conjunto de máquinas admisibles. La estrategia *Random* es un ejemplo de este nivel.

- Nivel 1: Adicionalmente a la información del nivel 0, se encuentra disponible la carga de tareas de cada máquina, es decir, el número de tareas esperando su ejecución en cada máquina. La estrategia *Min_Lp* corresponde a este nivel de conocimiento.
- Nivel 2: Adicionalmente a la información del nivel 1, el grado de paralelismo de cada tarea se utiliza durante los cálculos de la estrategia. Las estrategias *Min_PL* y *Min_LBal* pertenecen a este nivel.
- Nivel 3. Además de la información disponible en el nivel 2, se considera calendarización clarividente, esto es, el tiempo de ejecución de cada tarea está disponible desde el momento en que ésta llega al sistema. La estrategia *Min_LB* asigna una tarea a la máquina con la menor carga total restante de todas las tareas ya asignadas a esta máquina. Como el tiempo de ejecución real de la tarea no está disponible, se utiliza en su lugar un estimado de este tiempo. Tal estimado puede generarse automáticamente a partir de archivos históricos, o proporcionarlo el usuario cuando envía la tarea.
- Nivel 4: Se tiene acceso a toda la información del nivel 3 y también a todos los calendarios locales. Las estrategias *Min_CT*, *Min_SWCT*, *Min_WT*, *Min_WWT*, *Min_U*, *Min_ST*, *Min_TA*, *Min_WTA*, *Min_WWoTA* se encuentran en este nivel de conocimiento.

Los niveles 0 a 2 describen una calendarización no clarividente. Las estrategias de niveles 0 a 3 están basadas solamente en los parámetros de las tareas, y no necesitan información alguna de los calendarios locales.

III. 4. 2 Calendarización local

Según resultados previos, los algoritmos para calendarización local más utilizados actualmente en sistemas paralelos reales son *FCFS* y *EASY* (Extensible Argonne Scheduling sYstem) (Etsion y Tsafir, 2005). Este último es una implementación de *FCFS* con *Backfilling* (Lifka, 1995).

En el presente trabajo se utiliza únicamente *FCFS* como estrategia de calendarización local. Es una de las estrategias más simples. El calendarizador toma las tareas en el orden en que van llegando al sistema. Si no hay suficientes recursos disponibles, la tarea espera hasta que se liberen recursos y pueda iniciarse mientras las demás tareas permanecen detenidas en la cola. Al utilizar *Backfilling*, cuando no hay suficientes procesadores disponibles para ejecutar la tarea que se encuentra al inicio de la cola de espera, el calendarizador selecciona otras tareas de la cola que se adapten a los procesadores disponibles sin que se afecte el tiempo de inicio de ejecución de la tarea al inicio de la cola.

Capítulo IV

Carga de trabajo

Las cargas de trabajo reales (obtenidas de sistemas existentes) y las sintéticas (generadas a partir de modelos) son la principal fuente de información utilizada para las simulaciones. Sin embargo, esta información se debe estandarizar para generalizar su uso. Como resultado, aparecen varios formatos de cargas de trabajo como el Open Trace Format (Knüpfer *et al.*, 2006), el Standard Workload Format (Feitelson, 2005) y el Grid Workload Format (Anoep *et al.*, 2007).

A pesar de la existencia del Grid Workload Format, existe una escasez de cargas de trabajo para Grid computacional. En este capítulo se presenta una propuesta para la generación de cargas de trabajo semi-sintéticas a partir de registros independientes. Se basa en las siguientes consideraciones:

- Las cargas de trabajo para Grid contienen tareas sometidas por usuarios de diferentes sitios.
- El contexto de ejecución del Grid puede extenderse a estos sitios.
- La unificación de estos sitios a un Grid resulta en la combinación de todos los usuarios y sus tareas.

IV. 1. Cargas de trabajo reales y sintéticas

Las cargas de trabajo son necesarias para evaluar la eficiencia de las estrategias de calendarización emergentes en diferentes escenarios. Existen dos fuentes de cargas de trabajo: las cargas de trabajo sintéticas y las obtenidas a partir de sistemas reales (registros). La Tabla II muestra las ventajas y desventajas de cada una de ellas (Lublin y Feitelson, 2003).

Tabla II. Ventajas y desventajas de las cargas de trabajo reales y sintéticas.

Ventajas y desventajas de las cargas de trabajo reales y sintéticas		
Tipo	Ventajas	Desventajas
Real	Refleja tareas reales con todas sus complejidades	Representa un solo sitio con una infraestructura dada Los resultados no se pueden generalizar para modelar otras infraestructuras Puede contener datos erróneos
Sintética	Es posible cambiar los parámetros del modelo para generar otra carga de trabajo Es independiente del sitio	No refleja todas las características de las cargas de trabajo reales La complejidad del modelo aumenta al incluir nuevos detalles Se tiene que cambiar el modelo cuando se introducen nuevas características

Las cargas de trabajo sintéticas y las obtenidas de sistemas reales son la principal fuente de información utilizada para las simulaciones (Calzarossa y Serazzi, 1993 y Calzarossa *et al.*, 1995). Los registros obtenidos de sistemas reales tienen la particularidad de mostrar una imagen real del sistema; sin embargo, éstos pueden estar contaminados con datos erróneos (Lublin y Feitelson, 2003), no incluir toda la información necesaria, y además están limitados a la configuración de un solo sistema. Hoy en día, los registros obtenidos a partir de un Grid computacional son escasos y poco prácticos. La mayor parte de la información aún no se encuentra disponible. Algunos de los esfuerzos para proveer esta información es el Parallel Workloads Archive (PWA) (Feitelson, 2005) y el Grid Workloads Archive (GWA) (Anoep *et al.*, 2007).

Las cargas de trabajo sintéticas son relativamente fáciles de generar. Existen varios modelos propuestos para generar cargas de trabajo sintéticas (Lublin y Feitelson, 2003) que se pueden utilizar para la evaluación y comparación de los nuevos diseños de sistemas. Los

modelos se basan en registros de centros de cómputo reales tales como el San Diego Supercomputer Center, Los Alamos National Lab, el Swedish Royal Institute of Technology, etc. Estos modelos describen la generación de los siguientes componentes:

- Tamaño de la tarea, basado en una distribución uniforme de dos etapas con cuatro parámetros. Los dos primeros parámetros especifican el tamaño máximo y mínimo, los otros dos, la fracción de tareas secuenciales y las potencias de dos.
- Tiempo de ejecución de la tarea, basado en una distribución híper-gama.
- Tiempo de llegada de la tarea, basado en dos distribuciones gama, una para el periodo pico y la otra para el ciclo diario.

IV. 2. Formatos disponibles

La información relacionada a tareas independientes se encuentra disponible desde hace más de una década en la forma de registros de sistemas y de ejecución de aplicaciones. Tales registros se obtienen por instrumentación, ejecución y análisis de programas secuenciales y/o paralelos. Ciertas librerías y aplicaciones tales como TAU, PAPI y PERF se han utilizado para recopilar información relacionada a la ejecución de las tareas. Inicialmente no se estandarizaba esta información, pero en años recientes se ha puesto especial atención en este aspecto; como resultado, aparece el Open Trace Format (OTF), el Standard Workload Format (SWF) y el Grid Workload Format (GWF).

IV. 2. 1 Open Trace Format (OTF)

El Open Trace Format apareció como una especificación para crear y analizar registros de sistemas de Cómputo de Alto Desempeño (HPC). Pretende proveer una especificación amplia, flexible y eficaz que sirva como base para sistemas de Computación y Simulación Avanzadas (ASC). Los registros OTF se encuentran en formato ASCII, compuestos por un archivo maestro y varios archivos secundarios que almacenan información tal como:

- Tiempo de resolución, conteo de procesos, etc.
- Eventos de ejecución (entrada/salida de funciones), comunicación inter-procesos (envío/recepción) y contadores de hardware en diferentes grados de abstracción
- Información de estatus
- Información estadística, que provee un panorama de un intervalo de tiempo

El formato OTF soporta Fortran (77, 90 y 95) y C/C++. No provee información relacionada al formato interno de los archivos y no incluye información para una infraestructura de Grid.

IV. 2. 2 Standard Workload Format (SWF)

La motivación detrás del Standard Workload Format es proveer un formato estandarizado para registros y modelos de cargas de trabajo para tareas independientes y cadenas (Chapin *et al.*, 1999). Feitelson mantiene el archivo PWA (Feitelson, 2005), el cual provee cargas de trabajo para sistemas HPC. Actualmente este archivo almacena 10 modelos y 25 registros. El SWF utiliza un formato de archivo ASCII que es relativamente simple de interpretar. Los registros SWF proveen información sobre de ejecución de tareas tal como: número de tarea, tiempo de llegada, tiempo de ejecución, número de procesadores asignados, etc.

IV. 2. 3 Grid Workload Format (GWF)

El Grid Workload Format se diseñó para modelar datos de cargas de trabajo para Grid. Excluye algunos campos del formato original SWF e incluye campos para la información de origen (sitio en donde la tarea se ejecutó), estructura de la tarea, recursos solicitados e información de la organización virtual. La tarea puede ser compuesta (conjunto de tareas con relaciones de precedencia) o unitaria (independiente). Una tarea compuesta se almacena como DAG, DG, cadena o Bolsa de tareas.

El formato GWF almacena un DAG como un conjunto de tareas compuestas, pero no soporta otras estructuras. Una tarea de un DAG mantiene relaciones de precedencia al

almacenar una tarea predecesora y una sucesora. Para restaurar el DAG se requiere interpretar todo el registro.

IV. 3. Cargas de trabajo para Grid computacional

Existen pocos estudios enfocados en la recolección de información de sistemas de cómputo. Uno de estos esfuerzos para obtener información real de aplicaciones para Grid y presentarlas como cargas de trabajo es el GWA. Provee registros de varios Grids y sistemas paralelos. Estas cargas de trabajo reales contienen principalmente tareas independientes y algunas compuestas.

IV. 3. 1 Carga de trabajo semi-sintética

Existe una escasez de cargas de trabajo para simulaciones de calendarización en Grid computacional. Debido a esto se deben utilizar nuevos enfoques. Proponemos la generación de una carga de trabajo semi-sintética a partir de registros reales de súper computadoras paralelas. De esta manera se crea una carga de trabajo que se puede utilizar para simulaciones de calendarización en un Grid.

La premisa se basa en las siguientes consideraciones:

- Las cargas de trabajo para Grid contienen tareas sometidas por usuarios de diferentes sitios.
- El contexto de ejecución del Grid puede extenderse a estos sitios.
- La unificación de estos sitios a un Grid resulta en la combinación de todos los usuarios y sus tareas.

El unir registros de varios sitios independientes para simular una carga de trabajo de un Grid computacional no garantiza una adecuada representación del Grid con los mismos usuarios. Por ejemplo, si un sitio se vuelve parte de un Grid computacional, donde máquinas más grandes están disponibles, los usuarios pueden someter tareas más grandes que no están representadas en el registro original. A pesar de esto, se considera que es un

buen punto de inicio para evaluar estrategias de calendarización para Grid computacional en base a registros reales.

Algunos registros pueden no tener toda la información requerida para una simulación. Por ejemplo, el tiempo estimado del usuario (tiempo solicitado por el usuario para ejecutar la tarea) generalmente no se provee, pero muchas estrategias de calendarización lo requieren. Para tal caso, existen modelos que generan estos tiempos estimados (Tsafirir *et al.*, 2005).

Proceso de generación de cargas de trabajo semi-sintéticas	
1.	cargar la información de las cabeceras de los archivos fuentes (cargas de trabajo)
2.	cargar la información de tareas de los archivos fuentes (cargas de trabajo)
3.	para cada carga de trabajo fuente hacer
4.	normalizar la zona horaria de acuerdo a la zona horaria deseada
5.	fin
6.	para cada carga de trabajo fuente hacer
7.	encontrar la primera tarea admisible del registro
8.	encontrar la última tarea admisible del registro
9.	fin
10.	para cada carga de trabajo fuente hacer
11.	aplicar filtros para eliminar tareas no deseadas
12.	fin
13.	numero_de_usuarios := 0
14.	para cada carga de trabajo fuente hacer
15.	para cada usuario hacer
16.	numero_de_usuarios := numero_de_usuarios + 1
17.	identificador_de_usuario := numero_de_usuarios
18.	fin
19.	fin
20.	mientras (tareas guardadas < número total de tareas que serán guardadas) hacer
21.	tarea_a_guardar := tarea no guardada con el tiempo de llegada mínimo en todas las cargas de trabajo
22.	marcar la tarea como guardada
23.	si (tarea_a_guardar = la última tarea admisible de la carga de trabajo) entonces
24.	marcar como no guardadas todas las tareas admisibles de la carga de trabajo
25.	incrementar todos los tiempos de llegada de las tareas de la carga de trabajo, agregando el número de segundos en el periodo entre la primera y última tarea admisible.
26.	fin
27.	guardar la tarea y su información relacionada con un número de identificación progresivo
28.	fin
29.	guardar la cabecera de la nueva carga de trabajo generada

Figura 9. Algoritmo del proceso de generación de cargas de trabajo semi-sintéticas.

El utilizar registros reales y generar una carga de trabajo semi-sintética es mucho más fácil que obtenerla a partir de un monitoreo constante de un Grid computacional. Sin embargo, requiere una cuidadosa selección de los registros fuente y una definición de la configuración del Grid que se utilizará para la simulación. Este proceso de generación

necesita ser bien definido para obtener resultados comprobables, reproducibles y correctos. El proceso incluye seis etapas principales descritas a continuación. Los resultados estadísticos presentados en los siguientes párrafos se obtuvieron al analizar la información de los registros originales. La Figura 9 muestra el algoritmo de este proceso.

IV. 3. 1. 1. Normalización de la zona horaria

La normalización del tiempo de llegada de las tareas es importante. La cantidad de tareas sometidas por los usuarios depende de la hora del día (ver Figura 10 y Figura 11). Si se utilizan dos registros sin normalización (por ejemplo GMT-6 y GMT+6), la forma típica de la distribución de tareas sometidas por día no cambiaría. Al estar normalizados los registros, esta distribución será más uniforme debido al desplazamiento horario de los registros. En un Grid computacional global, los usuarios de diferentes zonas horarias contribuyen de una manera uniforme, caso contrario sería para los usuarios de un solo centro de cómputo. La Figura 10 y la Figura 11 muestran histogramas de cargas de trabajo del San Diego Supercomputer Center y el Swedish Royal Institute of Technology (Feitelson, 2005) sin una normalización de la zona horaria. Debido a la normalización, la distribución se vuelve más uniforme, ver Figura 12.

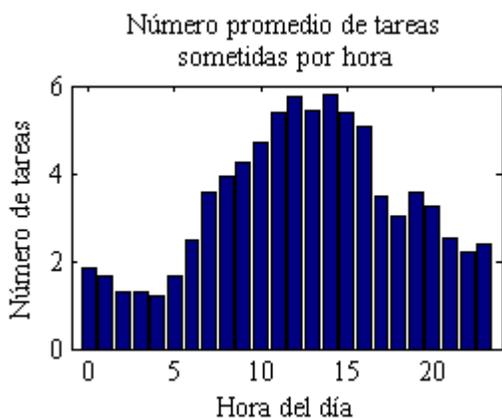


Figura 10. Número promedio de tareas sometidas por hora. Registro SDSC-SP2-1998-3.1-cln.swf, GMT-8.

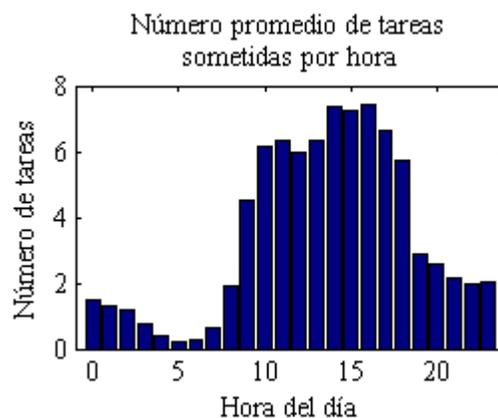


Figura 11. Número promedio de tareas sometidas por hora. Registro KTH-SP2-1996-2.swf, GMT+1.

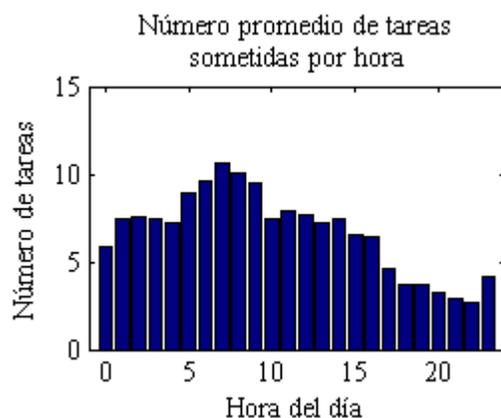


Figura 12. Número promedio de tareas sometidas por hora. Carga de trabajo semi-sintética resultante, GMT-8.

Se proponen tres formas para normalizar los registros fuentes.

- Zona horaria mínima: El valor numérico mínimo de todas las zonas horarias se usa como base para normalizar los registros. Para el ejemplo anterior, con GMT-6 y GMT+6, si ambos registros comienzan a media noche de la hora local, las zonas horarias resultantes serían GMT-6 00:00 horas para el primer registro y GMT-6 12:00 horas -1 día para el segundo. En esta normalización se evita modificar el valor de los tiempos de llegada de cada tarea en el registro.
- Zona horaria definida por el usuario: Es similar a la normalización anterior, pero en este caso el usuario define la zona horaria base para la normalización.
- Zona horaria moda: La zona horaria más común entre los registros se usa para normalizarlos.

IV. 3. 1. 2. Selección de tareas admisibles

Una *tarea admisible* es aquella que se considera y se puede incluir en la carga de trabajo semi-sintética generada. Se define el siguiente conjunto de condiciones para filtrar las tareas no admisibles y un pre procesamiento de los tiempos de llegada de cada tarea admisible.

- Desplazamiento semanal: Se definen la primera y última tarea admisible. La primera tarea admisible es la primera tarea del primer lunes del registro. La última

tarea admisible es la última tarea del último domingo del registro. Todos los tiempos de llegada de las tareas admisibles se recalculan como si su respectivo registro fuente comenzara en lunes.

- Desplazamiento anual: El número de tareas sometidas durante el intervalo de un año tiene un patrón particular. Por ejemplo, los usuarios someten menos tareas computacionales en Navidad. Por lo tanto, si se desea conservar este comportamiento, se necesita definir un conjunto de tareas admisibles adecuado. En lugar de utilizar el primer lunes y último domingo de cada registro, se usa el primer lunes más próximo al 1 de enero, y el último domingo más próximo al 31 de diciembre de cada registro. Esta manera tan particular de definir las tareas admisibles puede aplicarse únicamente a los registros que contienen la información correspondiente a un año completo.

IV. 3. 1. 3. Normalización de números de usuario

Los números de usuario también se deben normalizar. Esto elimina grandes rangos vacíos en la numeración y evita duplicidad en la carga de trabajo generada. Para el primer registro, los números de usuario se recalculan comenzando desde 1. Un proceso similar se aplica para los registros subsecuentes, excepto que los números de usuario de cada registro comenzarán con el siguiente número de usuario del registro anterior. Este proceso asegura que cada usuario se represente de manera única e independiente en la carga de trabajo generada; además evita la pérdida de información. Si todas las tareas de los registros originales se utilizan para la carga de trabajo semi-sintética, entonces todos los usuarios estarán presentes en la misma.

IV. 3. 1. 4. Filtrado de tareas no válidas

Las tareas irrelevantes para un experimento o con información incorrecta se eliminan de la carga de trabajo final. Varios filtros se aplican para este propósito. Si alguna de las siguientes condiciones se satisface, la tarea se remueve.

- Número de tarea menor o igual a cero
- Tiempo de llegada menor que cero

- Tiempo de ejecución menor o igual a cero
- Número de procesadores asignados menor o igual a cero
- Tiempo solicitado menor o igual a cero
- Número de usuario menor o igual a cero
- Estatus igual a cero
- Estatus igual a cuatro
- Estatus igual a cinco
- Tiempo de ejecución mayor que tiempo solicitado
- Número de procesadores asignados menor que lim_{inf}
- Número de procesadores asignados mayor que lim_{sup}

Es importante mencionar que los filtros se aplican opcionalmente. lim_{inf} y lim_{sup} son parámetros de la función generadora.

IV. 3. 1. 5. Generación de la carga de trabajo semi-sintética

Las tareas de todos los registros contribuyen en la generación de la carga de trabajo semi-sintética. Éstas se ordenan de acuerdo a su tiempo de llegada en forma no decreciente y se les asigna un nuevo número de tarea.

IV. 3. 1. 6. Agregación de nuevas tareas

Si un registro original no contiene suficientes tareas para completar la carga de trabajo semi-sintética, las tareas admisibles de este registro se duplican según sea necesario. Se supone que la calidad de la carga de trabajo no cambia significativamente al incrementar su longitud en el tiempo. El tiempo de llegada de estas tareas se recalcula adecuadamente.

Capítulo V

Experimentación

En este capítulo se presenta el análisis experimental de las diferentes estrategias de calendarización en un modelo jerárquico de dos niveles con un esquema de admisibilidad. Además, se describen las diferentes métricas utilizadas para el análisis de los algoritmos, un análisis completo de la carga de trabajo utilizada y los parámetros del experimento.

V. 1. Métricas para el análisis

Las métricas utilizadas para el análisis de las estrategias se dividen en tres grupos: métricas del sistema, métricas del usuario y métricas del algoritmo.

V. 1. 1 De acuerdo al criterio del sistema

Entre las métricas del sistema se encuentran:

- La longitud del calendario (C_{max}): Es el tiempo de finalización de la última tarea que termina su procesamiento. Esta métrica es importante desde el punto de vista del propietario del recurso. Se relaciona directamente con el factor de utilización de los procesadores.

$$\max_{1 \leq j \leq n} (C_j) \quad (19)$$

- Rendimiento de procesamiento (*Throughput*): Número de tareas terminadas por unidad de tiempo.

$$\frac{n}{C_{max}} \quad (20)$$

- Utilización del sistema (*Utilization*): Se define como la fracción del tiempo en la cual el sistema se utilizó.

$$\frac{\sum_{1 \leq j \leq n} (size_j \cdot p_j)}{C_{max} \cdot (\sum_{1 \leq i \leq m} m_i)} \quad (21)$$

- Trabajo total realizado (*Total work*): Es la cantidad de trabajo realizado por el sistema. Se obtiene al sumar el producto del tamaño por tiempo de ejecución de cada tarea.

$$\sum_{1 \leq j \leq n} (size_j \cdot p_j) \quad (22)$$

- Recursos no utilizados (*Total idle*): Suma de los tiempos de ocio de todos los procesadores en el sistema.

$$C_{max} \cdot \left(\sum_{1 \leq i \leq m} m_i \right) - \sum_{1 \leq j \leq n} (size_j \cdot p_j) \quad (23)$$

V. 1. 2 De acuerdo al criterio del usuario

Otras métricas son más importantes desde el punto de vista del usuario, entre ellas se encuentran:

- Tiempo promedio de permanencia (*Mean turnaround*): Tiempo promedio que una tarea permanece en el sistema, contando desde su llegada al sistema y hasta el término de su ejecución.

$$\frac{1}{n} \sum_{1 \leq j \leq n} (C_j - r_j) \quad (24)$$

- Tiempo promedio de permanencia ponderado (*Mean weighted turnaround*): Tiempo promedio que una tarea permanece en el sistema, contando desde su llegada al sistema y hasta el término de su ejecución. Este tiempo se pondera con el tamaño de la tarea.

$$\frac{1}{n} \sum_{1 \leq j \leq n} ((C_j - r_j) \cdot size_j) \quad (25)$$

- Tiempo promedio de permanencia ponderado por trabajo (*Mean weighted (work) turnaround*): Tiempo promedio que una tarea permanece en el sistema, contando desde su llegada al sistema y hasta el término de su ejecución. Este tiempo se pondera con el trabajo de la tarea.

$$\frac{1}{n} \sum_{1 \leq j \leq n} \left((C_j - r_j) \cdot (size_j \cdot p_j) \right) \quad (26)$$

- Tiempo promedio de espera (*Mean waiting time*): Tiempo promedio que una tarea espera antes de comenzar su ejecución, contando desde su llegada al sistema.

$$\frac{1}{n} \sum_{1 \leq j \leq n} (C_j - r_j - p_j) \quad (27)$$

- Tiempo promedio de espera ponderado (*Mean weighted waiting time*): Tiempo promedio que una tarea espera antes de comenzar su ejecución, contando desde su llegada al sistema. El tiempo se pondera con el tamaño de la tarea.

$$\frac{1}{n} \sum_{1 \leq j \leq n} \left((C_j - r_j - p_j) \cdot size_j \right) \quad (28)$$

- Tiempo de respuesta promedio (*Mean response time*): Es el tiempo transcurrido desde que la tarea llega al sistema hasta la primera respuesta al usuario por parte de la misma. En este caso, este tiempo se considera hasta la terminación de la tarea.

$$\frac{1}{n} \sum_{1 \leq j \leq n} (C_j - r_j) \quad (29)$$

- Cociente de respuesta promedio (*Mean response ratio*): Es el cociente promedio del tiempo de respuesta de una tarea y su tiempo de procesamiento.

$$\frac{1}{n} \sum_{1 \leq j \leq n} \frac{C_j - r_j}{p_j} \quad (30)$$

- Cociente de respuesta promedio acotado (*Mean bounded response ratio*): Es el cociente promedio del tiempo de respuesta de una tarea y su tiempo de procesamiento. Los tiempos de ejecución menores a diez se sustituyen por esta última cantidad.

$$\frac{1}{n} \sum_{1 \leq j \leq n} \frac{C_j - r_j}{\max(p_j, 10)} \quad (31)$$

- Cociente de respuesta del sistema (*System response ratio*): Es el cociente promedio de la suma total del tiempo de respuesta de las tareas y la suma total de sus tiempos de procesamiento.

$$\frac{1}{n} \left(\frac{\sum_{1 \leq j \leq n} C_j - \sum_{1 \leq j \leq n} r_j}{\sum_{1 \leq j \leq n} p_j} \right) \quad (32)$$

V. 1. 3 De acuerdo al criterio del algoritmo

Finalmente, la métrica del algoritmo más utilizada es:

- Cociente de competitividad (*Competitive ratio*): Está definido como el cociente de la longitud del calendario generado por la estrategia de calendarización y la longitud del calendario óptimo. Sin embargo, dado la dificultad para obtener el valor de la longitud del calendario óptimo, se utiliza una cota mínima. De esta manera el cociente de competitividad del algoritmo de calendarización se define como el cociente de la longitud del calendario generado por la estrategia de calendarización y la cota mínima de finalización del calendario.

$$\frac{C_{max}}{C_{max}^*} \quad (33)$$

V. 2. Carga de trabajo utilizada

El archivo *mix_gs_log_36770jobs_30exp(1111110000,0-32)r.swf* es una carga de trabajo semi-sintética, generada a partir de 5 registros reales de centros de cómputo de alto desempeño alrededor del mundo. Cuatro cargas de trabajo provenientes del SWA (Feitelson, 2005): Cornell Theory Center (Tabla III), High Performance Computing Center North (Tabla V), Swedish Royal Institute of Technology (Tabla VI) y Los Alamos National

Lab (Tabla VII). Una proveniente del GWA (Anoep *et al.*, 2007): The Advanced School for Computing and Imaging (Tabla IV).

Estos registros se incluyeron en la carga de trabajo semi-sintética debido a que contienen información importante acerca de las tareas como: número de usuario, tiempo de ejecución, tiempo solicitado, número de procesadores asignados, tiempo de llegada, tipo de tarea, etc. Además se incluye información relacionada al centro de cómputo de origen como: computadora, número de procesadores, zona horaria, periodo del registro, entre otras.

La carga de trabajo semi-sintética contiene 1,103,100 tareas, alrededor de 467 días. Equivalen a 30 experimentos de una semana cada uno; durante la primera semana se sometieron 36,770 tareas, se supone la misma cantidad en las semanas subsecuentes. La zona horaria después de la normalización de los registros originales es GMT-7, la mínima encontrada. Para los fines del experimento, las tareas con tamaños mayores a 32 procesadores se eliminaron de la carga de trabajo semi-sintética.

Tabla III. Registro del Cornell Theory Center (CTC).

CTC-SP2-1996-2.1-cls.swf	
Instalación	Cornell Theory Center (CTC)
Computadora y tamaño	IBM SP2, 430 procesadores
Zona horaria	-5 (EUA/Este)
Inicio del registro	Miércoles, 26 de junio de 1996, 17:06:00
Fin del registro	Sábado, 31 de mayo de 1997, 23:11:26
Número de tareas	79,302
Comentarios	Utiliza el calendarizador EASY
Información del registro en la carga de trabajo semi-sintética	
Números de usuario normalizados	1 - 673
Tareas admisibles (número de línea del registro original)	38 - 75,904
Número de tareas usadas	102,456
Número de tareas usadas antes de repetirlas	71,101
Número de veces que el registro fue utilizado	2
Número de tareas filtradas (número de tarea ≤ 0)	0
Número de tareas filtradas (tiempo de llegada < 0)	0
Número de tareas filtradas (tiempo de ejecución ≤ 0)	17
Número de tareas filtradas (número de procesadores asignados ≤ 0)	0

Número de tareas filtradas (tiempo solicitado ≤ 0)	6
Número de tareas filtradas (número de usuario ≤ 0)	0
Número de tareas filtradas (estatus == 0)	No se aplicó
Número de tareas filtradas (estatus == 4)	No se aplicó
Número de tareas filtradas (estatus == 5)	No se aplicó
Número de tareas filtradas (tiempo de ejecución $>$ tiempo solicitado)	No se aplicó
Número de tareas filtradas (número de procesadores asignados < 0)	No se aplicó
Número de tareas filtradas (número de procesadores asignados > 32)	4,743

Tabla IV. Registro de The Advanced School for Computing and Imaging (DAS-2 system).

Gwa-t-1 (anon_jobs_sqlite.zip)	
Instalación	The Advanced School for Computing and Imaging (DAS-2 system)
Computadora y tamaño	5 sitios, 400 procesadores
Zona horaria	UTC
Inicio del registro	Martes, 22 de febrero de 2005, 15:52:25 (actualizado al domingo, 26 de junio de 2005, 15:52:38)
Fin del registro	Miércoles, 13 de diciembre de 2006, 09:29:48
Número de tareas	1,124,772 (actualizado a 1,075,762)
Comentarios	Los primeros 124 días del registro se eliminaron debido a la escasez de tareas. 49,019 tareas sometidas durante este periodo, un promedio de 395.24 por día.
Información del registro en la carga de trabajo semi-sintética	
Números de usuario normalizados	674 - 961
Tareas admisibles (número de línea del registro original)	5,500 - 1,069,056
Número de tareas usadas	871,411
Número de tareas usadas antes de repetirlas	871,411
Número de veces que el registro fue utilizado	1
Número de tareas filtradas (número de tarea ≤ 0)	0
Número de tareas filtradas (tiempo de llegada < 0)	0
Número de tareas filtradas (tiempo de ejecución ≤ 0)	16,392
Número de tareas filtradas (número de procesadores asignados ≤ 0)	0
Número de tareas filtradas (tiempo solicitado ≤ 0)	651
Número de tareas filtradas (número de usuario ≤ 0)	0
Número de tareas filtradas (estatus == 0)	No se aplicó
Número de tareas filtradas (estatus == 4)	No se aplicó
Número de tareas filtradas (estatus == 5)	No se aplicó
Número de tareas filtradas (tiempo de ejecución $>$ tiempo solicitado)	No se aplicó
Número de tareas filtradas (número de procesadores asignados < 0)	No se aplicó
Número de tareas filtradas (número de procesadores asignados > 32)	3,775

Tabla V. Registro del High Performance Computing Center North, Sweden.

HPC2N-2002-1.1-cls.swf	
Instalación	High Performance Computing Center North, Sweden (HPC2N)
Computadora y tamaño	Cluster Linux (Seth), 240 procesadores
Zona horaria	1 (Europa/Estocolmo)
Inicio del registro	Domingo, 28 de julio de 2002, 09:04:05
Fin del registro	Lunes, 16 de enero de 2006, 20:31:24
Número de tareas	527,371
Comentarios	
Información del registro en la carga de trabajo semi-sintética	
Números de usuario normalizados	962 - 1,218
Tareas admisibles (número de línea del registro original)	4 - 202,856
Número de tareas usadas	55,115
Número de tareas usadas antes de repetirlas	55,115
Número de veces que el registro fue utilizado	1
Número de tareas filtradas (número de tarea ≤ 0)	0
Número de tareas filtradas (tiempo de llegada < 0)	0
Número de tareas filtradas (tiempo de ejecución ≤ 0)	51
Número de tareas filtradas (número de procesadores asignados ≤ 0)	0
Número de tareas filtradas (tiempo solicitado ≤ 0)	0
Número de tareas filtradas (número de usuario ≤ 0)	0
Número de tareas filtradas (estatus == 0)	No se aplicó
Número de tareas filtradas (estatus == 4)	No se aplicó
Número de tareas filtradas (estatus == 5)	No se aplicó
Número de tareas filtradas (tiempo de ejecución $>$ tiempo solicitado)	No se aplicó
Número de tareas filtradas (número de procesadores asignados < 0)	No se aplicó
Número de tareas filtradas (número de procesadores asignados > 32)	5,045

Tabla VI. Registro del Swedish Royal Institute of Technology (KTH).

KTH-SP2-1996-2.swf	
Instalación	The Swedish Royal Institute of Technology (KTH)
Computadora y tamaño	IBM SP2, 100 procesadores
Zona horaria	1 (Europa/Estocolmo)
Inicio del registro	Lunes, 23 de septiembre de 1996, 14:00:31
Fin del registro	Viernes, 29 de agosto de 1997, 10:55:01
Número de tareas	28,490
Comentarios	Utiliza el calendarizador EASY
Información del registro en la carga de trabajo semi-sintética	
Números de usuario normalizados	1,219 - 1,423
Tareas admisibles (número de línea del registro original)	1 - 28,092
Número de tareas usadas	35,980
Número de tareas usadas antes de repetirlas	26,938
Número de veces que el registro fue utilizado	2
Número de tareas filtradas (número de tarea ≤ 0)	0
Número de tareas filtradas (tiempo de llegada < 0)	0

Número de tareas filtradas (tiempo de ejecución ≤ 0)	8
Número de tareas filtradas (número de procesadores asignados ≤ 0)	0
Número de tareas filtradas (tiempo solicitado ≤ 0)	0
Número de tareas filtradas (número de usuario ≤ 0)	0
Número de tareas filtradas (estatus == 0)	No se aplicó
Número de tareas filtradas (estatus == 4)	No se aplicó
Número de tareas filtradas (estatus == 5)	No se aplicó
Número de tareas filtradas (tiempo de ejecución > tiempo solicitado)	No se aplicó
Número de tareas filtradas (número de procesadores asignados < 0)	No se aplicó
Número de tareas filtradas (número de procesadores asignados > 32)	1,146

Tabla VII. Registro de Los Alamos National Lab (LANL).

LANL-CM5-1994-3.1-cln.swf	
Instalación	Los Alamos National Lab (LANL)
Computadora y tamaño	Thinking Machines CM-5, 1024 procesadores
Zona horaria	-7 (EUA/Montañas)
Inicio del registro	Martes, 4 de octubre de 1994, 07:01:12
Fin del registro	Martes, 24 de septiembre de 1996, 06:44:55
Número de tareas	201,387
Comentarios	Utiliza DJM como administrador de tareas
Información del registro en la carga de trabajo semi-sintética	
Números de usuario normalizados	1,424 - 1,635
Tareas admisibles (número de línea del registro original)	1,535 - 121,922
Número de tareas usadas	38,138
Número de tareas usadas antes de repetirlas	38,138
Número de veces que el registro fue utilizado	1
Número de tareas filtradas (número de tarea ≤ 0)	0
Número de tareas filtradas (tiempo de llegada < 0)	0
Número de tareas filtradas (tiempo de ejecución ≤ 0)	5
Número de tareas filtradas (número de procesadores asignados ≤ 0)	0
Número de tareas filtradas (tiempo solicitado ≤ 0)	11,175
Número de tareas filtradas (número de usuario ≤ 0)	0
Número de tareas filtradas (estatus == 0)	No se aplicó
Número de tareas filtradas (estatus == 4)	No se aplicó
Número de tareas filtradas (estatus == 5)	No se aplicó
Número de tareas filtradas (tiempo de ejecución > tiempo solicitado)	No se aplicó
Número de tareas filtradas (número de procesadores asignados < 0)	No se aplicó
Número de tareas filtradas (número de procesadores asignados > 32)	53,593

V. 2. 1 Estadísticas de la carga de trabajo

Un análisis estadístico de la carga de trabajo es siempre importante, no sólo para interpretar los resultados de las simulaciones, sino también para sustentarlos. Si no se

conoce nada acerca de la carga de trabajo utilizada, no se puede decir si los resultados de las simulaciones son buenos o malos. Algunas estrategias de calendarización ofrecen buenos resultados con tareas grandes y otras con tareas pequeñas. Las características de la carga de trabajo afectan el desempeño del Grid computacional.

El número de tareas sometidas por hora del día tiene una distribución particular para un solo centro de cómputo. Durante la noche hay una menor cantidad de tareas sometidas. La Figura 13 muestra una distribución diferente, no existe un solo periodo intenso de llegada de tareas. Esto se debe a que los registros originales utilizados para generar la carga de trabajo semi-sintética provienen de zonas horarias diferentes. Al combinar estos registros, la distribución se vuelve más uniforme.

Similar al número de tareas sometidas por hora, el número de tareas sometidas por día de la semana también tiene una distribución particular. En este caso, la distribución se conserva por la selección de tareas admisibles. La mayor parte de las tareas se someten de lunes a viernes de manera regular. Durante los fines de semana, el número de tareas sometidas cae hasta un 40%. Esto significa que los usuarios trabajan menos en fines de semana, ver Figura 14.

El número de tareas sometidas por día del mes tiene una distribución uniforme, exceptuando el día 25 del mes, ver Figura 15. Esta irregularidad la causa la gran cantidad de tareas sometida en las horas 3,534, 3,536, 3,539, 3,541 y 3,542 de la carga de trabajo que coincidieron con el día 25 de algún mes, ver Figura 16. Únicamente existen 12 horas, distribuidas en la carga de trabajo, con más de 1,500 tareas sometidas en cada una de ellas. El resto de la carga de trabajo tiene un número más regular de tareas sometidas por hora.

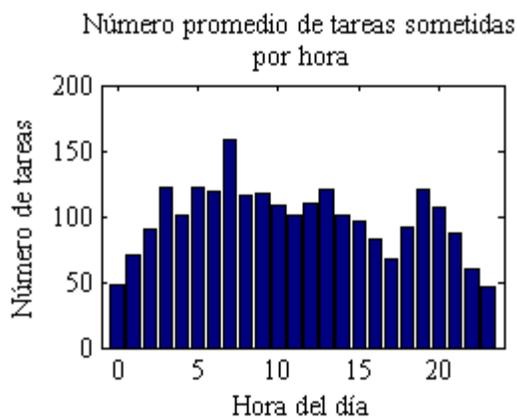


Figura 13. Número promedio de tareas sometidas por hora del día.

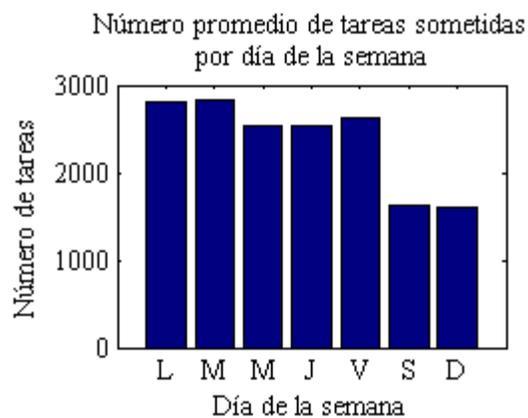


Figura 14. Número promedio de tareas sometidas por día de la semana.

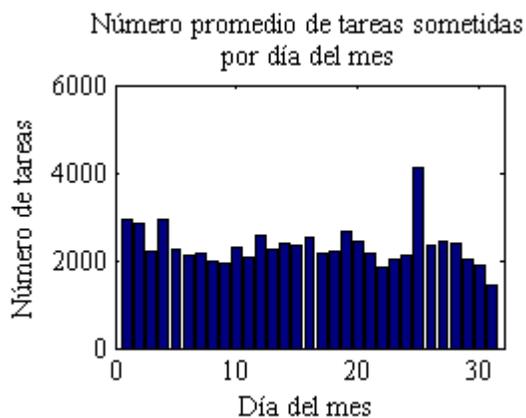


Figura 15. Número promedio de tareas sometidas por día del mes.

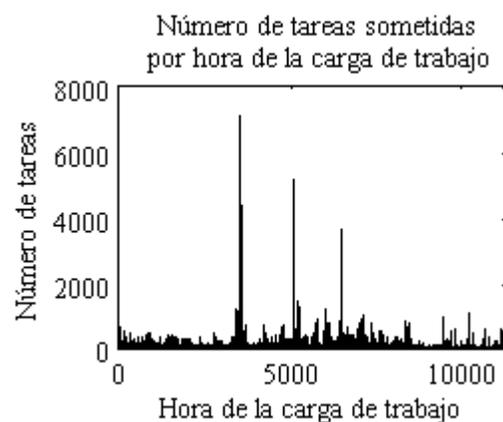


Figura 16. Número de tareas sometidas por hora de la carga de trabajo.

Pero el número de tareas sometidas no dice todo acerca de la carga de trabajo. También es importante saber la cantidad de recursos que requieren estas tareas. La Figura 17 muestra la distribución de recursos consumidos por hora del día, no es similar a la distribución de tareas sometidas por hora del día (Figura 13). Esto significa que la cantidad de tareas no define la cantidad de recursos consumidos. El periodo más intenso corresponde a las 10 horas.

La distribución de tareas sometidas por día de la semana (Figura 14) y los recursos consumidos por esas tareas (Figura 18) tienen una distribución similar. La distribución de

los recursos consumidos por día del mes es uniforme, exceptuando el día 31; no todos los meses tienen 31 días, ver Figura 19.

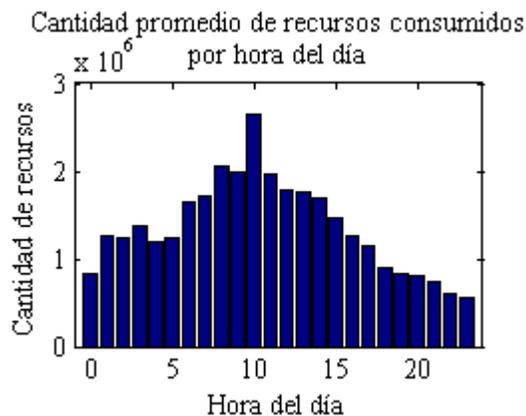


Figura 17. Cantidad promedio de recursos consumidos por hora del día.



Figura 18. Cantidad promedio de recursos consumidos por día de la semana.

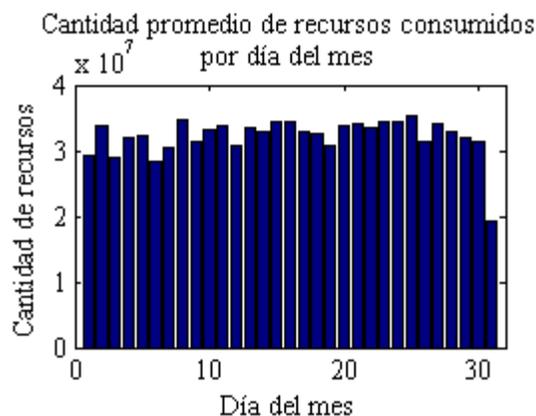


Figura 19. Cantidad promedio de recursos consumidos por día del mes.

La cantidad de tareas sometidas por cada usuario no es regular (Figura 20). Los usuarios del registro de The Advanced School for Computing and Imaging son los más activos con hasta 145,000 tareas por usuario (usuarios con identificadores entre 674 - 961); los demás usuarios contribuyen de una manera más uniforme, sobresaliendo los del registro de High Performance Computing Center North con aproximadamente el doble de tareas (usuarios con identificadores entre 962 - 1,218), ver Figura 21. La Figura 22 muestra una versión ordenada de la cantidad de tareas sometidas por cada usuario. Existen 9 usuarios

con más de 15,000 tareas cada uno, 13 con más de 10,000 y 31 con más de 5,000; el resto de los usuarios tienen una cantidad más uniforme de tareas sometidas, ver Figura 23.

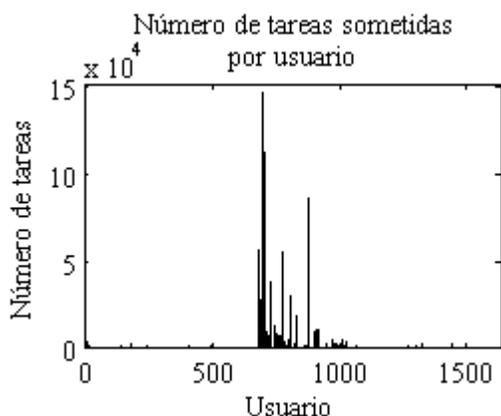


Figura 20. Número de tareas sometidas por usuario.

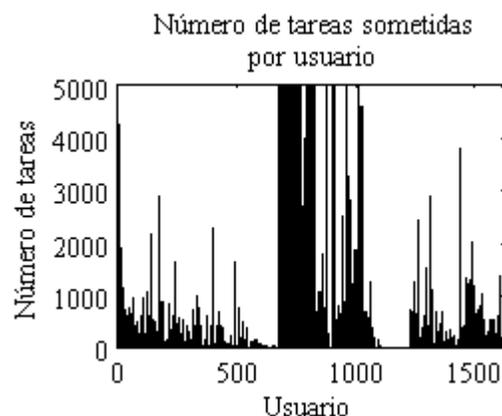


Figura 21. Número de tareas sometidas por usuario, vista detallada de los usuarios con menos de 5,000 tareas.

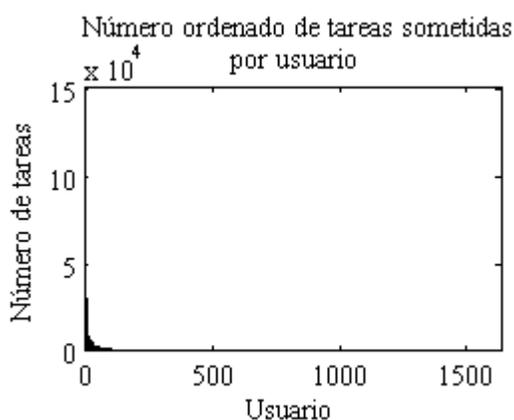


Figura 22. Número ordenado de tareas sometidas por usuario.



Figura 23. Número ordenado de tareas sometidas por usuario, vista detallada de los 100 usuarios con más tareas.

Una característica importante de la carga de trabajo es que el 37% de las tareas sometidas es de tamaño 1, el 23% de tamaño 2. Es principalmente una carga de trabajo secuencial en cuanto a número de tareas, ver Figura 24. Los tamaños más comunes son 1, 2, 4, 8 y 32 con 37%, 23%, 10%, 6% y 5%, respectivamente; para un total de 81% de las tareas sometidas. Los demás tamaños de tarea representan el 19% restante. Los tamaños con menos de 50,000 tareas pueden apreciarse en la Figura 25. Se puede observar una

relación entre el tamaño de la tarea y la cantidad sometida de las mismas, se someten mayor cantidad de tareas pequeñas, y las más comunes son aquellas con tamaño igual a una potencia de dos.

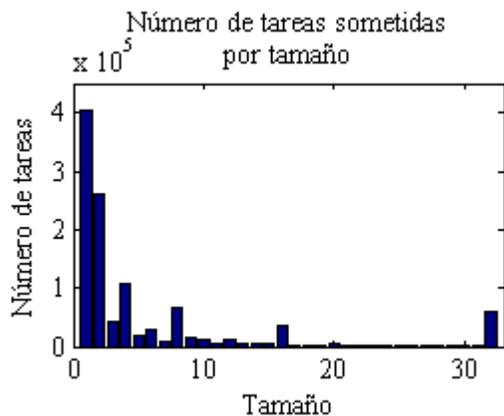


Figura 24. Número de tareas sometidas por tamaño.

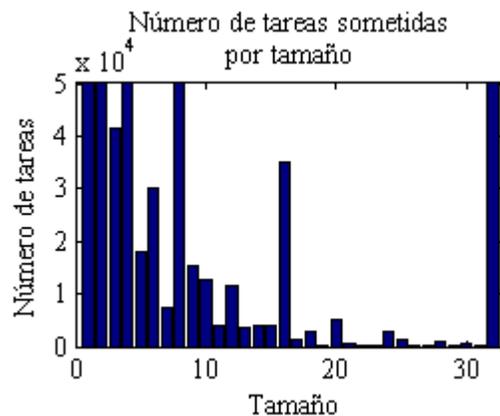


Figura 25. Número de tareas sometidas por tamaño, vista detallada de los tamaños con menos de 50,000 tareas.

Aproximadamente 92% de las tareas sometidas tienen un tiempo de ejecución menor a una hora, 2% entre una y dos horas, 1% entre dos y tres horas. El restante 5% de las tareas tienen tiempos de ejecución mayores a tres horas, ver Figura 26. El tiempo mayor de ejecución de una tarea es de 153 horas. La Figura 27 muestra una vista detallada de los tiempos de ejecución con menos de 10,000 tareas. Existe un contraste con el tiempo solicitado, 83% de tareas con menos de una hora de tiempo solicitado, 4% entre una y dos horas, 2% entre dos y tres horas, 2% entre tres y cinco horas, 2% entre dieciocho y diecinueve horas; ver Figura 28. Se excluyen 49 tareas en esta figura por cuestiones visuales: 1 tarea con tiempo solicitado de 160 horas, 23 con 600 horas y 25 con 16,666 horas. Una vista detallada se puede apreciar en la Figura 29.

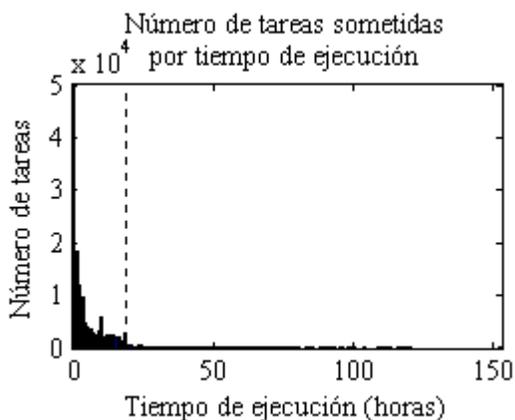


Figura 26. Número de tareas sometidas por tiempo de ejecución, vista detallada de los tiempos de ejecución con menos de 50,000 tareas.

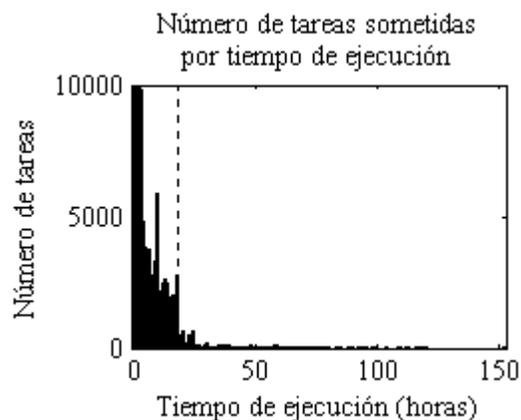


Figura 27. Número de tareas sometidas por tiempo de ejecución, vista detallada de los tiempos de ejecución con menos de 10,000 tareas.

Existe un “límite” de tiempo de ejecución de las tareas, 18 horas. Casi la totalidad de las tareas están dentro de este límite, ver Figura 27. Este patrón no se puede definir claramente en la Figura 29 con los tiempos solicitados por el usuario. Es fácil notar la diferencia entre el tiempo de ejecución y el tiempo solicitado por el usuario para ejecutar su tarea. Los usuarios solicitan un tiempo mayor al que requiere la tarea; quieren evitar que su tarea sea terminada por el sistema al ejecutarse por un tiempo mayor al solicitado. Pero esto no sucede en todos los casos, los registros originales utilizados para crear la carga de trabajo semi-sintética contienen tareas con tiempos de ejecución mayores al tiempo solicitado por el usuario.



Figura 28. Número de tareas sometidas por tiempo solicitado, vista detallada de los tiempos solicitados con menos de 50,000 tareas.



Figura 29. Número de tareas sometidas por tiempo solicitado, vista detallada de los tiempos solicitados con menos de 10,000 tareas.

El tipo de tarea más común es aquella con un tiempo de procesamiento de hasta 18 horas, siendo en mayor medida las tareas pequeñas. Estas mismas tareas pequeñas son las que representan la mayor cantidad de tareas con tiempos de ejecución de más de 18 horas, ver Figura 30. Únicamente las tareas con tamaños de 1, 4 y 8 son las que tienen tiempos de procesamiento mayores a 100 horas. En cambio, las tareas con tamaños 1, 2, 4, 6, 8, 16 y 24 tienen tiempos solicitados de más de 100 horas. También se puede apreciar el hecho que los usuarios solicitan más tiempo del necesario para sus tareas; las tareas se encuentran más dispersas, ver Figura 31.

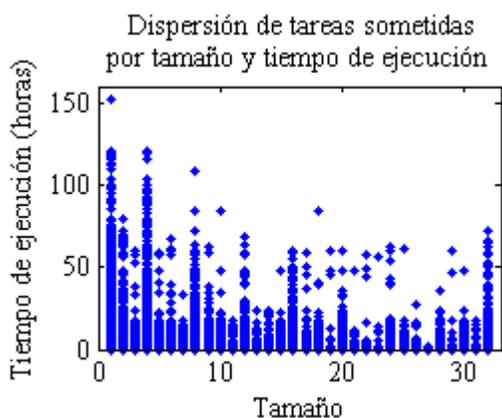


Figura 30. Dispersión de tareas sometidas por tamaño y tiempo de ejecución.

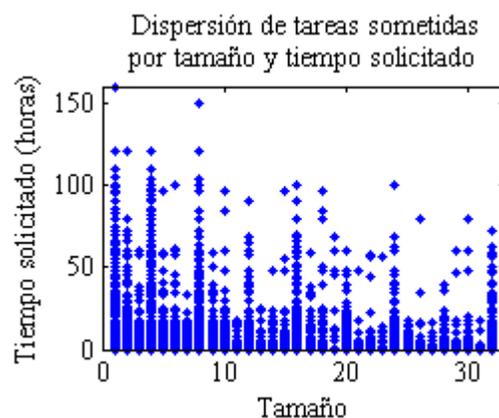


Figura 31. Dispersión de tareas sometidas por tamaño y tiempo solicitado.

Se sabe que los usuarios solicitan tiempos de ejecución mayores a los que realmente necesita la tarea para su ejecución. Por lo tanto, los recursos solicitados son también mayores a los requeridos. Pero ¿qué tan grande es esta diferencia? La relación de los recursos solicitados y consumidos se presenta en la Figura 32. Existen relaciones de hasta más de 2,000,000 de veces, significa un tiempo solicitado de más de 2,000,000 de veces el tiempo de ejecución de la tarea. Esta figura es muy similar a la Figura 24 que representa el número de tareas por tamaño. Esto significa que existe una proporción similar del número de tareas sometidas por tamaño y las malas estimaciones de tiempos de ejecución. Hay una relación entre el tamaño de la tarea y la relación de recursos solicitados y consumidos; al ser más comunes las tareas pequeñas, son las que presentan la mayor cantidad de malas estimaciones. Además es más común una estimación mucho más imprecisa de la duración de la tarea cuando ésta tiene un tamaño pequeño, según se aprecia en la Figura 33. Las tareas con tamaño 1 son las que tienen tiempos solicitados más distantes que los tiempos de ejecución, esto provoca una relación mayor entre los recursos solicitados y los consumidos.

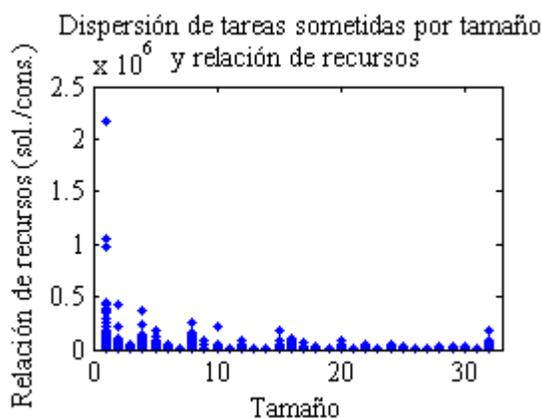


Figura 32. Dispersión de tareas sometidas por tamaño y relación de recursos (solicitados/consumidos).

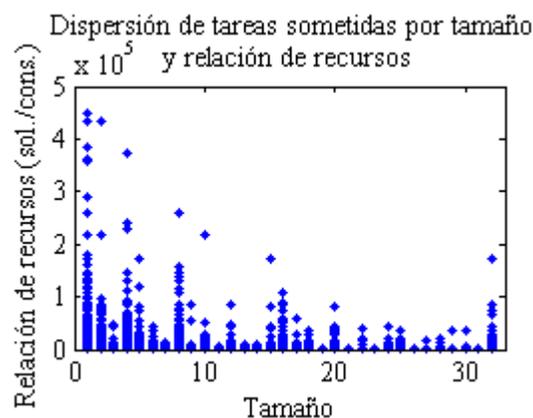


Figura 33. Dispersión de tareas sometidas por tamaño y relación de recursos (solicitados/consumidos), vista detallada de relaciones menores a 500,000.

Los usuarios pueden cambiar el tiempo solicitado para tarea, con respecto a la anterior, cuando la someten al sistema. Esto puede significar que el usuario somete una tarea completamente diferente a la anterior, o quizás está reajustando el tiempo solicitado de la misma. Estas variaciones en los tiempos solicitados son importantes para muchas estrategias de calendarización. Muchas utilizan el historial del usuario para crear estimaciones de tiempos de ejecución de las tareas de manera más precisa. Estas variaciones o cambios de tiempos solicitados se presentan en la Figura 34, se considera un cambio cuando hay una variación de $\pm 30\%$ con respecto a la tarea anterior. Existe una diferencia con la Figura 20 que muestra el número de tareas sometidas por usuario. Es normal creer que a mayor cantidad de tareas sometidas mayor cantidad de cambios en los tiempos solicitados, pero no es el caso para esta carga de trabajo. Los usuarios de The Advanced School for Computing and Imaging son los más activos en cuanto a tareas sometidas. Sin embargo, la proporción de los cambios de tiempos solicitados no es igual a la de tareas sometidas; aunque siguen siendo los que presentan más cambios. Un mayor detalle de este número de cambios se puede observar en la Figura 35.

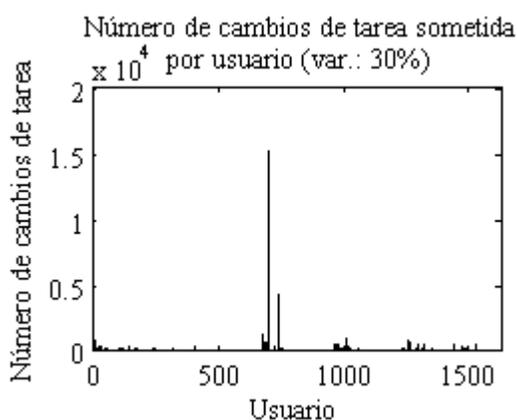


Figura 34. Número de cambios de tarea sometida por usuario, variación del 30%.

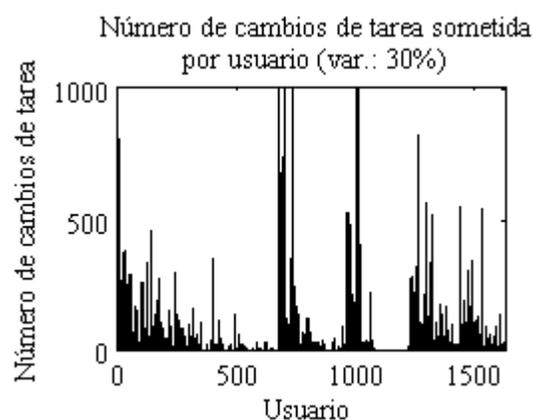


Figura 35. Número de cambios de tarea sometida por usuario, variación del 30%, vista detallada de usuarios con menos de 1,000 cambios de tarea.

A pesar de que las tareas con tiempo de ejecución menor a una hora son las más numerosas, no son las que consumen la mayor cantidad de recursos. Algo similar sucede con las tareas con menos de dos horas de tiempo de ejecución, Figura 36. El resto de las tareas presentan una relación más estrecha entre el número de tareas sometidas y los recursos consumidos por las mismas, a mayor cantidad de tareas sometidas mayor cantidad de recursos consumidos, ver Figura 37. Esta relación no se cumple con los recursos solicitados. Existe una cantidad mayor de recursos solicitados para las tareas, ver Figura 38. Un mayor detalle se presenta en la Figura 39. Es clara la diferencia entre los recursos solicitados y los consumidos, aunque es un comportamiento normal de los usuarios. Esta diferencia afecta a las estrategias de calendarización que utilizan los tiempos solicitados por los usuarios.

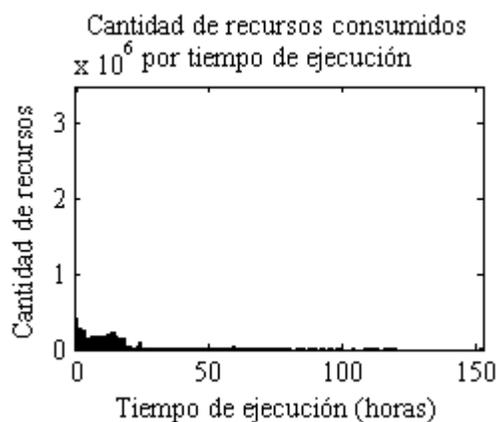


Figura 36. Cantidad de recursos consumidos por tiempo de ejecución.

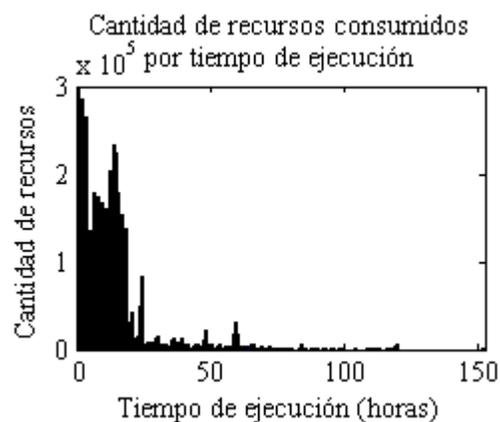


Figura 37. Cantidad de recursos consumidos por tiempo de ejecución, vista detallada de los tiempos de ejecución con recursos consumidos menores a 300,000.

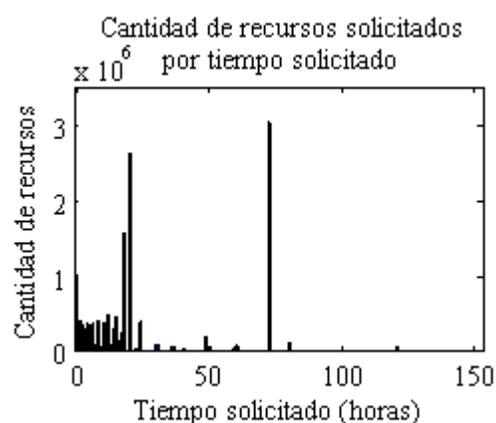


Figura 38. Cantidad de recursos solicitados por tiempo solicitado.



Figura 39. Cantidad de recursos solicitados por tiempo solicitado, vista detallada de los tiempos solicitados con recursos solicitados menores a 300,000.

Las tareas pueden ser de cualquier tamaño, pero las más comunes son las secuenciales o potencias de dos. Por consiguiente, estas tareas son las que más consumen recursos, ver Figura 40. También son estas mismas tareas las que solicitan mayor cantidad de recursos, ver Figura 41. Aunque existe un patrón muy similar entre ambas figuras; pero las tareas con tamaño 32 no siguen este patrón, solicitan casi de 5 veces más recursos que los que consumen.

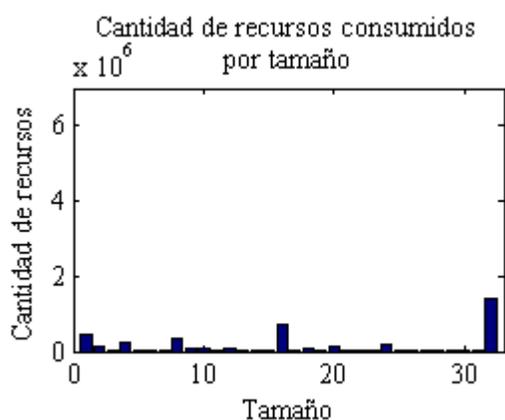


Figura 40. Cantidad de recursos consumidos por tamaño.

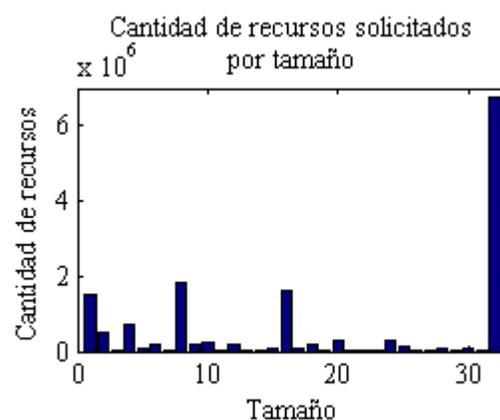


Figura 41. Cantidad de recursos solicitados por tamaño.

Aproximadamente 37,500 tareas terminaron con un estatus de error. Este error es el tipo 0, fallo. A pesar de que estas tareas fallaron no fueron filtradas de la carga de trabajo por la razón que consumen recursos. Por lo tanto, estas tareas las debe considerar el calendarizador, ver Figura 42. Los usuarios a los que pertenecen estas tareas se pueden apreciar en la Figura 43. Se puede observar que los usuarios de The Advanced School for Computing and Imaging (los más activos) presentan una proporción de errores muy pequeña en relación a las tareas sometidas.

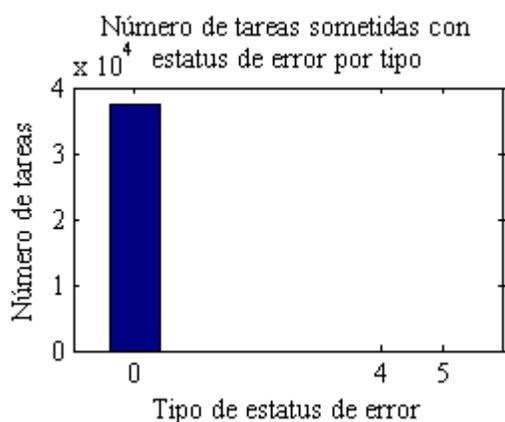


Figura 42. Número de tareas sometidas con estatus de error por tipo.

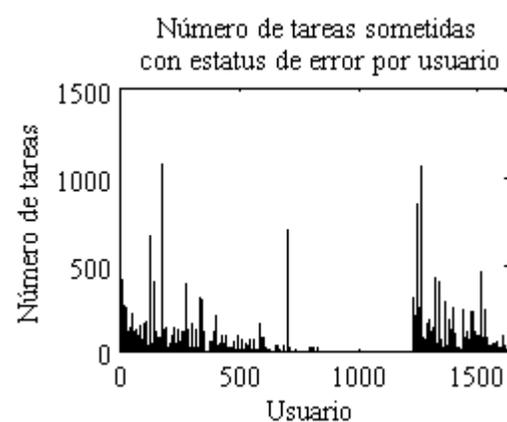


Figura 43. Número de tareas sometidas con estatus de error por usuario.

V. 3. Definición del experimento

V. 3. 1 Configuración del Grid

La configuración del Grid computacional utilizado para las simulaciones es el resultado de un análisis extenso de la carga de trabajo *mix_gs_log_36770jobs_30exp(1111110000,0-32)r.swf*, ver Tabla VIII. Esta carga de trabajo contiene tareas que requieren hasta 32 procesadores. Es la razón por la cual el tamaño máximo de las máquinas del Grid es 32. Generalmente las máquinas actuales tienen una cantidad de procesadores igual a una potencia de dos, por esta razón se seleccionaron de esta forma.

Tabla VIII. Configuración del Grid utilizado para los experimentos.

Configuración del Grid	
Tamaño máximo de tarea permitido	32
Número de máquinas	11
Tamaño de las máquinas	4
	4
	4
	4
	8
	8
	8
	16
	16
	32
	32

Esta configuración de Grid computacional fue seleccionada para simular la situación actual de los centros de cómputo donde existen incluso semanas de espera para ejecutar una tarea. No tiene sentido utilizar una configuración de Grid y una carga de trabajo que no genere colas de espera y la longitud del calendario la determine la última tarea en llegar. En este caso varios algoritmos de calendarización encontrarían el calendario óptimo. Aun cuando el tamaño del *cluster* máximo del Grid es relativamente pequeño, se

creo que es una imagen adecuada de un Grid computacional ya que las tareas se encuentran en la misma proporción.

Tabla IX. Parámetros de los experimentos.

Parámetros de los experimentos	
Número de experimentos	30
Número de tareas por experimento	36,770
Niveles de calendarización	2
Tipo de calendarización	0 (En línea)
Selección de tareas de la cola de espera del Grid	1 (FIFO)
Selección de la máquina a la cual se asignará la tarea	1 (<i>Random</i>) 2 (<i>Min_Lp</i>) 3 (<i>Min_PL</i>) 4 (<i>Min_LBal</i>) 5 (<i>Min_LB</i>) 6 (<i>Min_CT</i>) 7 (<i>Min_SWCT</i>) 8 (<i>Min_WT</i>) 9 (<i>Min_WWT</i>) 10 (<i>Min_U</i>) 11 (<i>Min_ST</i>) 12 (<i>Min_TA</i>) 13 (<i>Min_WTA</i>) 14 (<i>Min_WWoTA</i>)
Calendarizador local para el nivel de asignación de tareas (usado para las estrategias que requieren calendario tentativo)	1 (FCFS)
Grado de admisibilidad	0% 10% 20% 30% 40% 50% 60% 70% 80% 90% 100%
Tiempo de ejecución estimado	0 (Tiempo solicitado por usuario)
Calendarización local	1 (FCFS)

V. 3. 2 Carga de trabajo

La carga de trabajo *mix_gs_log_36770jobs_30exp(1111110000,0-32)r.swf* fue creada, a partir de información de registros reales, para la realización de los experimentos. La información proviene de 5 centros de cómputo de alto desempeño.

La carga de trabajo contiene información suficiente para realizar 30 experimentos de 36,770 tareas cada uno. Esto equivale a 30 experimentos de las tareas correspondientes a una semana; durante la primera semana se sometieron 36,770 tareas, se supone la misma cantidad en las semanas subsecuentes. Un total de 1,103,100 tareas en la carga de trabajo, correspondientes a 467 días.

V. 3. 3 Parámetros de los experimentos

Se realizaron 30 experimentos con las tareas correspondientes a una semana de la carga de trabajo, 36,770 tareas por semana. Se utilizó un modelo de calendarización jerárquico de dos niveles, ver Figura 8; catorce estrategias de asignación de tareas y una como calendarizador local. El esquema de admisibilidad usado incluye un grado de admisibilidad que va desde 0% hasta 100%, con incrementos de 10%. Se utilizó una calendarización en línea no clarividente, los tiempos de ejecución utilizados por algunas estrategias de asignación de tareas fueron los tiempos solicitados por el usuario. Los detalles completos del experimento se muestran en la Tabla IX.

V. 4. Análisis de resultados

El cociente de competitividad (ver página 48) es la principal métrica tratada en este trabajo de investigación. El objetivo del esquema de admisibilidad es reducir la longitud del calendario, por consiguiente, reducir el cociente de competitividad del algoritmo, incrementar la utilización y reducir la cantidad de recursos no utilizados. Los resultados experimentales muestran el comportamiento de las estrategias al utilizar el esquema de admisibilidad, así como los efectos en las diferentes métricas de análisis.

Tabla X. Resultados experimentales al aplicar el esquema de admisibilidad a las diferentes estrategias de calendarización. En blanco se muestra la métrica del algoritmo, en gris claro las métricas del sistema y en gris oscuro las métricas del usuario.

Estrategia		<i>Random</i>	<i>Min_Lp</i>	<i>Min_PL</i>	<i>Min_LBal</i>	<i>Min_LB</i>	<i>Min_CT</i>	<i>Min_SWCT</i>	<i>Min_WT</i>	<i>Min_WWT</i>	<i>Min_U</i>	<i>Min_ST</i>	<i>Min_TA</i>	<i>Min_WTA</i>	<i>Min_WWOTA</i>
Nivel de conocimiento		0	1	2	2	3	4	4	4	4	4	4	4	4	4
Porcentaje de mejora de la métrica y el grado de admisibilidad en el que se obtuvo	Cociente de competitividad	22.68 0.5	24.79 0.5	1.54 0.5	0.37 0.6	2.51 0.5	1.25 0.5	0.0 -	3.06 0.5	2.54 0.6	8.59 0.5	0.78 0.6	4.03 0.5	0.0 -	3.42 0.7
	Longitud del calendario	23.67 0.5	22.68 0.5	2.58 0.5	0.37 0.5	1.50 0.5	1.87 0.5	0.0 -	3.56 0.5	2.26 0.6	2.76 0.5	0.27 0.6	4.58 0.5	0.0 -	3.56 0.6
	Utilización del sistema	29.43 0.5	31.25 0.5	1.64 0.5	0.32 0.6	2.55 0.5	1.12 0.5	0.0 -	3.01 0.5	2.44 0.6	4.88 0.5	0.67 0.6	4.56 0.5	0.0 -	2.19 0.7
	Recursos no utilizados	37.30 0.5	35.95 0.5	6.18 0.5	0.86 0.5	3.38 0.5	5.35 0.5	0.0 -	7.60 0.5	4.20 0.6	4.23 0.5	0.80 0.6	9.67 0.5	0.0 -	5.65 0.6
	Rendimiento de procesamiento	25.62 0.5	34.68 0.5	0.0 -	0.00 0.6	3.14 0.5	0.51 0.5	0.0 -	1.19 0.5	3.35 0.5	12.31 0.5	1.57 0.6	3.11 0.6	0.0 -	0.73 0.7
	Tiempo promedio de permanencia	10.22 0.5	57.83 0.5	4.79 0.5	2.75 0.5	1.11 0.5	1.06 0.5	1.28 0.2	8.28 0.2	0.67 0.2	0.20 0.5	1.80 0.5	8.73 0.2	0.72 0.5	2.47 0.7
	Tiempo promedio de permanencia ponderado	29.81 0.5	36.06 0.5	4.78 0.5	2.46 0.5	4.13 0.5	1.41 0.5	0.49 0.6	7.09 0.5	0.81 0.5	2.28 0.5	0.67 0.5	6.61 0.5	0.41 0.6	3.99 0.5
	Tiempo promedio de permanencia ponderado por trabajo	30.58 0.5	29.87 0.5	4.31 0.5	1.91 0.5	4.22 0.5	1.83 0.5	0.74 0.6	7.13 0.5	0.53 0.6	5.07 0.5	0.76 0.5	6.70 0.5	0.39 0.6	6.35 0.6
	Tiempo promedio de espera	10.24 0.5	57.88 0.5	4.80 0.5	2.75 0.5	1.11 0.5	1.06 0.5	1.28 0.2	8.29 0.2	0.67 0.2	0.20 0.5	1.80 0.5	8.74 0.2	0.72 0.5	2.47 0.7
	Tiempo promedio de espera ponderado	29.84 0.5	36.09 0.5	4.79 0.5	2.46 0.5	4.13 0.5	1.41 0.5	0.49 0.6	7.10 0.5	0.81 0.5	2.28 0.5	0.67 0.5	6.62 0.5	0.41 0.6	4.00 0.5
	Tiempo de respuesta promedio	10.22 0.5	57.83 0.5	4.79 0.5	2.75 0.5	1.11 0.5	1.06 0.5	1.28 0.2	8.28 0.2	0.67 0.2	0.20 0.5	1.80 0.5	8.73 0.2	0.72 0.5	2.47 0.7
	Cociente de respuesta promedio	5.44 0.5	65.75 0.5	5.68 0.5	4.08 0.5	0.37 0.6	1.06 0.5	7.32 0.2	12.80 0.2	6.18 0.2	8.02 0.5	1.78 0.5	12.96 0.2	0.50 0.5	1.69 0.7
	Cociente de respuesta promedio acotado	7.76 0.5	62.69 0.5	5.20 0.5	3.12 0.5	0.80 0.2	0.96 0.5	5.08 0.2	11.64 0.2	4.49 0.2	5.65 0.5	1.78 0.5	12.03 0.2	0.89 0.5	1.89 0.7
	Cociente de respuesta del sistema	9.07 0.5	59.34 0.5	6.85 0.5	4.62 0.5	2.33 0.5	1.29 0.5	2.40 0.2	10.05 0.2	2.20 0.2	9.54 0.5	1.77 0.5	9.95 0.2	0.81 0.6	2.80 0.7

La Tabla X muestra un resumen de los resultados obtenidos durante la experimentación. Los porcentajes de mejora mostrados para cada métrica son los mayores obtenidos con los diferentes grados de admisibilidad utilizados. Algunas estrategias no

presentan mejoría en ciertas métricas y en algunos casos una misma estrategia tiene mejoras en las diferentes métricas con diferentes grados de admisibilidad. La información detallada se presenta más adelante; los resultados de cada métrica corresponden al mismo grado de admisibilidad usado para el cociente de competitividad, a menos que se indique lo contrario.

V. 4. 1 Estrategia *Random*

La estrategia *Random* (ver Página 30) no tiene un objetivo específico; no minimiza o maximiza ningún criterio. Por lo tanto, el esquema de admisibilidad afecta considerablemente el comportamiento de la misma. Con $a = 0.5$ se obtiene una mejora del 22.68% en el cociente de competitividad; de 2.74055 a 2.11909 (ver Apéndice, Figura 46). Esto significa una reducción de la longitud del calendario, en promedio de 23.67% (ver Apéndice, Figura 47). Al reducir el calendario se incrementa la utilización en 29.43% (ver Apéndice, Figura 60) y se reduce la cantidad de recursos no utilizados, 37.30% (ver Apéndice, Figura 58). Otra de las métricas relacionadas es el rendimiento de procesamiento, ésta presenta una mejora de 25.62% (ver Apéndice, Figura 57).

Los tiempos promedios de permanencia en el sistema también se reducen considerablemente. Al reducir tanto la longitud del calendario, el tiempo promedio de permanencia de una tarea también se ve reducido, 10.22% (ver Apéndice, Figura 50). El tiempo promedio de permanencia ponderado se reduce 29.81% (ver Apéndice, Figura 48) y el ponderado por trabajo 30.58% (ver Apéndice, Figura 49).

También es común reducir el tiempo promedio de espera al reducir la longitud del calendario. La reducción, en este caso, es de 10.24% (ver Apéndice, Figura 51) y para la versión ponderada es de 29.84% (ver Apéndice, Figura 52). Una reducción en el tiempo promedio de espera significa una reducción en el tiempo de respuesta promedio, 10.22% (ver Apéndice, Figura 55).

El cociente de respuesta promedio y el cociente de respuesta promedio acotado también tienen una mejora, 5.44% (ver Apéndice, Figura 54) y 7.76% (ver Apéndice, Figura 53), respectivamente. Con un cociente de respuesta del sistema 9.07% mejor (ver Apéndice, Figura 56).

La cantidad de trabajo total realizado es igual, para los diferentes grados de admisibilidad, al utilizar las mismas tareas para cada experimento (ver Apéndice, Figura 59).

V. 4. 2 Estrategia *Min_Lp*

El objetivo de *Min_Lp* (ver Página 30) es reducir el calendario al intentar balancear la carga de trabajo en cada máquina; sólo utiliza el tamaño de cada máquina y el número de tareas asignadas. El esquema de admisibilidad tiene un impacto incluso mayor que con la estrategia *Random*. Con $a = 0.5$ se obtiene una mejora del 24.79% en el cociente de competitividad; de 2.80035 a 2.10622 (ver Apéndice, Figura 61). La longitud del calendario de esta estrategia se reduce 22.68% en promedio (ver Apéndice, Figura 62). Por lo tanto, la utilización aumenta en 31.25% (ver Apéndice, Figura 75) y se reduce la cantidad de recursos no utilizados, 35.95% (ver Apéndice, Figura 73). El rendimiento de procesamiento, relacionado a las métricas anteriores, presenta una mejora de 34.68% (ver Apéndice, Figura 72).

Incluso los tiempos promedios de permanencia en el sistema presentan mejoras mayores a las de la estrategia *Random*. El tiempo promedio de permanencia de una tarea se reduce en 57.83% (ver Apéndice, Figura 65); el tiempo promedio de permanencia ponderado un 36.06% (ver Apéndice, Figura 63) y el ponderado por trabajo un 29.87% (ver Apéndice, Figura 64).

El tiempo promedio de espera sigue la misma tendencia que las métricas anteriores. En este caso, la reducción es de 57.88% (ver Apéndice, Figura 66) y para la versión ponderada es de 36.09% (ver Apéndice, Figura 67). Así, el tiempo de respuesta promedio,

en estrecha relación al tiempo promedio de espera, se reduce 57.83% (ver Apéndice, Figura 70).

Se aprecian mejoras aún más significativas para el cociente de respuesta promedio y el cociente de respuesta promedio acotado con 65.75% (ver Apéndice, Figura 69) y 62.69% (ver Apéndice, Figura 68), respectivamente. Como resultado, el cociente de respuesta del sistema mejora 59.34% (ver Apéndice, Figura 71).

La cantidad de trabajo total realizado es igual, para los diferentes grados de admisibilidad, al utilizar las mismas tareas para cada experimento (ver Apéndice, Figura 74).

V. 4. 3 Estrategia *Min_PL*

La estrategia *Min_PL* (ver Página 30) también tiene como objetivo reducir el calendario al intentar balancear la carga de trabajo; utiliza el tamaño de cada máquina, el número y tamaño de las tareas asignadas. Esta es una de las mejores estrategias sin el esquema de admisibilidad, es la razón por la cual no presenta una mejora importante. El cociente de competitividad mejora 1.54% con $\alpha = 0.5$; de 1.75421 a 1.72718 (ver Apéndice, Figura 76). La longitud del calendario también se reduce poco, 2.58% en promedio (ver Apéndice, Figura 77). Muy similar para la utilización que aumenta 1.64% (ver Apéndice, Figura 90) y la cantidad de recursos no utilizados bajan 6.18% (ver Apéndice, Figura 88). El rendimiento de procesamiento se mantiene igual (ver Apéndice, Figura 87).

También para los tiempos promedios de permanencia en el sistema se presentan mejoras modestas, muy similares entre ellos. El tiempo promedio de permanencia de una tarea se reduce 4.79% (ver Apéndice, Figura 80), el ponderado un 4.78% (ver Apéndice, Figura 78) y el ponderado por trabajo un 4.31% (ver Apéndice, Figura 79).

El tiempo promedio de espera también se encuentra en un rango similar a las métricas anteriores. Para este caso, la reducción es de 4.80% (ver Apéndice, Figura 81) y el ponderado disminuye 4.79% (ver Apéndice, Figura 82). Así, el tiempo de respuesta promedio se reduce 4.79% (ver Apéndice, Figura 85).

Se aprecian mejorías aún más significativas para el cociente de respuesta promedio y el cociente de respuesta promedio acotado con 5.68% (ver Apéndice, Figura 84) y 5.20% (ver Apéndice, Figura 83), respectivamente. Como resultado, el cociente de respuesta del sistema mejora 6.85% (ver Apéndice, Figura 86).

La cantidad de trabajo total realizado es igual, para los diferentes grados de admisibilidad, al utilizar las mismas tareas para cada experimento (ver Apéndice, Figura 89).

V. 4. 4 Estrategia *Min_LBal*

La estrategia *Min_LBal* (ver Página 30) tiene como objetivo reducir el calendario al reducir la desviación estándar de la distribución de la carga de trabajo; utiliza el tamaño de cada máquina, el número y tamaño de las tareas asignadas. Esta estrategia es casi tan buena como *Min_PL*, pero la mejoría al aplicar el esquema de admisibilidad es mucho menor. El cociente de competitividad sólo mejora 0.37% con $a = 0.6$; de 1.78609 a 1.77939 (ver Apéndice, Figura 91). La longitud del calendario se reduce igual, 0.37% ($a = 0.5$) (ver Apéndice, Figura 92). La utilización aumenta 0.32% (ver Apéndice, Figura 105), por lo tanto la cantidad de recursos no utilizados se reducen 0.86% ($a = 0.5$) (ver Apéndice, Figura 103). El rendimiento de procesamiento continúa igual (ver Apéndice, Figura 102). Los resultados de las métricas restantes corresponden a $a = 0.5$.

También para los tiempos promedios de permanencia en el sistema las mejoras son pequeñas. El tiempo promedio de permanencia de una tarea disminuye 2.75% (ver Apéndice, Figura 95); el tiempo promedio de permanencia ponderado baja 2.46% (ver

Apéndice, Figura 93) y el ponderado por trabajo se reduce 1.91% (ver Apéndice, Figura 94).

Para el tiempo promedio de espera también se presentan mejoras similares; la reducción es de 2.75% (ver Apéndice, Figura 96). El tiempo promedio ponderado disminuye 2.46% (ver Apéndice, Figura 97). El tiempo de respuesta promedio también se reduce, un 2.75% (ver Apéndice, Figura 100).

Las mejoras del cociente de respuesta promedio y el cociente de respuesta promedio acotado son también pequeñas, 4.08% (ver Apéndice, Figura 99) y 3.12% (ver Apéndice, Figura 98), respectivamente. Como resultado, el cociente de respuesta del sistema mejora 4.62% (ver Apéndice, Figura 101).

La cantidad de trabajo total realizado es igual, para los diferentes grados de admisibilidad, al utilizar las mismas tareas para cada experimento (ver Apéndice, Figura 104).

V. 4. 5 Estrategia *Min_LB*

La estrategia *Min_LB* (ver Página 30) tiene como objetivo reducir la cota mínima de terminación de los calendarios; utiliza el tamaño de cada máquina, el número, tamaño y tiempo solicitado de las tareas asignadas. Esta estrategia es casi tan buena como *Min_LBal*, la mejoría al aplicar el esquema de admisibilidad es mayor a la de *Min_PL*. El cociente de competitividad mejora 2.51% con $\alpha = 0.5$; de 1.83820 a 1.79214 (ver Apéndice, Figura 106). Por lo que la longitud del calendario disminuye 1.50% (ver Apéndice, Figura 107). Así, la utilización aumenta 2.55% (ver Apéndice, Figura 120) y la cantidad de recursos no utilizados bajan 3.38% (ver Apéndice, Figura 118). El rendimiento de procesamiento también se incrementa, 3.14% (ver Apéndice, Figura 117).

El tiempo promedio de permanencia de una tarea disminuye 1.11% (ver Apéndice, Figura 110); la versión ponderada baja 4.13% (ver Apéndice, Figura 108) y la ponderada por trabajo se reduce 4.22% (ver Apéndice, Figura 109).

Para el tiempo promedio de espera también se presentan mejoras similares; la reducción es de 1.11% (ver Apéndice, Figura 111). El tiempo promedio de espera ponderado disminuye 4.13% (ver Apéndice, Figura 112). El tiempo de respuesta promedio también se reduce, un 1.11% (ver Apéndice, Figura 115).

En el caso del cociente de respuesta promedio y el cociente de respuesta promedio acotado el esquema de admisibilidad no tiene mucho efecto, reducciones de 0.37% ($\alpha = 0.6$) (ver Apéndice, Figura 114) y 0.80% ($\alpha = 0.2$) (ver Apéndice, Figura 113), respectivamente. Pero, el cociente de respuesta del sistema mejora 2.33% (ver Apéndice, Figura 116).

La cantidad de trabajo total realizado es igual, para los diferentes grados de admisibilidad, al utilizar las mismas tareas para cada experimento (ver Apéndice, Figura 119).

V. 4. 6 Estrategia *Min_CT*

La estrategia *Min_CT* (ver Página 31) reduce el tiempo de terminación de cada tarea. Utiliza el tamaño de cada máquina, el número, tamaño y tiempo solicitado de las tareas asignadas; crea un calendario previo para seleccionar la máquina donde se asignará la tarea. Es la segunda mejor estrategia sin el esquema de admisibilidad. Con $\alpha = 0.5$ se obtiene una mejora del 1.25% en el cociente de competitividad; de 1.52976 a 1.51068 (ver Apéndice, Figura 121). La reducción de la longitud del calendario es de 1.87% en promedio (ver Apéndice, Figura 122). Al reducir el calendario se incrementa la utilización, en este caso de 1.12% (ver Apéndice, Figura 135); la cantidad de recursos no utilizados disminuye 5.35% (ver Apéndice, Figura 133). Para el rendimiento de procesamiento, relacionado a las métricas anteriores, una mejora de 0.51% (ver Apéndice, Figura 132).

La reducción de la longitud del calendario no es muy grande; aún así, los tiempos promedios de permanencia en el sistema se reducen 1.06% (ver Apéndice, Figura 125). El tiempo promedio de permanencia ponderado se reduce 1.41% (ver Apéndice, Figura 123) y el ponderado por trabajo 1.83% (ver Apéndice, Figura 124).

El tiempo promedio de espera se reduce en la misma proporción que las métricas anteriores. En este caso de 1.06% (ver Apéndice, Figura 126) y de 1.41% para la versión ponderada (ver Apéndice, Figura 127). La reducción en el tiempo promedio de espera implica una reducción en el tiempo de respuesta promedio; 1.06% (ver Apéndice, Figura 130).

Sin ser la excepción, el cociente de respuesta promedio y el cociente de respuesta promedio acotado también mejoran, 1.06% (ver Apéndice, Figura 129) y 0.96% (ver Apéndice, Figura 128), respectivamente. Así también el cociente de respuesta del sistema, 1.29% (ver Apéndice, Figura 131).

La cantidad de trabajo total realizado es igual, para los diferentes grados de admisibilidad, al utilizar las mismas tareas para cada experimento (ver Apéndice, Figura 134).

V. 4. 7 Estrategia *Min_SWCT*

La estrategia *Min_SWCT* (ver Página 31) reduce la suma total del tiempo de terminación de cada tarea ponderándolo por el tamaño de la misma. Utiliza el tamaño de cada máquina, el número, tamaño y tiempo solicitado de las tareas asignadas; además crea un calendario previo. Es una estrategia regular sin el esquema de admisibilidad al compararla con las demás. Los mejores comportamientos se presentan con $a = 0.8, 0.9, 1.0$. Por lo tanto, no hay mejoras al utilizar el esquema de admisibilidad en el cociente de competitividad (ver Apéndice, Figura 136), la longitud del calendario (ver Apéndice, Figura 137), la utilización (ver Apéndice, Figura 150), la cantidad de recursos

no utilizados (ver Apéndice, Figura 148) y el rendimiento de procesamiento (ver Apéndice, Figura 147).

Incluso sin mejoras en las métricas anteriores, los tiempos promedios de permanencia en el sistema disminuyen 1.28% ($a = 0.2$) (ver Apéndice, Figura 140). Para el tiempo promedio de permanencia ponderado hay una reducción de 0.49% ($a = 0.6$) (ver Apéndice, Figura 138) y para el ponderado por trabajo de 0.74% ($a = 0.6$) (ver Apéndice, Figura 139).

También el tiempo promedio de espera se reduce; 1.28% ($a = 0.2$) (ver Apéndice, Figura 141) y 0.49% ($a = 0.6$) para la versión ponderada (ver Apéndice, Figura 142). Al reducirse el tiempo promedio de espera también se reduce el tiempo de respuesta promedio; 1.28% en este caso (ver Apéndice, Figura 145).

El cociente de respuesta promedio y el cociente de respuesta promedio acotado también mejoran, 7.32% (ver Apéndice, Figura 144) y 5.08% (ver Apéndice, Figura 143), respectivamente. Por último, el cociente de respuesta del sistema presenta una mejora de 2.40% (ver Apéndice, Figura 146). Estas métricas corresponden a $a = 0.5$.

La cantidad de trabajo total realizado es igual, para los diferentes grados de admisibilidad, al utilizar las mismas tareas para cada experimento (ver Apéndice, Figura 149).

V. 4. 8 Estrategia *Min_WT*

La estrategia *Min_WT* (ver Página 31) reduce el tiempo de espera de cada tarea. Utiliza el tamaño de cada máquina, el número, tamaño y tiempo solicitado de las tareas asignadas; también crea un calendario previo. Es una estrategia regular sin el esquema de admisibilidad si se compara con las otras. Esta estrategia mejora un 3.06% en su cociente de competitividad con $a = 0.5$ (ver Apéndice, Figura 151), concretamente pasa de 1.88801 a 1.83030. Para las otras métricas relacionadas también se observan mejoras. La longitud

del calendario disminuye 3.56% (ver Apéndice, Figura 152) y la cantidad de recursos no utilizados 7.60% (ver Apéndice, Figura 163). La utilización aumenta 3.01% (ver Apéndice, Figura 165) y el rendimiento de procesamiento 1.19% (ver Apéndice, Figura 162).

A pesar de la pequeña reducción en la longitud del calendario, el tiempo promedio de permanencia en el sistema muestra una mejora de 8.28% ($\alpha = 0.2$) (ver Apéndice, Figura 155). El tiempo promedio de permanencia ponderado se reduce 7.09% (ver Apéndice, Figura 153) y el ponderado por trabajo 7.13% (ver Apéndice, Figura 154).

Mismo caso para el tiempo promedio de espera que se reduce 8.29% ($\alpha = 0.2$) (ver Apéndice, Figura 156) y 7.10% el ponderado (ver Apéndice, Figura 157). Así mismo, el tiempo de respuesta promedio disminuye 8.28% ($\alpha = 0.2$) (ver Apéndice, Figura 160).

Mejoras más importantes son las que presentan el cociente de respuesta promedio y el cociente de respuesta promedio acotado, 12.80% (ver Apéndice, Figura 159) y 11.64% (ver Apéndice, Figura 158), respectivamente. El cociente de respuesta del sistema también tiene un efecto positivo, 10.05% de mejora (ver Apéndice, Figura 161). Estas métricas corresponden a $\alpha = 0.2$.

La cantidad de trabajo total realizado es igual, para los diferentes grados de admisibilidad, al utilizar las mismas tareas para cada experimento (ver Apéndice, Figura 164).

V. 4. 9 Estrategia *Min_WWT*

La estrategia *Min_WWT* (ver Página 31) reduce el tiempo de espera ponderado de cada tarea. Utiliza el tamaño de cada máquina, el número, tamaño y tiempo solicitado de las tareas asignadas; igualmente crea un calendario previo. Esta estrategia es similar a la estrategia *Min_SWCT* en cuanto a su cociente de competitividad; con la excepción que en este caso sí existe una mejoría al aplicar el esquema de admisibilidad. La estrategia mejora un 2.54% en su cociente de competitividad con $\alpha = 0.6$ (ver Apéndice, Figura 166), pasa

de 2.16906 a 2.11387; además con $\alpha = 0.5$ también tenemos una mejoría, aunque menos significativa. Las demás métricas relacionadas al cociente de competitividad también presentan mejoras en grados similares. La longitud del calendario se reduce un 2.26% (ver Apéndice, Figura 167) así como la cantidad de recursos no utilizados un 4.20% (ver Apéndice, Figura 178). La utilización incrementa 2.44% (ver Apéndice, Figura 180) como también el rendimiento de procesamiento 3.35% ($\alpha = 0.5$) (ver Apéndice, Figura 177).

El tiempo promedio de permanencia en el sistema baja 0.67% ($\alpha = 0.2$) (ver Apéndice, Figura 170). El tiempo promedio de permanencia ponderado se reduce 0.81% ($\alpha = 0.5$) (ver Apéndice, Figura 168) y el ponderado por trabajo 0.53% (ver Apéndice, Figura 169). Estas pequeñas mejoras se presentan aún incluso con la pequeña reducción en el calendario, sin olvidar que el esquema de admisibilidad no tiene como objetivo directo el disminuir estos tiempos.

El tiempo promedio de espera tiene un efecto similar que las métricas anteriores, se reduce 0.67% ($\alpha = 0.2$) (ver Apéndice, Figura 171) y el ponderado 0.81% ($\alpha = 0.5$) (ver Apéndice, Figura 172). Por lo que el tiempo de respuesta promedio disminuye 0.67% ($\alpha = 0.2$) (ver Apéndice, Figura 175).

Mejoras más importantes para el cociente de respuesta promedio y el cociente de respuesta promedio acotado, 6.81% (ver Apéndice, Figura 174) y 4.49% (ver Apéndice, Figura 173), respectivamente. Similarmente, el cociente de respuesta del sistema también tiene un efecto positivo mejorando 2.20% (ver Apéndice, Figura 176). Estas métricas corresponden a $\alpha = 0.2$.

La cantidad de trabajo total realizado es igual, para los diferentes grados de admisibilidad, al utilizar las mismas tareas para cada experimento (ver Apéndice, Figura 179).

V. 4. 10 Estrategia *Min_U*

La estrategia *Min_U* (ver Página 31) intenta reducir la longitud del calendario al seleccionar la máquina con la menor utilización para asignar la tarea. Utiliza el tamaño de cada máquina, el número, tamaño y tiempo solicitado de las tareas asignadas; igualmente crea un calendario previo. Ésta es la que peor se desempeña de todas las estrategias analizadas; su cociente de competitividad, sin el esquema de admisibilidad, es mayor a 3. A pesar de eso, la estrategia sólo mejora un 8.59% en su cociente de competitividad con $\alpha = 0.5$ (ver Apéndice, Figura 181), pasa de 3.12233 a 2.85426. La longitud del calendario se reduce un 2.76% (ver Apéndice, Figura 182) y la cantidad de recursos no utilizados un 4.23% (ver Apéndice, Figura 193). La utilización incrementa 4.88% (ver Apéndice, Figura 195) y el rendimiento de procesamiento 12.31% (ver Apéndice, Figura 192). Esta última métrica es la que más se ve afectada por el esquema de admisibilidad.

Para el tiempo promedio de permanencia sólo hay una mejora de 0.20% (ver Apéndice, Figura 185). El tiempo promedio de permanencia ponderado mejora un poco más con 2.28% (ver Apéndice, Figura 183) y el ponderado por trabajo 5.07% (ver Apéndice, Figura 184). Únicamente mejoras pequeñas al utilizar el esquema de admisibilidad.

Los tiempos promedio de espera también tienen una mejoría pequeña. El tiempo promedio de espera se reduce 0.20% (ver Apéndice, Figura 186) y su correspondiente ponderado 2.28% (ver Apéndice, Figura 187). Por lo que el tiempo de respuesta promedio disminuye 0.20% (ver Apéndice, Figura 190).

Un efecto mayor para los cocientes de respuesta. El cociente de respuesta promedio y el cociente de respuesta promedio acotado mejoran 8.02% (ver Apéndice, Figura 189) y 5.65% (ver Apéndice, Figura 188), respectivamente. El cociente de respuesta del sistema tiene un mayor efecto al mejorar 9.54% (ver Apéndice, Figura 191).

La cantidad de trabajo total realizado es igual, para los diferentes grados de admisibilidad, al utilizar las mismas tareas para cada experimento (ver Apéndice, Figura 194).

V. 4. 11 Estrategia *Min_ST*

La estrategia *Min_ST* (ver Página 31) es una estrategia que tiene como objetivo reducir la longitud del calendario al asignar la tarea donde el inicio de su ejecución es más próximo. Utiliza el tamaño de cada máquina, el número, tamaño y tiempo solicitado de las tareas asignadas; requiere también de un calendario tentativo. Ésta es la mejor de las estrategias presentadas, aún así, muestra mejoras al aplicar el esquema de admisibilidad. Con $\alpha = 0.6$ se obtiene una mejora del 0.78% en el cociente de competitividad; de 1.50094 a 1.48922 (ver Apéndice, Figura 196). Existen otros grados de admisibilidad donde hay mejoras, aunque son menores: $\alpha = 0.5, 0.7$; con $\alpha = 0.8, 0.9, 1.0$ se obtiene el mismo cociente de competitividad. Así, la longitud del calendario se reduce en promedio 0.27% (ver Apéndice, Figura 197). La utilización se incrementa en 0.67% (ver Apéndice, Figura 210) y se reduce la cantidad de recursos no utilizados, 0.80% (ver Apéndice, Figura 208). Por consiguiente, el rendimiento de procesamiento presenta una mejora de 1.57% (ver Apéndice, Figura 207). Las métricas restantes corresponden a $\alpha = 0.5$.

Los tiempos promedios de permanencia en el sistema también se reducen en la misma proporción. El tiempo promedio de permanencia de una tarea se ve reducido un 1.80% (ver Apéndice, Figura 200). El tiempo promedio de permanencia ponderado se reduce 0.67% (ver Apéndice, Figura 198) y el ponderado por trabajo 0.76% (ver Apéndice, Figura 199).

Las reducciones en el tiempo promedio de espera siguen el mismo patrón. La reducción, en este caso, es de 1.80% (ver Apéndice, Figura 201) y para la versión ponderada es de 0.67% (ver Apéndice, Figura 202). También el tiempo de respuesta promedio se beneficia del esquema de admisibilidad con una reducción de 1.80% (ver Apéndice, Figura 205).

El cociente de respuesta promedio presenta una mejora de 1.78% (ver Apéndice, Figura 204) además el cociente de respuesta promedio acotado presenta el mismo resultado (ver Apéndice, Figura 203). El cociente de respuesta del sistema mejora 1.77% (ver Apéndice, Figura 206).

La cantidad de trabajo total realizado es igual, para los diferentes grados de admisibilidad, al utilizar las mismas tareas para cada experimento (ver Apéndice, Figura 209).

V. 4. 12 Estrategia *Min_TA*

La estrategia *Min_TA* (ver Página 32) es una estrategia que tiene como objetivo reducir el tiempo promedio de permanencia de una tarea. Utiliza el tamaño de cada máquina, el número, tamaño y tiempo solicitado de las tareas asignadas; también requiere un calendario tentativo. Es otra de las estrategias que no tienen un buen comportamiento sin el esquema de admisibilidad. Aún así, con $\alpha = 0.5$ se obtiene una mejora de sólo 4.03% en el cociente de competitividad; se reduce de 1.90301 a 1.82627 (ver Apéndice, Figura 211); inclusive, con $\alpha = 0.6$ también se presenta una mejora, pero menor. La longitud del calendario se reduce en un porcentaje similar, 4.58% (ver Apéndice, Figura 212). Con la reducción de la longitud del calendario, la utilización se incrementa en 4.56% (ver Apéndice, Figura 225) y se reduce la cantidad de recursos no utilizados, 9.67% (ver Apéndice, Figura 223). Por lo tanto, el rendimiento de procesamiento presenta una mejora de 3.11% ($\alpha = 0.6$) (ver Apéndice, Figura 222).

Los tiempos promedios de permanencia en el sistema tienen mejorías más grandes en general. El tiempo promedio de permanencia de una tarea se reduce 8.73% ($\alpha = 0.2$) (ver Apéndice, Figura 215), el ponderado 6.61% (ver Apéndice, Figura 213) y el ponderado por trabajo 6.70% (ver Apéndice, Figura 214).

Algo similar presentan los tiempos promedio de espera. La reducción es de 8.74% ($\alpha = 0.2$) (ver Apéndice, Figura 216) y de 6.62% para la versión ponderada (ver Apéndice, Figura 217). El tiempo de respuesta promedio sigue en el mismo rango con una reducción de 8.73% ($\alpha = 0.2$) (ver Apéndice, Figura 220).

Aún mejores resultados para los cocientes de respuesta promedio que presentan una mejora de 12.96% (ver Apéndice, Figura 219) y de 12.03% para el acotado (ver Apéndice, Figura 218). El cociente de respuesta del sistema mejora 9.95% (ver Apéndice, Figura 221). Estas métricas corresponden a $\alpha = 0.2$.

La cantidad de trabajo total realizado es igual, para los diferentes grados de admisibilidad, al utilizar las mismas tareas para cada experimento (ver Apéndice, Figura 224).

V. 4. 13 Estrategia *Min_WTA*

La estrategia *Min_WTA* (ver Página 32) es una estrategia que tiene como objetivo reducir el tiempo ponderado promedio de permanencia de una tarea. Utiliza el tamaño de cada máquina, el número, tamaño y tiempo solicitado de las tareas asignadas; requiere así mismo de un calendario tentativo. Es una estrategia promedio sin el esquema de admisibilidad al compararla con las demás. En este caso no existe mejoría alguna al aplicar el esquema de admisibilidad, con $\alpha = 0.8, 0.9, 1.0$ se obtienen los mismos resultados en el cociente de competitividad (ver Apéndice, Figura 226). Por lo tanto, la longitud del calendario (ver Apéndice, Figura 227), la utilización (ver Apéndice, Figura 240), la cantidad de recursos no utilizados (ver Apéndice, Figura 238) y el rendimiento de procesamiento (ver Apéndice, Figura 237) no tienen mejoría alguna.

A pesar de no presentar una reducción en la longitud del calendario, los tiempos promedio de permanencia en el sistema tienen una leve mejora. El tiempo promedio de permanencia de una tarea disminuye 0.72% ($\alpha = 0.5$) (ver Apéndice, Figura 230), el

ponderado 0.41% ($\alpha = 0.6$) (ver Apéndice, Figura 228) y por último, el ponderado por trabajo 0.39% ($\alpha = 0.6$) (ver Apéndice, Figura 229).

Es el mismo caso para los tiempos promedio de espera. Reducción de 0.72% ($\alpha = 0.5$) (ver Apéndice, Figura 231) y de 0.41% ($\alpha = 0.6$) para el ponderado (ver Apéndice, Figura 232). El tiempo de respuesta promedio también disminuye 0.72% ($\alpha = 0.5$) (ver Apéndice, Figura 235).

Al reducir poco los tiempos promedio de espera, los cocientes de respuesta promedio disminuyen en menor medida, 0.50% ($\alpha = 0.5$) (ver Apéndice, Figura 234) y 0.89% ($\alpha = 0.5$) para el acotado (ver Apéndice, Figura 233). El cociente de respuesta del sistema mejora 0.81% ($\alpha = 0.6$) (ver Apéndice, Figura 236).

La cantidad de trabajo total realizado es igual, para los diferentes grados de admisibilidad, al utilizar las mismas tareas para cada experimento (ver Apéndice, Figura 239).

V. 4. 14 Estrategia *Min_WWoTA*

La estrategia *Min_WWoTA* (ver Página 32) es similar a la estrategia *Min_WTA*, únicamente que la ponderación se hace por trabajo. Utiliza el tamaño de cada máquina, el número, tamaño y tiempo solicitado de las tareas asignadas; también requiere un calendario tentativo. Una de las malas estrategias sin el esquema de admisibilidad. Su mejora no es tampoco buena, con $\alpha = 0.7$ se reduce 3.42% el cociente de competitividad; de 2.68429 a 2.59262 (ver Apéndice, Figura 241). Pero, con $\alpha = 0.5, 0.6$ también presenta mejoría. La longitud del calendario disminuye 3.56% ($\alpha = 0.6$) (ver Apéndice, Figura 242). Con esa reducción de la longitud del calendario, la utilización se incrementa 2.19% (ver Apéndice, Figura 255) y la cantidad de recursos no utilizados se reduce 5.65% ($\alpha = 0.6$) (ver Apéndice, Figura 253). Por lo tanto, el rendimiento de procesamiento presenta una mejora de 0.73% (ver Apéndice, Figura 252).

A pesar de presentar una reducción pequeña en la longitud del calendario, los tiempos promedio de permanencia en el sistema mejoran en mayor medida. El tiempo promedio de permanencia de una tarea disminuye 2.47% (ver Apéndice, Figura 245), el ponderado 3.99% ($\alpha = 0.5$) (ver Apéndice, Figura 243) y el ponderado por trabajo 6.35% ($\alpha = 0.6$) (ver Apéndice, Figura 244).

Los tiempos promedio de espera también se reducen, 2.47% (ver Apéndice, Figura 246) y de 4.00% ($\alpha = 0.5$) para el ponderado (ver Apéndice, Figura 247). El tiempo de respuesta promedio disminuye 2.47% (ver Apéndice, Figura 250).

El cociente de respuesta promedio disminuye 1.69% (ver Apéndice, Figura 249) y 1.89% el acotado (ver Apéndice, Figura 248). El cociente de respuesta del sistema se reduce 2.80% (ver Apéndice, Figura 251).

La cantidad de trabajo total realizado es igual, para los diferentes grados de admisibilidad, al utilizar las mismas tareas para cada experimento (ver Apéndice, Figura 254).

V. 4. 15 Conclusiones

Al aplicar el esquema de admisibilidad se reducen los cocientes de competitividad de casi todas las estrategias analizadas. Las más afectadas son *Min_Lp* y *Random* con más de 20% de reducción. Esto significa que las longitudes de los calendarios se reducen en una proporción similar (ver Apéndice, Figura 257). Las dos estrategias que se mantienen invariantes son *Min_SWCT* y *Min_WTA*, (ver Apéndice, Figura 256). Para el caso de la utilización y de los recursos no utilizados, los efectos son incluso mayores al aplicar el esquema de admisibilidad. *Random* y *Min_Lp* disminuyen la cantidad de recursos no utilizados en más de 30% (ver Apéndice, Figura 259) y aumentan la utilización del sistema alrededor de la misma cantidad (ver Apéndice, Figura 258). *Min_SWCT* y *Min_WTA* se mantienen invariantes también para estas métricas. El rendimiento de procesamiento varía incluso más; *Random* y *Min_Lp* siguen siendo las más afectadas con más de 25% y 35%

respectivamente, además *Min_PL*, *Min_SWCT*, *Min_WTA* y *Min_LBal* no sufren cambio alguno (ver Apéndice, Figura 260).

El tiempo promedio de permanencia también se reduce al aplicar el esquema de admisibilidad. El patrón es diferente en este caso; *Min_Lp* tiene una reducción de casi 60%, *Min_WT*, *Min_TA* y *Random* presentan alrededor de 10% de reducción y *Min_U* menos de 1% (ver Apéndice, Figura 261). El caso del tiempo promedio de permanencia ponderado sigue un patrón más cercano a las primeras métricas mencionadas. *Min_Lp* y *Random* son las estrategias que muestran un mayor efecto al reducir alrededor de 30%, por otra parte, *Min_WTA*, *Min_SWCT*, *Min_ST* y *Min_WWT* sólo tienen una reducción de menos del 1% (ver Apéndice, Figura 262). El caso de la versión ponderada por trabajo es similar (ver Apéndice, Figura 263).

El tiempo promedio de espera, para la estrategia *Min_Lp*, también se reduce casi 60% como el tiempo promedio de permanencia. *Min_WT*, *Min_WTA* y *Random* están alrededor de 10% en la disminución de su tiempo de espera; *Min_U*, *Min_WWT* y *Min_WTA* con menos de 1% (ver Apéndice, Figura 264). En el caso del tiempo promedio de espera ponderado los efectos son menores, *Min_Lp* y *Random* tienen una reducción de alrededor de 30% y *Min_WTA*, *Min_SWCT*, *Min_ST* y *Min_WWT* menos de 1% (ver Apéndice, Figura 265).

Sólo *Min_Lp* es la estrategia que presenta el mayor efecto del esquema de admisibilidad con casi 60% en el tiempo de respuesta promedio. *Random*, *Min_TA* y *Min_WT* tienen una mejora de aproximadamente 10% y tres estrategias menos de 1% que son *Min_U*, *Min_WWT* y *Min_WTA* (ver Apéndice, Figura 266). El cociente de respuesta promedio es la métrica donde *Min_Lp* tiene el efecto positivo mayor, más de 65%; *Min_TA* y *Min_WT* le siguen con 13%; las estrategias con menor efecto son *Min_LB* y *Min_WTA* con menos de 1% (ver Apéndice, Figura 267). Un caso muy similar para el cociente de respuesta promedio acotado, únicamente que *Min_CT* también tiene menos de 1% de mejora (ver Apéndice, Figura 268). Para el cociente de respuesta del sistema, *Min_Lp* es

también la estrategia con mayor efecto al presentar una mejora de casi 60%, seguida por *Min_WT*. La que presenta la menor mejoría es *Min_WTA* (ver Apéndice, Figura 269).

Las dos estrategias que se benefician más con el esquema de admisibilidad son *Min_Lp* y *Random*, en algunas métricas con mejoras de hasta 60%. Por el contrario, *Min_WTA* es la estrategia que menos efectos tiene.

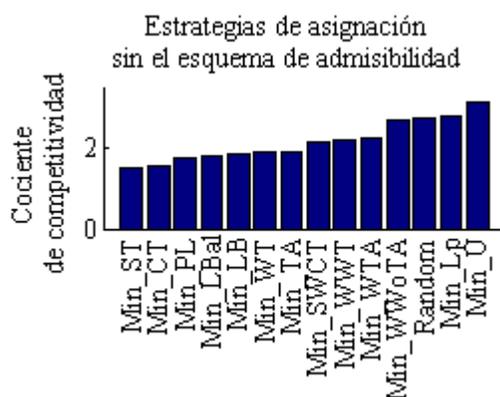


Figura 44. Cociente de competitividad de las estrategias de asignación sin el esquema de admisibilidad.

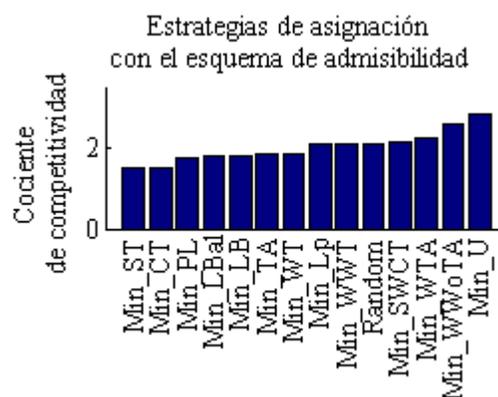


Figura 45. Cociente de competitividad de las estrategias de asignación con el esquema de admisibilidad.

Las estrategias que presentan los mejores cocientes de competitividad sin el esquema de admisibilidad se mantienen estables después de aplicarlo; *Min_ST*, *Min_CT*, *Min_PL*, *Min_LBal* y *Min_LB*. Algunas estrategias con cocientes de competitividad mayores muestran un caso diferente; *Min_Lp* pasa de ser la penúltima peor estrategia a la octava, baja 5 posiciones; *Random* baja 2 posiciones. *Min_U*, la peor estrategia, sigue sin cambios. La Figura 44 muestra los cocientes de competitividad de las estrategias antes de aplicar el esquema de admisibilidad y la Figura 45 los muestra después de haberlo aplicado.

Capítulo VI

Conclusiones

En este capítulo se presentan los resultados y las conclusiones de este trabajo de investigación y se proponen nuevas actividades y variantes del problema a resolver.

VI. 1. Conclusiones finales

El esquema de admisibilidad es un concepto muy simple y fácil de aplicar, basta con evitar la asignación de las tareas pequeñas a máquinas grandes. La consideración original es de restringir el 50% de los procesadores disponibles para una tarea. Se muestra que este porcentaje es el que más se acerca al óptimo si se generaliza para todas las estrategias de asignación de recursos. Sin embargo, algunas estrategias trabajan mejor con otro porcentaje. Un mejor ajuste, independiente para cada estrategia y métrica, ofrece mejores resultados como se aprecia en la Tabla X.

Se redujo la longitud de los calendarios generados; mejoras de hasta un 25% al aplicar el esquema de admisibilidad durante la etapa de selección de recursos. Este porcentaje varía dependiendo de la estrategia seleccionada, siendo *Min_Lp* la que obtiene el mayor beneficio. Las mejores estrategias sin el esquema de admisibilidad se mantienen estables al aplicarlo, aunque sí presentan mejoría. La hipótesis inicial del comportamiento del esquema de admisibilidad era que mejoría los calendarios de todas las estrategias. Este no es el caso, ya que dos estrategias, *Min_SWCT* y *Min_WTA*, no presentan mejora alguna. Se desconoce la razón, pero se observa que todas las estrategias que utilizan ponderación generan resultados más pobres comparadas con sus versiones no ponderadas.

Las dos estrategias que no utilizan el tamaño de cada tarea para asignarlas (nivel 0 y 1 de conocimiento) son las que obtienen mejores beneficios del esquema de admisibilidad. La razón de esta situación es la introducción de la selección de máquinas dependiendo del tamaño de la tarea. Al asignar las tareas a máquinas de manera indiscriminada se presenta más frecuentemente la situación en la que tareas grandes esperan por su ejecución; situación en la que se enfoca el esquema de admisibilidad. Se presenta un caso diferente con las estrategias correspondientes a los niveles 2 al 5 cuando se aplica el esquema de admisibilidad. Estas estrategias utilizan el tamaño de la tarea para seleccionar la máquina que la ejecutará y la nueva selección del esquema de admisibilidad no tiene el mismo efecto. El tamaño de la tarea juega un papel importante en la calendarización, además la única información relacionada a las tareas que utiliza el esquema de admisibilidad.

Algo muy particular sucede con algunas estrategias como lo son *Min_SWCT*, *Min_WT*, *Min_WWT*, *Min_TA*. En estos casos, las métricas relacionadas al tiempo de espera presentan mejoras con grados de admisibilidad muy pequeños, ($\alpha = .2$). Esto se debe a que el esquema de admisibilidad reduce el tiempo de espera de las tareas grandes, que seguramente toman poco tiempo en su ejecución. De esta forma, las tareas pequeñas esperan menos tiempo y por consiguiente el promedio de espera disminuye. Es probable que esto suceda con estas estrategias porque utilizan o se relacionan con el tiempo de espera de cada tarea para hacer las asignaciones correspondientes. Sin embargo, es extraño que las estrategias *Min_WTA* y *Min_WWoTA* no presenten el mismo comportamiento ya que también se relacionan.

En algunos casos, el utilizar el esquema de admisibilidad provee un cociente de competitividad constante para algunas estrategias como *Min_LB* y *Min_CT*. Esto es muy importante porque asegura que la longitud de los calendarios no se alejará del óptimo en más de cierto rango. Aunque la estrategia genere calendarios buenos en promedio, siempre existe la posibilidad de la generación de un calendario extremadamente malo, y algunas veces sin ningún límite conocido. Esto no sucede con un cociente de competitividad constante.

La métrica más importante tratada en este trabajo de investigación es el cociente de competitividad. Esta métrica se relaciona directamente con las métricas del sistema. Por lo tanto, al observar una mejora en el cociente de competitividad es común que ocurra con las métricas del sistema. Sin embargo, existen casos particulares en los que, al reducir la longitud del calendario en promedio, no se presentan mejoras en las métricas del sistema. Un caso como este es la estrategia *Min_PL*.

Treinta experimentos validan estadísticamente los resultados que se obtuvieron durante la etapa de experimentación. Sin embargo, estos resultados son susceptibles a los cambios en la configuración del Grid y la carga de trabajo como se apreció en los experimentos iniciales al estar verificando el funcionamiento del simulador. El problema que trata el esquema de admisibilidad es el retraso de tareas grandes a causa de las tareas pequeñas; al evitar esto, se puede reducir la longitud del calendario generado. Por lo tanto, al utilizar una carga de trabajo con la que no se creen colas de espera, el esquema de admisibilidad no tiene oportunidad de reducir la longitud del calendario. Incluso, con una carga de trabajo que cree colas de espera, si el tamaño de todas las máquinas del Grid es el mismo, el esquema de admisibilidad tampoco generará los resultados obtenidos en este trabajo de investigación.

Para corroborar esta información, se realizaron experimentos con cargas de trabajo conteniendo información de tareas reales. Estas cargas de trabajo semi-sintéticas se crearon, a partir de registros de centros de cómputo reales, con un nuevo proceso que utiliza la información de diferentes registros para crear una carga de trabajo compuesta. Comprobamos que la carga de trabajo es adecuada para la simulación de calendarización en Grid computacional. Los experimentos iniciales con los registros independientes mostraron que no son aptos para simulaciones en Grid, las características de las tareas son específicas a un solo centro de cómputo. Incluso las pocas cargas de trabajo de Grid que se encuentran disponibles no son buenas, no contienen la información necesaria o las características de las tareas son prácticamente irreales en la actualidad. El método propuesto para la generación

de cargas de trabajo semi-sintéticas es de gran ayuda para evaluar los diferentes algoritmos de calendarización en Grid.

Finalmente, se muestra que el esquema de admisibilidad es un concepto muy simple y de fácil implementación. Puede ayudar a mejorar el desempeño del Grid computacional si se configura adecuadamente el grado de admisibilidad. Además, para los sistemas que tienen como objetivo minimizar criterios del usuario, es una buena opción ya que reduce los tiempos de espera de las tareas. Sin embargo, al aplicarlo incorrectamente puede dar resultados no deseados; un grado de admisibilidad muy pequeño generalmente incrementa la longitud de los calendarios.

VI. 2. Trabajo futuro

Como trabajo futuro se pretende continuar con el estudio del esquema de admisibilidad al utilizar otras configuraciones de Grid. Se cree que este esquema es sensible a la configuración del sistema y de la carga de trabajo. De esta manera se desea conocer el grado de impacto que tiene el Grid computacional y la carga de trabajo en el esquema de admisibilidad.

Otro aspecto interesante es el comportamiento de otras estrategias de calendarización al utilizar el esquema de admisibilidad. Se desea introducir las al simulador para estudiarlas de manera experimental.

Este trabajo muestra resultados experimentales con un grado de admisibilidad constante durante la calendarización. Pero no necesariamente es la mejor opción, si la carga de trabajo es principalmente secuencial, entonces un grado de admisibilidad alto es mejor. En cambio, para una carga de trabajo principalmente paralela un grado de admisibilidad menor es el adecuado. El adaptar este grado durante la calendarización podría generar mejores resultados. Esta variación depende de la carga de trabajo y del estado del sistema al momento de calcular el mejor valor. Se planea introducir este esquema de admisibilidad

adaptable al simulador para realizar un análisis experimental y se esperan mejores resultados en la calendarización.

Una línea de investigación interesante y no explorada es la de calendarización en Grid jerárquico de tres niveles. En este nuevo modelo se pretende aprovechar el hecho de que algunas computadoras de alto desempeño actuales tienen particiones y se comportan como máquinas independientes, pero se encuentran en el mismo espacio geográfico. De esta manera, si se aplican diferentes algoritmos de calendarización, se puede mejorar su desempeño al crear un calendario colaborativo. Se definirá formalmente el modelo y se diseñarán algoritmos de calendarización adecuados. Se analizarán teórica y experimentalmente para comprobar su eficiencia.

Referencias

Andrieux, A., D. Berry, J. Garibaldi, S. Jarvis, J. MacLaren, D. Ouelhadj y D. Snelling. 2003. *Open Issues in Grid Scheduling*. National e-Science Centre. Edinburgh, Scotland. Reporte del workshop llevado a cabo en e-Science Institute, 21-22 de octubre ISSN 1751-5971.

Anoep, S., C. Dumitrescu, D. Epema, A. Iosup, M. Jan, H. Li y L. Wolters. 6 de septiembre de 2007. *The Grid Workload Format*. Technische Universiteit Delft. http://gwa.ewi.tudelft.nl/TheGridWorkloadFormat_v001.pdf (último acceso: 22 de diciembre de 2008)

Anoep, S., C. Dumitrescu, D. Epema, A. Iosup, M. Jan, H. Li y L. Wolters. 7 de septiembre de 2007. *The Grid Workloads Archive*. <http://gwa.ewi.tudelft.nl/pmwiki/pmwiki.php?n=Main.Home> (último acceso: 10 de mayo de 2009)

Berman, F. 1998. *High-performance schedulers*. En: I Foster y C. Kesselman (eds.). *The GRID. Blueprint for a New Computing Infrastructure*. Morgan Kaufmann Publishers. Primera edición. San Francisco. 279-310 p.

Błażewicz, J., K. H. Ecker, E. Pesch, G. Schmidt y J. Węglarz. 2007. *Handbook on Scheduling: From Theory to Applications*. Springer-Verlag. Primera edición. Berlin. 647 pp.

Buyya, R., D. Abramson y S. Venugopal. 2005. *The Grid Economy, Special Issue on Grid Computing*. Proceedings of the IEEE. 93(3): 698-714 p.

Calzarossa, M., A. P. Merlo, D. Tessaera, G. Haring y G. Kotsis. 1995. *A hierarchical approach to workload characterization for parallel systems*. High-Performance Computing and Networking, International Conference and Exhibition, HPCN Europe 1995. 3-5 de mayo. Milan, Italia.

Calzarossa, M. y G. Serazzi. 1993. *Workload Characterization: A Survey*. Proceedings of the IEEE. 81(8): 1136-1150 p.

Chapin, S. J., W. Cirne, D. G. Feitelson, J. P. Jones, S. T. Leutenegger, U. Schwiegelshohn, W. Smith y D. Talby. 1999. *Benchmarks and Standards for the Evaluation of Parallel Job Schedulers*. Lecture Notes in Computer Science. 1659(1999): 67-90 p.

Dong, F. y S. G. Akl. 2006. *Scheduling Algorithms for Grid Computing: State of the Art and Open Problems*. School of Computing, Queen's University. Kingston, Ontario, Canada. Reporte técnico 2006-504.

Etsion, Y. y D. Tsafrir. 2005. *A short survey of commercial cluster batch schedulers*. School of Computer Science and Engineering, the Hebrew University. Jerusalem, Israel. Reporte técnico 2005-13.

Feitelson, Dror G. 8 de diciembre de 2005. *Parallel Workloads Archive*. The Hebrew University, Jerusalem, Israel. <http://www.cs.huji.ac.il/labs/parallel/workload/> (último acceso: 22 de diciembre de 2008)

Feitelson, Dror G. 8 de diciembre de 2005. *The Standard Workload Format*. The Hebrew University, Jerusalem, Israel. <http://www.cs.huji.ac.il/labs/parallel/workload/swf.html> (último acceso: 22 de diciembre de 2008)

Foster, I. y C. Kesselman. 2004. *Concepts and Architecture*. En: I. Foster y C. Kesselman (eds.). *The Grid 2. Blueprint for a New Computing Infrastructure*. Morgan Kaufmann. Segunda edición. San Francisco. 37-64 p.

Foster, I. y C. Kesselman. 2003. *The Grid in a Nutshell*. En: J. Nabrzyski, J. M. Schopf y J. Weglarz (eds.). *GRID RESOURCE MANAGEMENT State of the Art and Future Trends*. Kluwer Academic Publisher. Primera edición. Norwell, Massachusetts. 3-14 p.

Foster, I. y C. Kesselman. 1998. *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann. Primera edición. San Francisco. 677 pp.

Foster, I. 20 de julio de 2002. *What is the Grid? A Three Point Checklist*. Argonne National Laboratory. <http://www-fp.mcs.anl.gov/~foster/Articles/WhatIsTheGrid.pdf> (último acceso: 28 de junio de 2008)

Galaviz, L. M., D. Madrigal, R. Sanchez, A. Lira, J. M. Ramírez, A. Tchernykh y J. A. Verduzco. 2006. *Evaluación Experimental de Estrategias de Calendarización para Selección de Recursos en Grid Computacional*. CICC 2006, 7º Congreso Internacional de las Ciencias Computacionales. 25 - 27 de octubre. Villa de Alvarez, Colima, México.

Garey, M. R. y R. L. Graham. 1975. *Bounds for Multiprocessor Scheduling with Resource Constraints*. SIAM Journal on Applied Mathematics. 4(2): 187-200 p.

Garey, M. R. y D. D. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W.H. Freeman and Company. Primera edición. San Francisco. 340 pp.

Graham, R. L. 1966. *Bounds on Multiprocessing Timing Anomalies*. Bell System Technical Journal. 45(9): 1563-1581 p.

Graham, R. L. 1969. *Bounds on Multiprocessing Timing Anomalies*. SIAM Journal on Applied Mathematics. 17(2): 416-429 p.

Graham, R. L., E. L. Lawler, J. K. Lenstra y A. H. G. Rinnooy Kan. 1979. *Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey*. Annals of Discrete Mathematics. 5(1979): 287-326 p.

Iovanella, A. 2002. *On-line algorithms for multiprocessor task scheduling*. Università degli Studi di Roma "La Sapienza". Roma, Italia. Tesis de doctorado .

Knüpfer, A., R. Brendel, H. Brunst, H. Mix y W. E. Nagel. 2006. *Introducing the Open Trace Format (OTF)*. Lecture Notes in Computer Science. 3992(2006): 526-533 p.

Kurowski, K., J. Nabrzyski, A. Oleksiak y J. Węglarz. 2008. *A multicriteria approach to two-level hierarchy scheduling in grids*. Journal of Scheduling. 11(5): 371-379 p.

Li, K. 2005. *Job scheduling and processor allocation for grid computing on metacomputers*. Journal of Parallel and Distributed Computing. 65(11): 1406-1418 p.

Lifka, David A. 1995. *The ANL/IBM SP scheduling system*. Workshop on Job Scheduling Strategies for Parallel Processing in conjunction with IPPS '95. 25 de abril. Santa Barbara, CA, Estados Unidos de América.

Lublin, U. y D. G. Feitelson. 2003. *The workload on parallel supercomputers: modeling the characteristics of rigid jobs*. Journal of Parallel and Distributed Computing. 63(11): 1105-1122 p.

Madrigal, D., L. M. Galaviz, J. M. Ramírez, R. Sánchez, A. Tchernykh y J. A. Verduzco. 2006. *Estrategias de Calendarización para un Grid Computacional*. CiComp'06, Primer Congreso Internacional de Ciencias Computacionales. 6 - 8 de noviembre. Ensenada, Baja California, México.

Naroska, E y U. Schwiegelshohn. 2002. *On an on-line scheduling problem for parallel jobs*. Information Processing Letters. 81(6): 297-304 p.

Pascual, F., K. Rzadca y D. Trystram. 2007. *Cooperation in multi-organization scheduling*. Euro-Par 2007 Parallel Processing In conjunction with the First CoreGRID. 28-31 agosto. Rennes, Francia.

Ramírez, J. M., A. I. Rodríguez, A. Tchernykh y J. A. Verduzco. 2007. *Rendimiento de Estrategias de Calendarización Considerando Fluctuación de Tiempo de Ejecución de Tareas en un Grid Computacional*. Conferencia Latinoamericana de Computación de Alto Rendimiento. 13 - 18 de agosto. Santa Marta, Colombia.

Schwiegelshohn, U., A. Tchernykh y R. Yahyapour. 2008. *Online Scheduling in Grids*. IPDPS 2008 Conference. 14-18 de abril. Miami, Florida, EUA.

Sgall, J. 1998. *On-line scheduling*. En: A. Fiat y G. J. Woeginger (eds.). *Lecture Notes in Computer Science*. Springer. Primera edición. Berlin. 196-231 p.

Tchernykh, A., U. Schwiegelshohn, R. Yahyapour y N. Kuzjurin. 2008. *Online hierarchical job scheduling on Grids*. En: Marco Vanneschi Thierry Priol (eds.). *From Grids To Service and Pervasive Computing*. Springer. Primera edición. Berlin. 77-91 p.

Tchernykh, A., J. M. Ramírez, A. Avetisyan, N. Kuzjurin, D. Grushin y S. Zhuk. 2006. *Two Level Job-Scheduling Strategies for a Computational Grid*. *Lecture Notes in Computer Science*. 3911(2006): 774-781 p.

Tsafir, D., Y. Etsion y D. G. Feitelson. 2005. *Modeling User Runtime Estimates*. *Lecture Notes in Computer Science*. 3834(2005): 1-35 p.

Zhang, G. y D. Ye. 2002. *A note on on-line scheduling with partial information*. *Computers & Mathematics with Applications*. 44(3): 539-543 p.

Zhu, Y. 2004. *A Survey on Grid Scheduling Systems*. Hong Kong University of Science and Technology. Kowloon, Hong Kong. Reporte técnico .

Zhuk, S., A. Tchernykh, N. Kuzjurin, A. Pospelov, A. Shokurov, A. Avetisyan, S. Gaissaryan y Grushin D.. 2004. *Comparison of Scheduling Heuristics for Grid Resource Broker*. PCS2004 Third International Conference on Parallel Computing Systems in conjunction with ENC'04 Mexican International Conference in Computer Science. 20 - 24 de septiembre. Colima, Mexico.

EC-cofunded GridTalk project. 1 de mayo de 2008. *The place for everybody to learn about the Grid*. EC-cofunded GridTalk project. <http://gridcafe.web.cern.ch/gridcafe/index.html> (último acceso: 27 de junio de 2008)

Platform Computing Inc. 1 de junio de 2008. *Platform Computing*. Platform Computing Inc.. <http://www.platform.com/> (último acceso: 28 de junio de 2008)

The University of Chicago. 1 de enero de 1997. *The Globus Alliance*. The University of Chicago. <http://www.globus.org/> (último acceso: 13 de agosto de 2008)

The University of Melbourne. 17 de junio de 2004. *The Grid Computing and Distributed Systems (GRIDS) Laboratory*. The University of Melbourne. <http://www.gridbus.org/> (último acceso: 13 de agosto de 2008)

The University of Virginia. 1 de agosto de 1993. *The Legion Project*. The University of Virginia. <http://legion.virginia.edu/> (último acceso: 13 de agosto de 2008)

The University of Wisconsin Madison. 1 de enero de 1988. *The Condor Project Homepage*.
The University of Wisconsin Madison. <http://www.cs.wisc.edu/condor/> (último acceso: 28 de junio de 2008)

APÉNDICE

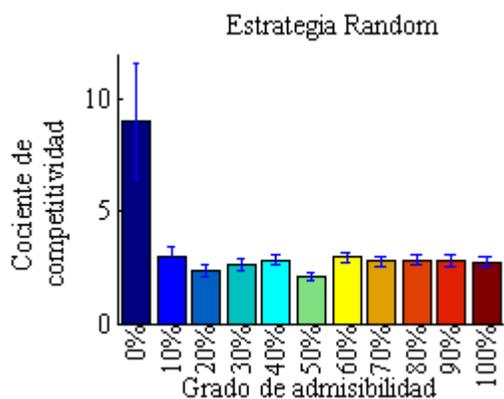
Estrategia *Random*

Figura 46. Cociente de competitividad, estrategia *Random* con admisibilidad.

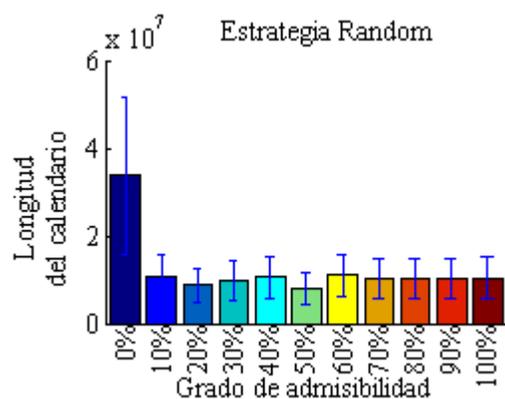


Figura 47. Longitud del calendario, estrategia *Random* con admisibilidad.

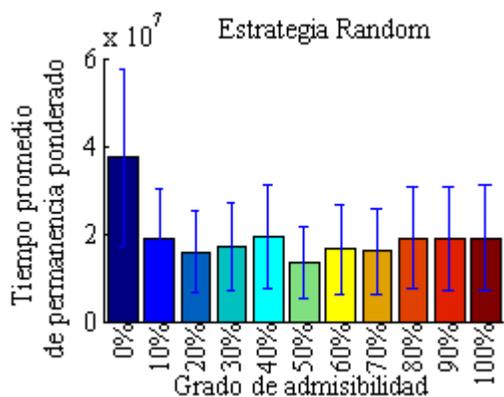


Figura 48. Tiempo promedio de permanencia ponderado, estrategia *Random* con admisibilidad.

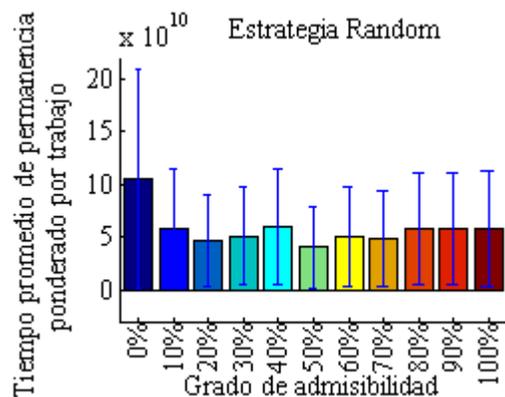


Figura 49. Tiempo promedio de permanencia ponderado por trabajo, estrategia *Random* con admisibilidad.

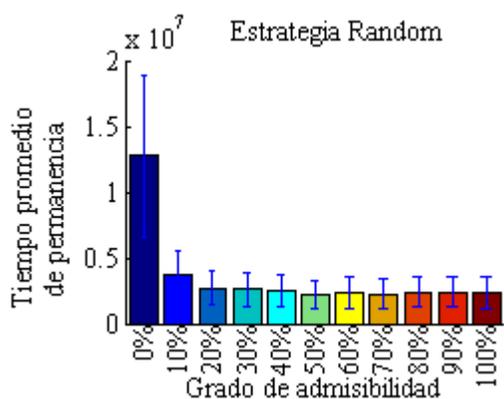


Figura 50. Tiempo promedio de permanencia, estrategia *Random* con admisibilidad.

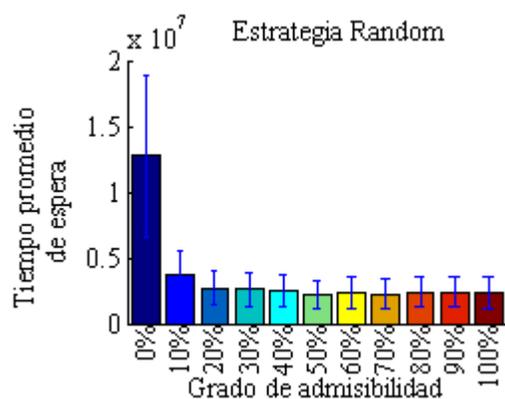


Figura 51. Tiempo promedio de espera, estrategia *Random* con admisibilidad.

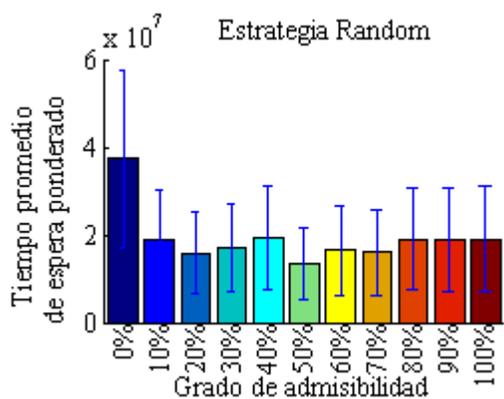


Figura 52. Tiempo promedio de espera ponderado, estrategia *Random* con admisibilidad.

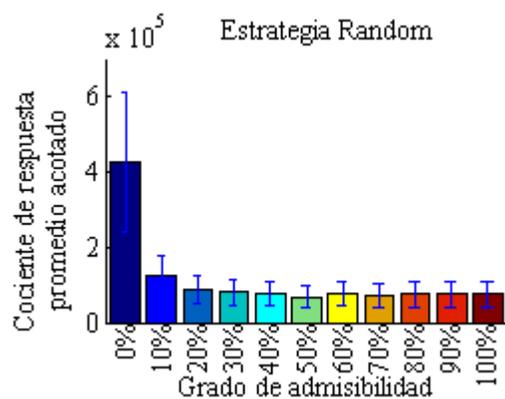


Figura 53. Cociente de respuesta promedio acotado, estrategia *Random* con admisibilidad.

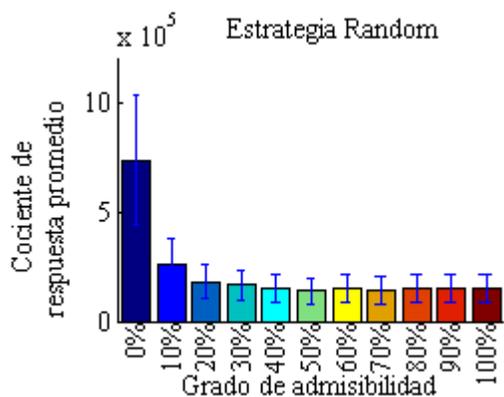


Figura 54. Cociente de respuesta promedio, estrategia *Random* con admisibilidad.

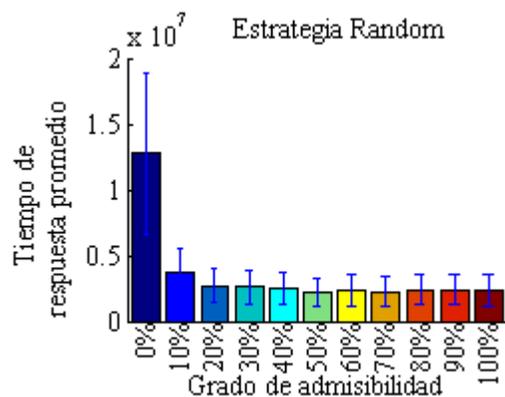


Figura 55. Tiempo de respuesta promedio, estrategia *Random* con admisibilidad.

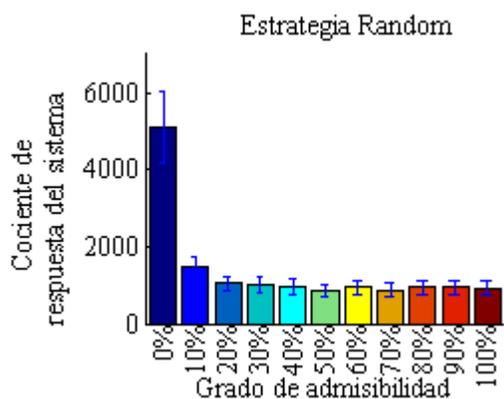


Figura 56. Cociente de respuesta del sistema, estrategia *Random* con admisibilidad.

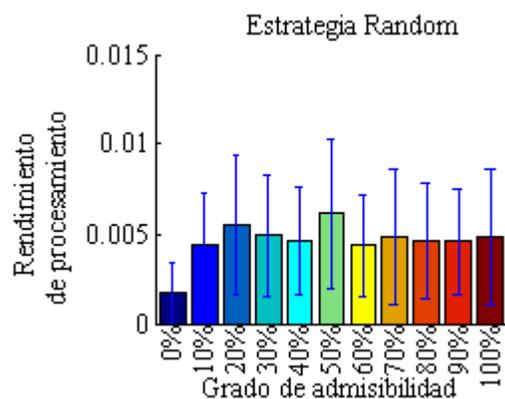


Figura 57. Rendimiento de procesamiento, estrategia *Random* con admisibilidad.

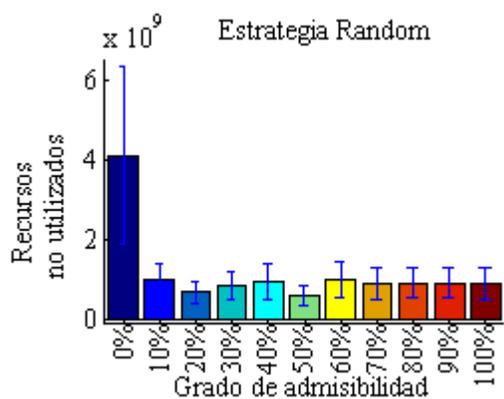


Figura 58. Recursos no utilizados, estrategia *Random* con admisibilidad.

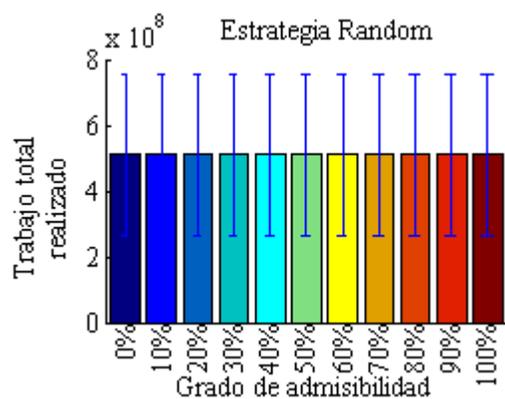


Figura 59. Trabajo total realizado, estrategia *Random* con admisibilidad.

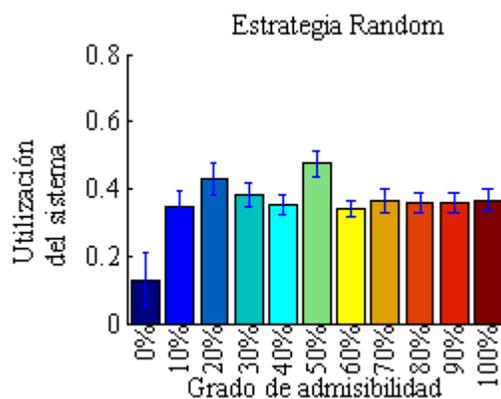


Figura 60. Utilización del sistema, estrategia *Random* con admisibilidad.

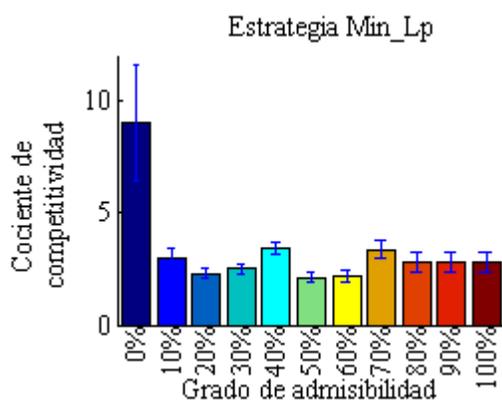
Estrategia *Min_Lp*

Figura 61. Cociente de competitividad, estrategia *Min_Lp* con admisibilidad.

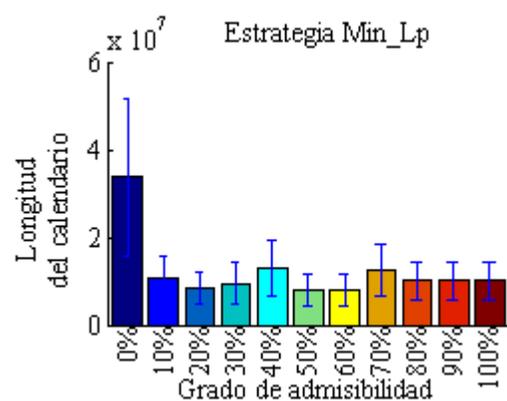


Figura 62. Longitud del calendario, estrategia *Min_Lp* con admisibilidad.

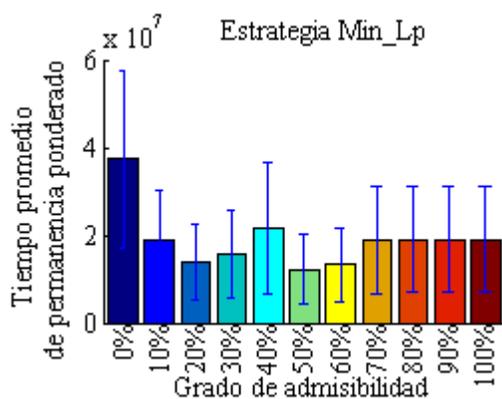


Figura 63. Tiempo promedio de permanencia ponderado, estrategia *Min_Lp* con admisibilidad.

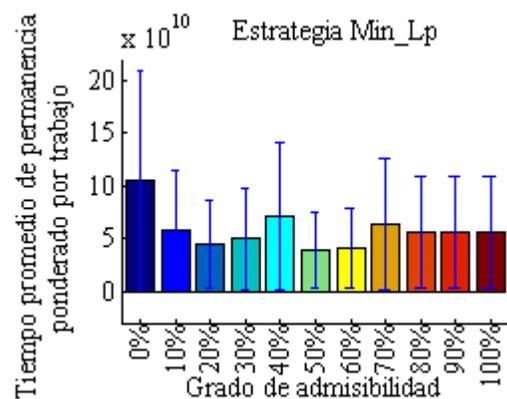


Figura 64. Tiempo promedio de permanencia ponderado por trabajo, estrategia *Min_Lp* con admisibilidad.

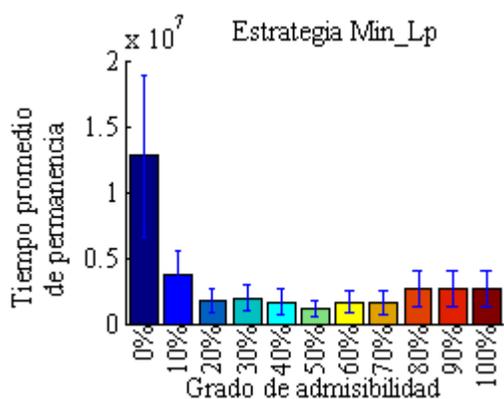


Figura 65. Tiempo promedio de permanencia, estrategia *Min_Lp* con admisibilidad.

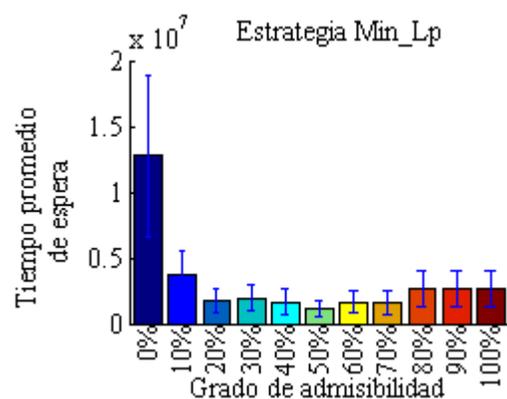


Figura 66. Tiempo promedio de espera, estrategia *Min_Lp* con admisibilidad.

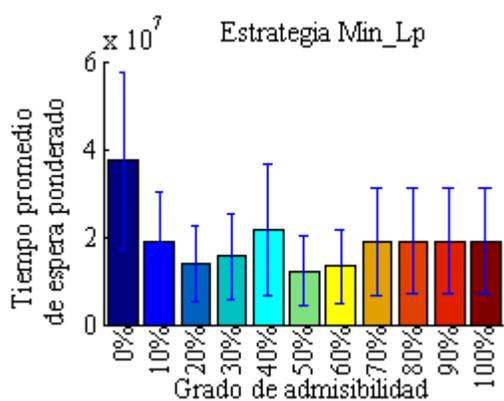


Figura 67. Tiempo promedio de espera ponderado, estrategia *Min_Lp* con admisibilidad.

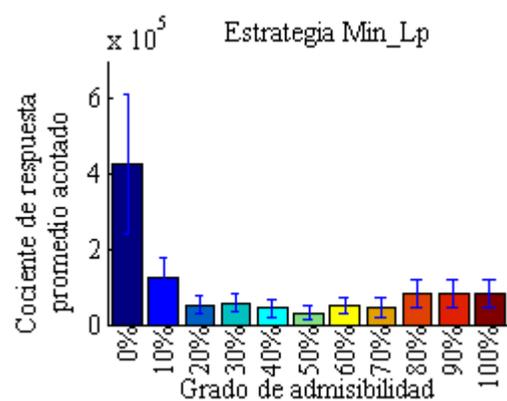


Figura 68. Cociente de respuesta promedio acotado, estrategia *Min_Lp* con admisibilidad.

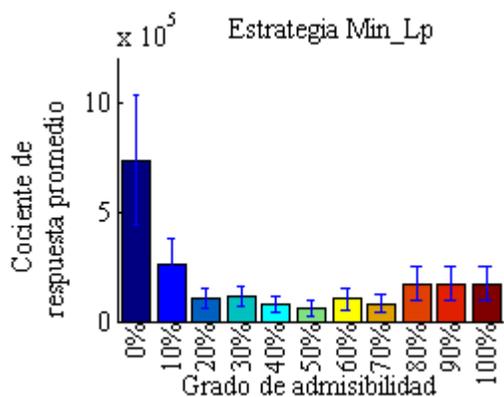


Figura 69. Cociente de respuesta promedio, estrategia *Min_Lp* con admisibilidad.

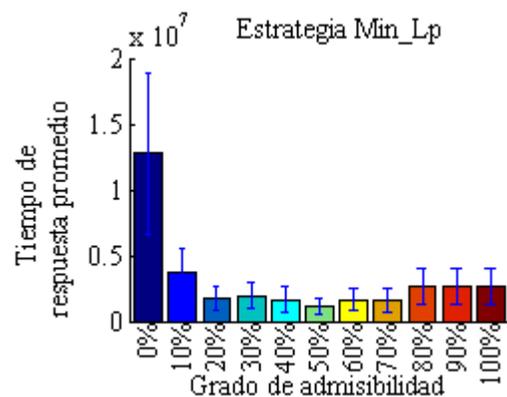


Figura 70. Tiempo de respuesta promedio, estrategia *Min_Lp* con admisibilidad.

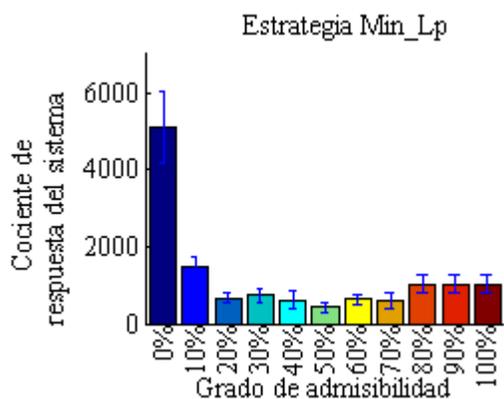


Figura 71. Cociente de respuesta del sistema, estrategia *Min_Lp* con admisibilidad.

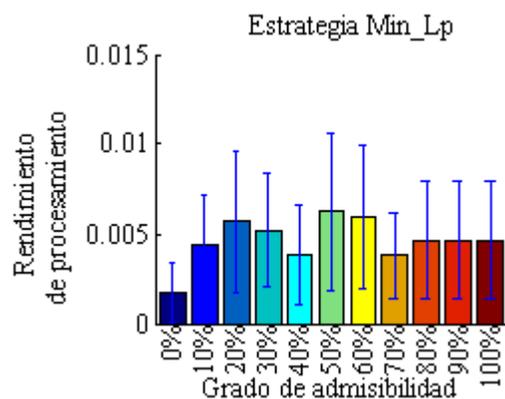


Figura 72. Rendimiento de procesamiento, estrategia *Min_Lp* con admisibilidad.

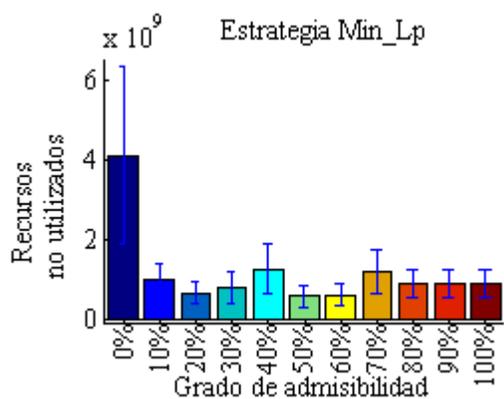


Figura 73. Recursos no utilizados, estrategia *Min_Lp* con admisibilidad.

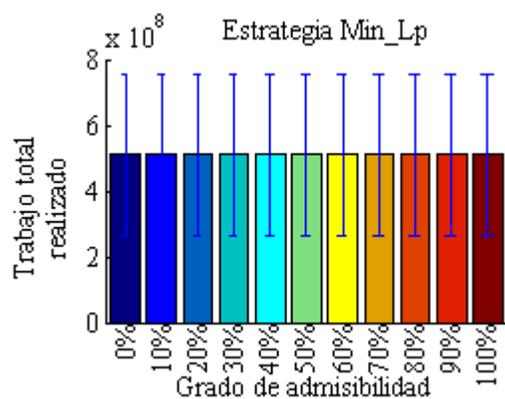


Figura 74. Trabajo total realizado, estrategia *Min_Lp* con admisibilidad.

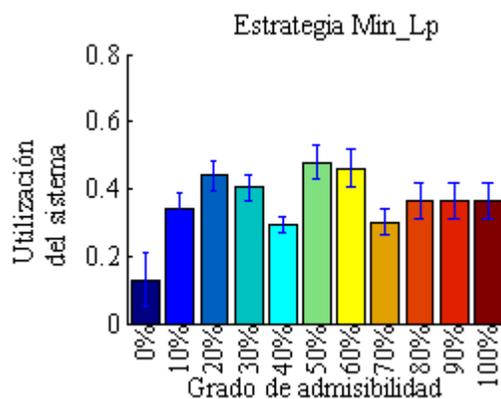


Figura 75. Utilización del sistema, estrategia *Min_Lp* con admisibilidad.

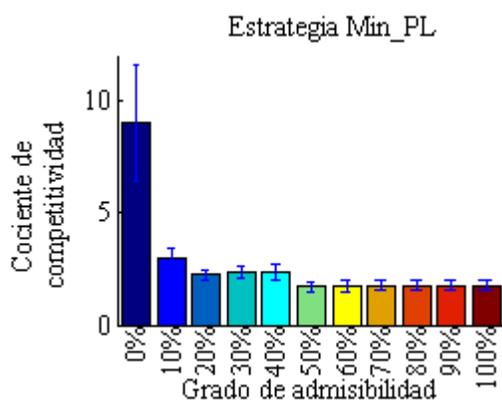
Estrategia *Min_PL*

Figura 76. Cociente de competitividad, estrategia *Min_PL* con admisibilidad.

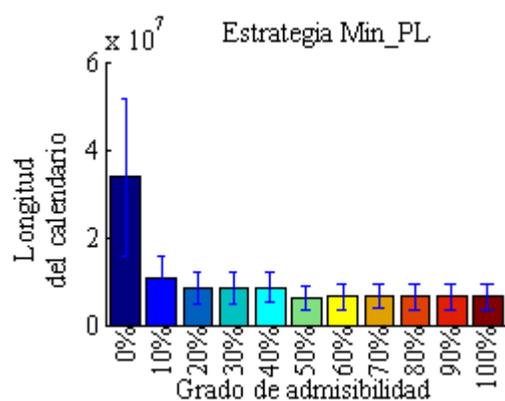


Figura 77. Longitud del calendario, estrategia *Min_PL* con admisibilidad.

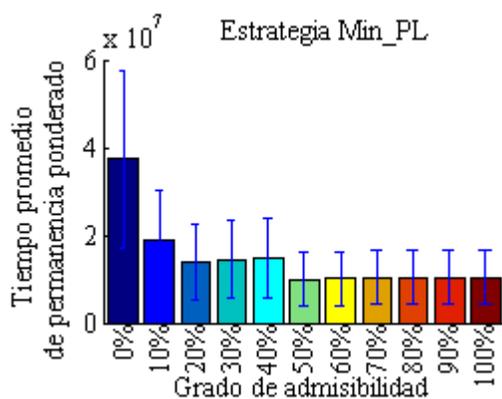


Figura 78. Tiempo promedio de permanencia ponderado, estrategia *Min_PL* con admisibilidad.

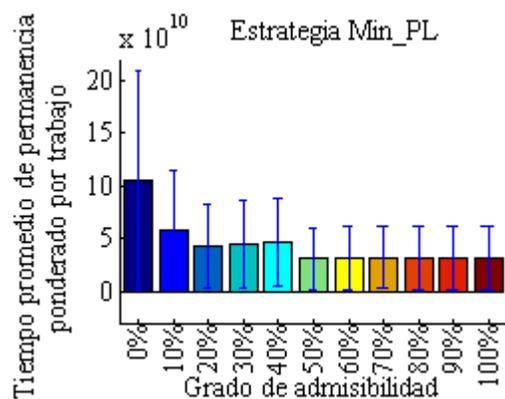


Figura 79. Tiempo promedio de permanencia ponderado por trabajo, estrategia *Min_PL* con admisibilidad.

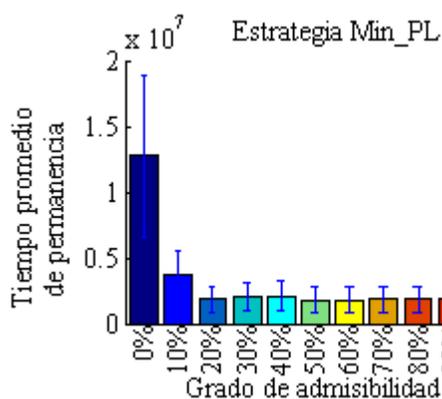


Figura 80. Tiempo promedio de permanencia, estrategia *Min_PL* con admisibilidad.

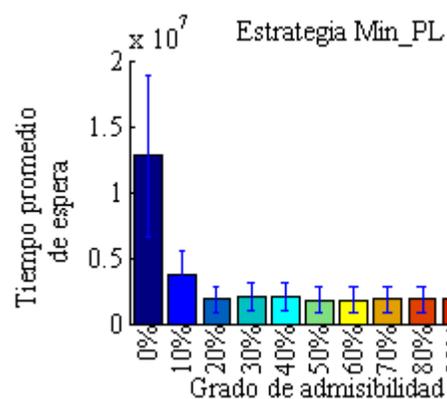


Figura 81. Tiempo promedio de espera, estrategia *Min_PL* con admisibilidad.

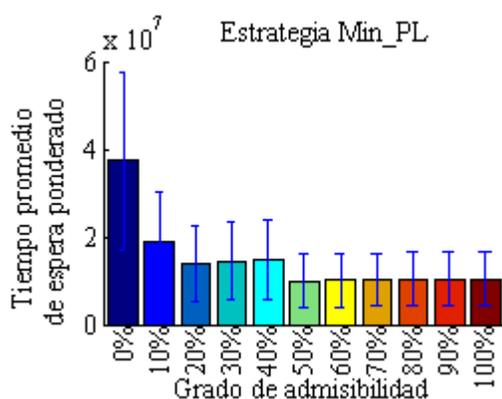


Figura 82. Tiempo promedio de espera ponderado, estrategia *Min_PL* con admisibilidad.

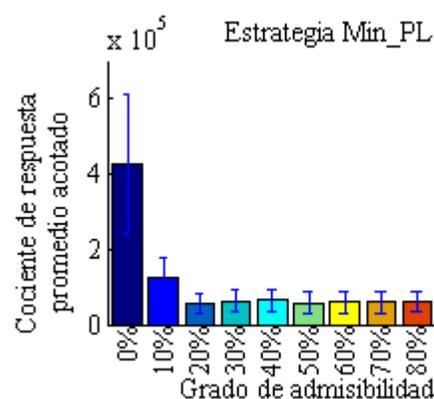


Figura 83. Cociente de respuesta promedio acotado, estrategia *Min_PL* con admisibilidad.

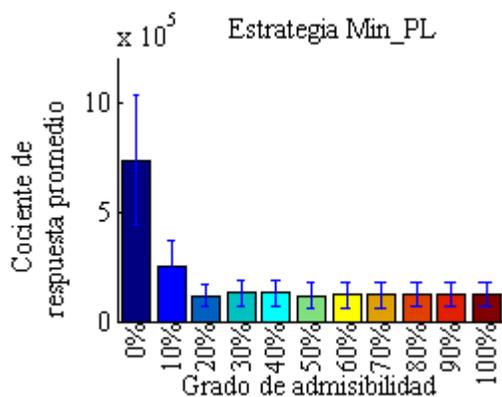


Figura 84. Cociente de respuesta promedio, estrategia *Min_PL* con admisibilidad.

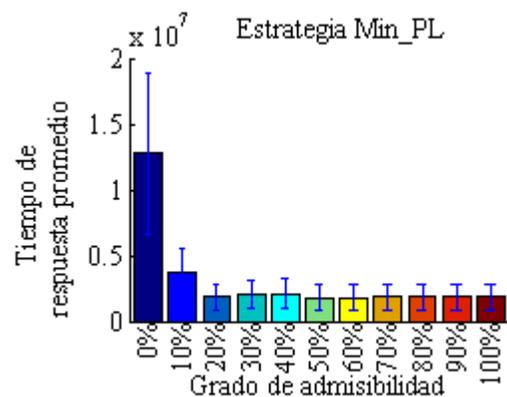


Figura 85. Tiempo de respuesta promedio, estrategia *Min_PL* con admisibilidad.

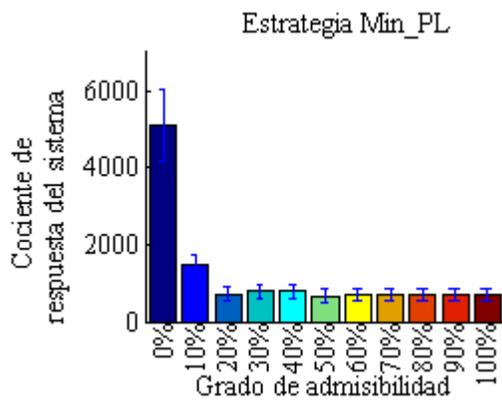


Figura 86. Cociente de respuesta del sistema, estrategia *Min_PL* con admisibilidad.

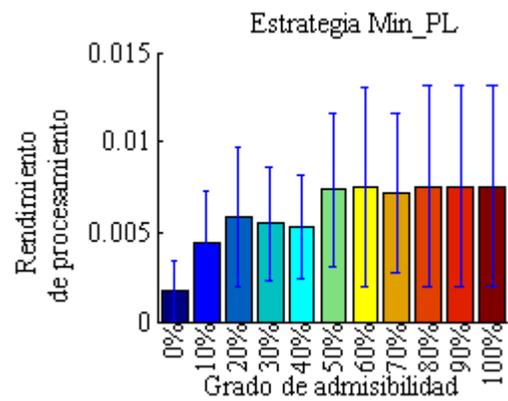


Figura 87. Rendimiento de procesamiento, estrategia *Min_PL* con admisibilidad.

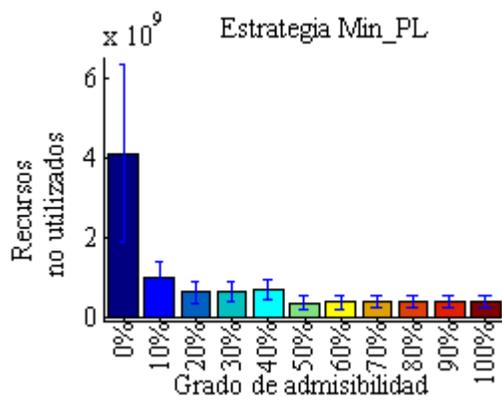


Figura 88. Recursos no utilizados, estrategia *Min_PL* con admisibilidad.

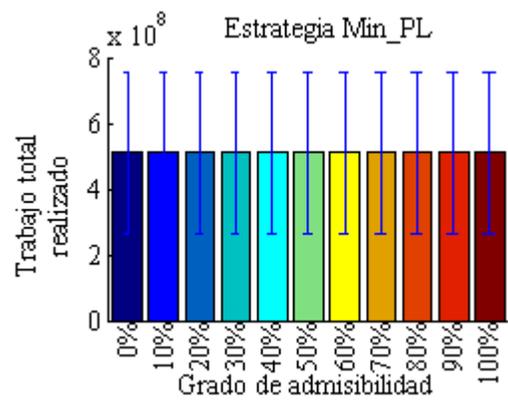


Figura 89. Trabajo total realizado, estrategia *Min_PL* con admisibilidad.

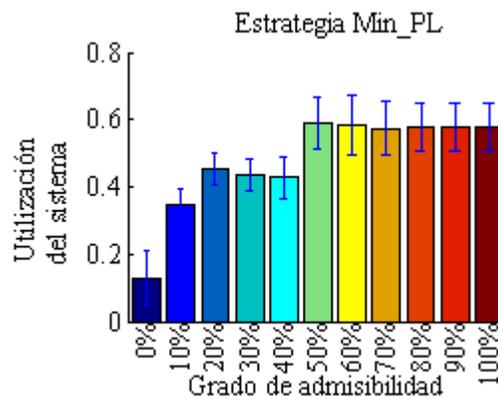


Figura 90. Utilización del sistema, estrategia *Min_PL* con admisibilidad.

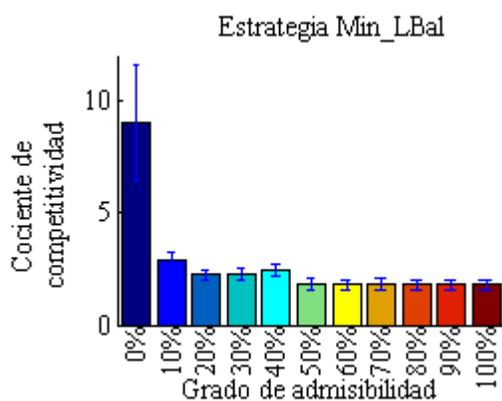
Estrategia *Min_LBal*

Figura 91. Cociente de competitividad, estrategia *Min_LBal* con admisibilidad.

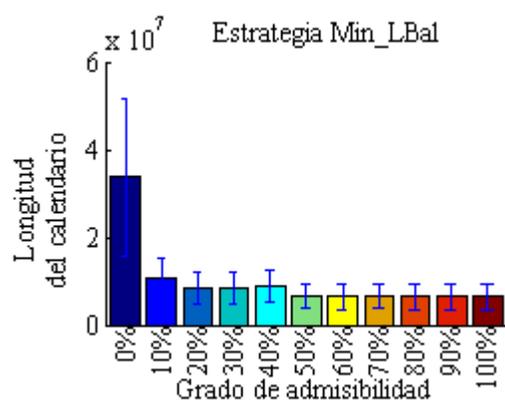


Figura 92. Longitud del calendario, estrategia *Min_LBal* con admisibilidad.

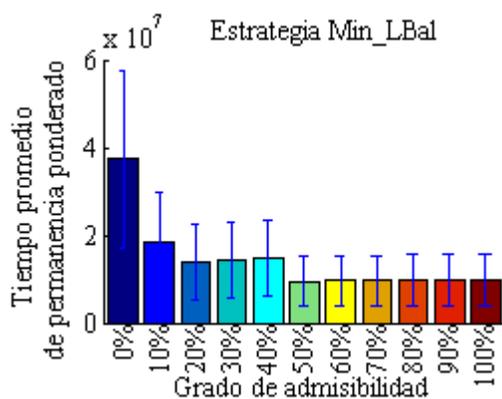


Figura 93. Tiempo promedio de permanencia ponderado, estrategia *Min_LBal* con admisibilidad.

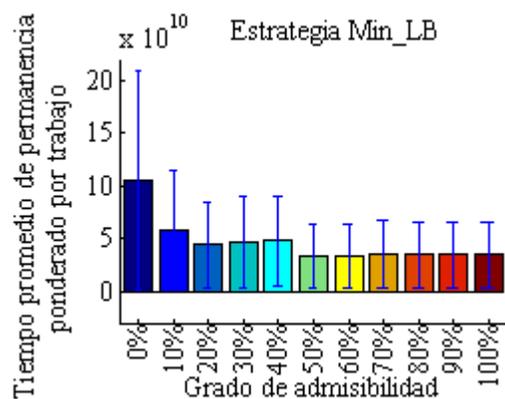


Figura 94. Tiempo promedio de permanencia ponderado por trabajo, estrategia *Min_LBal* con admisibilidad.

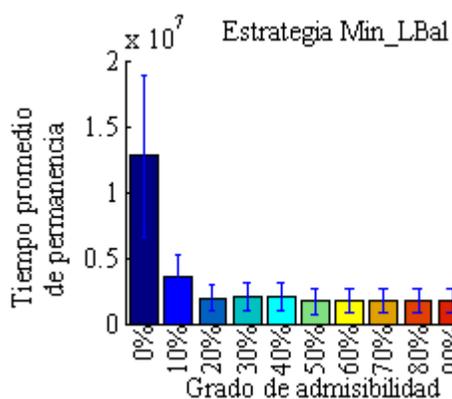


Figura 95. Tiempo promedio de permanencia, estrategia *Min_LBal* con admisibilidad.

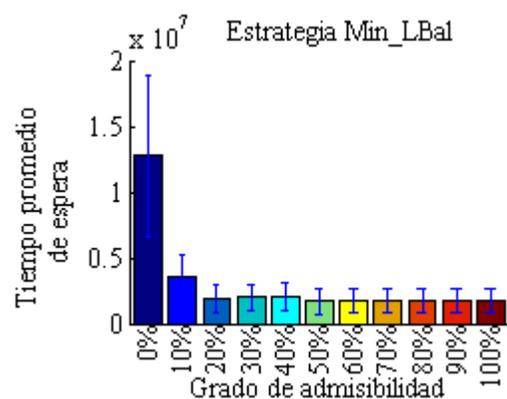


Figura 96. Tiempo promedio de espera, estrategia *Min_LBal* con admisibilidad.

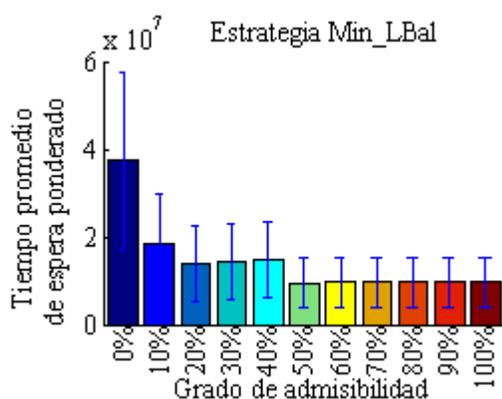


Figura 97. Tiempo promedio de espera ponderado, estrategia *Min_LBal* con admisibilidad.

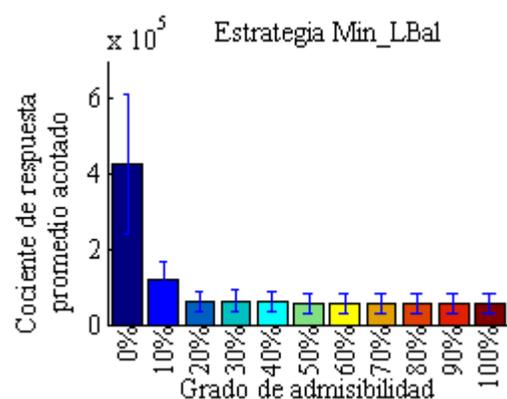


Figura 98. Cociente de respuesta promedio acotado, estrategia *Min_LBal* con admisibilidad.

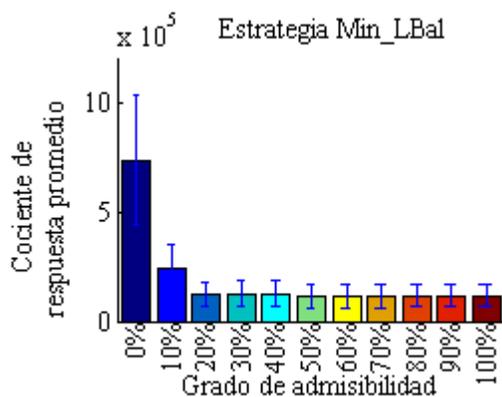


Figura 99. Cociente de respuesta promedio, estrategia *Min_LBal* con admisibilidad.

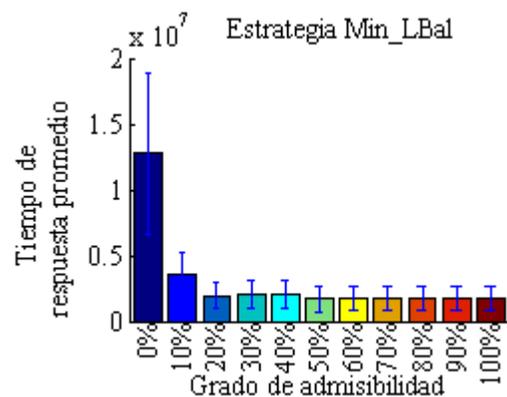


Figura 100. Tiempo de respuesta promedio, estrategia *Min_LBal* con admisibilidad.

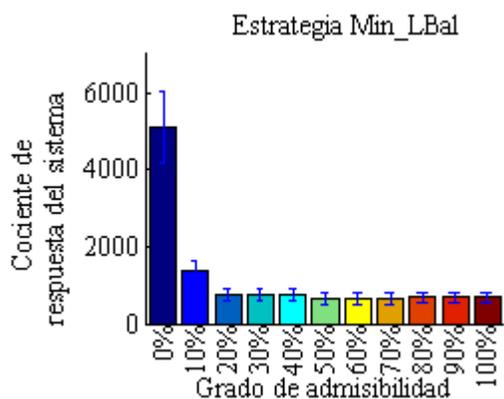


Figura 101. Cociente de respuesta del sistema, estrategia *Min_LBal* con admisibilidad.

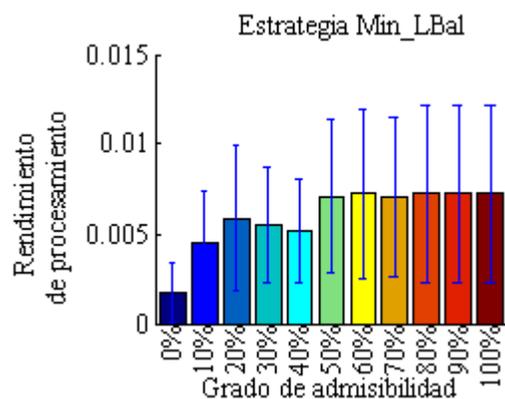


Figura 102. Rendimiento de procesamiento, estrategia *Min_LBal* con admisibilidad.

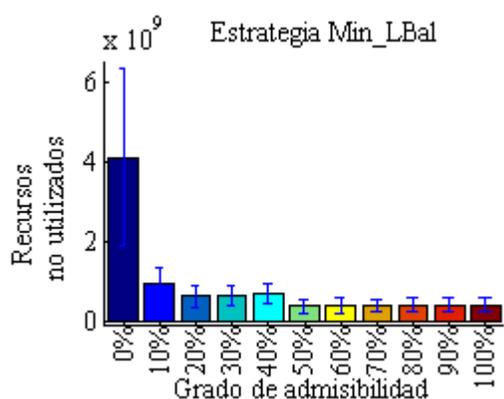


Figura 103. Recursos no utilizados, estrategia *Min_LBal* con admisibilidad.

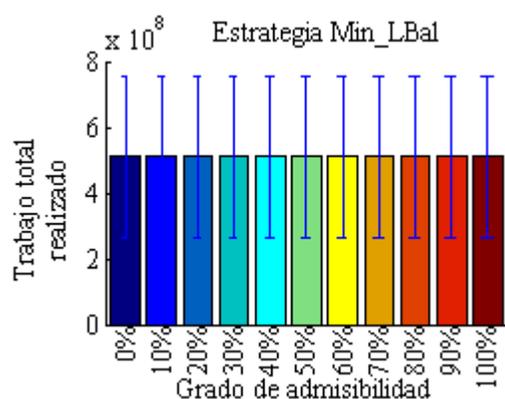


Figura 104. Trabajo total realizado, estrategia *Min_LBal* con admisibilidad.

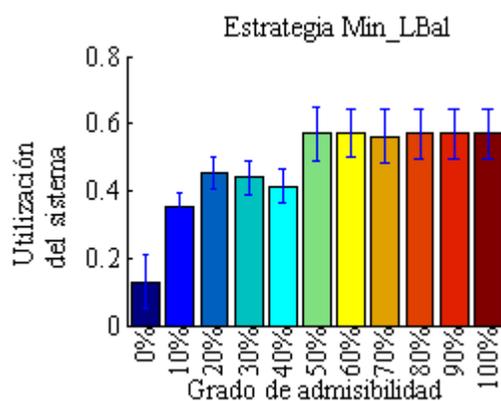


Figura 105. Utilización del sistema, estrategia *Min_LBal* con admisibilidad.

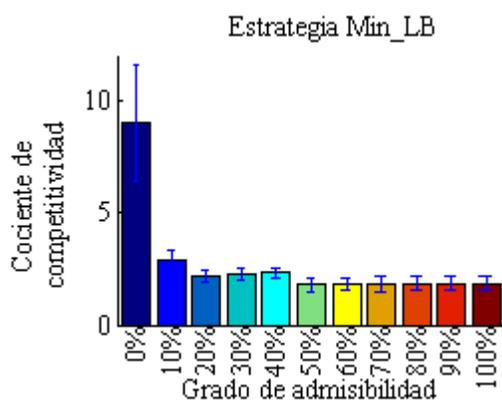
Estrategia *Min_LB*

Figura 106. Cociente de competitividad, estrategia *Min_LB* con admisibilidad.

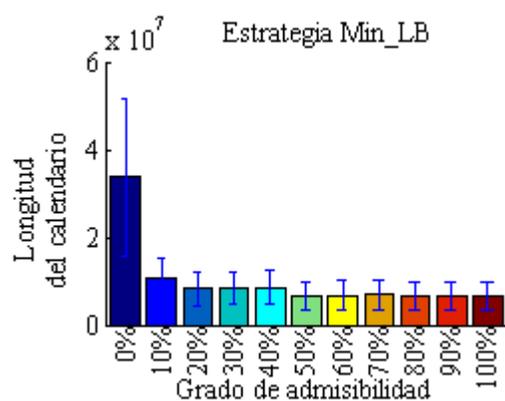


Figura 107. Longitud del calendario, estrategia *Min_LB* con admisibilidad.

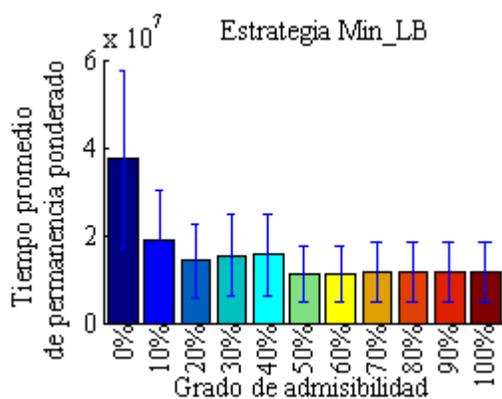


Figura 108. Tiempo promedio de permanencia ponderado, estrategia *Min_LB* con admisibilidad.

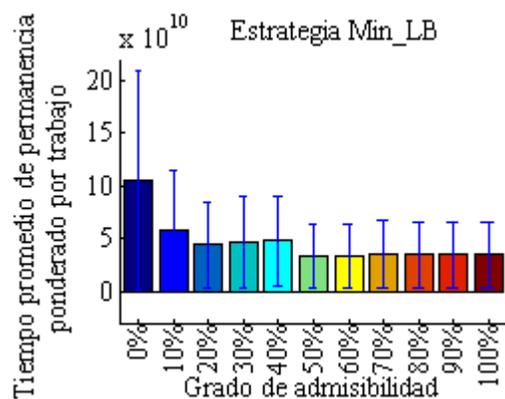


Figura 109. Tiempo promedio de permanencia ponderado por trabajo, estrategia *Min_LB* con admisibilidad.

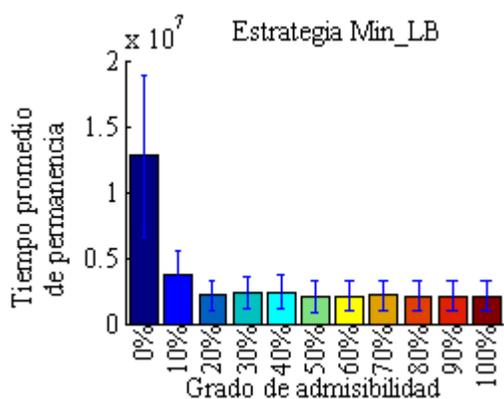


Figura 110. Tiempo promedio de permanencia, estrategia *Min_LB* con admisibilidad.

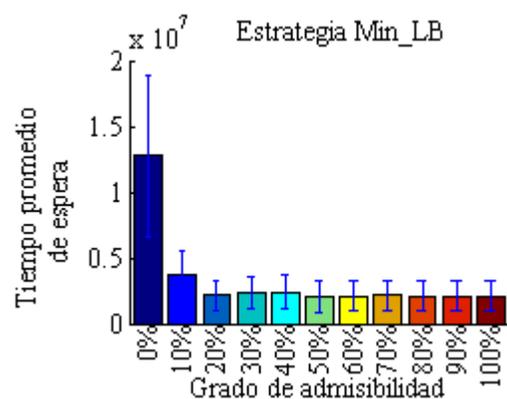


Figura 111. Tiempo promedio de espera, estrategia *Min_LB* con admisibilidad.

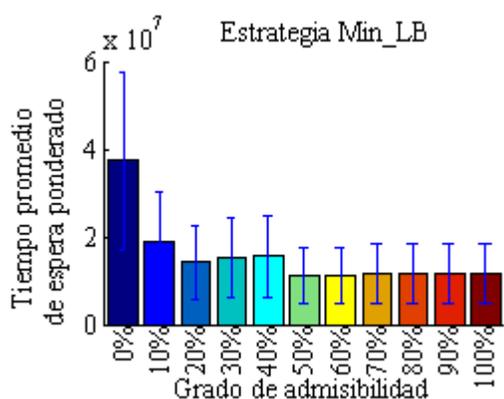


Figura 112. Tiempo promedio de espera ponderado, estrategia *Min_LB* con admisibilidad.

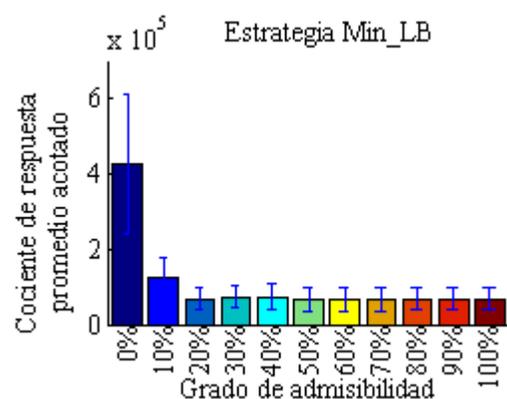


Figura 113. Cociente de respuesta promedio acotado, estrategia *Min_LB* con admisibilidad.

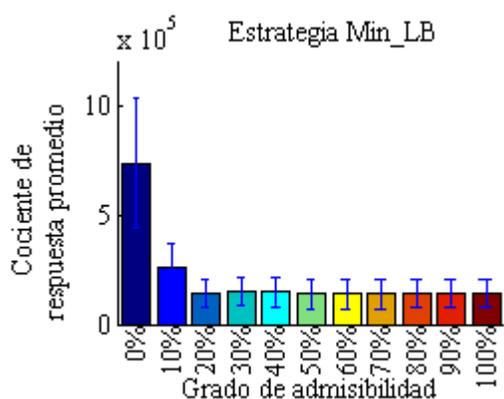


Figura 114. Cociente de respuesta promedio, estrategia *Min_LB* con admisibilidad.

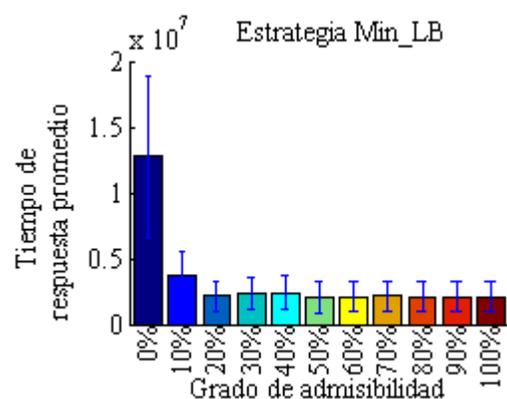


Figura 115. Tiempo de respuesta promedio, estrategia *Min_LB* con admisibilidad.

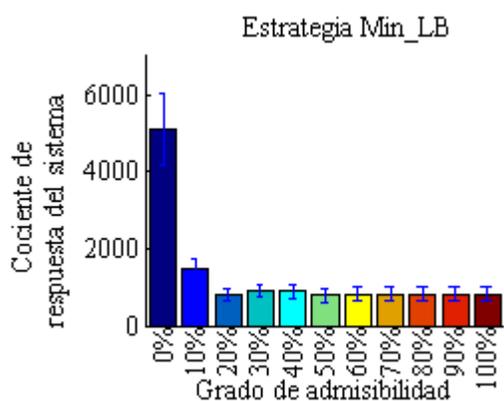


Figura 116. Cociente de respuesta del sistema, estrategia *Min_LB* con admisibilidad.

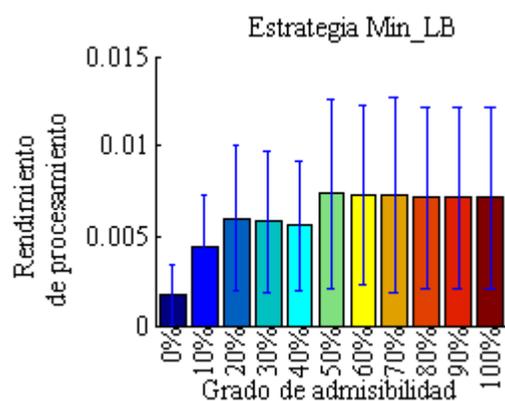


Figura 117. Rendimiento de procesamiento, estrategia *Min_LB* con admisibilidad.

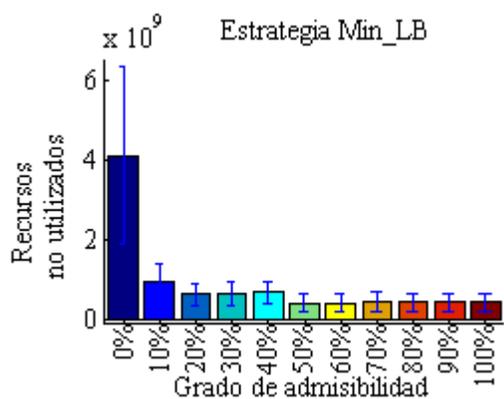


Figura 118. Recursos no utilizados, estrategia *Min_LB* con admisibilidad.

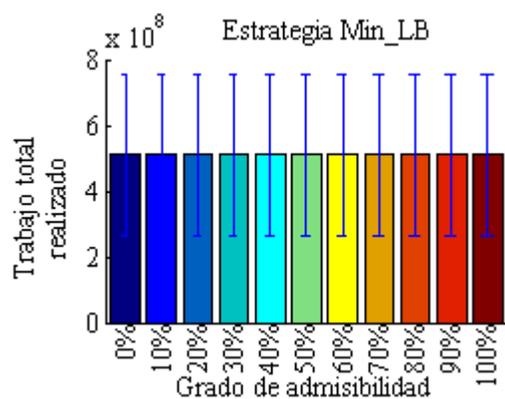


Figura 119. Trabajo total realizado, estrategia *Min_LB* con admisibilidad.

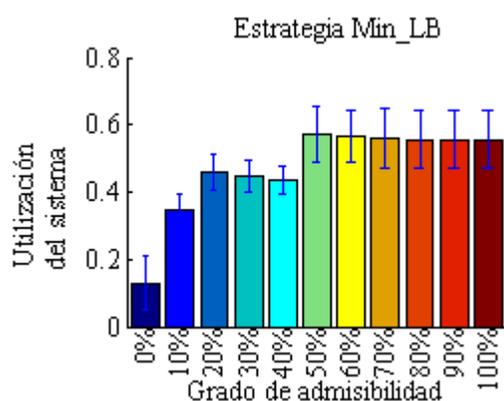


Figura 120. Utilización del sistema, estrategia *Min_LB* con admisibilidad.

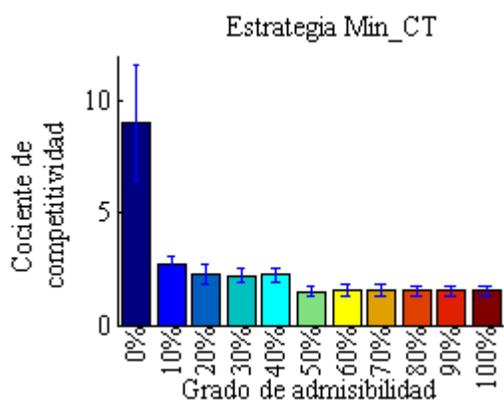
Estrategia *Min_CT*

Figura 121. Cociente de competitividad, estrategia *Min_CT* con admisibilidad.

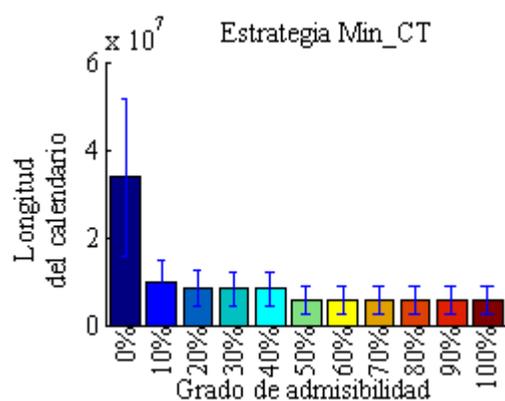


Figura 122. Longitud del calendario, estrategia *Min_CT* con admisibilidad.

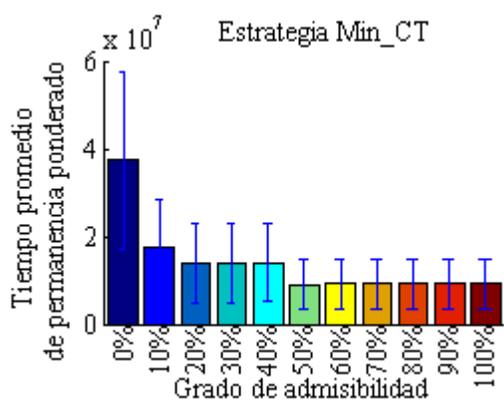


Figura 123. Tiempo promedio de permanencia ponderado, estrategia *Min_CT* con admisibilidad.

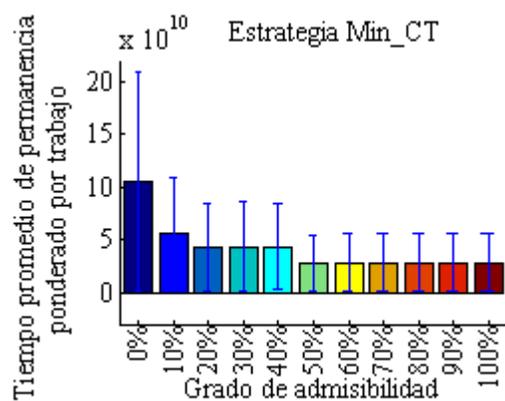


Figura 124. Tiempo promedio de permanencia ponderado por trabajo, estrategia *Min_CT* con admisibilidad.

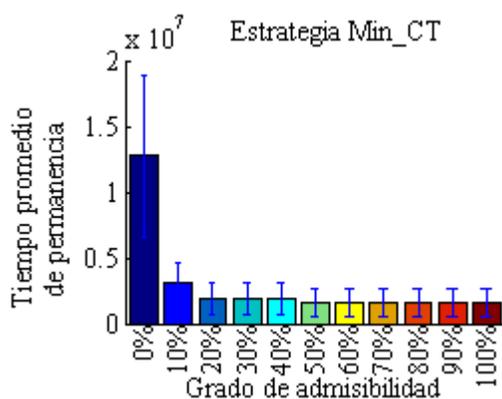


Figura 125. Tiempo promedio de permanencia, estrategia *Min_CT* con admisibilidad.

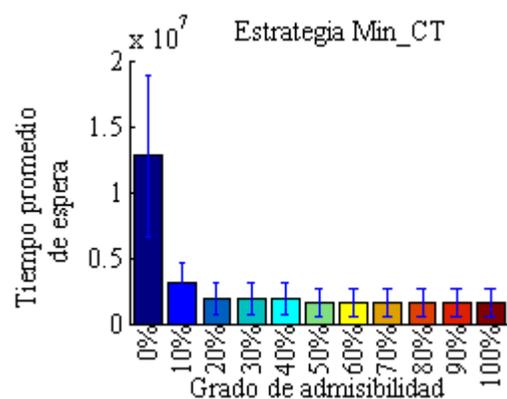


Figura 126. Tiempo promedio de espera, estrategia *Min_CT* con admisibilidad.

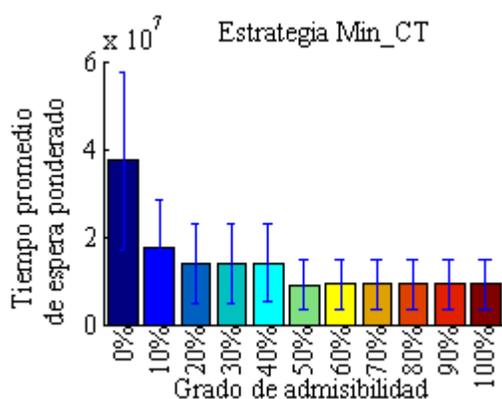


Figura 127. Tiempo promedio de espera ponderado, estrategia *Min_CT* con admisibilidad.

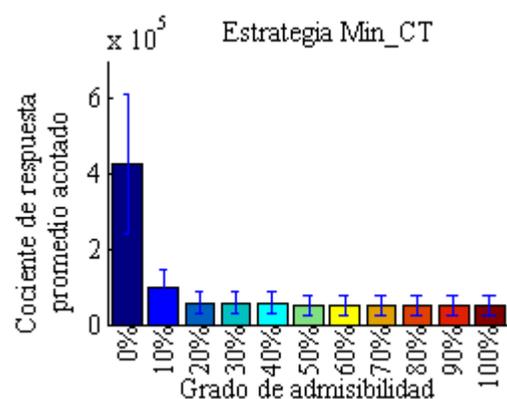


Figura 128. Cociente de respuesta promedio acotado, estrategia *Min_CT* con admisibilidad.

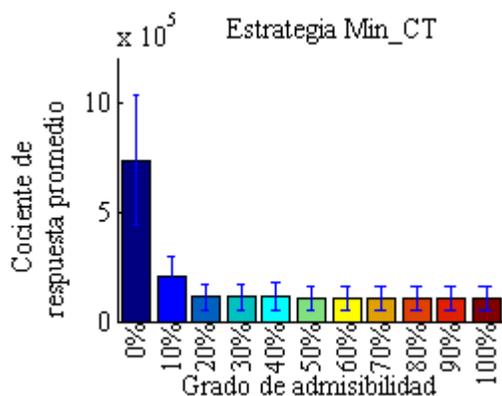


Figura 129. Cociente de respuesta promedio, estrategia *Min_CT* con admisibilidad.

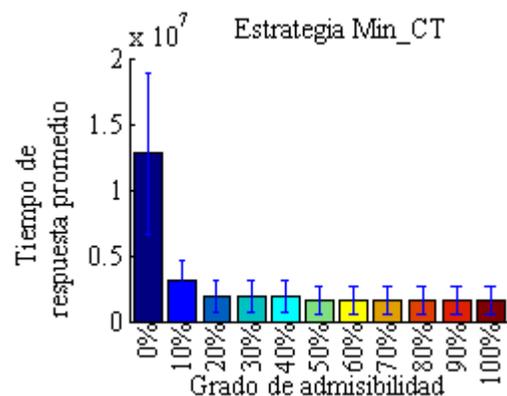


Figura 130. Tiempo de respuesta promedio, estrategia *Min_CT* con admisibilidad.

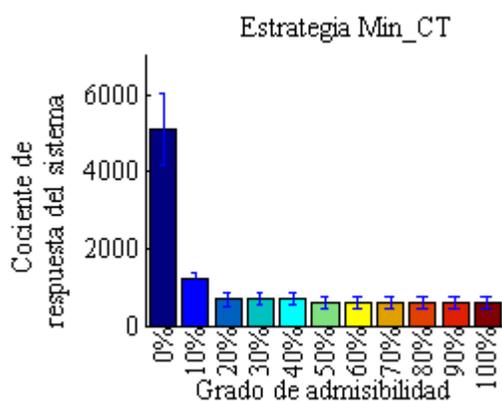


Figura 131. Cociente de respuesta del sistema, estrategia *Min_CT* con admisibilidad.

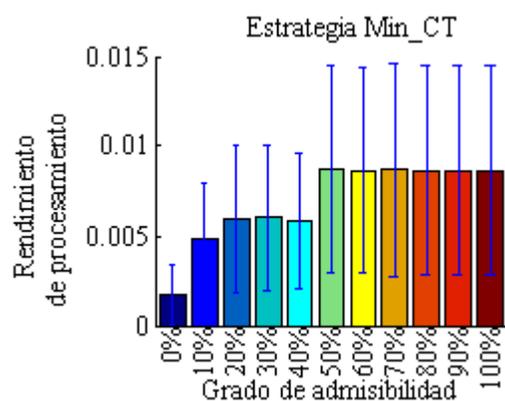


Figura 132. Rendimiento de procesamiento, estrategia *Min_CT* con admisibilidad.

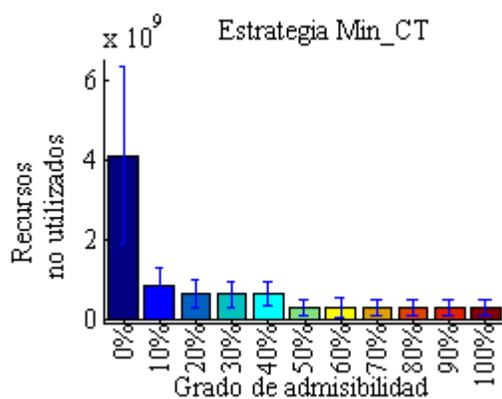


Figura 133. Recursos no utilizados, estrategia *Min_CT* con admisibilidad.

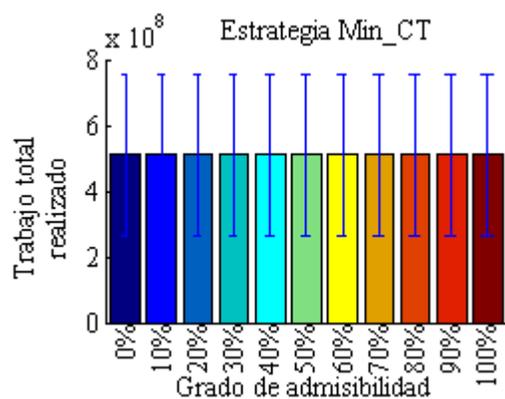


Figura 134. Trabajo total realizado, estrategia *Min_CT* con admisibilidad.

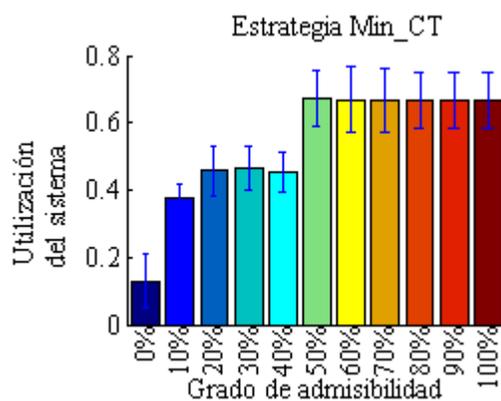


Figura 135. Utilización del sistema, estrategia *Min_CT* con admisibilidad.

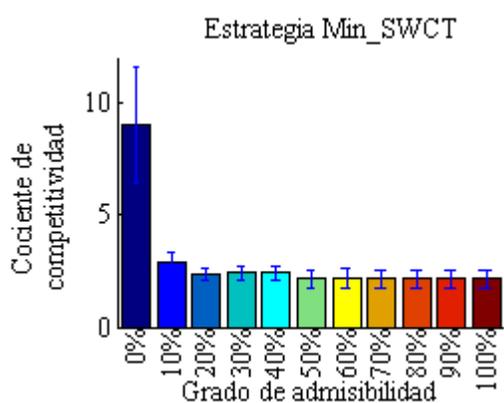
Estrategia *Min_SWCT*

Figura 136. Cociente de competitividad, estrategia *Min_SWCT* con admisibilidad.

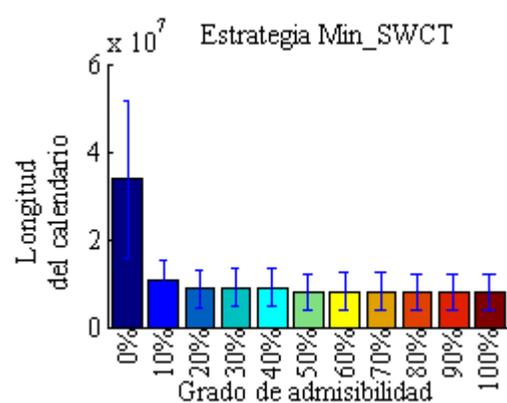


Figura 137. Longitud del calendario, estrategia *Min_SWCT* con admisibilidad.

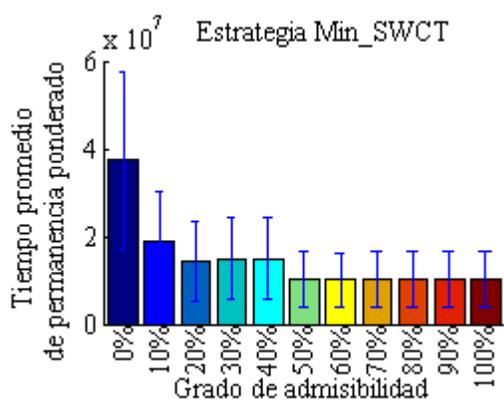


Figura 138. Tiempo promedio de permanencia ponderado, estrategia *Min_SWCT* con admisibilidad.

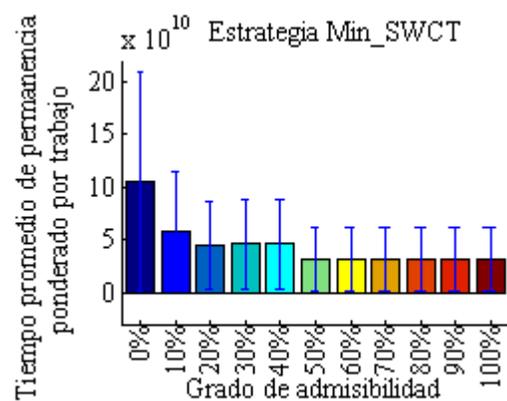


Figura 139. Tiempo promedio de permanencia ponderado por trabajo, estrategia *Min_SWCT* con admisibilidad.

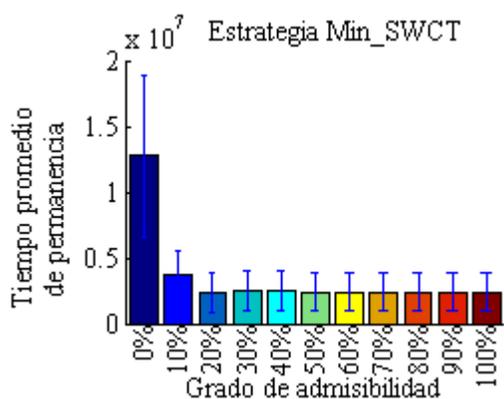


Figura 140. Tiempo promedio de permanencia, estrategia *Min_SWCT* con admisibilidad.

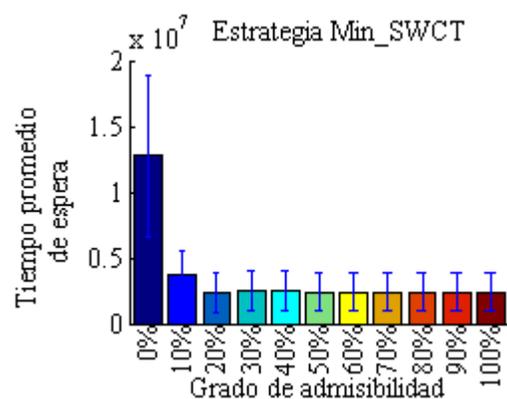


Figura 141. Tiempo promedio de espera, estrategia *Min_SWCT* con admisibilidad.

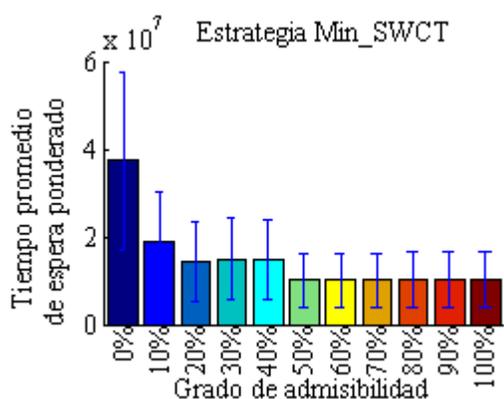


Figura 142. Tiempo promedio de espera ponderado, estrategia *Min_SWCT* con admisibilidad.

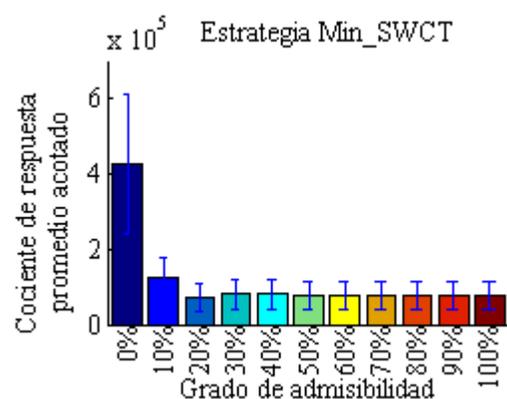


Figura 143. Cociente de respuesta promedio acotado, estrategia *Min_SWCT* con admisibilidad.

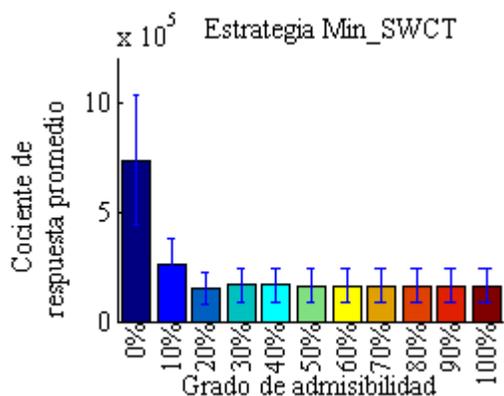


Figura 144. Cociente de respuesta promedio, estrategia *Min_SWCT* con admisibilidad.

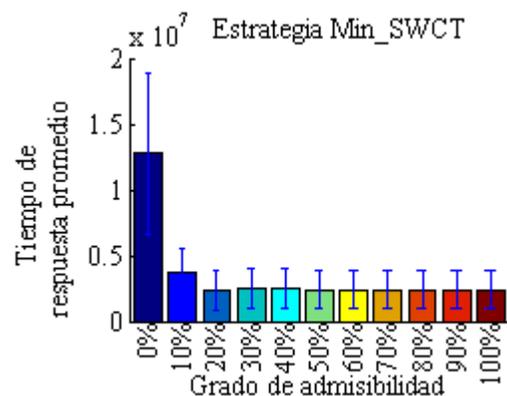


Figura 145. Tiempo de respuesta promedio, estrategia *Min_SWCT* con admisibilidad.

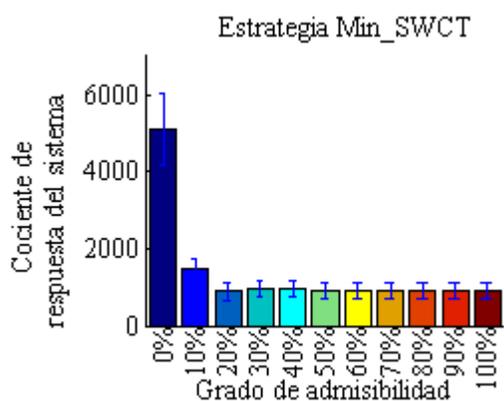


Figura 146. Cociente de respuesta del sistema, estrategia *Min_SWCT* con admisibilidad.

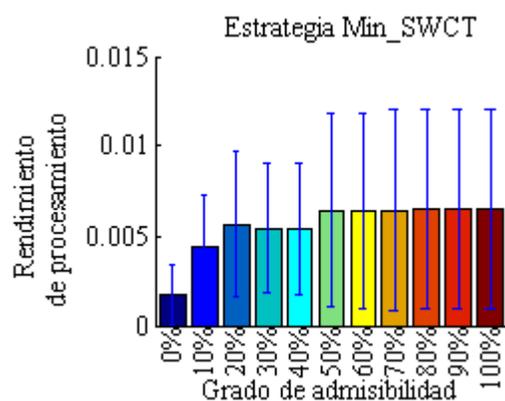


Figura 147. Rendimiento de procesamiento, estrategia *Min_SWCT* con admisibilidad.

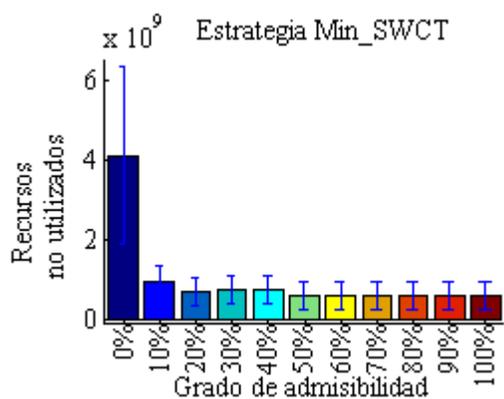


Figura 148. Recursos no utilizados, estrategia *Min_SWCT* con admisibilidad.

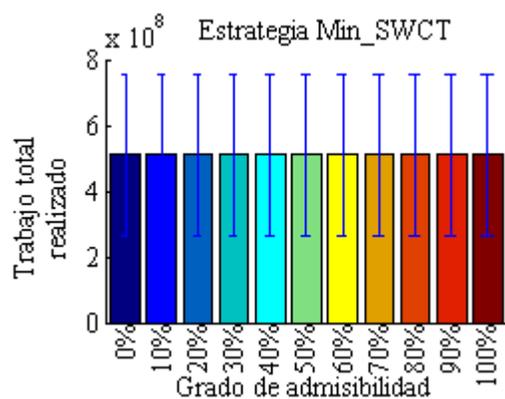


Figura 149. Trabajo total realizado, estrategia *Min_SWCT* con admisibilidad.

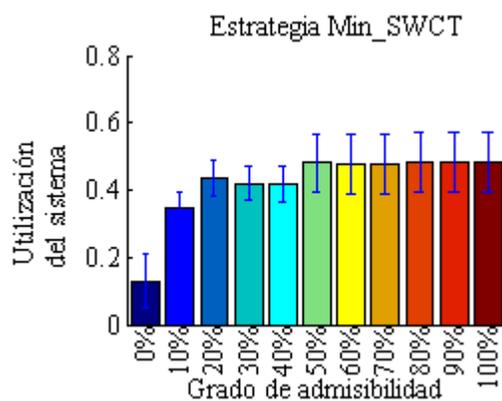


Figura 150. Utilización del sistema, estrategia *Min_SWCT* con admisibilidad.

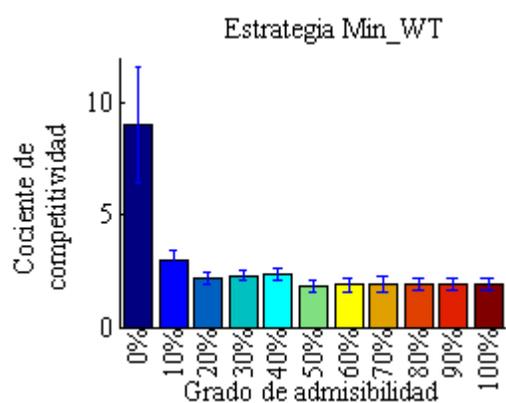
Estrategia *Min_WT*

Figura 151. Cociente de competitividad, estrategia *Min_WT* con admisibilidad.

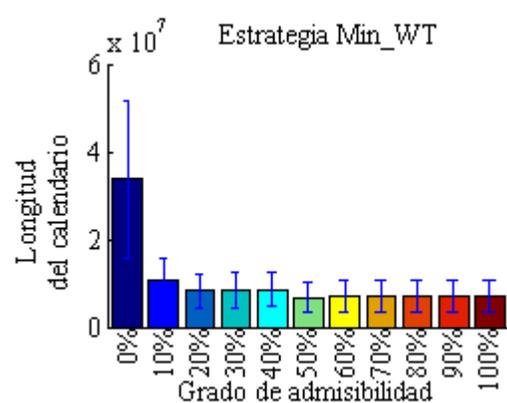


Figura 152. Longitud del calendario, estrategia *Min_WT* con admisibilidad.

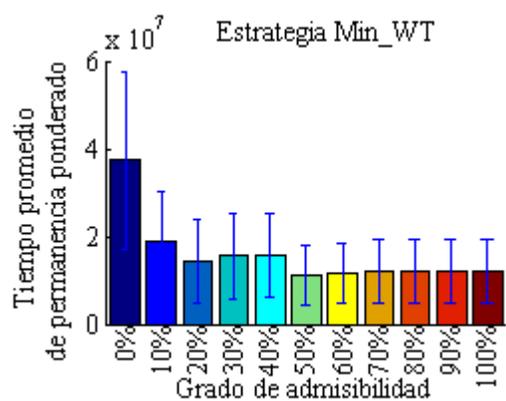


Figura 153. Tiempo promedio de permanencia ponderado, estrategia *Min_WT* con admisibilidad.

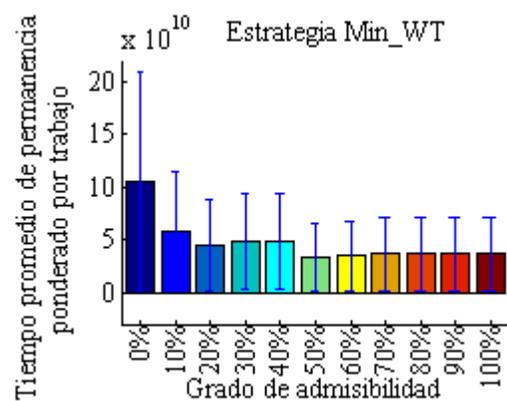


Figura 154. Tiempo promedio de permanencia ponderado por trabajo, estrategia *Min_WT* con admisibilidad.

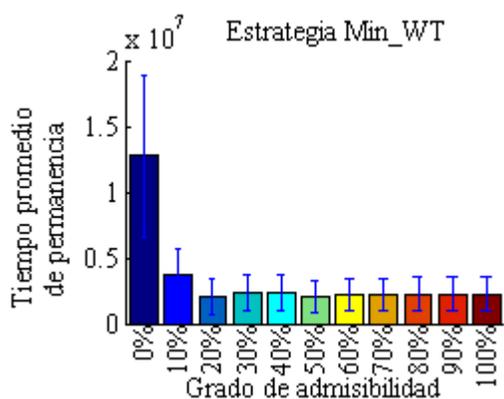


Figura 155. Tiempo promedio de permanencia, estrategia *Min_WT* con admisibilidad.

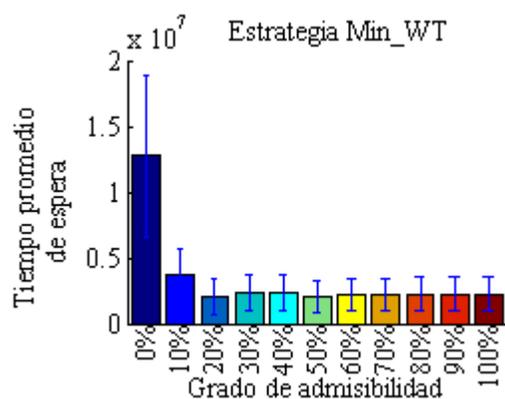


Figura 156. Tiempo promedio de espera, estrategia *Min_WT* con admisibilidad.

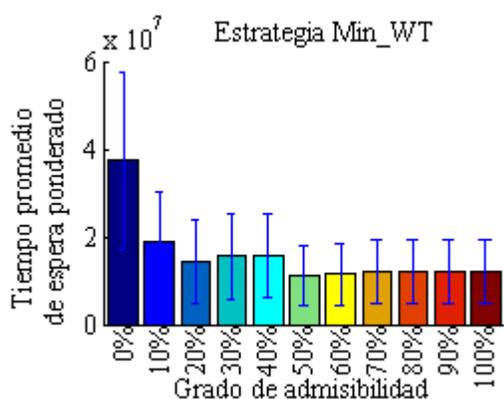


Figura 157. Tiempo promedio de espera ponderado, estrategia *Min_WT* con admisibilidad.

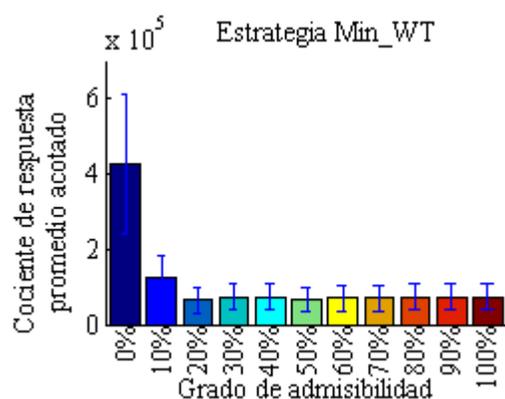


Figura 158. Cociente de respuesta promedio acotado, estrategia *Min_WT* con admisibilidad.

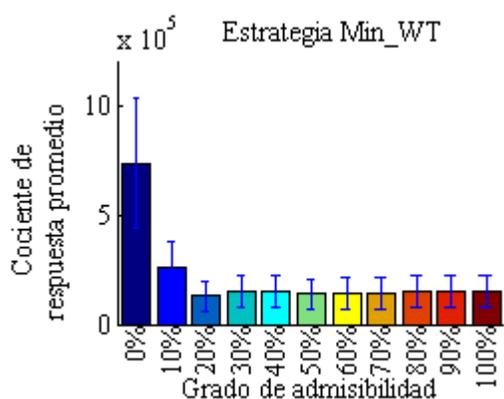


Figura 159. Cociente de respuesta promedio, estrategia *Min_WT* con admisibilidad.

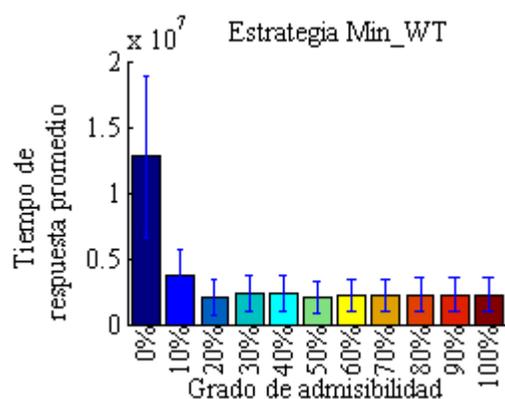


Figura 160. Tiempo de respuesta promedio, estrategia *Min_WT* con admisibilidad.

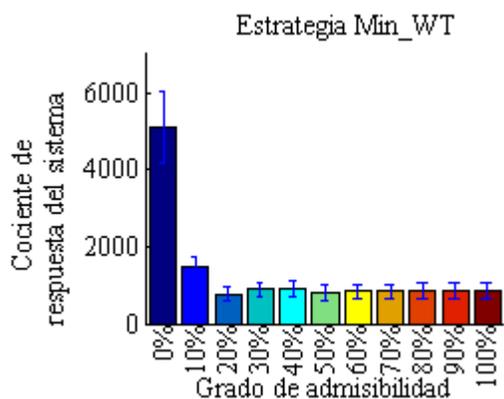


Figura 161. Cociente de respuesta del sistema, estrategia *Min_WT* con admisibilidad.

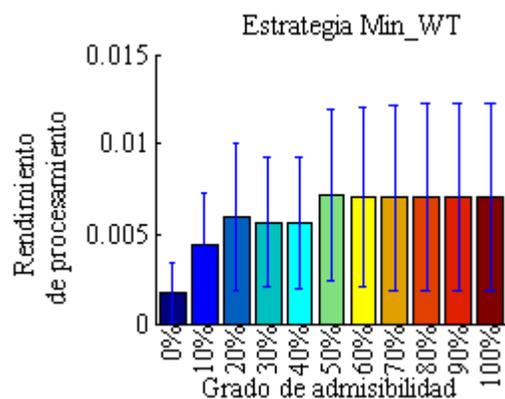


Figura 162. Rendimiento de procesamiento, estrategia *Min_WT* con admisibilidad.

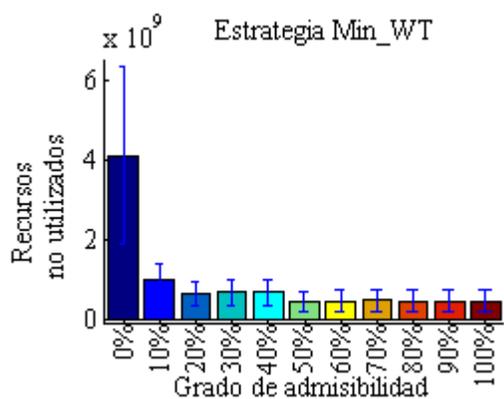


Figura 163. Recursos no utilizados, estrategia *Min_WT* con admisibilidad.

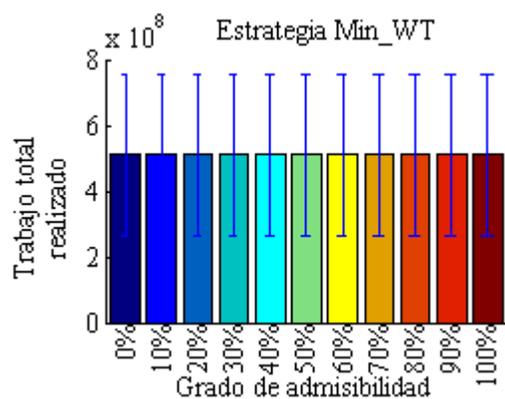


Figura 164. Trabajo total realizado, estrategia *Min_WT* con admisibilidad.

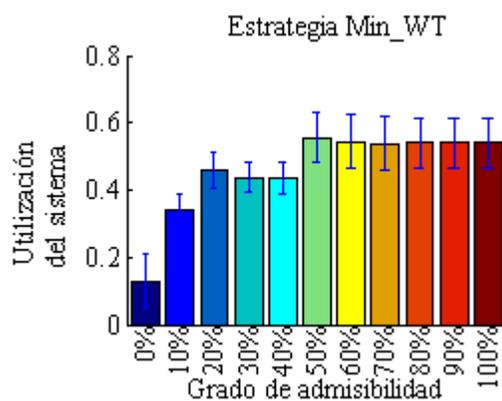


Figura 165. Utilización del sistema, estrategia *Min_WT* con admisibilidad.

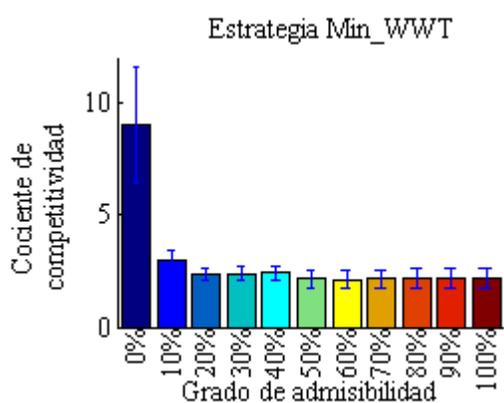
Estrategia *Min_WWT*

Figura 166. Cociente de competitividad, estrategia *Min_WWT* con admisibilidad.

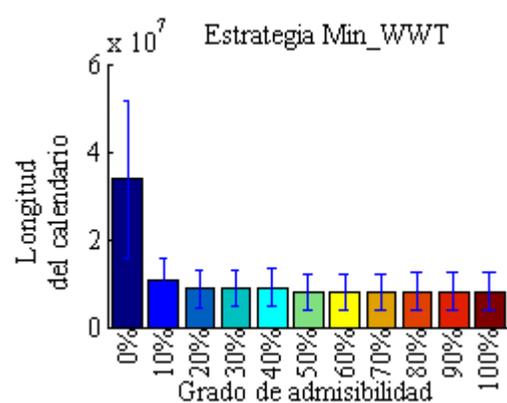


Figura 167. Longitud del calendario, estrategia *Min_WWT* con admisibilidad.

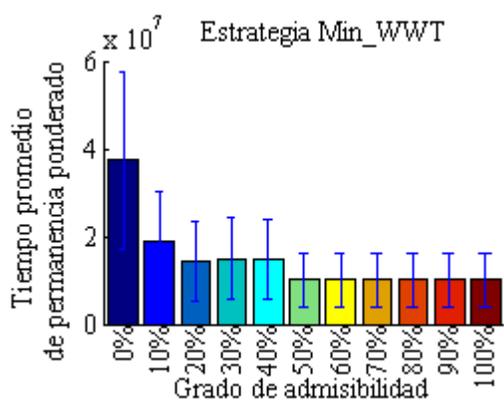


Figura 168. Tiempo promedio de permanencia ponderado, estrategia *Min_WWT* con admisibilidad.

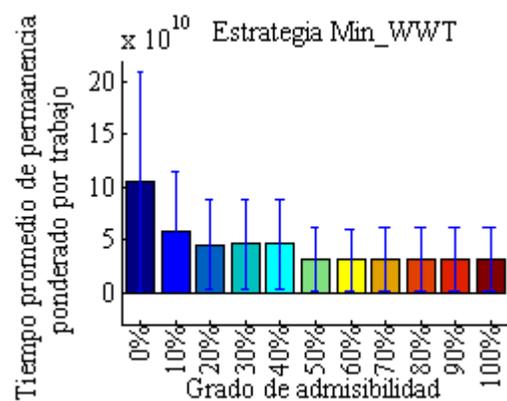


Figura 169. Tiempo promedio de permanencia ponderado por trabajo, estrategia *Min_WWT* con admisibilidad.

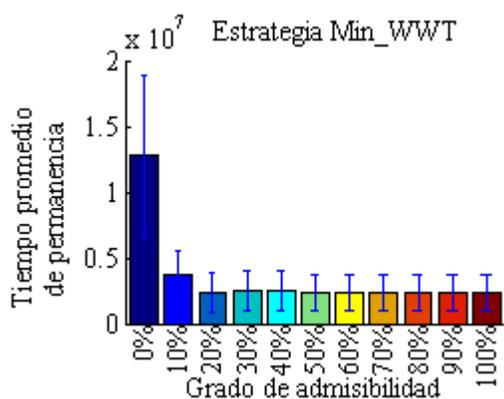


Figura 170. Tiempo promedio de permanencia, estrategia *Min_WWT* con admisibilidad.

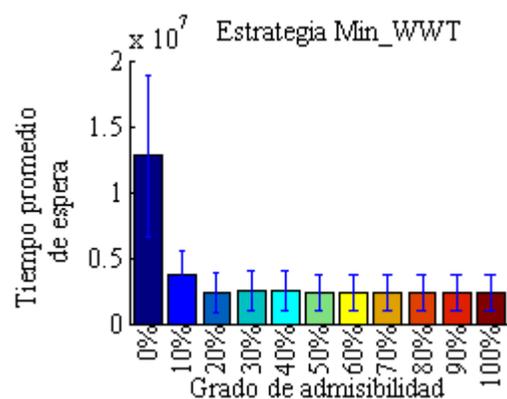


Figura 171. Tiempo promedio de espera, estrategia *Min_WWT* con admisibilidad.

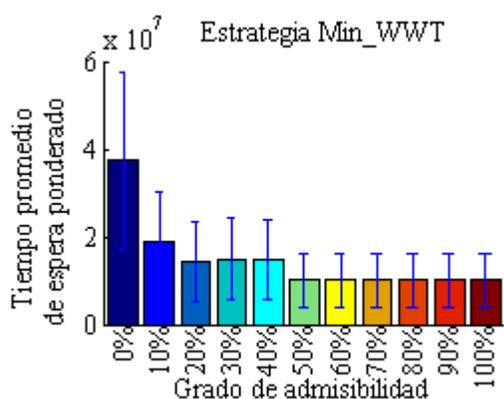


Figura 172. Tiempo promedio de espera ponderado, estrategia *Min_WWT* con admisibilidad.

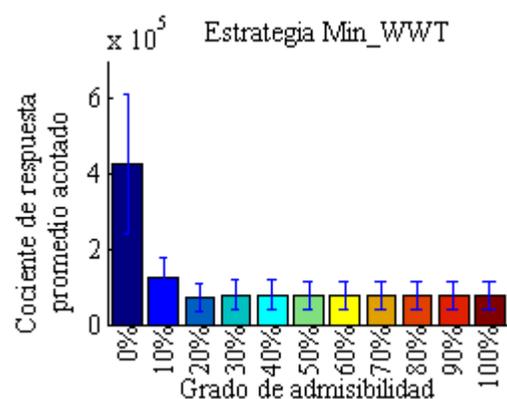


Figura 173. Cociente de respuesta promedio acotado, estrategia *Min_WWT* con admisibilidad.

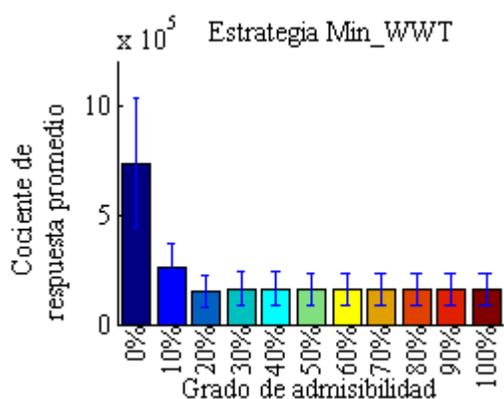


Figura 174. Cociente de respuesta promedio, estrategia *Min_WWT* con admisibilidad.

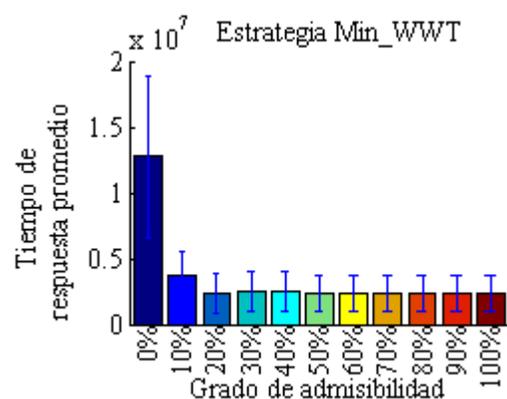


Figura 175. Tiempo de respuesta promedio, estrategia *Min_WWT* con admisibilidad.

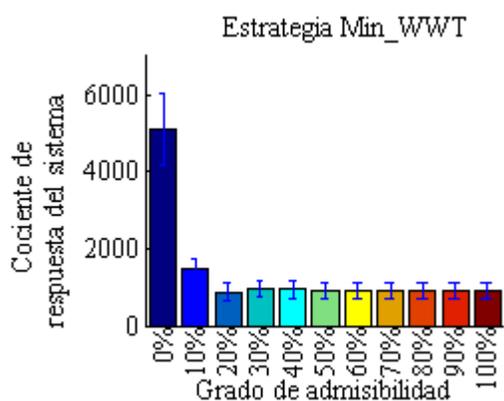


Figura 176. Cociente de respuesta del sistema, estrategia *Min_WWT* con admisibilidad.

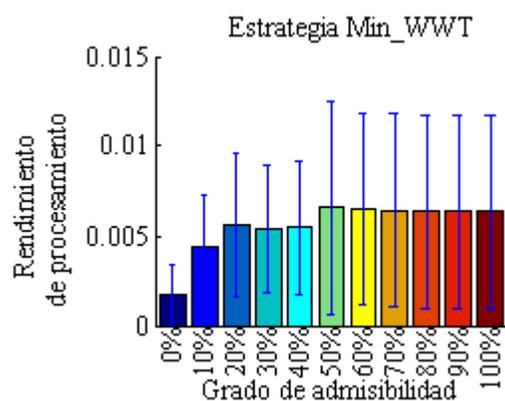


Figura 177. Rendimiento de procesamiento, estrategia *Min_WWT* con admisibilidad.

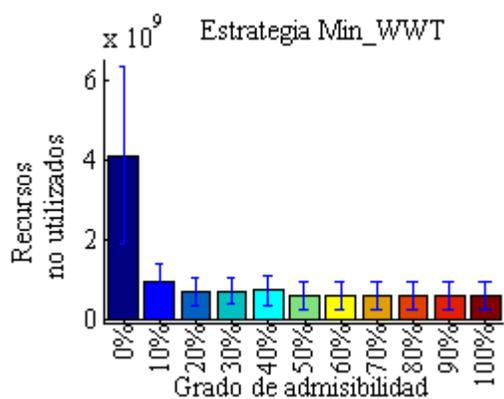


Figura 178. Recursos no utilizados, estrategia *Min_WWT* con admisibilidad.

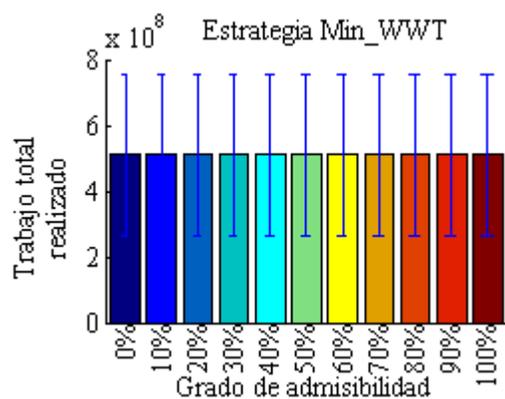


Figura 179. Trabajo total realizado, estrategia *Min_WWT* con admisibilidad.

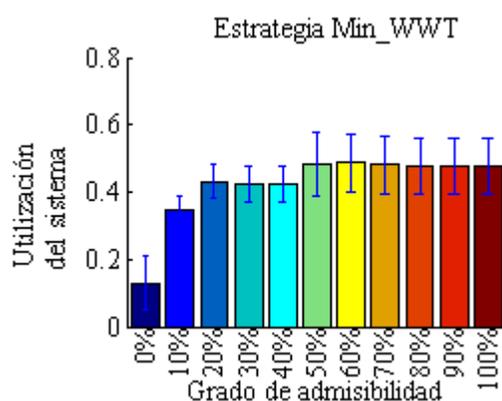


Figura 180. Utilización del sistema, estrategia *Min_WWT* con admisibilidad.

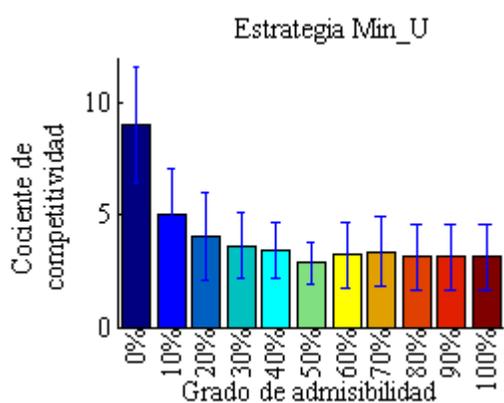
Estrategia *Min_U*

Figura 181. Cociente de competitividad, estrategia *Min_U* con admisibilidad.

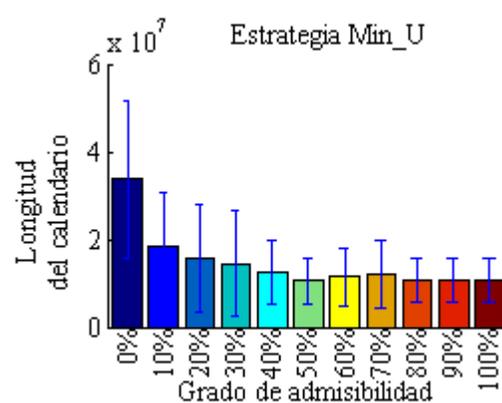


Figura 182. Longitud del calendario, estrategia *Min_U* con admisibilidad.

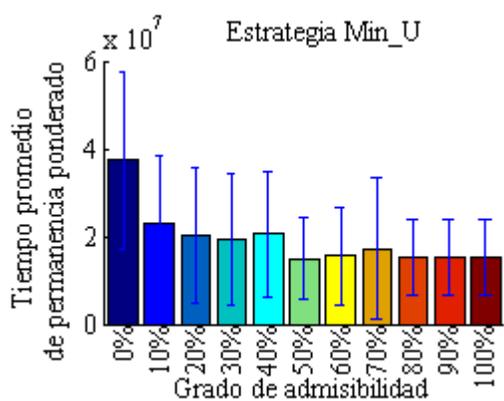


Figura 183. Tiempo promedio de permanencia ponderado, estrategia *Min_U* con admisibilidad.

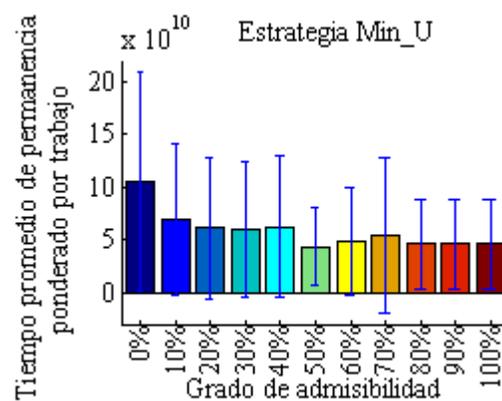


Figura 184. Tiempo promedio de permanencia ponderado por trabajo, estrategia *Min_U* con admisibilidad.

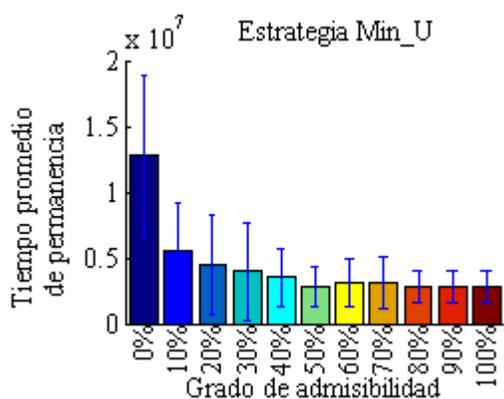


Figura 185. Tiempo promedio de permanencia, estrategia *Min_U* con admisibilidad.

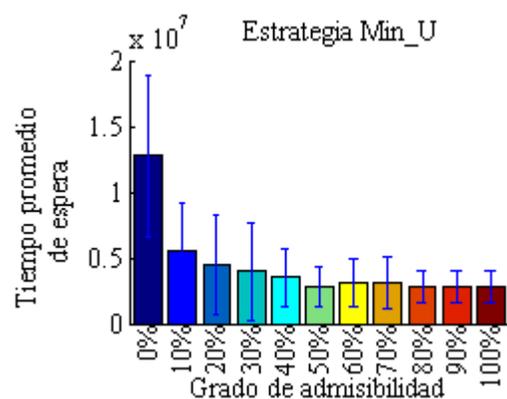


Figura 186. Tiempo promedio de espera, estrategia *Min_U* con admisibilidad.

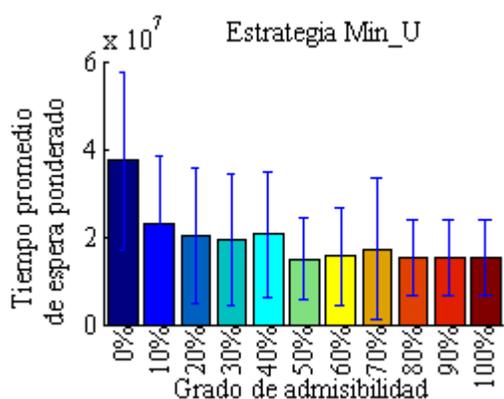


Figura 187. Tiempo promedio de espera ponderado, estrategia *Min_U* con admisibilidad.

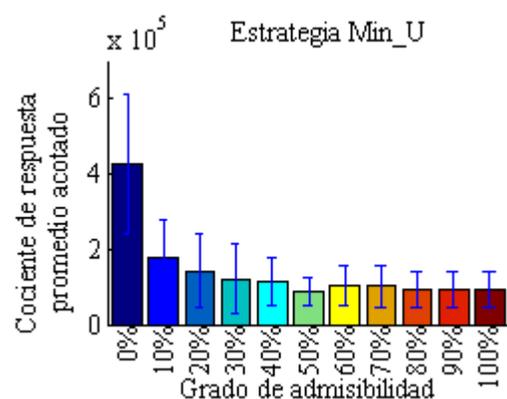


Figura 188. Cociente de respuesta promedio acotado, estrategia *Min_U* con admisibilidad.

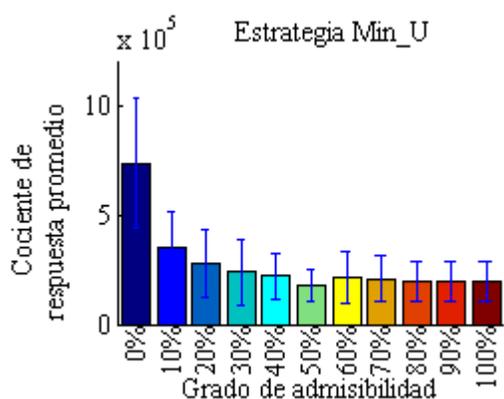


Figura 189. Cociente de respuesta promedio, estrategia *Min_U* con admisibilidad.

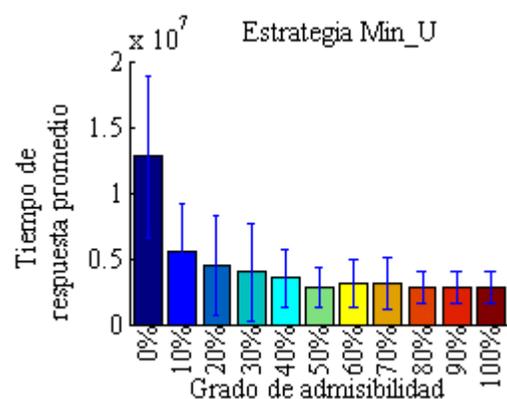


Figura 190. Tiempo de respuesta promedio, estrategia *Min_U* con admisibilidad.

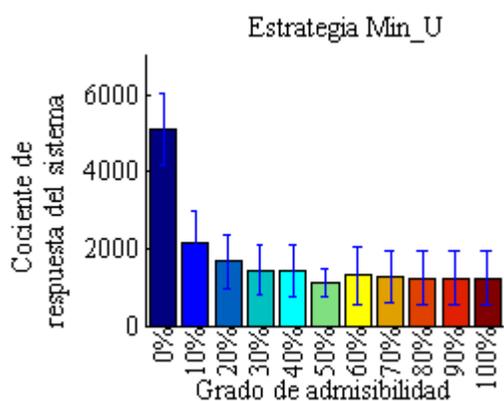


Figura 191. Cociente de respuesta del sistema, estrategia *Min_U* con admisibilidad.

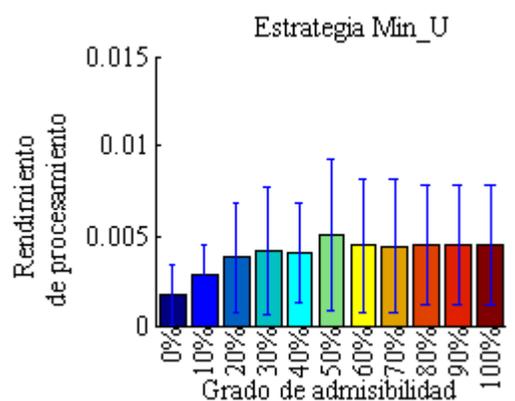


Figura 192. Rendimiento de procesamiento, estrategia *Min_U* con admisibilidad.

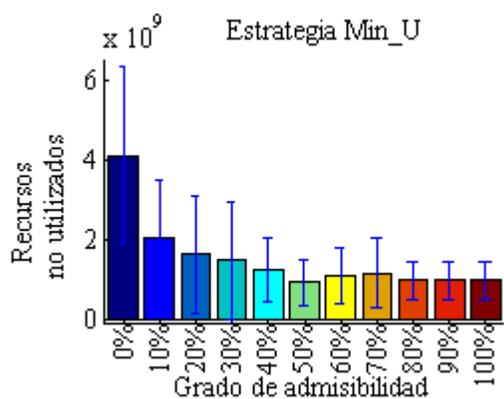


Figura 193. Recursos no utilizados, estrategia *Min_U* con admisibilidad.

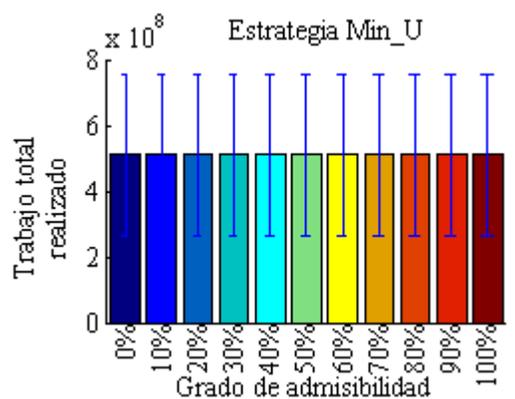


Figura 194. Trabajo total realizado, estrategia *Min_U* con admisibilidad.

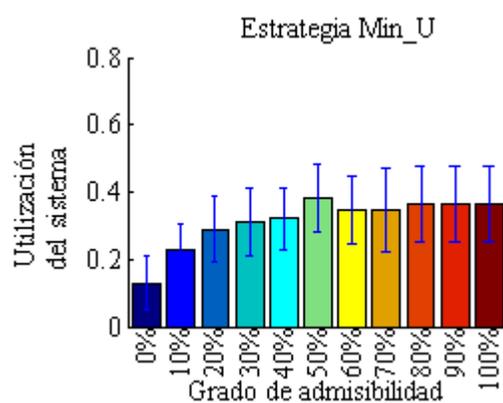


Figura 195. Utilización del sistema, estrategia *Min_U* con admisibilidad.

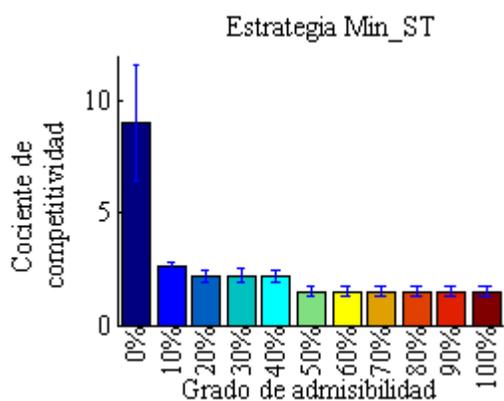
Estrategia *Min_ST*

Figura 196. Cociente de competitividad, estrategia *Min_ST* con admisibilidad.

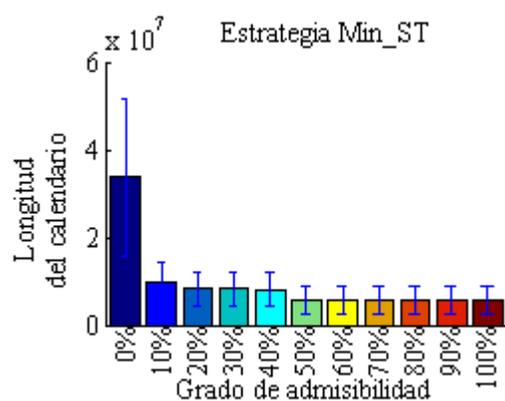


Figura 197. Longitud del calendario, estrategia *Min_ST* con admisibilidad.

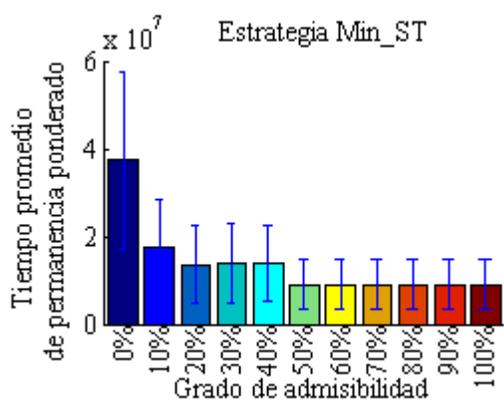


Figura 198. Tiempo promedio de permanencia ponderado, estrategia *Min_ST* con admisibilidad.

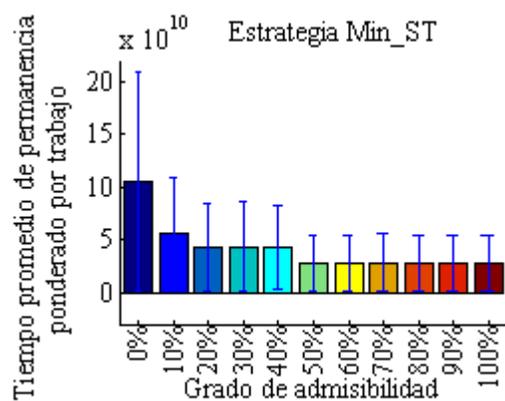


Figura 199. Tiempo promedio de permanencia ponderado por trabajo, estrategia *Min_ST* con admisibilidad.

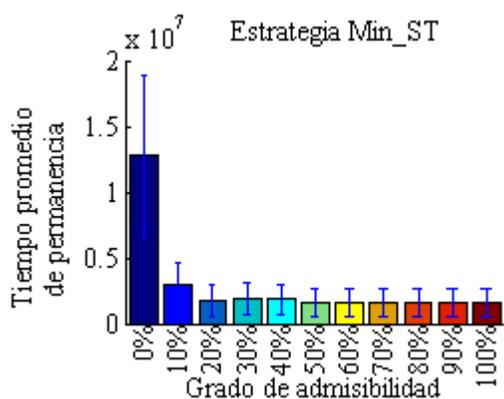


Figura 200. Tiempo promedio de permanencia, estrategia *Min_ST* con admisibilidad.

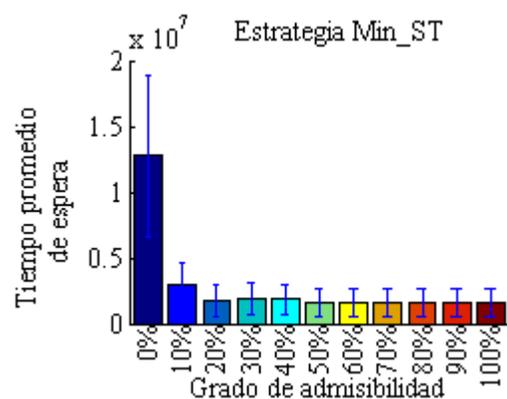


Figura 201. Tiempo promedio de espera, estrategia *Min_ST* con admisibilidad.

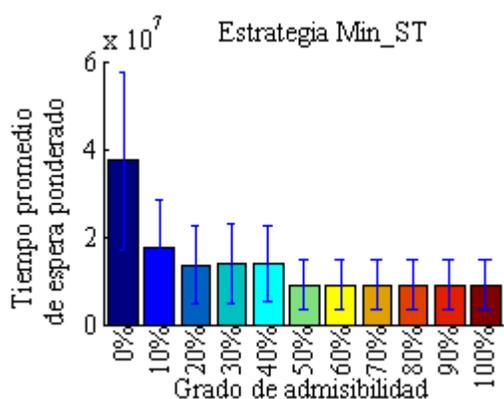


Figura 202. Tiempo promedio de espera ponderado, estrategia *Min_ST* con admisibilidad.

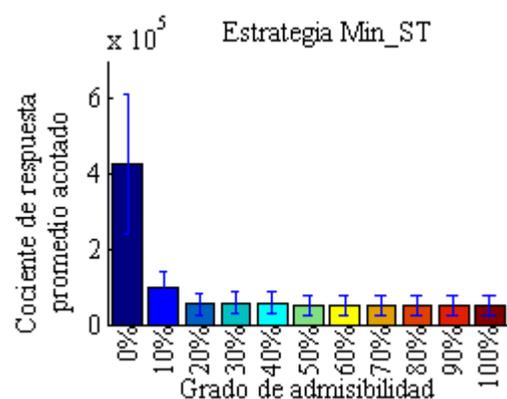


Figura 203. Cociente de respuesta promedio acotado, estrategia *Min_ST* con admisibilidad.

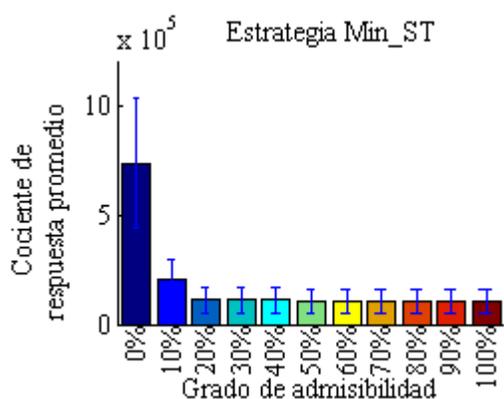


Figura 204. Cociente de respuesta promedio, estrategia *Min_ST* con admisibilidad.

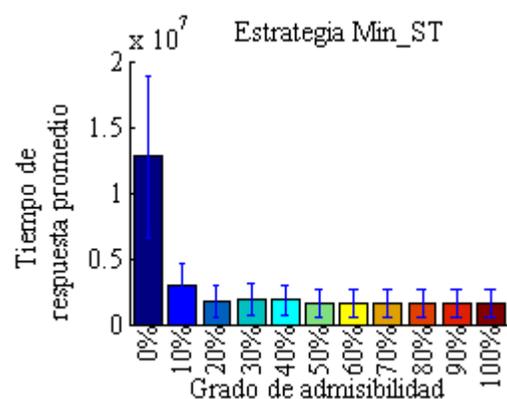


Figura 205. Tiempo de respuesta promedio, estrategia *Min_ST* con admisibilidad.

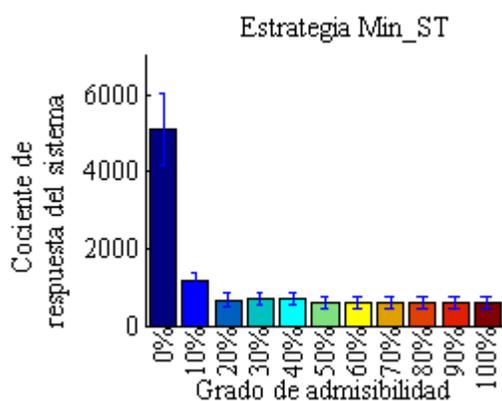


Figura 206. Cociente de respuesta del sistema, estrategia *Min_ST* con admisibilidad.

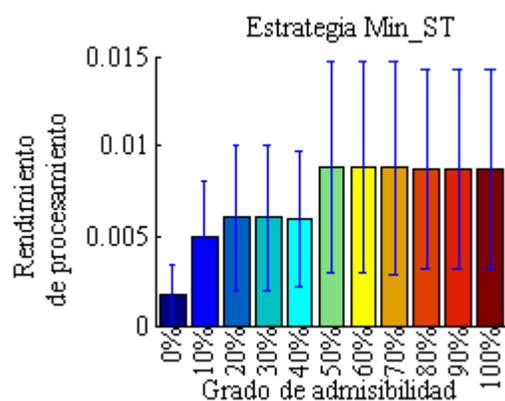


Figura 207. Rendimiento de procesamiento, estrategia *Min_ST* con admisibilidad.

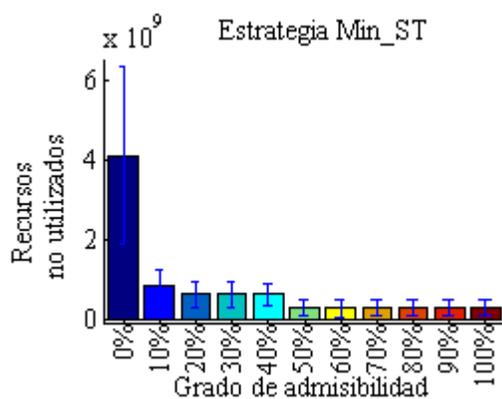


Figura 208. Recursos no utilizados, estrategia *Min_ST* con admisibilidad.

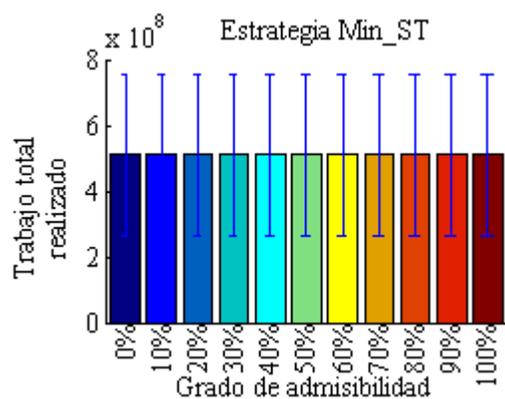


Figura 209. Trabajo total realizado, estrategia *Min_ST* con admisibilidad.

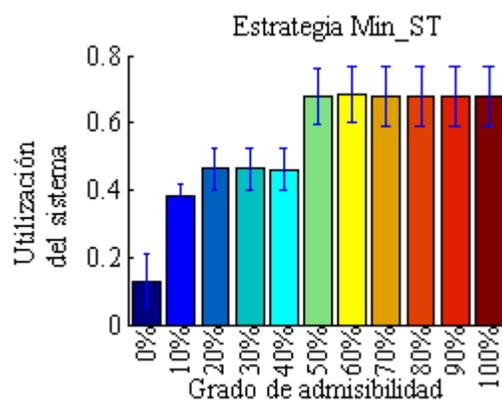


Figura 210. Utilización del sistema, estrategia *Min_ST* con admisibilidad.

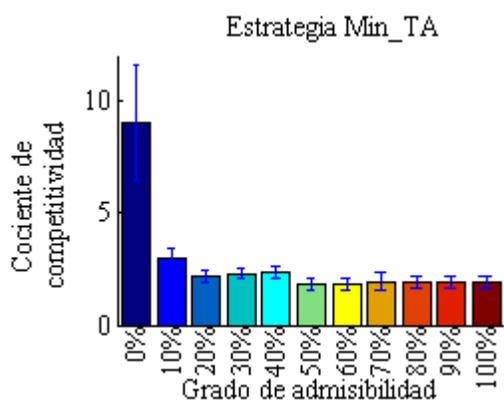
Estrategia *Min_TA*

Figura 211. Cociente de competitividad, estrategia *Min_TA* con admisibilidad.

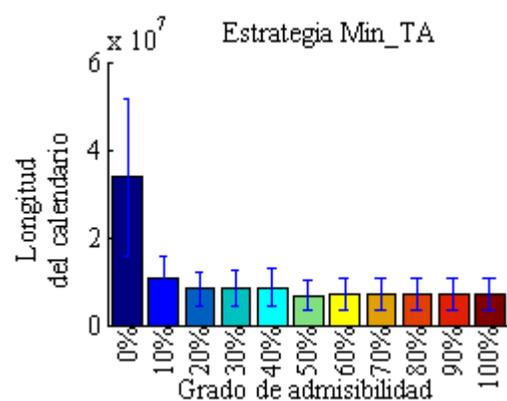


Figura 212. Longitud del calendario, estrategia *Min_TA* con admisibilidad.

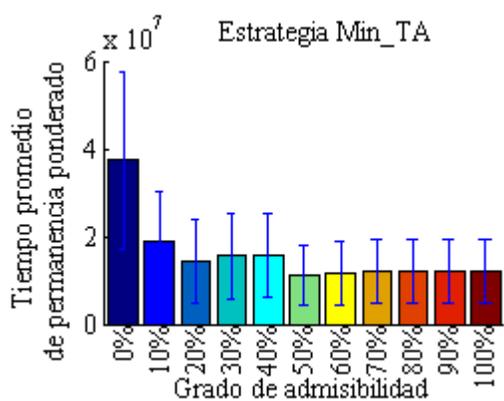


Figura 213. Tiempo promedio de permanencia ponderado, estrategia *Min_TA* con admisibilidad.

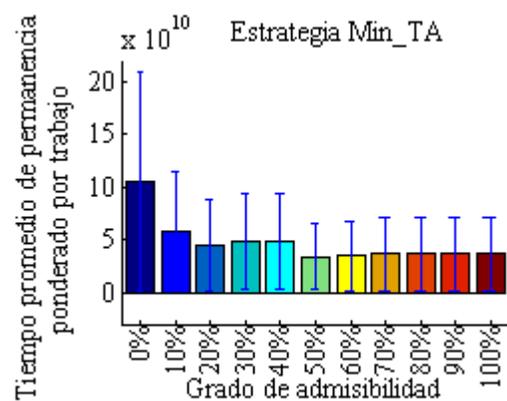


Figura 214. Tiempo promedio de permanencia ponderado por trabajo, estrategia *Min_TA* con admisibilidad.

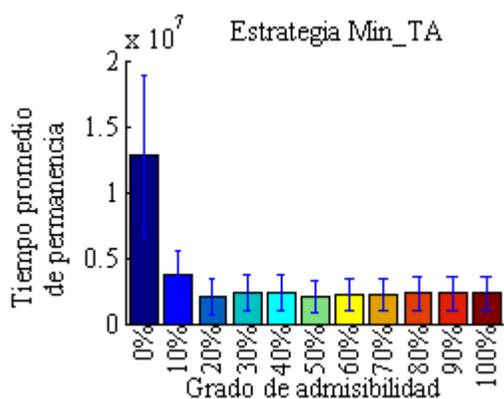


Figura 215. Tiempo promedio de permanencia, estrategia *Min_TA* con admisibilidad.

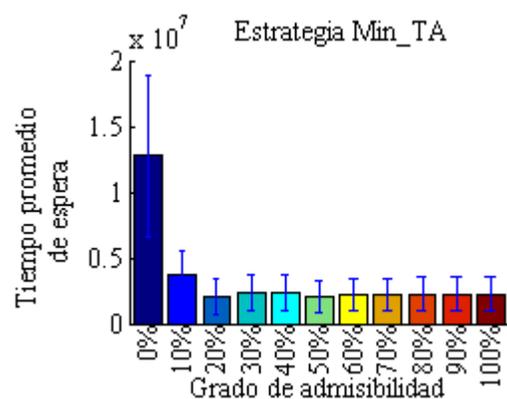


Figura 216. Tiempo promedio de espera, estrategia *Min_TA* con admisibilidad.

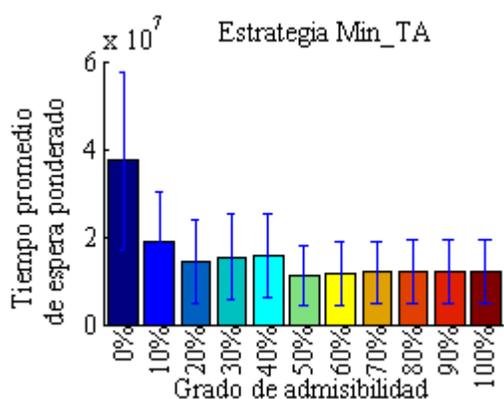


Figura 217. Tiempo promedio de espera ponderado, estrategia *Min_TA* con admisibilidad.

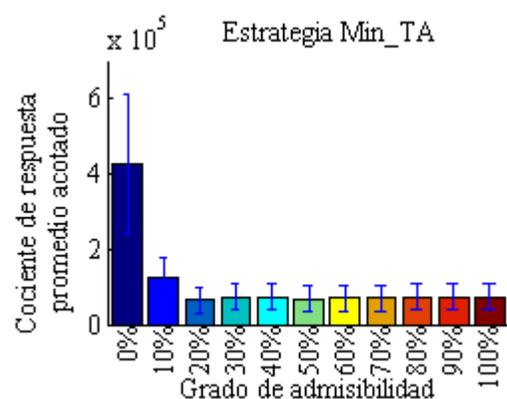


Figura 218. Cociente de respuesta promedio acotado, estrategia *Min_TA* con admisibilidad.

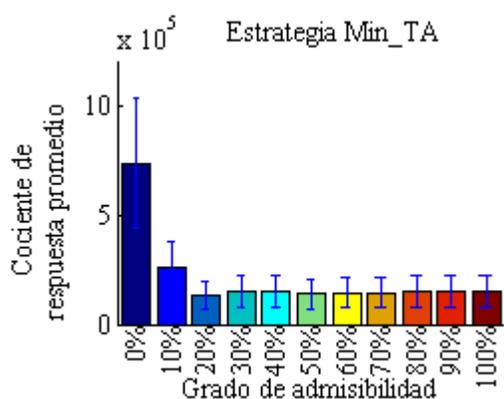


Figura 219. Cociente de respuesta promedio, estrategia *Min_TA* con admisibilidad.

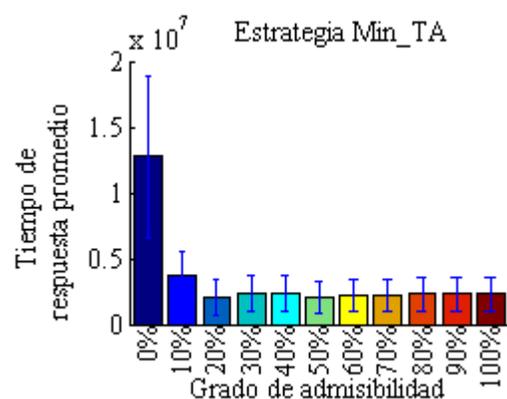


Figura 220. Tiempo de respuesta promedio, estrategia *Min_TA* con admisibilidad.

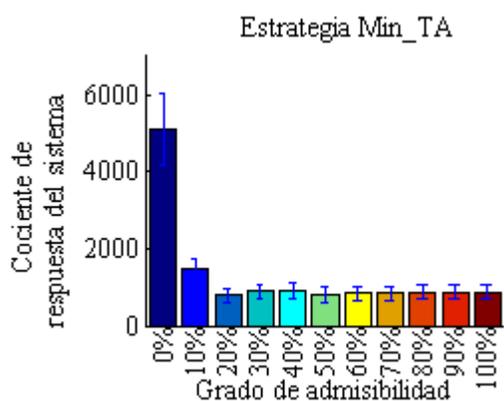


Figura 221. Cociente de respuesta del sistema, estrategia *Min_TA* con admisibilidad.

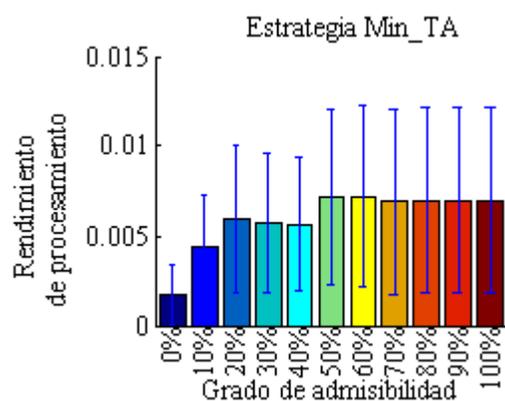


Figura 222. Rendimiento de procesamiento, estrategia *Min_TA* con admisibilidad.

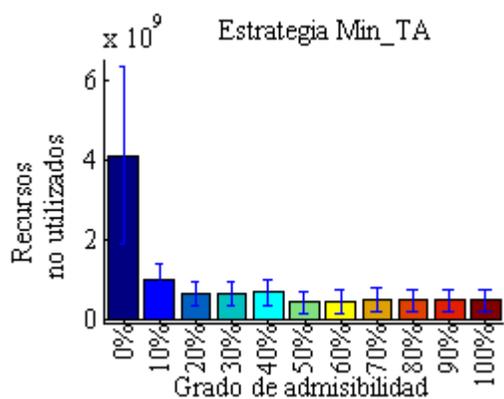


Figura 223. Recursos no utilizados, estrategia *Min_TA* con admisibilidad.

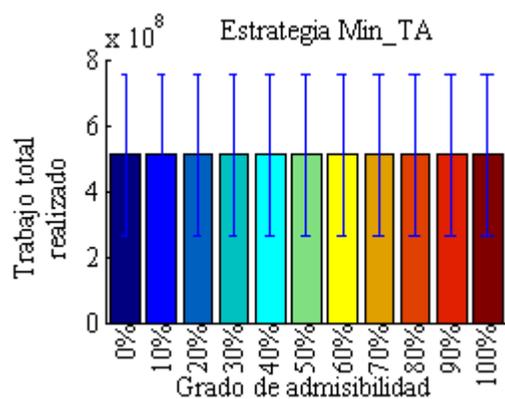


Figura 224. Trabajo total realizado, estrategia *Min_TA* con admisibilidad.

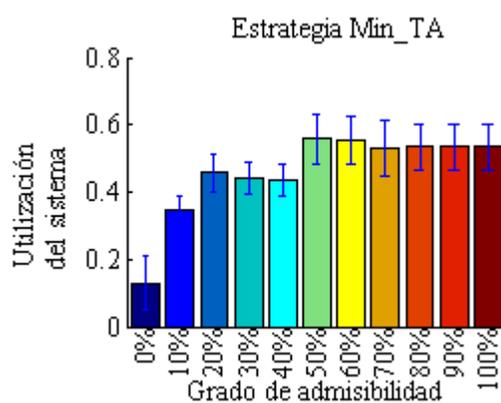


Figura 225. Utilización del sistema, estrategia *Min_TA* con admisibilidad.

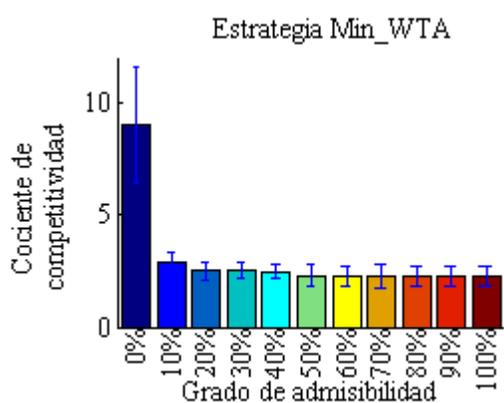
Estrategia *Min_WTA*

Figura 226. Cociente de competitividad, estrategia *Min_WTA* con admisibilidad.

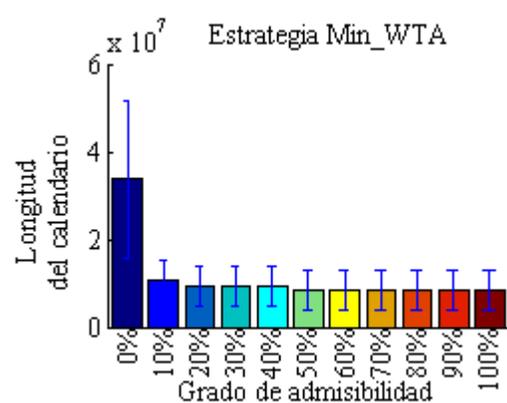


Figura 227. Longitud del calendario, estrategia *Min_WTA* con admisibilidad.

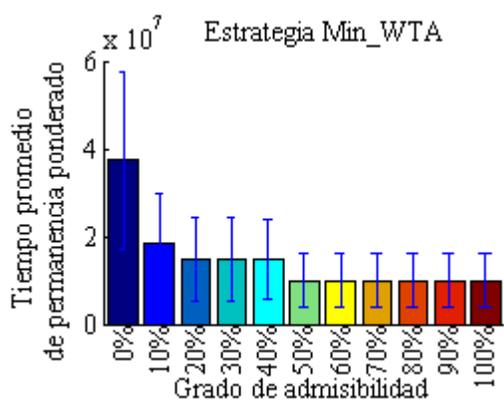


Figura 228. Tiempo promedio de permanencia ponderado, estrategia *Min_WTA* con admisibilidad.

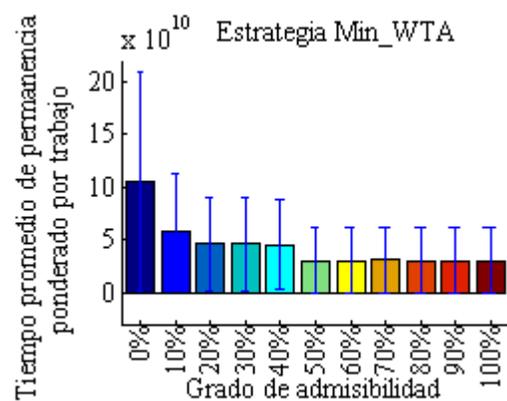


Figura 229. Tiempo promedio de permanencia ponderado por trabajo, estrategia *Min_WTA* con admisibilidad.

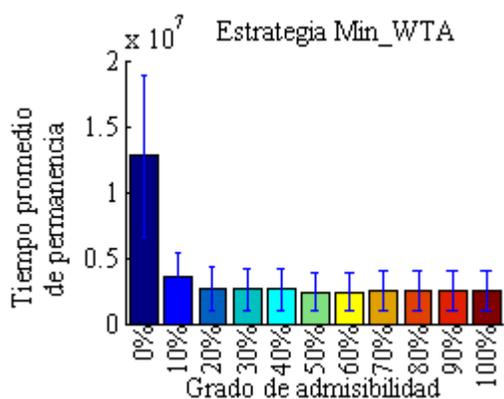


Figura 230. Tiempo promedio de permanencia, estrategia *Min_WTA* con admisibilidad.

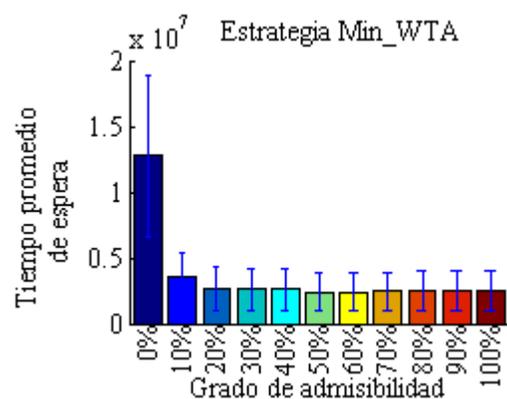


Figura 231. Tiempo promedio de espera, estrategia *Min_WTA* con admisibilidad.

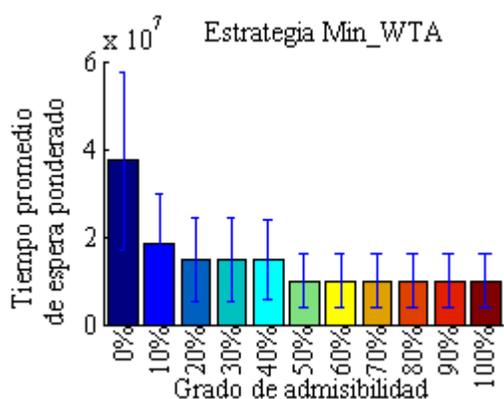


Figura 232. Tiempo promedio de espera ponderado, estrategia *Min_WTA* con admisibilidad.

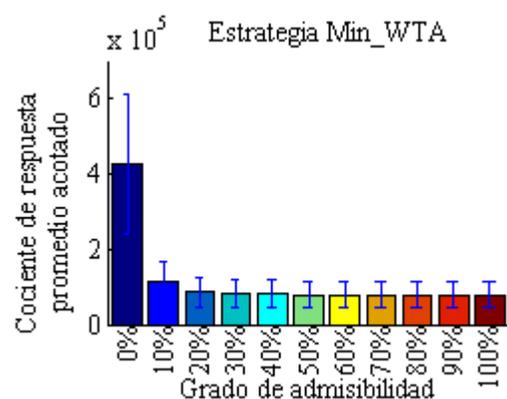


Figura 233. Cociente de respuesta promedio acotado, estrategia *Min_WTA* con admisibilidad.

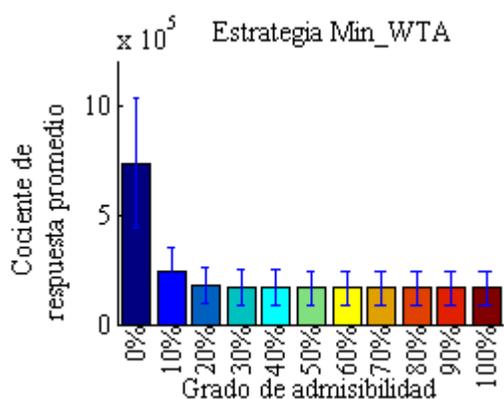


Figura 234. Cociente de respuesta promedio, estrategia *Min_WTA* con admisibilidad.

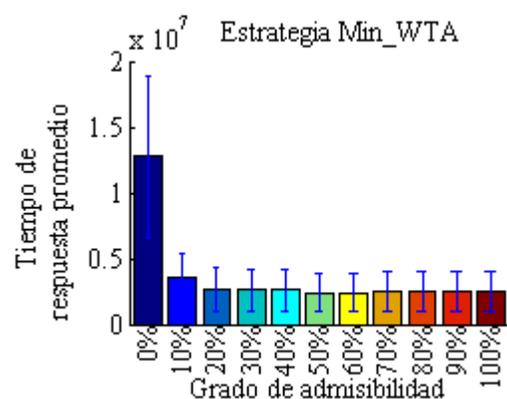


Figura 235. Tiempo de respuesta promedio, estrategia *Min_WTA* con admisibilidad.

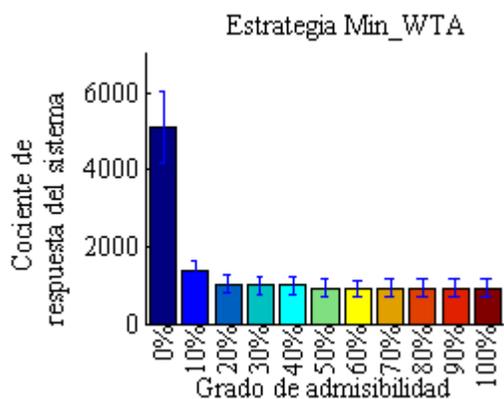


Figura 236. Cociente de respuesta del sistema, estrategia *Min_WTA* con admisibilidad.

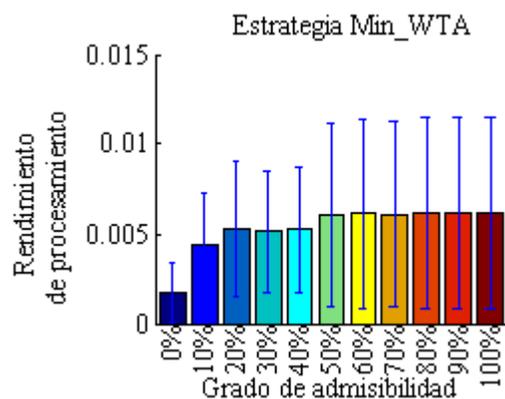


Figura 237. Rendimiento de procesamiento, estrategia *Min_WTA* con admisibilidad.

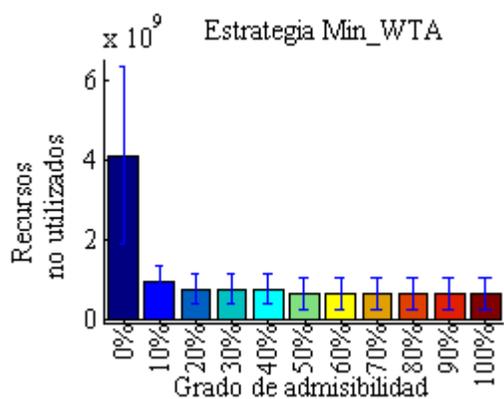


Figura 238. Recursos no utilizados, estrategia *Min_WTA* con admisibilidad.

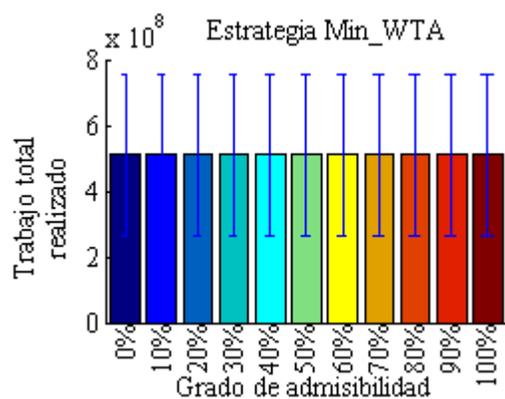


Figura 239. Trabajo total realizado, estrategia *Min_WTA* con admisibilidad.

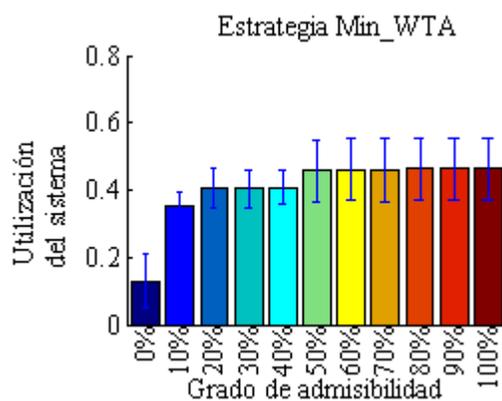


Figura 240. Utilización del sistema, estrategia *Min_WTA* con admisibilidad.

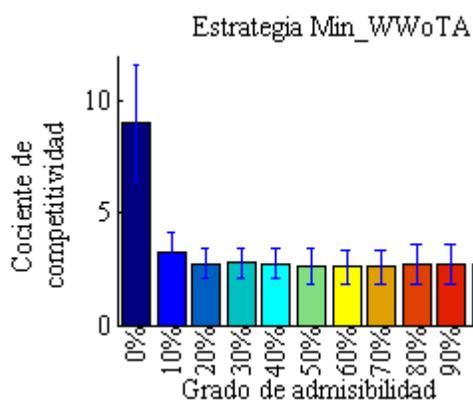
Estrategia *Min_WWoTA*

Figura 241. Cociente de competitividad, estrategia *Min_WWoTA* con admisibilidad.

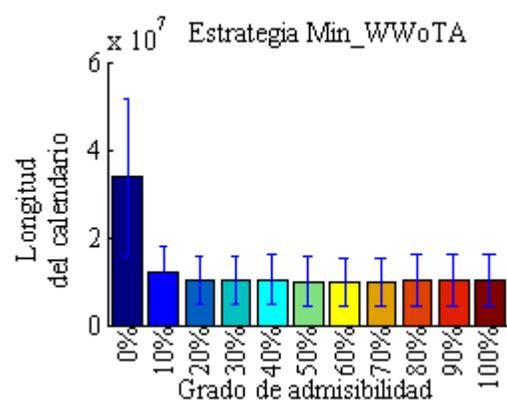


Figura 242. Longitud del calendario, estrategia *Min_WWoTA* con admisibilidad.

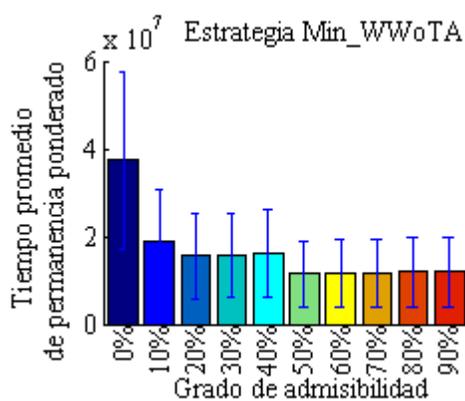


Figura 243. Tiempo promedio de permanencia ponderado, estrategia *Min_WWoTA* con admisibilidad.

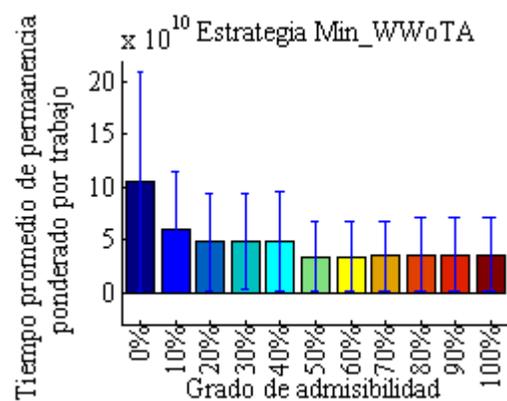


Figura 244. Tiempo promedio de permanencia ponderado por trabajo, estrategia *Min_WWoTA* con admisibilidad.

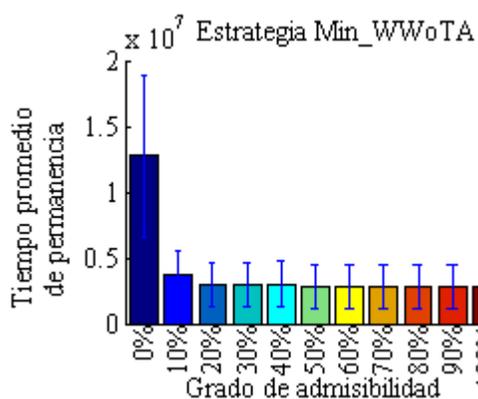


Figura 245. Tiempo promedio de permanencia, estrategia *Min_WWoTA* con admisibilidad.

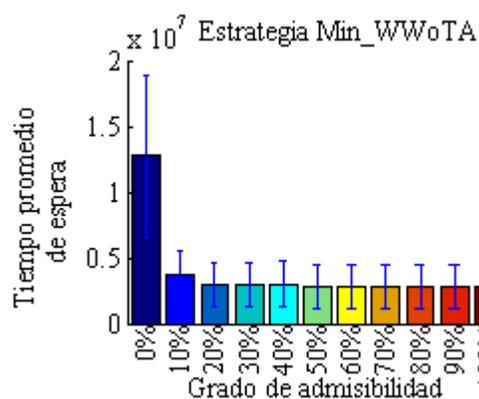


Figura 246. Tiempo promedio de espera, estrategia *Min_WWoTA* con admisibilidad.

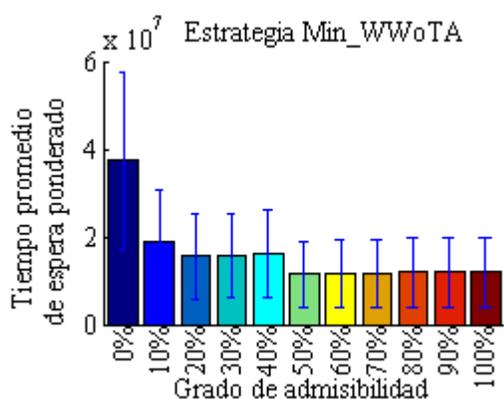


Figura 247. Tiempo promedio de espera ponderado, estrategia *Min_WWoTA* con admisibilidad.

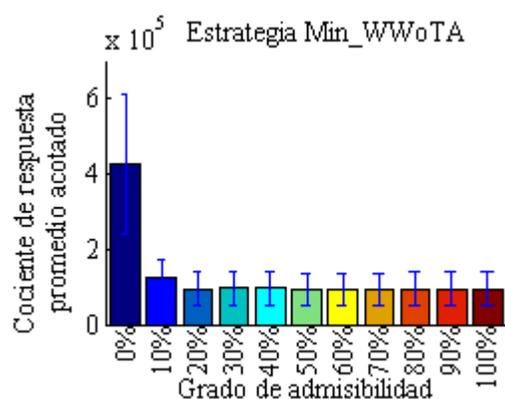


Figura 248. Cociente de respuesta promedio acotado, estrategia *Min_WWoTA* con admisibilidad.

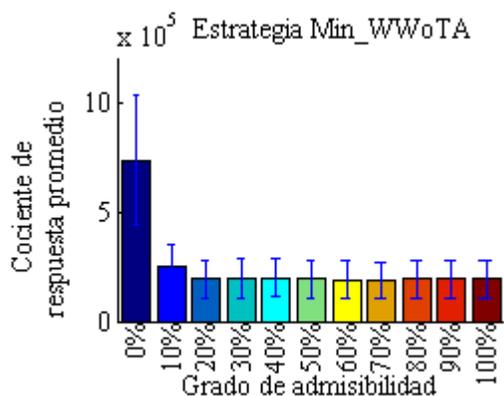


Figura 249. Cociente de respuesta promedio, estrategia *Min_WWoTA* con admisibilidad.

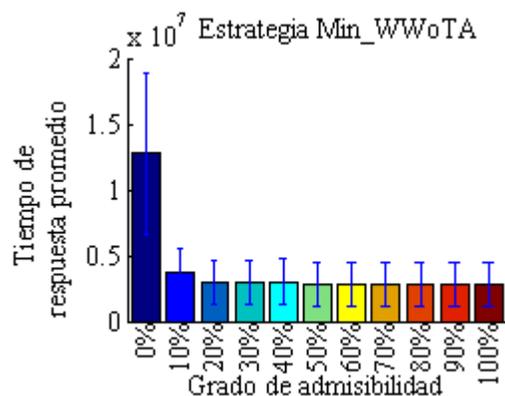


Figura 250. Tiempo de respuesta promedio, estrategia *Min_WWoTA* con admisibilidad.

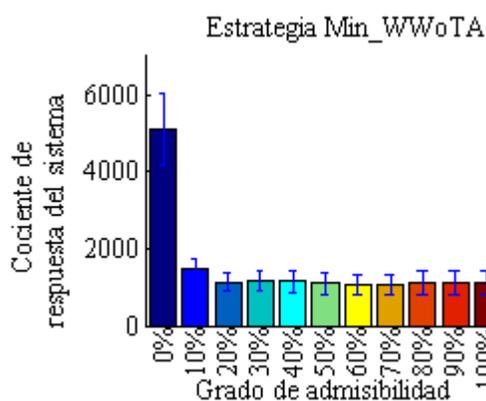


Figura 251. Cociente de respuesta del sistema, estrategia *Min_WWoTA* con admisibilidad.

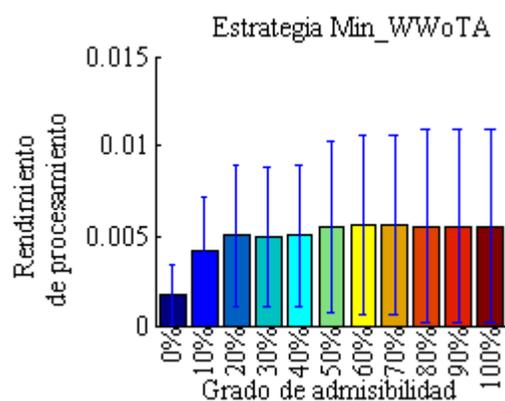


Figura 252. Rendimiento de procesamiento, estrategia *Min_WWoTA* con admisibilidad.

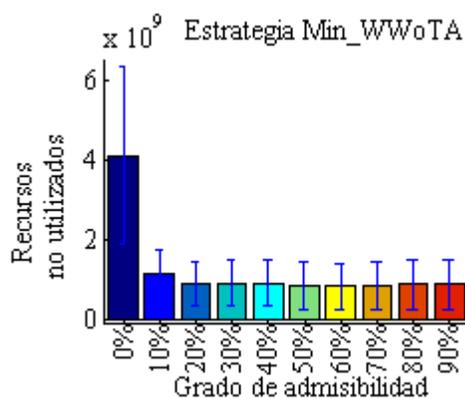


Figura 253. Recursos no utilizados, estrategia *Min_WWoTA* con admisibilidad.

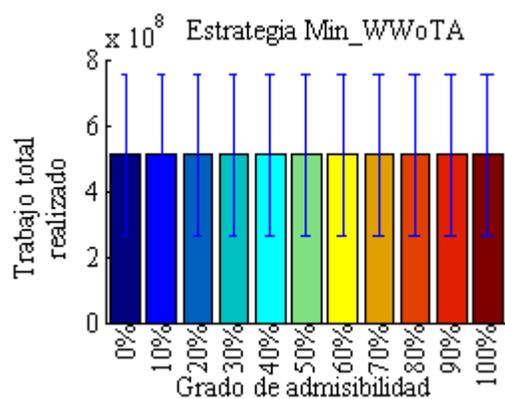


Figura 254. Trabajo total realizado, estrategia *Min_WWoTA* con admisibilidad.

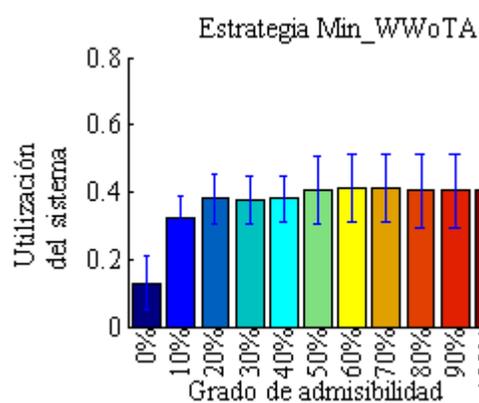


Figura 255. Utilización del sistema, estrategia *Min_WWoTA* con admisibilidad.

Porcentaje de mejora de los algoritmos

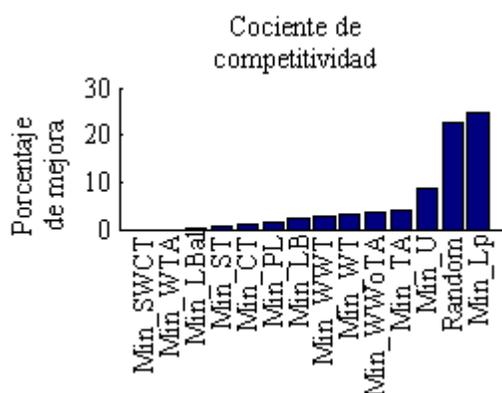


Figura 256. Porcentaje de mejora de los algoritmos, en el cociente de competitividad, al utilizar el esquema de admisibilidad.

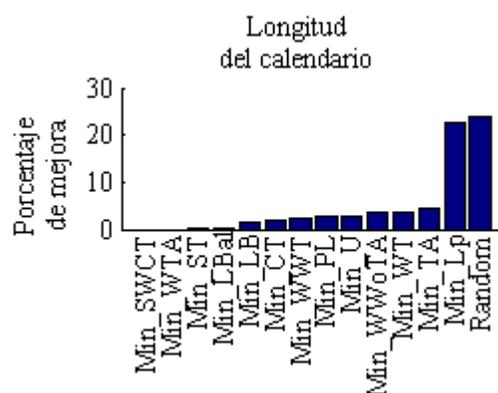


Figura 257. Porcentaje de mejora de los algoritmos, en la longitud del calendario, al utilizar el esquema de admisibilidad.

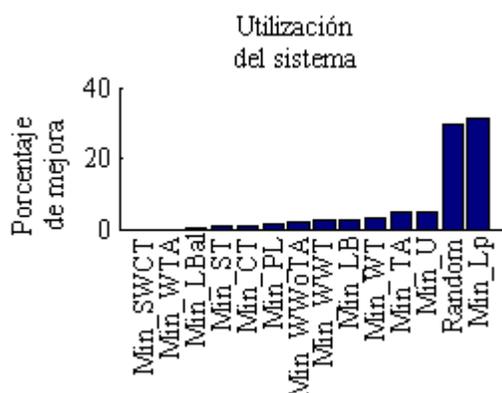


Figura 258. Porcentaje de mejora de los algoritmos, en la utilización del sistema, al utilizar el esquema de admisibilidad.

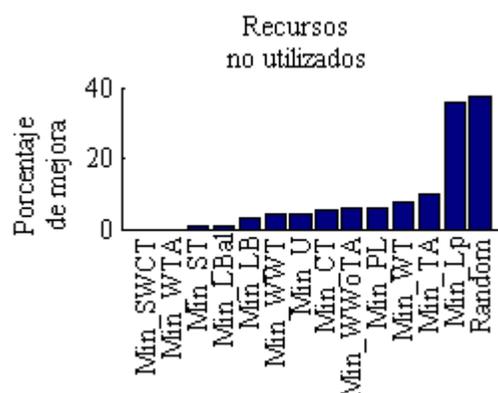


Figura 259. Porcentaje de mejora de los algoritmos, en los recursos no utilizados, al utilizar el esquema de admisibilidad.

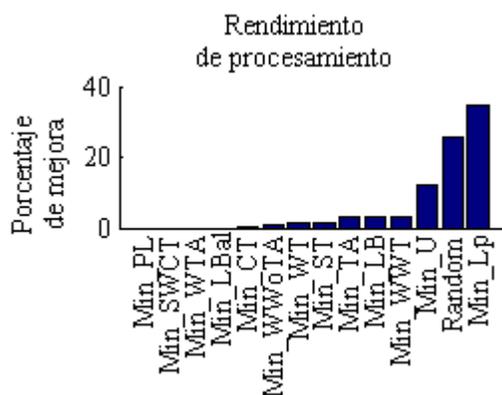


Figura 260. Porcentaje de mejora de los algoritmos, en el rendimiento de procesamiento, al utilizar el esquema de admisibilidad.

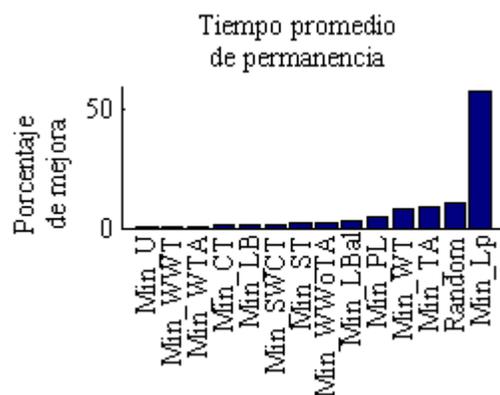


Figura 261. Porcentaje de mejora de los algoritmos, en el tiempo promedio de permanencia, al utilizar el esquema de admisibilidad.

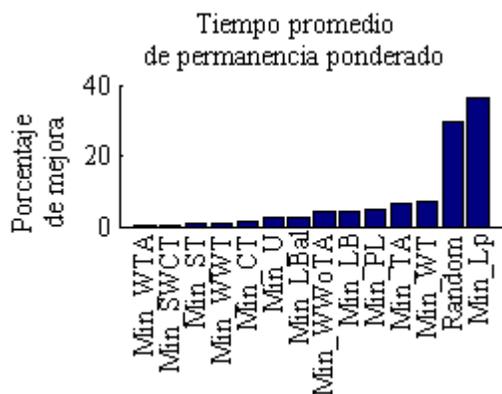


Figura 262. Porcentaje de mejora de los algoritmos, en el tiempo promedio de permanencia ponderado, al utilizar el esquema de admisibilidad.

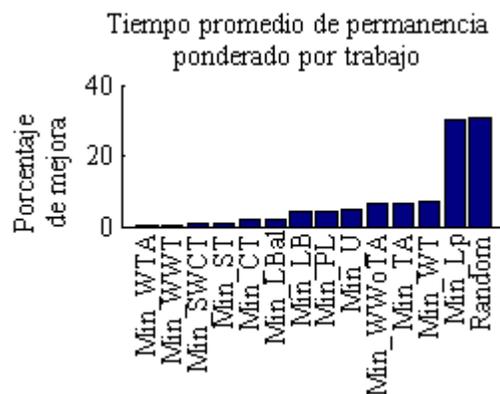


Figura 263. Porcentaje de mejora de los algoritmos, en el tiempo promedio de permanencia ponderado por trabajo, al utilizar el esquema de admisibilidad.

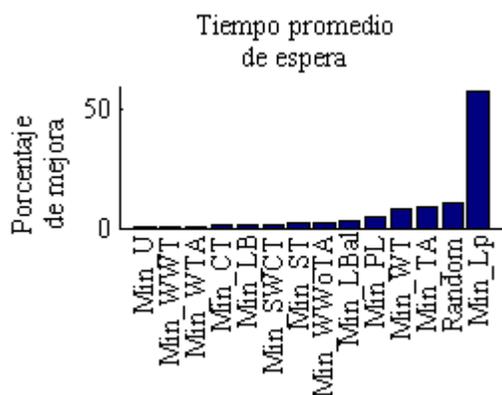


Figura 264. Porcentaje de mejora de los algoritmos, en el tiempo promedio de espera, al utilizar el esquema de admisibilidad.

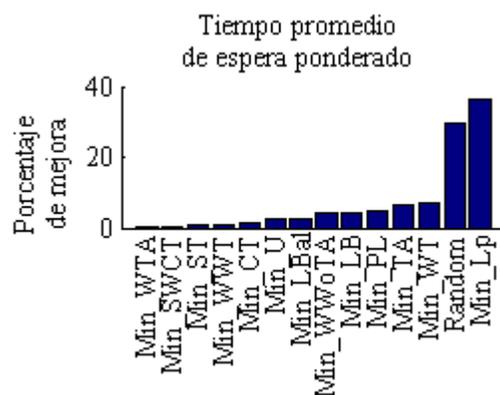


Figura 265. Porcentaje de mejora de los algoritmos, en el tiempo promedio de espera ponderado, al utilizar el esquema de admisibilidad.

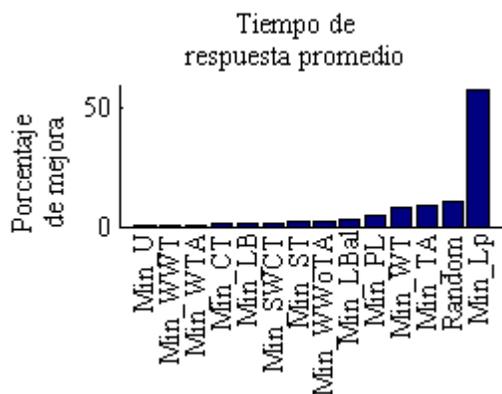


Figura 266. Porcentaje de mejora de los algoritmos, en el tiempo de respuesta promedio, al utilizar el esquema de admisibilidad.

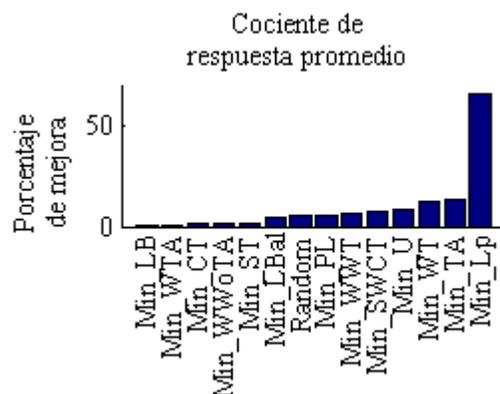


Figura 267. Porcentaje de mejora de los algoritmos, en el cociente de respuesta promedio, al utilizar el esquema de admisibilidad.

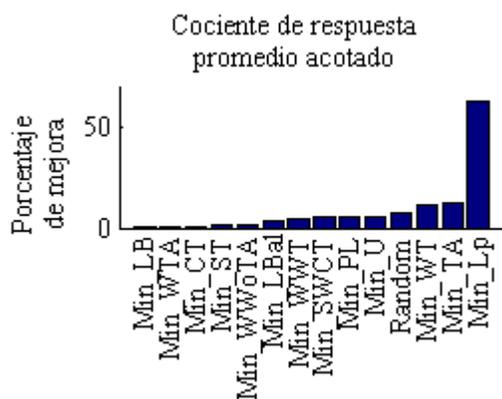


Figura 268. Porcentaje de mejora de los algoritmos, en el cociente de respuesta promedio acotado, al utilizar el esquema de admisibilidad.

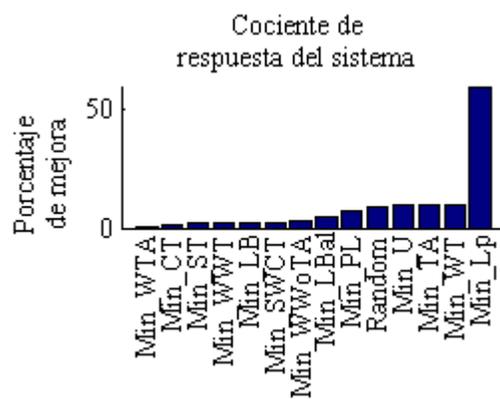


Figura 269. Porcentaje de mejora de los algoritmos, en el cociente de respuesta del sistema, al utilizar el esquema de admisibilidad.