



**Instituto Politécnico Nacional**

---

---

**Centro de Investigación en Computación**

**Secretaría de Investigación y Posgrado**

**CALENDARIZACIÓN DE TAREAS EN TIEMPO REAL  
CON RESTRICCIONES TEMPORALES ESTOCÁSTICAS  
MEDIANTE NIVELES DE ENTROPÍA**

**T E S I S**

**PARA OBTENER EL GRADO DE  
DOCTOR EN CIENCIAS DE LA COMPUTACIÓN**

**P R E S E N T A**

**M. C. MIGUEL ANGEL TOLEDO CUEVAS**



**DIRECTOR DE TESIS: Dr. JOSÉ DE JESÚS MEDEL JUÁREZ**

**MÉXICO, D.F.**

**AGOSTO 2007**

## **Dedicatoria**

En todo momento de mi vida he contado con personas que me han brindado su confianza. Agradezco a Dios por la oportunidad de convivir y aprender de ellas. Expreso mi agradecimiento a la familia, quien es la base de todo logro obtenido. Con este respaldo dedico este trabajo a las que llamo mi Familia:

- A mis padres por sus enseñanzas y apoyo incondicional,
- A mis hermanos por su comprensión y cariño,
- A mi esposa, quien es el amor de mi vida, por su amor y tolerancia ante las dificultades.

Mi Familia:

Albino Toledo de la Paz, María del Refugio Cuevas Ávila, Roberto Javier Toledo Cuevas, Elva Mayra Toledo Cuevas, Hugo Eduardo Toledo Cuevas, Claudia Xochitl Toledo Cuevas, María del Pilar Loya Zurita.

Así mismo dedico la culminación de presente trabajo a la familia como anifestación de nuestro espíritu de “guerreros” que llevamos en nuestra sangre. Les ofrezco mi sincero reconocimiento y cariño.

## Agradecimientos

Muchas y distintas personas han influido en mi. Son ellas quienes me han ayudado a formar mi carácter. He aprendido de cada una de ellas a proyectar mis anhelos de superación. A cada paso de mi vida he contado con ellas y ahora quiero agradecerles por todo lo recibido. Agradezco en particular:

- A mis amigos quienes se han arriesgado a involucrarse más allá de lo cotidiano,
- A mis maestros por su solidaridad en la realización de este anhelo,
- A mis compañeros de estudio quienes han compartido conmigo el privilegio de recibir una educación,
- A mis compañeros de trabajo quienes han permitido crear lazos de amistad.

Agradezco también a las instituciones que me dieron educación. En especial al Instituto Politécnico Nacional quien me acogió como estudiante desde nivel medio superior. Al Centro de Investigación en Computación por su apoyo y dirección; cumpliendo en un servidor, su compromiso con la investigación. A las personas que apoyaron para la realización de este trabajo. Algunas de ellas: Ing. Hugo Gamboa, Dr. Alejandro Tello, Dr. Jesús Figueroa, Dr. Jesús Medel y Dr. Julián Tapia.

# Índice general

Dedicatoria . . . . .	I
Agradecimientos . . . . .	II
Resumen . . . . .	III
Abstract . . . . .	V
Índice de Figuras . . . . .	XI
Simbología y Abreviaciones . . . . .	XII
Glosario de Términos . . . . .	XIV
<b>1. Introducción</b>	<b>1</b>
1.1. Motivación . . . . .	1
1.2. Planteamiento del Problema . . . . .	4
1.3. Objetivos de la Tesis . . . . .	5
1.3.1. Objetivo General . . . . .	5
1.3.2. Objetivos Específicos . . . . .	6

1.3.3. Objetivos Particulares . . . . .	7
1.4. Hipótesis . . . . .	7
1.5. Justificación . . . . .	8
1.6. Aportaciones . . . . .	10
1.7. Alcances y Limitaciones . . . . .	10
1.8. Organización de la Tesis . . . . .	11
1.9. Resumen del Capítulo . . . . .	12
<b>2. Estado del Arte de la Calendarización de Tareas en Tiempo Real</b>	<b>13</b>
2.1. Introducción . . . . .	13
2.2. Tareas en Tiempo Real del tipo Suave . . . . .	14
2.3. Calendarización de Sistemas en Tiempo Real . . . . .	16
2.4. Métricas de Planificadores . . . . .	16
2.5. Planificadores Estáticos . . . . .	17
2.6. Planificadores Dinámicos . . . . .	21
2.7. Calendarización Estocástica . . . . .	23
2.7.1. Sistemas con Tiempos de Demanda Estocástica . . . . .	24
2.7.2. Sistemas con Tiempos de Ejecución Estocásticos . . . . .	24
2.8. Resumen del Capítulo . . . . .	26
<b>3. Formulación del Modelo</b>	<b>27</b>

3.1. Proceso de Tarea en Tiempo Real . . . . .	28
3.2. Principio de Máxima Entropía . . . . .	30
3.3. Restricciones Temporales . . . . .	35
3.4. Asignación de Prioridad . . . . .	42
3.5. Resumen del Capítulo . . . . .	44
<b>4. Modelo Probabilístico de Planificador de Tareas</b>	<b>45</b>
4.1. Planeación de Tareas en Tiempo Real . . . . .	46
4.2. Modelo de Planificador Estocástico por Plazo . . . . .	46
4.2.1. Viabilidad . . . . .	49
4.3. Optimalidad . . . . .	50
4.4. Resumen del Capítulo . . . . .	53
<b>5. Algoritmo de Planificación Estocástico por Plazo</b>	<b>54</b>
5.1. Construcción del Algoritmo de Calendarización . . . . .	55
5.2. Experimentación y Resultados . . . . .	57
5.2.1. Comparación entre Algoritmos de Calendarización . . . . .	60
5.2.2. Análisis del Algoritmo bajo Condiciones Críticas . . . . .	62
5.3. Resumen del Capítulo . . . . .	66

<b>6. Conclusiones</b>	<b>67</b>
6.1. Conclusiones de los Alcances . . . . .	68
6.2. Trabajos Futuros . . . . .	69
6.3. Publicaciones . . . . .	71

## Índice de Figuras

5.1	Construcción del algoritmo planificador estocástico por plazo	56
5.2	Asignación de prioridades	58
5.3	Calendarización de tareas	59
5.4	Diferencia entre plazos y tiempos de finalizado	61
5.5	Asignación de prioridades	63
5.6	Calendarización de Tareas	64
5.7	Diferencia entre plazos y tiempos de finalizado	65

## Simbología y Abreviaciones

$A$	Tiempo de arribo
$A(t, \omega)$	Proceso de arribo
$C$	Tiempo de cálculo
$C(t, \omega)$	Proceso de cálculo
$D$	Plazo
$D(t, \omega)$	Proceso de plazo
$E_\pi$	Esperanza matemática para política $\pi$
$F$	Tiempo de finalizado
$F(t, \omega)$	Proceso de finalizado
$f_A(t)$	Función de densidad del tiempo de arribo
$f_C(t)$	Función de densidad del tiempo de cálculo
$f_D(t)$	Función de densidad del plazo
$f_F(t)$	Función de densidad del tiempo de finalizado
$\phi(z)$	Función de restricción
$H$	Entropía
$H_M$	Máxima entropía
$H(X)$	Entropía del proceso $X$ , o $H_X$
$I(X; Y)$	Transinformación de $Y$ en función de $X$
$I(A; D)$	Viabilidad de operación $A$ en función de $D$
$J$	Jerarquía
$J_M$	Máxima Jerarquía
$\lambda$	Unidad de tiempo de cálculo
$K_a$	Unidad de tiempo de arribo

$K_c$	Variación en la unidad de tiempo de cálculo
$K_d$	Unidad de tiempo de plazo
$\tau$	Proceso de tarea
$T$	Periodo, $p_i$
$U$	Utilización del procesador
$W$	Tiempo de demanda
$X(t, \omega)$	Proceso estocástico continuo

DTA	Delay Time Algorithm
EDF	Earliest Deadline Fist algorithm
EDS	Earliest Deadline Scheduler
LLF	Least Laxity Fist algorithm
PEP	Planificador Estocástico por Plazo
RM	Rate Monotonic algorithm
RTS	Real Time System
SRMS	Statical Rate Monotonic Scheduling
STDA	Stochastic Time Demand Algorithm
u.d.t.	Unidades de Tiempo
WCET	Worst Case Execution Time

## Glosario de Términos

**Algoritmo.** Conjunto ordenado de instrucciones con características como: preciso, finito, con resultados únicos, con entradas y salidas finitas, generalizable.

**Algoritmo óptimo.** Algoritmo de calendarización que minimiza alguna función de costo definida para el conjunto de tareas.

**Asignación de prioridad.** Operación del planificador que interrumpe la ejecución del trabajo actual y asigna el procesador a un trabajo mas urgente listo para ejecutarse.

**Calendarización.** Asignación de tareas en el procesador a modo que cada tarea es ejecutada hasta su finalización.

**Calendarización de plazo monotónico.** Calendarización con prioridad fija y derecho de procesamiento, que asigna una prioridad mayor a una tarea con un plazo más corto.

**Calendarización de tasa monotónica.** Calendarización con prioridad fija y derecho de procesamiento, que asigna una prioridad mayor a una tarea con un periodo más corto.

**Conjunto de tareas planificables.** Conjunto de tareas en el cual existe una calendarización factible.

**Desborde de tiempo.** Falta en el cumplimiento del plazo. Una situación en la cual la ejecución de una tarea continúa después de su plazo.

**Evento.** Un suceso que requiere una reacción del sistema.

**Exceso en tiempo.** Intervalo de tiempo en el cual un trabajo permanece activo después de su plazo.

**Factor de utilización.** Fracción de tiempo de procesador utilizada por un conjunto de tareas periódicas.

**Instancia.** Una ejecución particular de una tarea. Un único trabajo perteneciente a la secuencia de trabajos que caracteriza a una tarea periódica o aperiódica

**Instante crítico.** Tiempo en el cual la liberación de un trabajo produce el mayor tiempo de respuesta.

**Línea de espera.** Conjunto de tareas en espera de un tipo de recurso dado, ordenadas de acuerdo con algún parámetro de prioridad.

**Periodo.** Intervalo de tiempo comprendido entre la activación de dos instancias consecutivas de una tarea periódica.

**Plazo.** Tiempo en el cual una tarea en tiempo real debe completar su ejecución.

**Predecibilidad.** Una propiedad importante de un sistema de tiempo real que permite anticipar la consecución de cualquier decisión planificada.

**Prioridad.** Número asociado a una tarea y utilizado por el planificador para establecer u orden de precedencia entre tareas que compiten por un mismo recurso.

**Procesador.** Proceso de la computadora dedicado al manejo de un recurso compartido.

**Proceso.** Cálculo en el cual las operaciones son ejecutados por el procesador una a la vez. Un proceso puede consistir de una secuencia de trabajos idénticos llamados instancias.

**Programación de prioridad.** Forma de planificar en la cual los trabajos pueden ser interrumpidos en cualquier momento y el procesador es asignado a trabajos más urgentes que se encuentran listos para ejecutarse

**Sistema de tiempo real.** Cualquier sistema donde la salida producida es crítica. Esto es generalmente por que la entrada corresponde a algún desplazamiento en el mundo físico, y la salida tiene relación con ese mismo desplazamiento. El retraso entre el tiempo de entrada

y el tiempo de salida debe ser suficientemente pequeño para una puntualidad aceptable.

**Sobrecarga.** Condición de carga excepcional en el procesador, dado que el tiempo de cómputo demandado por las tareas en un cierto intervalo exceda el tiempo de procesamiento disponible en el mismo.

**Tarea.** Un cálculo en el cual las operaciones son ejecutadas por el procesador en un tiempo. Una tarea puede consistir en una secuencia de trabajos idénticos también llamados instancias. Las palabras proceso y tarea son usadas generalmente como sinónimos.

**Tarea aperiódica.** Tarea consistente en una secuencia de instancias, activados a intervalos irregulares.

**Tarea del tipo duro.** Tarea cuyas instancias deben garantizar a priori su cumplimiento antes del plazo establecido. Una falta en el plazo produce una falla definitiva en el sistema.

**Tarea del tipo firme.** Una tarea en la cual cada instancia debe garantizar su completa ejecución antes del plazo o será rechazada en su totalidad.

**Tarea del tipo suave.** Tarea cuya ejecución pueden ser posiblemente completado en su plazo sin tener consecuencias serias si ocurre una falta.

**Tarea esporádica.** Una tarea es esporádica si sus arribos son irregulares con cada arribo separado del precedente por el lapso de tiempo mínimo llamado tiempo de separación mínimo.

**Tarea periódica.** Tipo de tarea que consiste de una secuencia de trabajos idénticos (instancias), activados a intervalos regulares. Dos tareas periódicas están separadas por exactamente el mismo lapso de tiempo, llamado periodo  $p_i$ .

**Tiempo de arribo.** Instancia de tiempo en la cual un trabajo o tarea ingresa a la fila de procesos preparados. También se le conoce como tiempo de solicitud.

**Tiempo de computación.** Cantidad de tiempo requerida por el procesador para ejecutar un trabajo sin interrupción. También es llamado tiempo de servicio o tiempo de procesamiento.

**Tiempo de interarribo.** Intervalo de tiempo entre la activación de dos tareas consecutivas.

**Tiempo de liberación.** Tiempo asociado a una tarea para el inicio de su ejecución.

**Tiempo de procesamiento.** Cantidad de tiempo requerido por el procesador para ejecutar un trabajo sin interrupción. También llamado tiempo de cálculo o tiempo de servicio.

**Tiempo de relajación.** Retraso máximo que un trabajo puede experimentar después de su activación y todavía termine dentro de su plazo. En el tiempo de arribo, la relajación es igual al plazo relativo menos el tiempo de computación ( $D - C$ ). También es llamado tiempo flojo.

**Tiempo de finalizado.** Tiempo en el cual un trabajo finaliza su ejecución. También es llamado tiempo de completado.

**Trabajo.** Un cómputo en el cual las operaciones, en ausencia de otras actividades, son ejecutadas secuencialmente por el procesador hasta ser completadas.

**Transinformación.** En teoría de la probabilidad, y en teoría de la información, la información mutua o transinformación de dos variables aleatorias es una cantidad que mide la dependencia mutua de dos variables.

**Viabilidad.** Para un sistema  $X$  de tareas y una calendarización  $S$ ;  $X$  es viable en  $S$  sí y solo sí todos los plazos de las tareas en  $X$  son cumplidos en  $S$ .

**Zona crítica.** Es el intervalo comprendido entre el instante crítico de un trabajo y el tiempo de finalizado correspondiente.

## Resumen

Los equipos computacionales involucran procesos llamados tareas o instancias; cuando estos tienen interacción con sistemas del mundo exterior se dice que operan en tiempo real. Las tareas computacionales son eventos que requieren de recursos para llevarse a cabo. Al realizarse estas tareas acorde con sistemas de tiempo real están sujetas a restricciones dadas por: su origen, sistema donde se ejecutan, entradas de activación y plazos; entre otros.

Siendo las restricciones temporales, las características fundamentales que debe tomar en cuenta un planificador al operar en un sistema de tiempo real. La política que implementa el planificador tiene como objetivo satisfacer las restricciones temporales establecidas. El cumplimiento de estas las restricciones temporales mantiene “útil” al sistema computacional respecto del entorno donde se desenvuelve. Cuando existen variaciones en los tiempos que caracterizan la ejecución de una tarea, se dice que la tarea tiene restricciones temporales estocásticas; o es una tarea del tipo suave.

El presente trabajo de tesis investiga el problema de calendarización de tareas concurrentes para sistemas de tiempo real del tipo suave. En los cuales, una falta en alguna de las restricciones temporales es tolerable. En este trabajo se propone un modelo de planificador dinámico para tareas del tipo suave con un procesador. Donde las tareas son analizadas como procesos estocásticos con restricciones temporales. Estas restricciones se expresan como variables aleatorias con funciones de probabilidad conocida. Siendo la máxima entropía la base de la asignación de prioridad de las tareas en el modelo probabilístico. Así, asignando una política de calendarización factible sustentada en un modelo de planificador estocástico por plazo, se genera un algoritmo de calendarización para tareas con restricciones temporales estocásticas a través en dos técnicas desarrolladas:

- Asignación de prioridad dinámica a través del principio de entropía en un sentido de probabilidad.
- Calendarización dinámica con base en la viabilidad de la tarea a través del principio de transinformación.

Se construye, entonces, una técnica para la calendarización de tareas del tipo suave a través de la diferenciación entre ellas de acuerdo con una jerarquía asignada dinámicamente. Se establece una política de calendarización óptima para tareas planificables. El algoritmo de calendarización resultante se describe y evalúa en función de las propiedades de los procesos de tareas estocásticas concurrentes.

## Abstract

Computational devices involve processes called tasks or jobs; when these have interaction with real world systems they work in real time. Computational tasks are events which need resources to be carried out. At the moment to being executed tasks in real time systems, they are subject to certain constraints given by: sources or inputs, deadlines, for example.

Being time constraints, the main characteristics that a scheduler should take into account. The policy that a scheduler implements seeks to satisfy the established constraints. The fulfillment of these temporary constraints maintains “useful” the computational system respect to the surroundings where it operates. When variations in time exist that characterize the execution of a task, it says that the task has stochastic temporary constraints or the task is a soft kind.

The present thesis investigates the scheduling problem for concurrent stochastic tasks in soft real time systems; where a fault in one constraint is tolerable. In the thesis a dynamic scheduling model for stochastic tasks with one processor is set out; where tasks are analyzed as stochastic processes with temporary constraints. These constraints are expressed as stochastic variables with well-known probability functions. Thus, assigning a scheduling policy feasible supported in a model of stochastic scheduler by deadline, generates a scheduling algorithm for tasks with stochastic temporary constraints based on two techniques developed:

- Dynamic task allocation through the principle of entropy in a probability sense.
- Dynamic scheduling with base in the viability of the task through the transinformation principle.

A technique for soft real time tasks is, then, developed through the differentiation among tasks according to hierarchy allocated dynamically. The scheduling algorithm is described and evaluated based on the properties of the concurrent stochastic tasks processes.

# Capítulo 1

## Introducción

En este capítulo se presentan las premisas para el estudio de los sistemas en tiempo real del tipo suave a través de un modelo probabilístico de planificador, considerando la información contenida en las tareas. Partiendo del problema de investigación, la definición de objetivos y alcances del trabajo de tesis; se expone en este capítulo, la metodología que rige el proyecto de investigación. Justificando la importancia del tema de investigación se establece en el presente capítulo la organización del trabajo de tesis, en donde se establecen los parámetros para tener una medida de eficiencia de los resultados generados en el proceso de planificación.

### 1.1. Motivación

Un problema de actualidad en los procesos computacionales es la calendarización efectiva de tareas computacionales en ambientes con restricciones. Normalmente estas restricciones son impuestas por el mundo real. Dado que un sistema computacional interactúa con el

mundo real durante su evolución. El tiempo del sistema debe ser medido en la misma escala del sistema controlado. A estos sistemas se les llama “sistemas en línea”. Cuando estos sistemas cumplen con las restricciones temporales del sistema real, se les llama “sistemas en tiempo real” [44].

La calendarización de tareas involucra algoritmos de planificación estáticos o dinámicos que consideran variaciones en los parámetros del modelo de calendarización; ya sea en tiempo, cantidad o complejidad. Como ejemplo, los sistemas del mundo real en donde este tipo de procesos se presentan y tienen su campo de aplicación en: en equipos médicos, telecomunicaciones, control de procesos, entre otros. Donde los sistemas cuentan con restricciones del tipo duro, firme o suave dependiendo del grado con el cual, el mal funcionamiento del proceso de calendarización afecta al sistema.

Los sistemas en tiempo real del tipo suave son una generalización de los sistemas del tipo duro; entre estos se encuentran los sistemas de transmisión de información. La herramienta analizada en el trabajo de tesis tiene su aplicación en sistemas donde una falta temporal en el cumplimiento de las restricciones, disminuye el desempeño del sistema sin que se genere mayor problema en el sistema.

En últimas décadas el comportamiento de las tareas computacionales en su proceso de calendarización ha sido objeto de diversos estudios [6]. Desde los trabajos sobre la formalización de los procesos de calendarización hasta los de aplicación más recientes en la materia, han buscado dar respuesta a este problema a través de varios modelos. No obstante, la mayoría de los trabajos establecen modelos estáticos [34], [39]; es decir, presuponen condiciones estables durante el proceso de planificación de las tareas. Todos estos definen un conjunto de suposiciones sobre el comportamiento de las tareas que limitan la implementación del

algoritmo seleccionado de manera general. Esto especializa al planificador para operar con tareas de una clase particular [51]. Por lo que, cuando se involucran en un proceso de calendarización diferentes tipos de tareas, el algoritmo de planificación se vuelve más complejo, con grandes niveles de incertidumbre.

El estudio de diferentes modelos de planificadores, entre los cuales se pueden consultar [1], [25], [30], [35] y [55]; además de experiencias con algoritmos de control dinámicos, dieron la pauta para el desarrollo del trabajo de investigación. El trabajo de tesis introduce un planificador dinámico que maneja diversas clases de tareas descritas en los modelos clásicos coexistiendo durante el proceso de calendarización. El fundamento del algoritmo es un modelo probabilístico propuesto con base en la medida de la información contenida en las tareas, a través de la entropía para dar una jerarquía al conjunto planificable de tareas. A logro de establecer los parámetros de operación del planificador desarrollado se presenta un algoritmo de calendarización con base en la entropía en un sentido de probabilidad, del cual se analizan sus alcances.

Dentro del marco de sistemas en tiempo real del tipo suave, el modelo garantiza la planificación de tareas con distribución de probabilidad conocida en procesos con un procesador. Se plantean dos políticas de control de acuerdo con restricciones del tiempo de arribo y del plazo. Así, dentro de los alcances que se buscan en esta tesis es procesar un conjunto de tareas concurrentes en sistemas con un procesador, desarrollando un modelo formal e implementando un algoritmo de simulación para probar su eficiencia respecto de los algoritmos clásicos de calendarización de tareas: “Rate Monotonic” (RM), y “Earliest Deadline First” (EDF), planteados por Liu y Layland [34].

## 1.2. Planteamiento del Problema

En ambientes computacionales que interactúan con sistemas del mundo real se generan procesos de servicio de tareas en tiempo real. Este servicio se ve sujeto a restricciones propias de procesos de líneas de espera y tiempo real [28]. Un sistema computacional efectivo debe cumplir las siguientes propiedades en su operación:

1. Servicio a tareas planificables de diferente tipo en tiempo y forma.
2. Capacidad de servicio a un número finito de tareas concurrentes existentes.
3. Proceso de servicio dinámico de acuerdo con condiciones cambiantes del sistema.
4. Predicción paramétrica de las restricciones temporales.

Debido a las limitaciones propias del proceso de servicio, las tareas deben ser planificadas de acuerdo con una política de calendarización a modo de cumplir las propiedades arriba descritas.

Un planificador es un algoritmo que busca este fin, de manera eficaz y en la medida de lo posible efectiva. Para sistemas de tiempo real con tareas estocásticas con restricciones temporales del tipo suave, se establece la siguiente pregunta con respecto del problema de calendarización:

- ¿Es factible desarrollar un planificador óptimo en un sentido temporal, que opere con tareas concurrentes en tiempo real del tipo suave para un procesador?

A partir de esta pregunta principal se generan las siguientes preguntas particulares:

- ¿Qué características debe cumplir un modelo probabilístico de calendarización para ser óptimo respecto de un conjunto de tareas estocásticas viables?
- ¿Cuán efectiva puede ser la política de calendarización del algoritmo de calendarización propuesto, sustentado en un modelo probabilístico, en función de los algoritmos de calendarización clásicos RM y EDF?
- ¿Es factible expresar la optimalidad de un planificador de tareas estocásticas en función de la máxima entropía para la asignación de prioridades?

### **1.3. Objetivos de la Tesis**

La finalidad de la realización de este trabajo de tesis se expresa en un objetivo general que conlleva a tres objetivos específicos para orientar el método seguido en la investigación realizada.

#### **1.3.1. Objetivo General**

- Desarrollar un algoritmo de calendarización dinámico de tareas computacionales con base en un modelo probabilístico generado para tareas concurrentes con restricciones temporales estocásticas con función de distribución conocida en procesos de tiempo real del tipo suave con un solo procesador.

Para alcanzar el objetivo general se requieren realizar los siguientes objetivos específicos que marcan la pauta del proceso de investigación presente.

### 1.3.2. Objetivos Específicos

En primera instancia se establece un criterio para establecer la jerarquía del conjunto de tareas. Se propone una técnica de asignación de prioridad para las tareas con restricciones temporales definidas como variables aleatorias continuas con función de distribución conocida que las describe en un sentido de probabilidad. Donde la medida de incertidumbre está acotada de acuerdo con el nivel de entropía que se presenta en las restricciones temporales para tareas que coexisten en el proceso de calendarización con un procesador.

- Desarrollar un modelo probabilístico de asignación de prioridad para las tareas con restricciones de tiempo real del tipo suave analizadas como procesos estocásticos con función de distribución conocida de acuerdo con el principio de máxima entropía.

La capacidad de ordenar las tareas de acuerdo con una prioridad asignada dinámicamente establece una base para un modelo de calendarización que asegure una política de ejecución viable en un sentido de probabilidad.

- Desarrollar un modelo de probabilístico para planificar tareas estocásticas con funciones de distribución conocida a través de las restricciones temporales.

Este modelo permite implementar una forma práctica de operación a través de un algoritmo de planificación de tareas con el que puede compararse su eficiencia respecto a los algoritmos clásicos; RM y EDF.

- Aplicar el modelo de planificador a un algoritmo de calendarización estocástico para tareas de tiempo real del tipo suave comparando los resultados obtenidos con dos algoritmos clásicos de calendarización; RM y EDF.

### 1.3.3. Objetivos Particulares

Se ha de establecer el tiempo de cálculo de una tarea como una función de la información contenida en la tarea así como la capacidad de procesamiento del servidor para parametrizar el algoritmo propuesto.

- Definir las restricciones temporales de tareas en tiempo real del tipo suave como procesos estocásticos continuos con funciones de distribución de probabilidad conocida.
- Determinar las funciones de máxima entropía para las restricciones temporales de las tareas bajo consideraciones establecidas de función de probabilidad e independencia entre tareas. Establecer una unidad de medida temporal a partir de un tiempo de cálculo medio.
- Asignar una jerarquía de manera dinámica a las tareas de acuerdo con la máxima entropía como medida de uniformidad como una forma diferenciación y ordenamiento.

## 1.4. Hipótesis

El análisis de tareas en tiempo real del tipo suave puede verse como un proceso estocástico continuo con función de distribución conocida que permite definir un modelo probabilístico para la calendarización de tareas con restricciones temporales estocásticas con base en niveles de entropía . Así mismo es posible establecer una solución matemática viable de calendarización si se conocen las funciones de probabilidad durante la operación del sistema, tal como sucede con los procesos de líneas de espera. Con esta herramienta se desarrolla una técnica de calendarización de tareas capaz de procesar de manera efectiva, en un sentido de probabilidad, diversas tareas que coexisten en un sistema con un procesador.

## 1.5. Justificación

Desde el desarrollo de los primeros planificadores de tiempo real se han propuesto diversos modelos para la calendarización de tareas de acuerdo con el tipo de las mismas. Considerando sus restricciones temporales, así como sus aplicaciones. En varios algoritmos de planificación se requiere especificar el tipo de tarea que arriba al procesador a ser atendida; estableciendo medidas y parámetros de prioridad. Dentro de los modelos clásicos se tienen: los planificadores de prioridad fija como el planificador de “Tasa monotónica” (RM); y los planificadores de prioridad dinámica como el “Primero el de plazo más cercano” (EDF) [34], o el “Primero el de tiempo más laxo” (LLF) [39]; ver también [8].

El conocimiento del medio con el que interactúa el procesador es parte fundamental en la selección del algoritmo de planificación [38]. En sistemas embebidos donde el medio circundante es controlado por un sistema computacional, el cumplimiento de las tareas en tiempo y forma es vital en este tipo de aplicaciones, de donde es necesario distinguir los requerimientos específicos de las tareas para darles el servicio adecuado en algún sentido definido. Asignar prioridades a las tareas de acuerdo con criterios de operación que le permitan al planificador dar servicio de manera eficiente en un sentido temporal, es un paso fundamental en los diversos estudios de calendarización [63]. La propiedad más importante de un sistema en tiempo real, es la predicción [56]. Es decir, la determinación previa de las características de las tareas y sus restricciones por cumplir para poder ser planificados. Es por lo tanto importante en cualquier sistema de calendarización de tiempo real desarrollar una herramienta que cumpla con este requerimiento.

Se ha visto que la naturaleza de las tareas computacionales determina en mayor medida la técnica de calendarización a utilizar [62]. El trabajo de investigación propone un modelo de

calendarización para el tratamiento de tareas que coexisten en un sistema con un procesador. Siendo uno de los objetivos generar un modelo de asignación de prioridades a tareas con base en la información contenida en la tarea en un sentido de probabilidad.

El estudio se enfoca a aquellos sistemas en tiempo real del tipo suave donde existe una variación en las restricciones temporales considerándose del tipo suave [31]. Así los resultados del trabajo son útiles en aplicaciones dominadas por actividades semiperiódicas o aperiódicas. Ejemplos de aplicación se pueden mencionar los siguientes sistemas:

- Transmisión de audio y video en tiempo real,
- Comunicación en redes digitales,
- Problemas de control robótico,
- Balanceo de líneas en manufactura flexible,
- Acceso a bases de datos,
- Compresión y reproducción de archivos multimedia.

En varias aplicaciones se requiere que el planificador siga la dinámica de la tarea para cumplir con el control del sistema del mundo real [13]. Como sucede en sistemas de manufactura o en redes de comunicaciones digitales. Existen en la actualidad varios modelos de planificadores dinámicos para tareas en tiempo real del tipo suave [6]. El modelo propuesto realiza el tratamiento de las tareas de acuerdo con una medida de uniformidad en las restricciones correspondientes de manera dinámica. Tomando como base la información contenida

de las tareas analizadas como procesos estocásticos. La generación de este algoritmo, a partir de los modelos propuestos, abre una ventana en sistemas de calendarización dinámica de tareas para estudios posteriores.

## 1.6. Aportaciones

Como aportes, la tesis especifica:

- Desarrollo de un modelo de asignación de prioridad dinámica para tareas estocásticas con base en niveles de entropía en un sentido de probabilidad.
- Desarrollo de un modelo óptimo de calendarización de tareas estocásticas por plazo con base en la transinformación.
- Implementación de los modelos desarrollados en un algoritmo de calendarización por plazo para tareas estocásticas.

El trabajo de tesis provee de un análisis de tareas en tiempo real del tipo suave como procesos estocásticos. Propone un modelo de jerarquías dinámicas de tareas de acuerdo con parámetros obtenidos de sus restricciones temporales.

## 1.7. Alcances y Limitaciones

Como todo modelo propuesto está sujeto a limitaciones tanto de aplicación como de establecimiento de parámetros de operación. La viabilidad del desarrollo de la investigación surge a partir de resultados matemáticos previos como: [30], [29], o [21], así como de experimentos propios de calendarización de tareas. Los alcances de la investigación cubren tareas

de sistemas en tiempo real del tipo suave con un procesador con propiedades dinámicas acotadas. Los resultados expresados a lo largo del trabajo de tesis establecen la viabilidad de implementación del modelo planteado en sistemas reales a través de modelos de simulación para los casos analizados.

## **1.8. Organización de la Tesis**

De acuerdo con la metodología establecida en [9], se plantea la estructura y desarrollo de esta tesis:

Se presenta en este capítulo 1 el planteamiento del problema y los objetivos del trabajo de investigación. Se establece también la justificación de la hipótesis planteada así como los alcances y limitaciones del tema bajo estudio.

En el capítulo 2, se describe el estudio del estado del arte de los planificadores de tareas en tiempo real, con esto se construye el marco teórico. Concretando con los sistemas de tareas a analizar en los capítulos del 3 al 5.

En el capítulo 3 se presenta la propuesta del modelo de asignación de prioridades para las tareas de tiempo real del tipo suave con restricciones temporales estocásticas. Éste incluye el modelado del proceso de tarea, la asignación de prioridades de acuerdo con sus niveles de entropía y el planteamiento de un planificador soportado por estas prioridades.

En el capítulo 4 se desarrolla un modelo de planificador estocástico con base en el plazo. Estableciendo los casos para cuando el tiempo de finalizado sea estrictamente menor que el plazo, así como cuando el plazo marca el periodo de la tarea.

En el capítulo 5 se concreta el valor práctico del trabajo, estableciendo las consecuencias particulares del estudio. Como prueba de viabilidad del algoritmo desarrollado, se lleva a cabo el análisis computacional del modelo de simulación con respecto a los algoritmos clásicos de RM y EDF. Se presentan los resultados de simulación para datos teóricos con los que se puede concluir sobre del modelo de calendarización probabilístico, sus limitaciones y alcances.

## **1.9. Resumen del Capítulo**

En este capítulo se introduce la motivación para el trabajo de tesis. Se plantea el problema de la calendarización de tareas con restricciones temporales operando en un procesador. Se plantea un modelo matemático de calendarización óptimo en su política de asignación de prioridad para tareas concurrentes viables.

Se expone la justificación al problema planteado en su aplicación a sistemas flexibles de manufactura, redes y sistemas multimedia. Donde las tareas deben ser completadas en plazos establecidos por el entorno.

Para lograr el objetivo planteado, se consideran a las restricciones temporales de las tareas como procesos estocásticos con funciones de distribución determinadas a partir de la información contenida en los parámetros estadísticos del comportamiento.

A través de la metodología planeada en este capítulo se estructura el trabajo de tesis con la finalidad de establecer sus alcances y limitaciones.

## Capítulo 2

# Estado del Arte de la Calendarización de Tareas en Tiempo Real

En este capítulo se presenta una retrospectiva de trabajos de investigación referentes a la teoría de calendarización de tareas computacionales en tiempo real. Se realiza un estudio sobre las características de algunos planificadores de tareas en tiempo real con la finalidad de presentar un panorama general del estado del arte en esta disciplina. Así como, se hace un análisis de las consideraciones empleadas por los modelos previos que permita sustentar la técnica de análisis desarrollada en los capítulos subsecuentes.

### 2.1. Introducción

Los sistemas en tiempo real están definidos como aquellos sistemas donde la respuesta correcta del sistema depende no solo de los resultados lógicos del cálculo, sino también del tiempo en el que estos resultados se generan [38]. Ejemplos de este tipo de sistemas son:

sistemas de control de procesos, sistemas de comunicaciones, sistemas multimedia, etc.

Un sistema en tiempo real es aquel con requerimientos temporales explícitos determinísticos o probabilísticos [56]. Históricamente, la investigación en sistemas de tiempo real surgió de la necesidad de entender, diseñar, predecir, y analizar aplicaciones críticas tales como control de plantas industriales, robótica, o sistemas de control de vuelo, por citar algunos. En los cuales una falta en el cumplimiento de alguna restricción temporal causaría daños serios al equipo o a vidas humanas. Los primeros estudios sobre esta clase de sistemas en tiempo real del tipo duro donde la única manera de asegurar que ningún requerimiento de tiempo real sea quebrantado es considerando restricciones determinísticas acerca de las tareas.

En sistemas en tiempo real del tipo duro, se asume que cada instancia de una tarea opera en un intervalo finito y acotado de tiempo [40] para el peor de los casos, a éste se le llama el “peor tiempo de ejecución” (WCET). Esta aproximación es a menudo la única aplicable para la clase de sistemas embebidos críticos. Lo que significa que el planificador deba tener la habilidad de predecir a priori las características de las tareas a modo de que pueda cumplir con todos los requerimientos temporales de forma crítica [56]. No así, para muchos otros sistemas reales donde el objetivo es la calendarización de tareas en tiempo real del tipo suave, se asegura estadísticamente el buen desempeño de los procesos de las tareas a ejecutarse.

## **2.2. Tareas en Tiempo Real del tipo Suave**

Las restricciones temporales son particulares de los sistemas de tiempo real. Este tipo de restricciones expresan un requerimiento en el tiempo que expresa cuándo una tarea debe ser completada. Estas restricciones son generalmente expresadas a través de un plazo. El

plazo representa el tiempo máximo de cálculo en el que una tarea debe ser completada por el procesador. Se distinguen dos tipos de plazo:

- **Plazo Duro.** Sí el cumplimiento del plazo de una tarea es determinístico para el funcionamiento del sistema, entonces la tarea se dice que es del tipo *duro* (perder un plazo duro es considerado una falla definitiva, y resulta en consecuencias catastróficas).
- **Plazo Suave.** Sí el cumplimiento del plazo en sentido de probabilidad está acotado en sus vecindades por las varianzas, entonces el plazo se dice que es *suave* (una tarea con plazo suave espera ser completada ya sea antes que el plazo o lo más pronto posible después de él).

Para una gran cantidad de clases de sistemas en tiempo real del tipo suave, la aproximación de un tiempo crítico de ejecución WCET significa una subutilización de recursos de computo. Ya que, la falta en una restricción temporal en una vecindad no repercute de manera perceptible en la calidad de servicio.

La predecibilidad de las tareas es un principio que le permite a un sistema de tiempo real operar de manera adecuada. Para los sistemas de tiempo real del tipo suave las restricciones temporales deben cumplirse de manera probabilística para mantener la calidad de servicio. Por ejemplo en aplicaciones multimedia; en ellos, la falta en el cumplimiento de algún plazo no perjudica el desempeño general de la reproducción de tales archivos debido a que no es perceptible a los sentidos.

### **2.3. Calendarización de Sistemas en Tiempo Real**

La calendarización de tareas involucra la asignación de tiempos del procesador en función de mantener los requerimientos de desempeño [34]. La teoría de calendarización de tareas ha sido uno de los tópicos más investigados en sistemas de tiempo real. Aunque la teoría de la calendarización nació en el dominio de la Investigación de Operaciones [15]. La calendarización de tareas en tiempo real aparece ocasionalmente en esta literatura, la razón es por la diferencia en la naturaleza del análisis. En Investigación de Operaciones se busca minimizar el tiempo total del procesador; independientemente del cumplimiento local de cada proceso dentro de la tarea. Los estudios clásicos buscan maximizar la medida de desempeño pero sin garantizar que las restricciones temporales se cumplieran siempre; es decir, la predecibilidad del sistema [10]. Además, el planteamiento clásico consiste principalmente en la calendarización de tareas que se ejecutan una sola vez en lugar de una cantidad infinita. Por lo que es necesario realizar el análisis de la planificación y la comprobación de viabilidad para cada tarea durante el proceso en las que las tareas están activas en el sistema.

### **2.4. Métricas de Planificadores**

La teoría de calendarización clásica utiliza métricas de manera local para: minimizar la suma de los tiempos de finalizado (ó suma ponderada), minimizar el tiempo de calendarización, minimizar el tiempo de procesadores requeridos, o minimizar los tiempos de latencia máximos. Cuando se consideran a los plazos añadidos como restricciones creando un tiempo de calendarización máximo sujeto a que todos los tiempos de finalizado deben cumplir con esta restricción. Si una o más tareas pierden su plazo, no se plantea una solución factible [57].

Relacionado a las métricas está la complejidad misma de los problemas de calendarización. Varios resultados de planificadores resultan ser NP-completos o NP-duros [56]. Donde NP es la clase de todos los problemas de decisión que pueden resolverse en tiempo polinomial por una máquina determinística. Un problema reconocido  $R$  es NP-completo si  $R \in NP$  y todos los otros problemas en  $NP$  son transformables polinomialmente a  $R$ . Un problema de optimización  $R$  es NP-duro si todos los problemas en  $NP$  son polinomialmente transformables a  $R$ , pero no se puede mostrar que  $R \in NP$ .

En el marco de este trabajo de tesis se enfocará a la calendarización de sistemas con un procesador único y tareas independientes. En esta situación, el algoritmo de calendarización puede ser entendido como la regla que selecciona, en cada instante, una petición específica entre todas las peticiones activas (si las hay) de acuerdo con una política establecida. Por lo que las tareas esperan por la disponibilidad del procesador para iniciar o continuar su ejecución. Si no existe alguna petición, el procesador permanece sin ser asignado, o se dice que está desocupado.

Entre varios algoritmos de calendarización propuestos en el análisis de tareas en tiempo real con un procesador se pueden distinguir los planteamientos siguientes:

## 2.5. Planificadores Estáticos

Los primeros resultados en la calendarización de tareas en tiempo real se han obtenido considerando restricciones sobre el comportamiento de las tareas o la arquitectura que los soporta [51]. En esta primera literatura los procesos de tareas se han asumido con propiedades específicas para su estudio.

En 1973, la teoría de calendarización de tareas en tiempo real con prioridad fija fue

propuesta por Liu y Layland [34], quienes establecieron las reglas fundamentales para las tareas periódicas. En su trabajo, los autores establecieron las primeras consideraciones de operación para los planificadores. Sha [51] resume estas consideraciones como sigue:

1. Todas las tareas son periódicas.
2. Todas las tareas son liberadas al comienzo de su periodo.
3. Todas las tareas tienen un tiempo de terminación igual a su periodo.
4. Todas las tareas son independientes, por lo que no tienen relación de precedencia.
5. Todas las tareas tienen un tiempo de cálculo o límite fijo, menor o igual a su periodo.
6. Ninguna tarea puede suspenderse voluntariamente a sí misma.
7. Todas las tareas pueden interrumpirse en su ejecución en cualquier momento de acuerdo a su prioridad relativa.
8. Todos los tiempos de administración son cero.
9. Existe un único procesador.

Enfocando esta definición al trabajo de investigación, se define a las tareas periódicas expresadas como un proceso parametrizado por restricciones temporales características de las tareas en tiempo real.

### **Definición 2.1** Tarea Periódica en Tiempo Real

*Un proceso de tareas periódicas en tiempo real  $X = \tau_1, \tau_2, \dots, \tau_n$  es un conjunto de tareas con las siguientes características [37]:*

- *Las tareas periódicas son independientes en su inicio.*
- *Cada tarea  $\tau_i$  es periódica, con un periodo constante  $p_i$  entre el inicio de cada tarea.*
- *Cada tarea  $\tau_i$  tiene un tiempo de inicio  $I_i(\geq 0)$ . Considerando que  $\min_{1 \leq i \leq n} I_i = 0$ . Las tareas son iniciadas en  $I_i + kp_i$  ( $k = 0, 1, \dots$ ).*
- *Cada tarea  $\tau_i$  tiene un plazo  $d_i(\leq p_i)$ . Una tarea iniciada en  $I_i + kp_i$  debe ser completada antes de  $I_i + kp_i + d_i$ .*
- *El requerimiento de cálculo en cada tarea  $\tau_i$  es una constante  $c_i(\leq d_i)$ .*
- *Las tareas  $\tau_i$  pueden ser interrumpidas en cualquier momento y el tiempo de administración puede considerarse nulo.*

Cuando  $d_i = p_i(1 \leq i \leq n)$ , la calendarización de plazo monotónico se transforma en la calendarización de tasa monotónica.

En los modelos de calendarización estáticos, las tareas son ejecutadas por un planificador de acuerdo con una política de asignación estática de prioridades fuera de línea. Esta asignación de prioridades se realiza, por ejemplo, de acuerdo con el algoritmo RM. El planificador de tasa monotónica es un planificador estático para tareas con derechos de ejecución; el cual asigna una prioridad a cada tarea en proporción inversa a su tasa de arribo. Liu y Layland plantearon la utilización del procesador como base para su criterio de calendarización. Ellos también probaron que la asignación de prioridades por RM es óptimo para el esquema fijo fuera de línea. Definiendo a  $C_i$  como el tiempo de cálculo, a  $T_i$  como el período y a  $N$  como el número de tareas, se dice que una tarea es planificable si:

$$\sum_{i=1}^N \frac{C_i}{T_i} \leq N \cdot (2^{1/N} - 1) \quad (2.1)$$

Una condición necesaria y suficiente para la calendarización acorde con las consideraciones arriba mencionadas fue elaborada por Lehoczky *et al* [32], quienes permitieron la abstracción de la prueba de factibilidad para la zona crítica del planificador RM, observando que si un conjunto de tareas son liberadas en un tiempo zero, la tarea de mayor prioridad completará su primera ejecución antes de su tiempo de finalizado si existe un tiempo  $t$ ,  $0 < t \leq T_i$  tal que la demanda en el procesador  $W_i(t)$  para la tarea  $i$  de mayor prioridad sea:

$$W_i(t) = \sum_{j=1}^i \left( \frac{t}{T_j} \right) \cdot C_j \leq t \quad (2.2)$$

Donde la relación  $\lfloor \frac{t}{T_j} \rfloor$  debe ser menor que 1. Lehoczky mostró que es suficiente revisar la condición de planificación en el tiempo de liberación de las tareas con mayor prioridad. Esta prueba es también aplicable a los procesos de tarea con prioridades fijas arbitrarias. Joseph y Pandya [27] derivaron una condición necesaria y suficiente sobre el análisis en el tiempo de respuesta para el “Algoritmo de Dilatación del Tiempo” (DTA). El algoritmo calcula el tiempo de respuesta para peor caso  $R_i$  de la tarea  $\tau_i$  como la solución del punto fijo menor con la ecuación recursiva:

$$R_i = C_i + \sum_{j=1}^{i-1} \left\lceil \frac{R_i}{T_j} \right\rceil \cdot C_j \quad (2.3)$$

Una propiedad fundamental para el algoritmo de tasa monotónica es su optimalidad. El RM es aplicable si se cumple con el análisis de viabilidad para el proceso de calendarización. Este análisis puede ser caracterizado por un número de factores como las restricciones temporales o el modelo computacional. Según Liu y Layland:

1. Un conjunto de tareas se dice planificable con una prioridad asignada, sí todos los tiempos de terminación de todos los requerimientos de las tareas se cumplen.

2. Una asignación de prioridad se dice que es viable sí, con esta asignación de prioridad, la tarea es planificable.

El algoritmo RM es probablemente es más usado en aplicaciones de tiempo real con prioridad asignada debido a su facilidad de implementación en aquellos kernels comerciales que no soportan restricciones de tiempo explícitas en el conjunto de tareas. Esta facilidad de implementación unida a un menor tiempo de administración facilita su análisis, y es más predecible en condiciones de sobre carga; además produce un menor deface en los tiempos de ejecución.

## 2.6. Planificadores Dinámicos

En el trabajo de Liu y Layland se analiza la calendarización de tareas en línea, de acuerdo con un planificador que asigna las prioridades a las tareas de manera dinámica. Esta asignación se realiza mediante el algoritmo EDF. De acuerdo con el algoritmo EDF, las prioridades son asignadas dinámicamente y son inversamente proporcionales a los tiempos absolutos del plazo de las tareas activas [11]. Es decir, la tarea próxima en cumplir su plazo se le asigna la mayor prioridad.

Liu and Layland demostraron que para la asignación de prioridades dinámicas el algoritmo de EDF es óptimo. lo que significa que si existe un método de asignación de prioridad para tarea en línea la cual es planificable entonces es planificable por el algoritmo EDF.

Con las condiciones anteriores; un conjunto de  $N$  tareas periódicas es planificable por el algoritmo EDF, siendo  $U_i$  la utilización del procesador; sí y solo sí:

$$\sum_{i=1}^N U_i \leq 1 \tag{2.4}$$

Sha y Goodenough [52] mostraron que la capacidad de calendarización puede ser mejorada significativamente. Sí el plazo es menor que el periodo en la asignación de prioridades con la política EDF. Condición necesaria y suficiente dada por Baruah *et al.* [5]. Quien establece que:

**Teorema 2.1** Capacidad de Realización

*Para un conjunto de tareas de distintos tipos con cardinalidad acotada, el problema de capacidad de realización puede resolverse en un tiempo pseudo-polinomial. Baruah.*

Mok presentó en 1983 [39], un algoritmo óptimo, “Primero el de Menor Holgura” (LLF); el cual asigna al procesador la tarea activa con periodo de relajación más pequeño. El LLF y el EDF son algoritmos óptimos; aunque el LLF maneja un tiempo de administración mayor debido al constante cambio entre tareas por los cambios del tiempo de relajación. En el 2000 Aydin *et al* [4] presentaron un algoritmo óptimo que maximiza el peso promedio de las tareas asociadas con un ganancia en el esquema EDF. En 2003 Goossens [20] aplicó el esquema del algoritmo EDF a sistemas multiprocesador demostrando la optimalidad de la política de calendarización para tareas periódicas.

Considerando los diferentes métodos y técnicas presentados con anterioridad se puede observar que no existe un criterio unificado en la aproximación práctica de los sistemas de tareas con parámetros temporales para el diseño y análisis de políticas óptimas entre los modelos de calendarización. Varias formulaciones han obtenido resultados aplicados de acuerdo con sus argumentos específicos al problema. Más no ofrecen una extensión a modelos semejantes. Sin embargo, algunas técnicas, como el intercambio de argumentos han sido aplicados satisfactoriamente a varios modelos, P. Nain [41].

## 2.7. Calendarización Estocástica

La teoría de la calendarización estocástica orienta este problema a varios modelos de sistemas con servicios estocásticos. Las características aleatorias de las tareas, como los tiempos de procesamiento, son modelados por sus distribuciones de probabilidad [18]. Suponiendo que estas son conocidas por el manejador del sistema, las consideraciones del modelo varían en diferentes dimensiones, incluyendo las clases de políticas de planificación admisibles, las distribuciones para los tiempos de arribo y cálculo, el tipo y arreglo de los recursos de servicio, y el objetivo del desempeño a ser optimizado.

Los modelos de calendarización estocástica pueden clasificarse en tres categorías generales de acuerdo con su autonomía:

- modelos para planificar un conjunto de tareas estocásticas,
- modelos de “máquina múltiple”,
- modelos para planificar sistemas de líneas de espera.

Recientemente Niño - Mora [42] propuso un modelo de planificador de tareas estocásticas a través de la esperanza matemática como base para la asignación de prioridad. En su trabajo, el autor calcula la esperanza matemática  $E_\pi[\cdot]$  bajo una política  $\pi$  establecida. Así como para una clase  $\Pi$  de las políticas dadas. Entonces el planificador se busca minimizar el tiempo de cálculo de las  $n$  tareas representado con la expresión:

$$\min_{\pi \in \Pi} w_1 E_\pi[\tilde{C}_1] + \dots + w_n E_\pi[\tilde{C}_n] \quad (2.5)$$

Donde el objetivo principal de desempeño se establece a través de minimizar el “tiempo de flujo”. Expresado como:

$$\min_{\pi \in \Pi} E_{\pi} \left[ \sum_{j=1}^n \tilde{C}_j \right] \quad (2.6)$$

### 2.7.1. Sistemas con Tiempos de Demanda Estocástica

A través de una frontera temporal para el “Algoritmo de Tiempos de Demanda Menores” STDA; Gardner [16] presenta un algoritmo para planificar tareas estocásticas con prioridad fija considerando la distribución de probabilidad de los tiempos de ejecución de acuerdo con el tiempo de demanda restante para cumplir con el plazo. En su trabajo Gardner establece que el tiempo de demanda de una tarea  $\omega_{i,j}(t) \in t$  está en función de la probabilidad de cumplimiento del tiempo de demanda;  $W_{i,j}(t) = \Pr[\omega_{i,j}(t) \leq t]$  siendo:

$$W_{i,1}(t) = \Pr[\omega_{i,1}(t) \leq t] \quad (2.7)$$

Para la tarea con mayor prioridad; y para las tareas con prioridad menor en el intervalo  $[r_{i,j}, c_{i,j})$ , de los tiempos de inicio y cálculo respectivamente,

$$W_{i,j}(t) = \Pr[\omega_{i,1}(t) \leq t \mid \omega_{i,j-1}(r_{i,j}) > r_{i,j}] \quad (2.8)$$

Partiendo del análisis de Tia [58] para el tiempo de demanda probabilístico, Gardner aplica su algoritmo al planificador estadístico de tasa monotónica de Altas y Bestavros [3].

### 2.7.2. Sistemas con Tiempos de Ejecución Estocásticos

Considerando que los sistemas digitales con un procesador tienen variaciones en el tiempo de ejecución, se han desarrollado varios modelos para operar bajo estas condiciones. Atlas

y Bestavros [3] extendieron la política de calendarización de tasa monotónica para manejar tareas con tiempo de ejecución estocástico SRMS. Ellos analizaron la calidad de servicio de la calendarización resultante y su dependencia con los parámetros de admisión del procesador. Aunque su propuesta se limitó al análisis de tasa monotónica asumiendo la presencia de un control de admisión durante el tiempo de operación.

El trabajo de Abeni y Butazzo [2] se enfocó al análisis del desempeño y la calendarización de tareas con parámetros estocásticos. Enfocando su trabajo en la calendarización de tareas en tiempo real del tipo duro y del tipo suave en un mismo procesador; de manera que las tareas del tipo duro no se vieran afectadas por las tareas del tipo suave. Aquí, el método del análisis de desempeño es utilizado para valorar su política de calendarización propuesta, estando restringida al campo de sus consideraciones.

Lehoczky [29], [30] modeló un conjunto de tareas como un proceso de Markov. La ventaja de su propuesta es que es aplicable a políticas de calendarización arbitrarias. El estado del proceso es un vector de “tiempo de avance” (tiempo restante antes del plazo). Como este espacio es potencialmente infinito, Lehoczky analizó el sistema en condiciones de tráfico pesado, donde el sistema provee una solución simple. Las principales limitaciones de esta propuesta son que las consideraciones para los tiempos de ejecución e interarribo no son realistas.

En 2006 Niño - Mora [43], sustenta su trabajo de manejo de políticas para la calendarización de líneas de espera multiclase con base en la teoría de líneas de espera y la esperanza matemática de los tiempos de relajación. En su trabajo el autor determina los “índices de productividad marginal” a través del análisis de costo marginal de cada clase de tarea en el sistema.

Otras técnicas se ha propuesto en el desarrollo de sistemas con prioridad dinámica con base en los algoritmos RM y EDF. Dependiendo de la aplicación en sistemas de comunicaciones, análisis de señales, y multimedia se establece en el siguiente capítulo una técnica que maneja de manera más adecuada estos sistemas de tiempo real del tipo suave.

## **2.8. Resumen del Capítulo**

Los estudios realizados en el área de calendarización de tareas son amplios. En este capítulo se da una visión general de diversos trabajos de acuerdo con su enfoque. Las consideraciones hechas por diversos autores dificulta la comparación ente resultados aplicables de forma general para tareas en tiempo real.

Así mismo se dan los antecedentes de trabajos para el tratamiento de tareas con restricciones temporales estocásticas. Enfocando el estudio de tesis a aquellos planificadores dinámicos de tareas en tiempo real del tipo suave.

# Capítulo 3

## Formulación del Modelo

En este capítulo se presentan las definiciones y notaciones utilizadas a lo largo del trabajo de tesis. Se establece una formulación exacta del modelo con las consideraciones que marcarán su aplicación en sistemas de tiempo real del tipo suave.

Sea un conjunto de eventos en un proceso de tareas computacionales. Donde un evento puede ser el arribo, ejecución o plazo de las tareas. Así mismo, un evento puede referirse a una secuencia de acciones, como por ejemplo, el inicio de una tarea  $\tau$  al tiempo  $t_1$  y su finalización al tiempo  $t_2$ . Ahora bien, se describe una variable aleatoria como un mapeo que asocia un número real a un evento. La distribución de una variable aleatoria  $X$  es una función de  $F_X$ , ó bien,  $F(x) = \Pr(X \leq x)$ . La función  $F(x)$  indica cuál es la probabilidad de que los eventos tengan un mapeo a los reales menores o iguales que  $x$ . Esta función es monótona creciente y cumple con  $\lim_{x \rightarrow -\infty} F(x) = 0$  y  $\lim_{x \rightarrow \infty} F(x) = 1$ . La densidad de probabilidad para la variable  $X$  está expresada por la primera derivada de  $F(x)$ , siendo  $f(x) = \frac{dF(x)}{dx}$ . Si la función de distribución es continua, la variable aleatoria es continua. Un proceso de tarea está formado por aquellas restricciones temporales definidas por la familia de variables aleatorias que la

integran. Dentro de las principales características que conforman una tarea en tiempo real del tipo suave. En [59], el autor hace una definición formal de este proceso.

### 3.1. Proceso de Tarea en Tiempo Real

Una tarea en tiempo real del tipo suave  $\tau$ , se describe como un proceso estocástico continuo a través de una colección de variables aleatorias  $\{X_i(t, \omega)\} \in \mathbb{R}^+$  expresadas por sus restricciones temporales. Para el modelo bajo estudio se establecen las siguientes consideraciones:

1. Una tarea se activa un instante después de su arribo al sistema.
2. Todas las tareas tienen derecho de ejecución y los tiempos de administración son despreciables.
3. El plazo es resultado de los tiempo de arribo anterior y posterior.
4. El tiempo de inicio se considera despreciable.
5. La precedencia de tareas se considera nula.

Sea la tupla  $(\Omega, \mathcal{F}, \Pr)$  el espacio de probabilidad que describe el proceso de una tarea. El conjunto  $\Omega$  arroja sus resultados durante el tiempo de muestra.  $\mathcal{F}$  es el  $\sigma$ -campo sobre  $\Omega$ . Entonces  $\Pr$  es la función de medida de la probabilidad en este espacio. Las restricciones temporales para el proceso estocástico  $\tau$ , (vea Papoulis [46] y Parzen [47]), se asignan de acuerdo con las siguientes definiciones:

**Definición 3.1** Variable Aleatoria

Sea  $(\Omega, \mathcal{F}, \Pr)$  un espacio de probabilidad,  $(S, \mathcal{S})$  un espacio medible y  $X : \Omega \rightarrow S$  una función de medida; entonces  $X$  es una variable aleatoria.

**Definición 3.2** Distribución de Probabilidad

Sea  $(\Omega, \mathcal{F}, \Pr)$  un espacio de probabilidad,  $(S, \mathcal{S})$  un espacio medible y  $X : \Omega \rightarrow S$  una función de medida. La medida de probabilidad  $P : \mathcal{S} \rightarrow \mathbb{R}$ ,  $P(A) = \Pr(X^{-1}(A)) = \Pr\{\omega : X(\omega) \in A\} \forall A \in \mathcal{S}$  es la distribución de  $X$ .

**Teorema 3.1** Espacio de Medida

Sea  $(\mathbb{R}, \mathcal{B})$  un espacio real medible, donde  $\mathcal{B}$  la  $\sigma$ -álgebra de Borel, i.e. la mínima  $\sigma$ -álgebra de todos los conjuntos abiertos. Para toda función real continua por la derecha y creciente  $F : \mathbb{R} \rightarrow \mathbb{R}$  con  $\lim_{x \rightarrow -\infty} F(x) = 0$  y  $\lim_{x \rightarrow \infty} F(x) = 1$ , existe una única medida de probabilidad  $\Pr(F)$ . Que es la extensión sobre los conjuntos de Borel de la única función de medida definida para intervalos semiabiertos y disjuntos, dada por  $P((a, b]) = F(b) - F(a)$ . Al revés, para toda función medida de probabilidad  $P$  en  $(\mathbb{R}, \mathcal{B})$  existe una única función real continua por la derecha y creciente  $F_P : \mathbb{R} \rightarrow \mathbb{R}$  tal que  $\lim_{x \rightarrow -\infty} F_P(x) = 0$ ,  $\lim_{x \rightarrow \infty} F_P(x) = 1$ , y  $F_P(x) = \Pr((-\infty, x]), \forall x \in \mathbb{R}$ .

**Definición 3.3** Variable Aleatoria Continua

Sea  $(\Omega, \mathcal{F}, \Pr)$  un espacio de probabilidad y  $(\mathbb{R}, \mathcal{B})$  un espacio medible, donde  $\mathcal{B}$  son los conjuntos de Borel sobre la línea. Una variable aleatoria  $X : \Omega \rightarrow \mathbb{R}$  es continua si su función de distribución de probabilidades es una función continua. Equivalentemente, sí y sólo sí  $\Pr\{X = x\} = 0$  para toda  $x \in \mathbb{R}$ .

**Definición 3.4** Proceso Estocástico Continuo

Sea  $(\Omega, \mathcal{F}, \Pr)$  un espacio de probabilidad. La familia de variables aleatorias continuas  $\{X(t) : t \in \mathbb{R}^+, t \leq \infty\}$  en algún espacio real medible  $(S, \mathcal{S})$ , es un proceso estocástico continuo.

**Definición 3.5** Función Muestra

Sea  $(\Omega, \mathcal{F}, \Pr)$  un espacio de probabilidad y sea  $\{X(t) : t \in \mathbb{R}^+\}$  un proceso estocástico. Para  $\omega \in \Omega$  fija,  $x_\omega : \mathbb{R}^+ \rightarrow \mathbb{R}$ ,  $x_\omega(t) = X(t, \omega)$  es una trayectoria muestra o una función muestra del proceso estocástico.

Si  $P(X \leq t + u \mid X > t) = P(X \leq u)$  entonces  $X$ , o su distribución, se dice que “no tiene memoria”. Si la distribución de la variable  $X$  es de la forma  $F(t) = 1 - e^{-t/\lambda}$ , entonces  $X$  está exponencialmente distribuida. Por ejemplo, si  $X$  es el tiempo de finalizado de una tarea  $\tau$  y  $X$  no tiene memoria, entonces la probabilidad de que  $\tau$  complete su ejecución antes de un tiempo  $t_2$ , observando que para el tiempo  $t_1$  todavía no ha terminado, es igual a la probabilidad de que  $\tau$  termine en  $t_2 - t_1$  unidades de tiempo. Entonces, si el tiempo de ejecución está distribuido de acuerdo con una distribución sin memoria, la probabilidad de que  $\tau$  termine antes de  $t$  unidades de tiempo es independiente del tiempo que lleva ejecutado.

## 3.2. Principio de Máxima Entropía

En 1948 Claude Shannon [53] publicó su teoría de la comunicación donde derivó una medida de la incertidumbre llamada “Entropía”; refiriéndose a sistemas con entidades físicas o conceptuales. Estos sistemas producen mensajes con significado. Shannon propuso una medida de la cantidad de información “producida” por un proceso. Mejor dicho, la tasa de

velocidad a la que se produce dicha información. En probabilidad, la entropía establece la uniformidad de la función de distribución de una variable aleatoria en un rango.

La entropía es una función de medida de la incertidumbre de que un evento ocurra. La medida sobre un conjunto de todos los posibles eventos cuyas probabilidades de ocurrencia son  $p_1, p_2, \dots, p_n$  se denota por  $H(p_1, p_2, \dots, p_n)$ . La entropía como medida de uniformidad requiere cumplir con las siguientes propiedades:

1.  $H$  debe ser continua en  $p_i$ .
2. Sí todas las  $p_i$  son iguales,  $p_i = 1/n$ , entonces  $H$  es una función creciente monotónica de  $n$ : con eventos igualmente probables existe mayor incertidumbre que cuando se seleccionan eventos con mayor probabilidad.
3. Sí un conjunto de alternativas puede ser reformulado como un conjunto consecutivo múltiple de alternativas, entonces la  $H$  original debe ser la suma ponderada de los valores consecutivos de  $H$ :  $H(p_1, p_2, p_3) = H(p_1, 1 - p_1) + (1 - p_1) H(p_2, p_3)$ .

Estas propiedades significan que la función de entropía de Shannon se comporta de acuerdo con la Teoría de la Información. La función de Shannon representa la incertidumbre esperada [12]. La función de entropía de Shannon es el negativo del valor esperado del logaritmo de la función másica de probabilidad.

**Definición 3.6** Entropía de una Variable Aleatoria Discreta

*La única función que satisface las propiedades establecidas por Shannon [53] es la función:*

$$H(p_1, p_2, \dots, p_n) = -K \sum_{i=1}^n p_i \log p_i \tag{3.1}$$

Sahnnon definió la información en un mensaje como la diferencia entre la entropía antes del mensaje y la entropía después del mensaje. De acuerdo con Shannon, el mensaje produce un nuevo estatuto acerca del estado del sistema y permite la asignación de nuevas probabilidades  $p_i$  que actualizan el valor de la entropía. La forma  $H = -\sum_{i=1}^n p_i \log p_i$  es reconocida como la entropía de las probabilidades  $p_i$  por la similitud matemática con la definición en termodinámica de entropía [24]. La suma de estas probabilidades es uno, por lo que el factor de escala,  $K$ , se considera como la unidad. En la Teoría de la Información para sistemas binarios, se establece el logaritmo con base 2 de acuerdo con la unidad de información en bits. Sin embargo, en términos de probabilidad, permite ajustar la base del logaritmo. Por lo que de acuerdo con Jaynes [24] se tomará el logaritmo natural.

Para variables aleatorias discretas, la medida de incertidumbre absoluta se toma del conjunto de probabilidades correspondientes. Para el caso de variables aleatorias continuas la medida de incertidumbre es una medida relativa al sistema de coordenadas del sistema seleccionado. La entropía, para el caso continuo, es la esperanza matemática de la incertidumbre en distribuciones de probabilidad [49].

**Definición 3.7** Entropía de una Variable Aleatoria Continua

Sea  $X$  una variable aleatoria real definida en  $(\Omega, \mathcal{F}, \text{Pr})$ , la entropía de  $X$  se denota como

$$H(X) = E[-\ln f(X)] = -\int_{-\infty}^{\infty} f(x) \ln f(x) dx \quad (3.2)$$

La entropía de  $X$  se expresa como la esperanza del logaritmo de su función de distribución,  $-\ln f(X)$ . Siguiendo la misma notación, las entropías de dos variables y la entropía condicional entre las variables aleatorias  $X$  y  $Y$  se denotan como:

1.  $H(X, Y) = E[-\ln f(X, Y)],$
2.  $H(Y) = E[-\ln f_1(X)],$
3.  $H(Y) = E[-\ln f_2(Y)],$
4.  $H(X | Y) = E[-\ln f_y(X | Y)],$
5.  $H(Y | X) = E[-\ln f_x(Y | X)],$
6.  $H(X) + H(Y) \geq H(X, Y).$

Para funciones continuas específicas; como la función exponencial (definición 3.10), la entropía conserva una medida de incertidumbre. En otros casos la función de Transinformación retiene el significado de la Teoría de la Información [48], definida a continuación.

**Definición 3.8** Transinformación de dos Variables Aleatorias Continuas

Sean  $X(t, \omega)$  y  $Y(t, \omega)$  dos variables aleatorias continuas en  $(\Omega, \mathcal{F}, \text{Pr})$ , con una función conjunta  $F(x, y)$ . La transinformación entre las variables  $X$  y  $Y$  está definida por:

$$I(X; Y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(x, y) \ln \frac{f(x, y)}{f_1(x)f_2(y)} dx dy \quad (3.3)$$

Expresada como funciones de entropía

$$I(X; Y) = H(X) - H(X | Y) \quad (3.4)$$

Las propiedades elementales de la Transinformación de  $I(X; Y)$  son:

1.  $I(X; Y) = I(Y; X),$

2.  $I(X;Y) \geq 0$ ,
3.  $I(X;Y) = 0$ , si  $X$  y  $Y$  son variables aleatorias independientes,
4.  $I(X;Y) = H(X) - H(X | Y) = H(Y) - H(Y | X)$
5.  $I(X;Y) = H(X) + H(Y) - H(X, Y)$

Otro concepto que expresa la dependencia de una variable aleatoria en función de una variable de restricción es la capacidad de operación definida como:

**Teorema 3.2** Viabilidad de Operación

Sean  $Y(t, \omega)$  y  $X(t, \omega)$  variables aleatorias en  $(\Omega, \mathcal{F}, \text{Pr})$ . Si  $Y$  está sujeta a una función de restricción  $\phi(z)$ ; la viabilidad de operación de  $Y$  en función de la variable  $X$  está descrita por la transformación entre las variables  $Y$  y  $X$  cuando  $f_x(z+x) = \phi(z)$ . Siendo:

$$I(X;Y) = H(Y) + \int_{-\infty}^{\infty} \phi(z) \ln \phi(z) dz \quad (3.5)$$

Expresada como funciones de entropía

$$I(X;Y) = H(Y) - H(Z) \quad (3.6)$$

**Demostración.** Sean  $X$  y  $Y$  variables aleatorias; y sea  $Z$  una variable aleatoria añadida cuya función de densidad es  $\phi(z)$ . Entonces expresando  $f_x(z+x | x) = \phi(z)$  en términos de probabilidad se tiene que  $P\{z_0 < Z < z_0 + dz_0\} = \phi(z_0)dz_0$ . Así mismo la probabilidad condicional de  $Y$  se tiene que  $P\{y_0 < Y < y_0 + dy_0 | X = x_0\} = P\{z_0 < Z < z_0 + dz_0 | X = x_0\}$ ; como  $P\{y_0 < Y < y_0 + dy_0 | X = x_0\} = P\{z_0 < Z < z_0 + dz_0\}$ , y esto es  $\phi(z_0)dz_0$ . Entonces  $f_x(y | x) = f_x(x+z | x) = \phi(z)$ . Expresado en términos de entropía  $H(Y | X) = H(X+Z | X) = H(Z)$ ; por lo que  $I(X;Y) = H(Y) - H(Y | X) = H(Y) - H(Z)$

■

### 3.3. Restricciones Temporales

Los requerimientos temporales para las tareas en tiempo real del tipo suave son aquellos en donde la falta en su cumplimiento conlleva a un detrimento en el desempeño del sistema, más no perjudica al sistema [10]. Esto es, el cumplimiento estadístico de las restricciones temporales es aceptable [34].

**Definición 3.9** Tarea en Tiempo Real del tipo Suave

*Sea  $(\Omega, \mathcal{F}, \Pr)$  un espacio de probabilidad, con  $t \in \mathbb{R}^+$ , y  $\omega \in \Omega$ . Una tarea en tiempo real del tipo suave es un proceso estocástico descrito por la colección  $\tau = \{A(t, \omega), C(t, \omega), D(t, \omega)\}$  caracterizada por la familia de variables aleatorias: Tiempo de arribo  $A(t, \omega)$ , Tiempo de cálculo  $C(t, \omega)$  y Plazo  $D(t, \omega)$ .*

Para la tarea,  $\tau$ , se parte de un modelo simple de línea de espera  $M/M/1$ . Las funciones de densidad definidas a continuación representan las restricciones temporales del modelo bajo análisis. Para estas funciones, la entropía como una medida de uniformidad es desarrollada. A través del método de multiplicadores de Lagrange se designa la función de probabilidad que maximiza la entropía [26]. Cumpliendo los requerimientos de las funciones sin memoria definidas anteriormente. Las restricciones de las variables aleatorias obedecen al comportamiento estadístico desarrollado en la Teoría de Líneas de Espera [28], [45]. Así, la máxima entropía representa la medida de uniformidad para el conjunto de restricciones de las tareas [23].

**Definición 3.10** Variable Aleatoria Exponencial

*Sea  $F : [0, \infty) \rightarrow \mathbb{R}$  la distribución de una variable aleatoria real  $X$ . Si  $F$  es de la forma*

$F(t) = 1 - e^{-t/\lambda}$  entonces  $X$  está exponencialmente distribuida o es una variable aleatoria exponencial.

Para las variables aleatorias exponenciales, las técnicas de Multiplicadores de Lagrange son utilizadas para determinar una medida de certidumbre con base en el principio de máxima entropía [48].

**Definición 3.11** Tiempo de Cálculo

El tiempo de cálculo es una variable aleatoria  $\{C(t, \omega)\}$  definida como el tiempo necesario por el procesador para ejecutar una tarea sin interrupciones. Sea  $f_C(t)$ , la función de densidad para el tiempo de cálculo está sujeta a las siguientes restricciones:

1.  $H(C) = - \int_0^\infty f_C(t) \ln f_C(t) dt$ ,
2.  $\int_0^\infty f_C(t) dt = 1$ ,
3.  $\int_0^\infty t f_C(t) dt = \lambda$ ,
4.  $\lambda > 0$ .

La función de densidad para el tiempo de cálculo es una función del tipo exponencial

$$f_C(t) = \frac{1}{\lambda} e^{-t/\lambda} \tag{3.7}$$

El tiempo de cálculo con media  $\lambda$ , representa la unidad de tiempo como base para las restricciones temporales siguientes. Se enfatiza que la velocidad del procesador da la referencia para la ejecución de las tareas. Siendo las demás restricciones temporales múltiplos de  $\lambda$ .

**Definición 3.12** Tiempo de Arribo

El tiempo de arribo es una variable aleatoria  $\{A(t, \omega)\}$  definida como el tiempo de espera requerido para observar el arribo de una tarea. Esto ocurre a un tiempo de  $K_a \lambda > 0$  unidades de tiempo. También es referido como el tiempo de liberación cuando la tarea está lista para ejecutarse. Sea  $f_A(t)$ , la función de densidad que satisface las siguientes restricciones:

1.  $H(A) = - \int_0^\infty f_A(t) \ln f_A(t) dt$ ,
2.  $\int_0^\infty f_A(t) dt = 1$ ,
3.  $\int_0^\infty t f_A(t) dt = K_a \lambda$ .

La función para el tiempo de arribo es una función del tipo exponencial con tasa  $K_a$

$$f_A(t) = \frac{1}{K_a \lambda} e^{-t/K_a \lambda} \quad (3.8)$$

En la colección de variables aleatorias  $A$  y  $D$  del proceso de tarea  $\tau$ , sus tiempos de arribo  $K_a$  y finalizado  $K_d$  respectivamente, son múltiplos del tiempo de cálculo definido en 3.11 y acotados globalmente. Por lo que se establece:

$$K_a > 1 \quad (3.9)$$

$$K_d > 1 \quad (3.10)$$

$$K_a \geq K_d \quad (3.11)$$

**Definición 3.13** Plazo

El plazo es una variable aleatoria  $\{D(t, \omega)\}$  definida como el tiempo en el cual una tarea debe ser completada para no afectar el desempeño del sistema. La función de densidad del plazo

está descrito por la convolución de la función del tiempo de arribo  $f_{A,i}(t)$  más la función del siguiente arribo  $f_{A,j}(t)$  para  $j > i$ . Sea  $f_D(t)$  la función de densidad del plazo cumple con las siguientes restricciones:

1.  $H(D) = - \int_0^\infty f_D(t) \ln f_D(t) dt,$

2.  $\int_0^\infty f_D(t) dt = 1,$

3.  $f_D(t) = \int_0^x f_{A,i}(t) f_{A,j}(x-t) dt.$

Sí  $K_a = K_d$  la función para el plazo es del tipo gamma

$$f_D(t) = \frac{t e^{-t/K_d \lambda}}{K_d^2 \lambda^2} \quad (3.12)$$

Sí  $K_a > K_d$  la función para el plazo es del tipo exponencial

$$f_D(t) = \frac{e^{-t/K_a \lambda} - e^{-t/K_d \lambda}}{\lambda(K_a - K_d)} \quad (3.13)$$

### **Definición 3.14** Tiempo de Finalizado

El tiempo de finalizado es una variable aleatoria  $\{F(t, \omega)\}$ , definido como el tiempo en que una tarea debe ser completada una vez que está activa; se establece como la convolución entre las densidades del tiempo de cálculo  $f_C$  y el tiempo de arribo  $f_A$ . Sea  $f_F$ , la función de densidad para el tiempo de cálculo sujeta a las siguientes restricciones:

1.  $H(F) = - \int_0^\infty f_F(t) \ln f_F(t) dt,$

2.  $\int_0^\infty f_F(t) dt = 1,$

3.  $f_F(t) = \int_0^x f_A(t) f_C(x-t) dt.$

Con  $K_a > 1$  la función para el tiempo de finalizado es del tipo exponencial

$$f_F(t) = \frac{e^{-t/K_a\lambda} - e^{-t/\lambda}}{\lambda(K_a - 1)} \quad (3.14)$$

Se ha establecido para las funciones de probabilidad del proceso de tareas, que la entropía es una medida de la uniformidad de la distribución sobre el rango de tiempo de cada variable aleatoria de la colección  $\tau$  [26]. De acuerdo con las definiciones 3.12 y 3.13, se desarrolla el valor de medida de máxima entropía para estas funciones de probabilidad, a partir del método de multiplicadores de Lagrange.

**Teorema 3.3** Máxima Entropía del Tiempo de Arribo

Sea  $f_A(t)$  la función para el tiempo de arribo de una tarea sujeta a las siguientes restricciones:

1.  $H(A) = - \int_0^\infty f_A(t) \ln f_A(t) dt$ ,
2.  $\int_0^\infty f_A(t) dt = 1$ ,
3.  $\int_0^\infty t f_A(t) dt = K_a \lambda$ .

La máxima entropía de la función  $f_A(t)$  con un primer momento  $K_a \lambda$  especificado en  $t[0, \infty)$  es  $H_{A,i}$  dada por

$$H_{A,i} = H(A) = \ln \lambda + \ln K_a + 1 \quad (3.15)$$

**Demostración.** Para el proceso estocástico  $\tau$ , la función  $f_A(t)$  está definida como una función del tipo exponencial de acuerdo con Lehoczky [30] y Lin [33]. Utilizando el método de multiplicadores de Lagrange, se desarrolla una función de densidad que satisface las restricciones 1, 2 y 3 de la definición 3.12. De acuerdo con Lagrange la suma de las derivadas

parciales de estas funciones debe ser cero;  $\frac{\partial}{\partial f}(-f_A \ln f_A) + \alpha \frac{\partial}{\partial f}(f_A) + \beta \frac{\partial}{\partial f}(t f_A) = 0$  con coeficientes  $\alpha$  y  $\beta$ . Lo que resulta en  $-(1 + \ln f_A) + \alpha + \beta t = 0$ , de donde  $f_A(t) = e^{\alpha-1+\beta t} = e^{\alpha-1} e^{\beta t}$ . Para determinar los coeficientes  $\alpha$  y  $\beta$  se evalúa  $f_A(t)$  en las restricciones 2 y 3 del teorema 3.3. De donde se obtiene  $\beta = -\frac{1}{K_a \lambda}$  y  $e^{\alpha-1} = -\frac{1}{K_a \lambda}$ . Entonces  $f_A(t) = \frac{1}{K_a \lambda} e^{-t/K_a \lambda}$ . Entonces, una función que maximiza la entropía es la función exponencial con parámetro  $K_a \lambda$ . La máxima entropía para la función  $f_A$  se obtiene de la restricción 1 del teorema 3.3. Siendo  $H_M(f_A(t)) = -\int_0^\infty \frac{1}{K_a \lambda} e^{-t/K_a \lambda} \ln \frac{1}{K_a \lambda} e^{-t/K_a \lambda} dt$ , así  $H_M(f_A(t)) = \ln K_a \lambda \int_0^\infty \frac{1}{K_a \lambda} e^{-\frac{1}{K_a \lambda} t} dt + \frac{1}{K_a \lambda} \int_0^\infty \frac{t}{K_a \lambda} e^{-\frac{1}{K_a \lambda} t} dt$ . Entonces, realizando la integración correspondiente se obtiene finalmente  $H_M(A) = \ln \lambda + \ln K_a + 1$ . ■

### Teorema 3.4 Entropía del Plazo

Sea  $f_D(t)$ , la función de densidad que describe el plazo de una tarea, sujeta a las siguientes restricciones:

1.  $H(D) = -\int_0^\infty f_D(t) \ln f_D(t) dt$
2.  $\int_0^\infty f_D(t) dt = 1$
3.  $f_D(t) = \int_0^x f_{A, i}(t) f_{A, j}(x-t) dt$

La entropía de la función  $f_D(t)$  con primer momento especificado en  $t[0, \infty)$  es  $H_{D, i}$  expresada por

$$H_{D, i} = H(D) = \begin{cases} \gamma + \ln \lambda + \ln K_d + 1, & K_a = K_d \\ \gamma + \ln \lambda + (K_a - K_d) - \frac{1}{2} \ln(K_a - K_d), & K_a > K_d \end{cases} \quad (3.16)$$

**Demstración.** Dado que el plazo está definido como la convolución de dos funciones de arriba consecutivas, su expresión depende de los parámetros de las funciones  $K_a$  y  $K_d$  respectivamente en  $f_D(t) = \int_0^x f_{A, i}(t) f_{A, j}(x-t) dt$ . La función  $f_D(t)$  cumple con la restricción

de densidad 2, de la definición 3.13. Para el caso en que  $K_a = K_d$ , la restricción 1 del teorema 3.4, produce la función  $f_D(t) = \frac{t}{K_d^2 \lambda^2} e^{-t/K_d \lambda}$ . Aplicando la función de entropía dada por la restricción 3, la función de entropía resulta en:  $H(D) = \gamma + \ln \lambda + \ln K_d + 1$ . Con esta misma secuencia se resuelve el caso en que  $K_a > K_d$ , donde  $f_D(t) = \frac{1}{\lambda(K_d - K_a)} (e^{-t/K_d \lambda} - e^{-t/K_a \lambda})$ . Con lo que la expresión de entropía resulta en:  $H(D) \approx \gamma + \ln \lambda + (K_a - K_d) - \frac{1}{2} \ln(K_a - K_d)$

■

### **Teorema 3.5** Entropía del Tiempo de Finalizado

Sea  $f_F(t)$  la función de densidad del tiempo de finalizado de una tarea sujeta a las siguientes restricciones:

1.  $H(F) = - \int_0^\infty f_F(t) \ln f_F(t) dt$ ,
2.  $\int_0^\infty f_F(t) dt = 1$ ,
3.  $f_F(t) = \int_0^x f_A(t) f_C(x - t) dt$ .

La entropía de la función  $f_F(t)$  está dada por

$$H_{F, i} = H(F) = \gamma + \ln \lambda + K_a - \frac{1}{2} \ln(K_a - 1) \quad (3.17)$$

**Demostración.** El tiempo de finalizado, definido como la convolución de las funciones de tiempo de arribo y tiempo de cálculo expresada en la restricción 3 de la definición 3.14. La función de densidad resultante cumple con la restricción 2 y está descrita por  $f_F(t) = \frac{1}{\lambda(K_a - 1)} (e^{-t/K_a \lambda} - e^{-t/\lambda})$ . Aplicando la función de entropía, dada por 3.14, sobre la función  $f_F$ ,  $H(F) = - \int_0^\infty f_F(t) \ln f_F(t) dt$  se obtiene  $H(F) = \gamma + \ln \lambda + K_a - \frac{1}{2} \ln(K_a - 1)$ . ■

Se establece para una variable aleatoria del proceso  $\tau$  una medida de uniformidad descrita por la máxima entropía. Donde los parámetros a observar son los tiempo de arribo  $K_a$  y plazo  $K_d$ .

### 3.4. Asignación de Prioridad

En la sección anterior se estableció un criterio de uniformidad en un sentido de probabilidad para las tareas con restricciones temporales estocásticas. Con el que se relaciona la función de densidad con una medida de su distribución; siendo mayor para variables aleatorias con parámetros  $K_a$  o  $K_d$  respecto a la media de referencia  $\lambda$ .

Desde los primeros trabajos en el área de la calendarización se han establecido dos criterios en la asignación de prioridad para las tareas computacionales. Estos son el tiempo de arribo y el plazo. En función de estos tiempos se han establecido criterios para manejar las tareas como periódicas o esporádicas.

Una propiedad principal de una tarea es su periodo. En su trabajo, pionero en esta área, Liu y Lyland [34], definieron el periodo como el tiempo entre tareas sucesivas  $p_i$ ; considerándolo constante. Una tarea periódica se dice que tiene un tiempo de liberación regular, o es regularmente activada. Si una tarea no cumple con este criterio es llamada tarea no periódica o esporádica. La periodicidad de una tarea juega un rol relevante en el diseño de algoritmos de calendarización debido a las consideraciones hechas para su desarrollo; en función de obtener los resultados esperados [32]. La periodicidad se considera como una medida del orden temporalmente hablando. Una tarea con mayor prioridad, o jerarquía, corresponde a una tarea con mayor periodicidad. Lo que está definida en su función de distribución.

La propiedad del plazo es considerada en los planificadores dinámicos como el algoritmo EDF [54]. De acuerdo con la utilización del procesador, la prioridad de la tarea es determinada por el plazo absoluto. La tarea con el plazo más corto tendrá la prioridad mayor.

**Definición 3.15** Máxima Jerarquía de Tareas en un sentido de Probabilidad

Sea  $J_i = J(\tau(t, \omega))$  el parámetro de una tarea que describe la prioridad relativa entre las tareas que conforman un sistema de calendarización de tareas  $\tau$ . El parámetro de jerarquía de una tarea está definido como:

$$\text{máx}\{J_i\} = \begin{cases} \text{mín}\{H_{A,i}\} \\ \text{mín}\{H_{D,i}\} \end{cases} \quad (3.18)$$

**Teorema 3.6** Jerarquía de Tareas en base al criterio de Máxima Entropía

Sea  $\tau_i(t, \omega)$  una tarea con restricciones temporales establecidas por la familia de variables aleatorias  $\{A_i(t, \omega), C_i(t, \omega), D_i(t, \omega)\}$ . La jerarquía  $J_i$  de la tarea  $\tau_i$  respecto a la jerarquía  $J_j$  de la tarea  $\tau_j$ , de acuerdo con el principio de máxima entropía, se establece por:

$$J_i > J_j \quad \text{sí y solo sí} \quad H_{M,i} < H_{M,j} \quad (3.19)$$

**Demostración.** De los resultados establecidos por los teoremas 3.3 y 3.4; para la máxima entropía de una tarea como medida de la uniformidad en su función de distribución; de acuerdo con la definición 3.15, la tarea con mayor jerarquía es aquella con menor medida de máxima entropía. Así para dos tareas  $\tau_i$  y  $\tau_j$  la prioridad queda establecida en función de las entropías respectivas  $H_{M,i}$  y  $H_{M,j}$ . ■

### 3.5. Resumen del Capítulo

En este capítulo se fundamenta el modelo estocástico para las tareas en tiempo real del tipo suave consideradas. Se establece aquí la base para la asignación de prioridad entre tareas que han de convivir en un sistema de calendarización. La técnica aplicada permite trabajar con tareas de índole diverso para los casos de análisis sobre tiempos de arribo o plazos a través de sus funciones de distribución.

A partir del trabajo de Shannon se establece una medida de distinción entre tareas. Con base en el nivel de máxima entropía como elemento diferenciador de tareas, el planteamiento presentado analiza los procesos de tareas en forma dinámica.

# Capítulo 4

## Modelo Probabilístico de Planificador de Tareas

El problema general de la calendarización es asignar recursos disponibles en un periodo de tiempo para ejecutar un conjunto de tareas, en particular el procesador; así como tener alguna medida del desempeño (vea [14] y [10] para una introducción a la teoría de la calendarización). La meta de la calendarización de tareas en tiempo real del tipo suave es cumplir con el plazo estadísticamente para minimizar las amonestaciones; asegurando un buen desempeño del sistema.

De acuerdo con las consideraciones presentadas anteriormente; en este capítulo se desarrolla un modelo de planificador dinámico para tareas en tiempo real del tipo suave con base en el “Planificador Estocástico operado por el Plazo” (PEP). El cual, calcula las prioridades de las tarea durante la ejecución del sistema. Aquí, la prioridad de cada tarea depende del estado del sistema y de las características de la tarea. Es decir la medida de uniformidad dada por

la máxima entropía de las variables aleatorias que conforman las restricciones temporales del proceso de tarea. Para los casos analizados Así mismo, se establece un criterio de optimalidad para los casos analizados [60].

## 4.1. Planeación de Tareas en Tiempo Real

En un sistema de tiempo real del tipo suave donde convergen tareas que requieren ser atendidas; la política de calendarización debe de establecerse a modo de lograr un desempeño adecuado del sistema.

### Definición 4.1 Calendarización de Tareas

Sea  $\tau_i$ , una tarea en tiempo real. Dado un conjunto de tareas  $\tau = \{\tau_1, \tau_2, \dots, \tau_n \mid n \in \mathbb{Z}^+\}$ , la calendarización de estas tareas es una relación  $\sigma(t) : \mathbb{R}^+ \rightarrow \mathbb{Z}^+$  que determina la actividad del procesador de manera temporal.

### Definición 4.2 Algoritmos con Manejo de Prioridad

Un algoritmo de calendarización es un algoritmo con manejo de prioridad sí satisface la condición de que para todo par de tareas  $\tau_i$  y  $\tau_j$ , se cumple en algún instante de tiempo  $J_i > J_j$ , sí y solo sí  $\tau_i$  tiene una mayor prioridad que  $\tau_j$ . Ha y Liu [22].

## 4.2. Modelo de Planificador Estocástico por Plazo

Presentado el planificador de mayor popularidad llamado EDF; también denominado “Planificador del Plazo más Corto” (EDS), el algoritmo de prioridad dinámica tiene como base de su política el plazo más corto. Este algoritmo especificado básicamente por Liu y

Layland [34]; está definido como algoritmo de planificación controlado por el plazo y es un algoritmo de calendarización dinámico que otorga (en todo instante) la prioridad mayor (por lo tanto el uso del procesador) a la tarea activa con el plazo más corto.

Establecida la entropía como una función de medida de dispersion sobre las funciones que componen una tarea. Para un proceso de tarea definido en 3.9; la medida propuesta cumple con las propiedades de Máxima Entropía de acuerdo con Shannon [53]. La base temporal de referencia para las funciones de la colección  $\tau$  fue el tiempo de cálculo  $\lambda$  expresado en la definición 3.11. Aquí los parámetros  $K_a$  y  $K_d$  se contemplaron en función de la base de tiempo  $\lambda$ . a partir de las ecuaciones 3.9, 3.10, y 3.11; considerando ahora una variación de este tiempo expresado por la tasa  $K_c\lambda$  la cual está acotada por:

$$K_d \geq K_c \geq \lambda \quad (4.1)$$

**Teorema 4.1** Máxima Entropía del Tiempo de Cálculo

Sea  $f_C(t)$  la función para el tiempo de cálculo de la tarea cuya máxima entropía con primer momento acotado por  $K_c\lambda$  especificado para  $t \in [0, \infty)$ ; su máxima entropía está dada por

$$H_C = H(C) = \ln \lambda + \ln K_c + 1 \quad (4.2)$$

**Demostración.** Para el proceso estocástico  $\tau = \{\tau_i | \in \mathbb{Z}^+\}$ , la función  $f_C(t)$  es una función del tipo exponencial con las siguientes propiedades

1.  $H(C) = - \int_0^\infty f_C(t) \ln f_C(t) dt$ ,
2.  $\int_0^\infty f_C(t) dt = 1$ ,
3.  $\int_0^\infty t f_C(t) dt = K_c\lambda$ ,

4.  $K_d \geq K_c \geq 1$ .

tal que al utilizar los multiplicadores de Lagrange se obtiene la función de densidad en relación a  $f_C(t)$  por lo que  $\frac{\partial}{\partial f}(-f_C \ln f_C) + \alpha \frac{\partial}{\partial f}(f_C) + \beta \frac{\partial}{\partial f}(f_C t) = 0$  que se reduce a  $-(1 + \ln f_C) + \alpha + \beta t = 0$ , de donde  $f_C(t) = e^{\alpha-1+\beta t} = e^{\alpha-1} e^{\beta t}$ . Para determinar los coeficientes  $\alpha$  y  $\beta$  se evalúa  $f_C(t)$  en las restricciones 2 y 3. Con lo que se obtiene que  $\beta = -\frac{1}{K_c \lambda}$  y  $e^{\alpha-1} = -\frac{1}{K_c \lambda}$ . Entonces  $f_C(t) = \frac{1}{K_c \lambda} e^{-t/K_c \lambda}$ . La máxima entropía para la función  $f_C$  se obtiene de 1. Siendo  $H_M(f_C(t)) = -\int_0^\infty \frac{1}{K_c \lambda} e^{-t/K_c \lambda} \ln \frac{1}{K_c \lambda} e^{-t/K_c \lambda} dt$ , así  $H(C) = \ln K_c \lambda \int_0^\infty \frac{1}{K_c \lambda} e^{-\frac{1}{K_c \lambda} t} dt + \frac{1}{K_c \lambda} \int_0^\infty \frac{t}{K_c \lambda} e^{-\frac{1}{K_c \lambda} t} dt$ . Realizando la integración correspondiente se obtiene 4.2. ■

Del teorema 4.1 se observa que en 4.2 la entropía añade el parámetro  $K_c$  requerido para  $\beta$  y  $\alpha$ . En la relación 4.1, sí  $K_c = 1$ , el resultado coincide con la máxima entropía para un tiempo de cálculo unitario. En la expresión para la medida de incertidumbre del tiempo de finalizado descrito en 3.5; considerando que  $K_a > K_c \geq 1$ , la función  $H(F)$  se redescrive como

$$H(F) = \gamma + \ln \lambda + (K_a - K_c) - \frac{1}{2} \ln(K_a - K_c) \quad (4.3)$$

A partir de la definición 3.8 se tiene que para una tarea con funciones  $f_A$ ,  $f_D$  y  $f_C$  conocidas, la viabilidad de operación de la tarea se puede definir de la siguiente manera:

**Teorema 4.2** Viabilidad de Operación de una Tarea

*Sean  $D(t, \omega)$  y  $A(t, \omega)$  variables aleatorias de la colección  $\tau$  con funciones de densidad conocidas. Sí la función de densidad  $f_D(d_\omega)$  está sujeta a la restricción  $f_C(t, c_\omega)$ , entonces la viabilidad de operación para  $D(t, \omega)$  en función de  $A(t, \omega)$  se describe por la transformación  $I(A; D)$ , cuando  $f_A(c_\omega + a_\omega) = f_C(c_\omega)$  es decir:*

$$I(A; D) = H(D) - H(C) \quad (4.4)$$

**Demostración.** Sean  $D(t, \omega)$  y  $A(t, \omega)$  variables aleatorias del proceso  $\tau$ ; y sea  $C(t, \omega)$  una variable aleatoria independiente añadida estadísticamente tal que  $D = A + C$ . La función de densidad de  $C$  es  $f_C(c_\omega)$ . Expresando  $f_A(c_\omega + a_\omega | a_\omega) = f_C(c_\omega)$  en términos de probabilidad. Se escribe  $P\{c_0 < C < c_0 + dc_0\} = f_C(c_0) dc_0$ . Así mismo la probabilidad condicional de  $D$  en  $A$  es  $P\{d_0 < D < d_0 + dd_0 | A = a_0\} = P\{c_0 < C < c_0 + dc_0 | A = a_0\}$ . Como  $P\{d_0 < D < d_0 + dd_0 | A = a_0\} = P\{c_0 < C < c_0 + dc_0\}$ ; y esto es  $f_C(c_0) dc_0$ . Entonces  $f_a(d | a) = f_a(a + c | a) = f_C(c_\omega)$ . Expresado en términos de la medida de entropía  $H(D | A) = H(D + C | A) = H(C)$ ; por lo que  $I(A; D) = H(D) - H(D | A) = H(D) - H(C)$ . Finalmente expresando la transinformación en términos de las funciones de densidad se cumple 4.4. ■

#### 4.2.1. Viabilidad

Consideremos  $n$  procesos de tareas  $\tau = \{\tau_1, \tau_2, \dots, \tau_n | \in \mathbb{Z}^+\}$  siendo atendidas por un único procesador. Ordenando las tareas de acuerdo con la definición 3.15. Se tiene que  $\tau_i$  tiene una prioridad mayor a  $\tau_j$  sí y solo sí  $H_{M, i} < H_{M, j}$ , como lo establece el teorema 3.6. Partiendo de este hecho, la tarea con  $J_i$  se ejecutará primero, seguida de la siguiente con prioridad inmediata inferior. Sí se toma el tiempo de ejecución unitario con una media de  $\lambda$ , la tarea que se ejecuta en segundo lugar tendrá que comenzar en  $\lambda$  unidades después; y así sucesivamente. Así para la  $i$ -ésima tarea, se tiene:

#### **Teorema 4.3** Viabilidad de una Tarea

*El proceso de tarea  $\tau_i$  para ser atendido por un procesador su viabilidad de que el proceso de la tarea sea planificable en un tiempo  $t \in [0, \infty)$  está dada por la relación*

$$I(A_i; D_i) \geq 0. \tag{4.5}$$

**Demstración.** Considerando a 4.4 donde  $I(A; D)$  está descrita por la diferencia entre  $H(D)$  y  $H(D | A)$  que se reduce a la forma  $H(D) - H(C)$ , tal que se al cumplirse que  $K_a > K_c \geq 1$  de acuerdo con 4.3 se cumple 4.5. ■

### 4.3. Optimalidad

Liu y Layland describieron varias de las principales propiedades para los algoritmos de calendarización del tipo EDF; sus principales resultados conciernen a la optimalidad de este planificador. Los resultados dados por Liu y Layland se basan en el hecho que, para tareas del tipo síncrono, el análisis para el “peor caso” (WCET) debe ser contemplado. Al cual se le llama óptimo. Ellos también probaron que el EDF es el algoritmo de asignación de prioridad óptimo. Optimalidad, en el contexto de calendarización, significa que si existe un método de asignación de prioridad en línea; o fuera de línea, donde las tareas son planificables, entonces estas tareas también son planificables por el método EDF.

Según Liu y Layland, el sentido de optimalidad para sistemas de calendarización controlados por el plazo se distingue en dos tipos de optimalidad.

#### **Definición 4.3** Optimalidad Fuerte

*Una regla de prioridad dinámica es óptima en un sentido fuerte para una familia de tareas  $\{\tau_i\}$ , si la asignación de prioridad es dinámica entonces es factible para cualquier calendarización, sin importar la manera en que las ambigüedades sean resueltas.*

#### **Definición 4.4** Optimalidad Débil

*Una regla de prioridad dinámica es óptima en un sentido débil para una familia de tareas*

$\{\tau_i\}$ , sí por lo menos alguna calendarización es resuelta, entonces es factible para el conjunto de tareas en donde de alguna manera son resueltas las ambigüedades.

Liu y Layland no probaron explícitamente que el caso de calendarización síncrono es el WCET desde un punto de vista de planificadores para sistemas con plazo tardío. Goossens [19] establece que para un caso general, es aplicable el sentido de optimalidad para sistemas asíncronos.

**Teorema 4.4** Calendarización de Tareas Asíncronas

Sea  $S = \{\tau_i = \{C_i, D_i, T_i\} \mid i = 1, \dots, n\}$ , con plazos arbitrarios. Sí  $S$  es planificable en el caso síncrono utilizando el planificador manejado por plazo; también es planificable para situaciones del caso asíncrono. Goossens.

En el presente trabajo la base del método de calendarización es el análisis de probabilidad en función del caso “con mayor certidumbre” como parámetro de estudio. Estableciendo la optimalidad de un planificador en función directa con la manera como el planificador maneja la calendarización de todas y cada una de las tareas que atiende. En el trabajo de Goossens [19] se establece que el planificador de tareas asíncronas controlado por el plazo es óptimo en un sentido fuerte. Donde se estable un plazo arbitrario para tareas asíncronas. Es decir, para tareas donde se cumple que  $t_d \leq T$ . Siendo  $t_d$  es el tiempo del plazo y  $T$  es el periodo.

**Teorema 4.5** Calendarización de Tareas Estocásticas por Plazo

Sea  $S = \{\tau_i = \{A_i, C_i, D_i\} \mid i = 1, \dots, n\}$ , el conjunto de tareas  $\tau_i$  ordenado de acuerdo con  $H_{A, i}$ .  $S$  es planificable por plazo sí y solo sí para toda tarea  $\tau_i$  se cumple

$$I_i(A_i; D_i) \geq 0, \tag{4.6}$$

$$H(D_i) - H(F_i) \geq 0 \quad (4.7)$$

**Demostración.** Sea  $\{\tau_i\} \in S$ ; sí para cada tarea se cumple la condición del teorema 4.3, y el conjunto de tareas ha sido ordenado según el teorema 3.6; entonces se tiene que para la tarea  $\tau_i$ , la variable aleatoria para tiempo de finalizado  $F_i$  establecida por el teorema 4.1, está definido por  $F_i = A_i + \sum_{k=1}^i C_k$ . Cuando la tarea es viable con  $K_d = K_a$ , el plazo se define como  $D_i = 2 A_i$ . Entonces  $H(D_i) - F(F_i)$  implica que,  $2 A_i - (A_i + \sum_{k=1}^i C_k) = A_i + \sum_{k=1}^i C_k \geq 0$ . Entonces, como se establece en el teorema 4.2,  $I_i(A_i; D_i) \geq 0$  implica 4.7.

■

#### **Corolario 4.1** Optimalidad del Planificador de Tareas Estocásticas por Plazo

Sea  $S = \{\tau_i = \{A_i, C_i, D_i\} \mid i = 1, \dots, n\}$  un conjunto de tareas ordenado de acuerdo con  $H_{A, i}$ , Entonces el planificador de tareas estocásticas por plazo es óptimo en un sentido fuerte de acuerdo con la ecuación 4.4.

**Demostración.** Consideremos la existencia de un algoritmo de calendarización  $Q$  tal que el sistema de tareas es planificable en él. Sea  $\sigma$  la relación de calendarización viable producida por el planificador  $Q$  en el sistema. Debemos de observar que cualquier calendarización  $\sigma^i \subset \sigma$  producida por el planificador de tareas estocásticas por plazo es también viable, de acuerdo con el teorema . Por lo que por inducción debemos observar que siempre es posible transformar la calendarización  $\sigma^t$  en el intervalo  $[0, t)$  resultante en  $\sigma^t(t)$ , así mismo como con  $\sigma^1(t)$ . Permaneciendo  $\sigma^t(t)$  viable para el mismo conjunto de tareas. Considerando que  $\sigma^t(t) = \sigma^1(t)$  entonces  $\sigma^{t+1} = \sigma^1$ . Por el contrario sí  $\sigma^t(t) = 0 \neq \sigma^1(\lambda) = (i, k)$  para un primer instante  $\lambda$  después de  $t$ , tenemos que  $\sigma^{t+1}(e) = 0$ . Entonces no es posible tener que  $\sigma^t(t) \neq 0 = \sigma^1(t)$  dado que  $\sigma^1(t) = 0$  significa que no existe una petición activa al tiempo  $t$  en  $\sigma^1$  como en  $\sigma^t$ . Considerando el planificador de tareas estocásticas controlado

por plazo y suponiendo que la solicitud de la tarea  $\tau_i = (\delta_i^k)$  se ejecuta al tiempo  $t$  en  $\sigma^t$ , (i.e.  $\sigma^t(t) = (i, k)$ ); mientras que otra solicitud activa de la tarea  $\tau_j = (\delta_j^{k'})$  con un plazo mayor (o igual), se ejecuta en  $\sigma^1$ , (i.e.  $\sigma^1(t) = (i, k')$ ), con  $D_i$  y  $D_j$  los plazos para las solicitudes  $\delta_i^k$  y  $\delta_j^{k'}$ , respectivamente. Por la definición 3.15, tenemos que  $H(D_i) \leq H(D_j)$ , tal que la solicitud  $\delta_i^k$  debe ejecutarse al menos una unidad de tiempo  $\lambda_i$  antes que  $\delta_j^{k'}$ , en  $\sigma^t$  al tiempo  $\lambda_i$ , (i.e.  $\sigma^{t'+1}(\lambda_i) = (i, k)$ ), donde  $\lambda_i$  es la primer unidad de tiempo después del tiempo  $t$  en  $\sigma^t$  asignado a la solicitud  $\delta_j^k$ . Esto es valido, siendo que  $\delta_j^k$  está todavía activa al tiempo  $t$  en  $\sigma^{t'+1}$ , y  $H(C_i) < H(D_i) \leq H(D_j)$  por el teorema 4.3. La calendarización resultante  $\sigma^{t'+1} = \sigma^1$  en otra parte, permanece viable en el intervalo  $[0, t' + \lambda_i)$  que es la calendarización producida por  $\sigma^1$ . Consecuentemente, la calendarización  $\sigma$  es viable en  $\sigma^1$ .

■

#### 4.4. Resumen del Capítulo

En el presente capítulo se estableció el modelo del Planificador Estocástico por Plazo PEP, para tareas con restricciones temporales estocásticas. Para esto fue necesario plantear los principios de viabilidad y optimalidad que rigen el modelo. A partir del algoritmo establecido de calendarización de tareas controlado por plazo, EDF. Así mismo se plantea una optimalidad del tipo fuerte para el algoritmo PEP. Sustentado en el teorema de la transinformación, el modelo planteado es óptimo para un conjunto de tarea planificables.

En este mismo sentido se desarrollan las definiciones formales para el algoritmo de calendarización propuesto. Este algoritmo se aplicará a un algoritmo de simulación; analizando sus resultados en el siguiente capítulo.

# Capítulo 5

## Algoritmo de Planificación Estocástico por Plazo

Una manera de vislumbrar la aplicación del algoritmo a desarrollar es a través de la simulación de trayectorias para los casos previstos. En el presente capítulo se presentan los resultados del trabajo de simulación obtenidos. El modelo de calendarización planteado en el capítulo anterior establece dos análisis para su construcción:

1. Asignación de la prioridad de las tareas del sistema de acuerdo con el principio de máxima entropía  $H_M$ .
2. Calendarización de las tareas por la regla del plazo más corto de acuerdo con la viabilidad de la tarea  $I(A_i; D_i)$ .

Para un número de  $n$  tareas que operan en un sistema con un procesador. A un tiempo  $t$  se de observan los tiempos en los que la tarea  $\tau_i$  llega al procesador; además de otros

estadísticos. Con esto se calcula  $H_{M, i}$  (vea el teorema 3.6). Junto con el conjunto  $S$  dado por  $S = \{\tau_i = \{A_i, C_i, D_i\} \mid i = 1, \dots, n\}$  ordenado creciente en  $i$ .

Sea  $\tau_i$ , si  $S$  está ordenado de manera ascendente de acuerdo con  $H_{M, i}$ , significa que  $J_i > J_j$  (con  $i < j$ ). Entonces, el tiempo requerido para que la tarea  $\tau_i$  concluya depende de que se ejecuten  $i - 1$  tareas antes que ella; siempre y cuando se garantice su viabilidad,  $I_i = I(A_i, D_i) \geq 0$ .

### **Ejemplo 5.1** Ejemplo de Análisis

*Consideremos  $S$ , como un conjunto de tres tareas  $\{\tau_1, \tau_2, \tau_3\}$  las cuales tienen una tasa de arribo  $K_{a,1} = 5,0$ ,  $K_{a,2} = 5,1$  y  $K_{a,3} = 5,2$ . La prioridad de las tareas se establece como:  $J_3 = J(\tau_1) = H(A_1) = \ln(5,0\lambda e)$ , así mismo  $J_2 = J(\tau_2) = H(A_2) = \ln(5,1\lambda e)$  y  $J_1 = J(\tau_3) = H(A_3) = \ln(5,2\lambda e)$ .*

*Sean los tiempos de plazo establecidos por  $K_{d,i} = 2K_{a,i}$ , por lo que  $K_{d,1} = 10$ ,  $K_{d,2} = 10,2$  y  $K_{d,3} = 10,4$  respectivamente. Además, consideremos los tiempos de cálculo de 5 unidades. Es decir,  $K_{c,1} = K_{c,2} = K_{c,3} = 5\lambda$ .*

Teniendo un planteamiento simple dado por el ejemplo 5.1, se analiza el algoritmo construido para visualizar de una manera sencilla los resultados obtenidos.

## **5.1. Construcción del Algoritmo de Calendarización**

Para el sistema de tareas  $S$ , se desarrolla una política de calendarización controlado por plazo  $\sigma$ , que cumple con los parámetros de desempeño.

$S = 0$

**Mientras:**  $S < S_{MAX}$

$J = \{\emptyset\}$

**Para:**  $i = 1$  a  $n$

Registra  $\tau_i = \{K_{a,i}, K_{c,i}, K_{d,i}\}$

$\tau = \{\tau_1, \tau_2, \dots, \tau_n\}$

**Para:**  $i = 1$  a  $n$

Asigna prioridad  $J_j = \min \tau(H(A_i))$

Actualiza  $J = \{J\} + \tau_j$

Actualiza  $\tau = \{\tau\} - \tau_j$

$C_0 = 0$

$i = 1$

**Mientras:**  $i < n$

**Sí:**  $I(D_i; A_i) - H(C_0) > 0$

Calcula  $I(D_i; A_i) - H(C_0)$

Actualiza  $C_0 = C_0 + C_i$

**No:**

Intercambia  $J_i$  con  $J_{i-1}$

$i = i - 2$

Actualiza  $i = i + 1$

$S = S + 1$

**Fin**

**Figura 5.1. Construcción del algoritmo planificador estocástico por plazo**

Para el análisis del algoritmo de calendarización estocástico por plazo, se considera el ejemplo de estudio 5.1. A través de una metodología similar a la que presenta Manolache en [38]. En la Figura 5.1 se presenta la propuesta del algoritmo de calendarización. En él se observa la asignación previa de prioridades  $J_i$  para las  $n$  tareas.

En la segunda parte del algoritmo se establecen los requerimientos que deben cumplir las tareas para ser planificadas. En caso de no cumplirse  $I(A_i; D_i) \geq 0$  el algoritmo modifica la jerarquía entre las tareas restantes y por consiguiente el orden de su planificación [61].

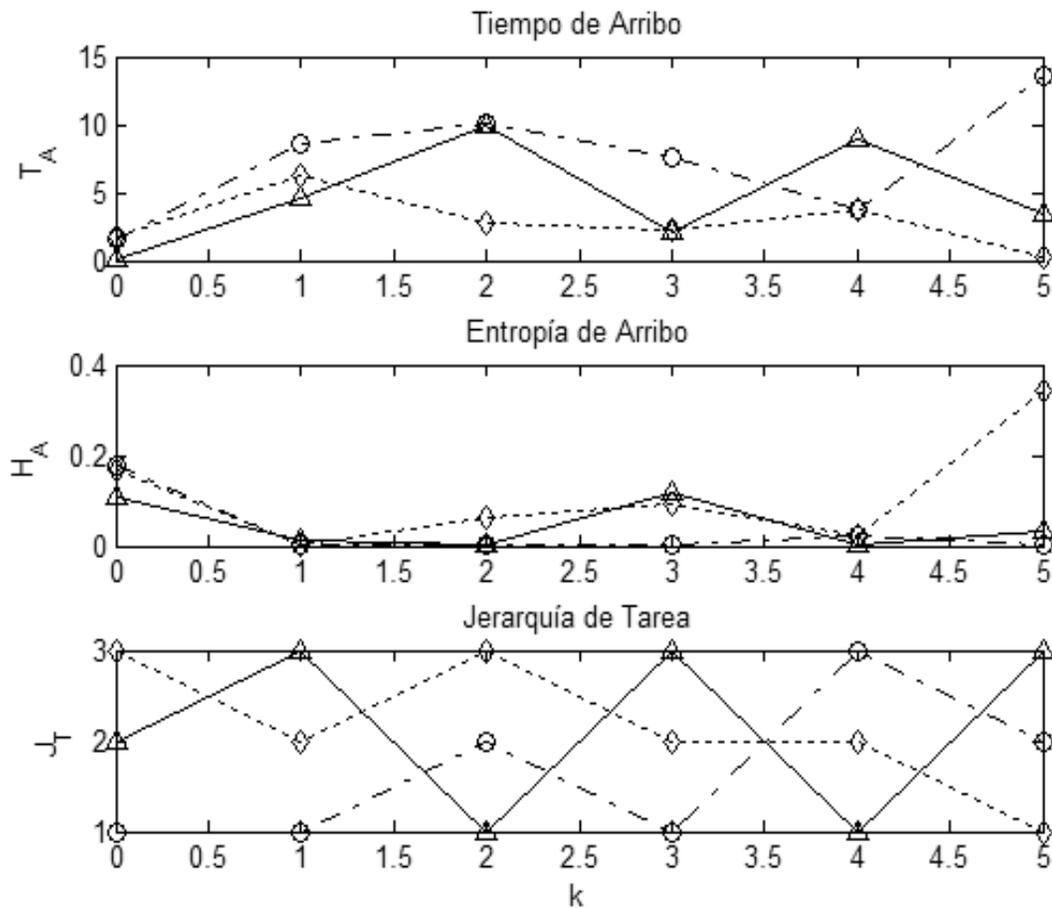
## 5.2. Experimentación y Resultados

Para efectos de claridad, se considera la unidad de tiempo  $\lambda = 1[u.d.t]$ , por lo que las tasas para las restricciones temporales del ejemplo 5.1 quedarían simplemente como  $K_{a,i}$ ,  $K_{c,i}$  y  $K_{d,i}$  respectivamente.

Se toma ahora un tiempo de simulación para visualizar la asignación de prioridades. En la Figura 5.1, se observan los valores del cálculo de la entropía correspondiente a los valores de arriba de las tareas que conforman el sistema  $S$ .

- $\tau_1$ , símbolo ( $\Delta$ ),
  
- $\tau_2$ , símbolo ( $\diamond$ ),
  
- $\tau_3$ , símbolo ( $\circ$ ).

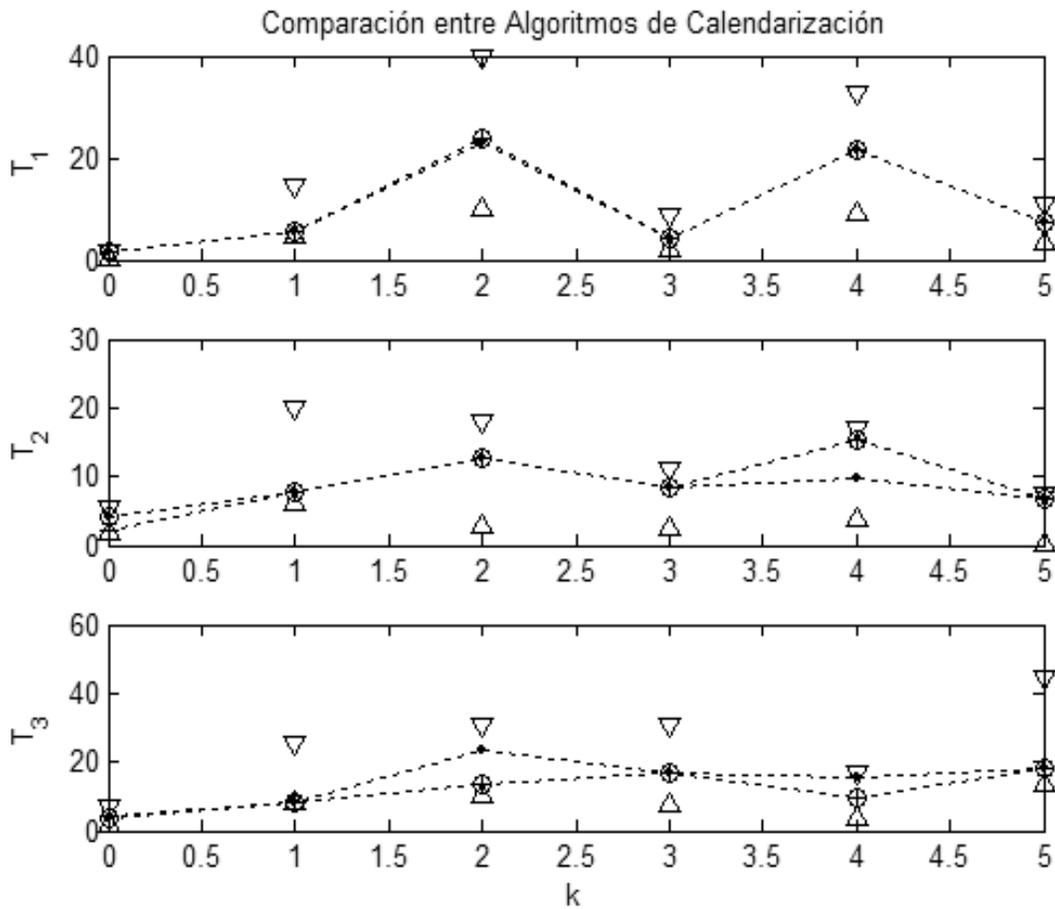
La jerarquía es asignada a cada tarea; como se propone en el ejemplo 5.1. En la Figura 5.2, se considera el tiempo máximo de simulación como  $T_k = 5$  [u.d.t.]. Los cambios en la prioridad de cada tarea obedecen a las variaciones de la entropía del tiempo de arribo  $H(A_i)$ .



**Figura 5.2. Simulación 1: Asignación de prioridades.**

Por ejemplo, para el tiempo  $\lambda_i = 5$  [u.d.t.], se observa en la Figura 5.2, que la tarea  $\tau_1$  con entropía menor, resulta con la prioridad mayor  $J_1 = 3$ . En este mismo punto, el algoritmo asigna una prioridad  $J_3 = 1$  para la tarea  $\tau_3$ .

Con el sistema de tareas  $S = \{\tau_1, \tau_2, \tau_3\}$  ordenado de manera creciente para el instante  $\lambda_i = 5$  se procede a calcular la viabilidad de la tarea en función del orden asignado. De esta manera en el tiempo  $\lambda_i = 5$ . Siguiendo el ejemplo, se observa en la Figura 5.3, que la tarea  $\tau_1$  se ejecuta primero, siendo su jerarquía la mayor  $J_3 = 3$ . Siempre y cuando cumpla con  $H(D_3) - H(C_3) \geq 0$ .



**Figura 5.3. Simulación 1: Calendarización de tareas.**

La Figura 5.3 muestra una ejecución de las tareas del ejemplo 5.1. Donde se muestra para cada tiempo  $\tau_i$  los tiempos de arribo, finalizado y plazo, con la nomenclatura:

- Tiempo de Arribo, símbolo ( $\Delta$ ),
  
- Tiempo de Finalizado, símbolos ( $\cdot$ ,  $+$ , ó  $\circ$ ),
  
- Plazo, símbolo ( $\nabla$ ).

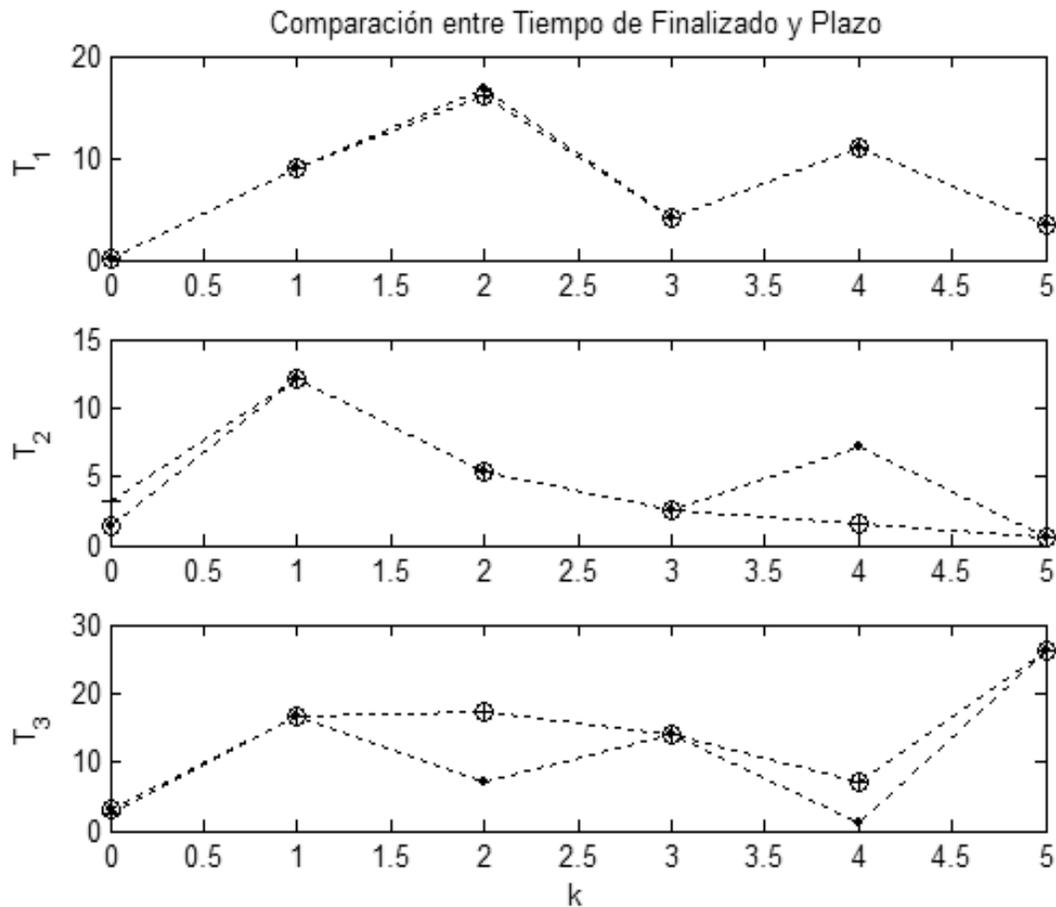
### 5.2.1. Comparación entre Algoritmos de Calendarización

Para sustentar los resultados que arroja el algoritmo propuesto se realiza un análisis comparativo entre dos algoritmos de calendarización clásicos y el “Planificador Estocástico por Plazo” (PEP) [11]. Considerando la siguiente simbología para cada algoritmo:

- Algoritmo de Tasa Monotónica (RM), símbolo ( $\cdot$ ),
  
- Algoritmo de Plazo más Corto (EDF), símbolo ( $+$ ),
  
- Algoritmo de Planificación Estocástico por Plazo (PEP), símbolo ( $\circ$ ).

A partir del ejemplo 5.1 en la Figura 5.3 se muestran los resultados de una ejecución de las tareas  $\tau_1$ ,  $\tau_2$  y  $\tau_3$ . Se muestra, así, la respuesta de los tres algoritmos de acuerdo con su política respectiva a cada tiempo  $\lambda_i$ . En ella se aprecia que los tres algoritmos planifican las

tareas de manera similar. Siendo los tres óptimos, bajo las condiciones establecidas. Y en casos específicos difieren en el orden de ejecución.



**Figura 5.4. Simulación 1: Diferencia entre plazos y tiempos de finalizado.**

Sí se obtiene la diferencia entre el plazo y le tiempo de finalizado en cada evento de calendarización, entonces se tiene cuando el resultado es positivo, la política de calendarización fue realizadas sobre un conjunto de tareas viable. En la figura 5.4 se muestran los resultados de una simulación con los parámetros planteados en el ejemplo 5.1. Siendo todos los valores positivos se concluye que el conjunto de tareas analizadas es viable.

### 5.2.2. Análisis del Algoritmo bajo Condiciones Críticas

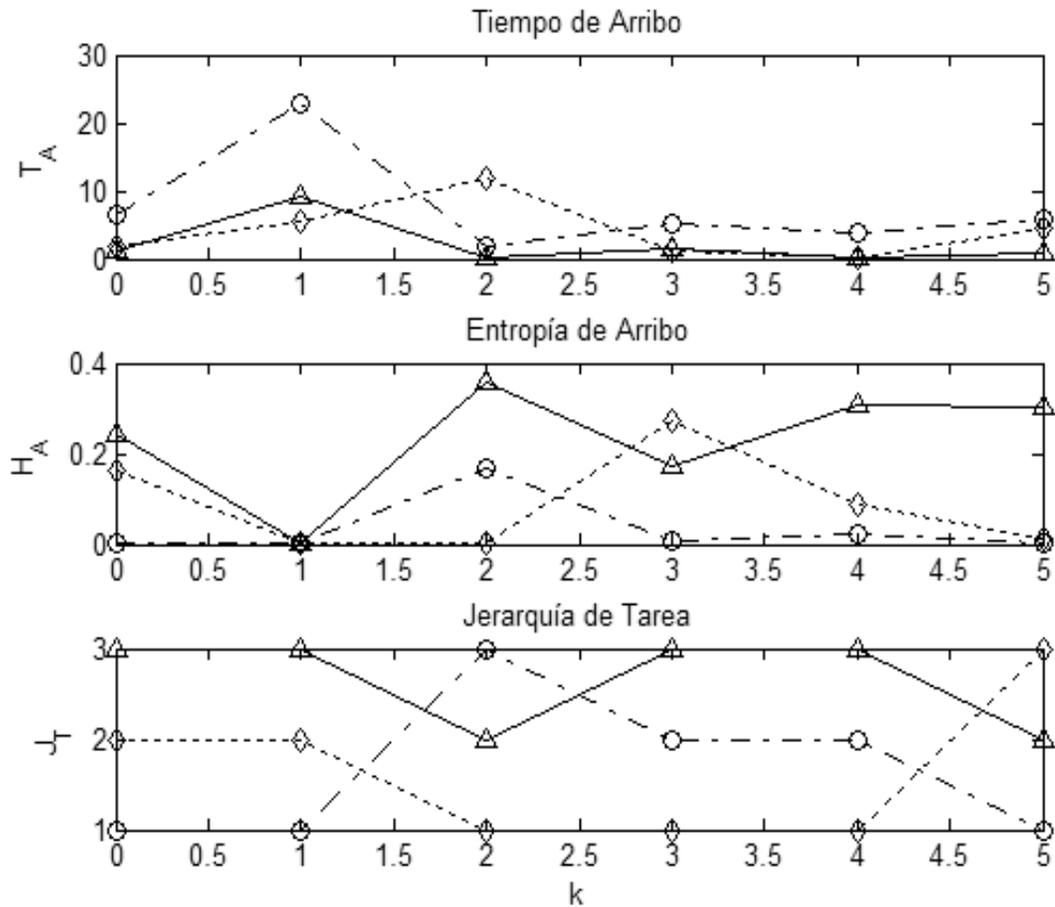
El caso crítico de planificación de tareas sucede cuando las restricciones temporales de tiempo de finalizado y plazo convergen [17]. Es decir  $T_F \approx T_D$ . En este caso existe la posibilidad que el proceso de tareas de no planificable por lo que alguna de las tareas excederá el plazo en su ejecución. La política seleccionada por el algoritmo Planificador Estocástico contempla este caso y asigna las prioridades respectivas de acuerdo con el cumplimiento de la máxima entropía. Siendo mínimo el plazo excedido por el tiempo de finalizado. A continuación se presenta un ejemplo en condiciones críticas para su análisis de resultados.

#### **Ejemplo 5.2** Ejemplo de Análisis en Condición Crítica

*Consideremos  $S$ , como un conjunto de tres tareas  $\{\tau_1, \tau_2, \tau_3\}$  las cuales tienen una tasa de arribo  $K_{a,1} = 5,0$ ,  $K_{a,2} = 5,1$  y  $K_{a,3} = 5,2$ . Sean los tiempos de plazo establecidos por  $K_{d,i} = K_{a,i}$ . Además, consideremos los tiempos de cálculo de 5 unidades. Es decir,  $K_{c,i} \approx K_{d,i}$ .*

De acuerdo con la política del algoritmo de la Figura 5.1, se establece la jerarquía de las tareas en cada instante  $\lambda_i$ . En la Figura 5.5 se muestra una trayectoria de simulación para el ejemplo 5.2. En esta Figura 5.5 se aprecia la similitud en los tiempos de arribo. Aun así se establece una diferencia en la jerarquía asignada por el algoritmo, siendo mayor, en promedio, para la tarea  $\tau_1$ .

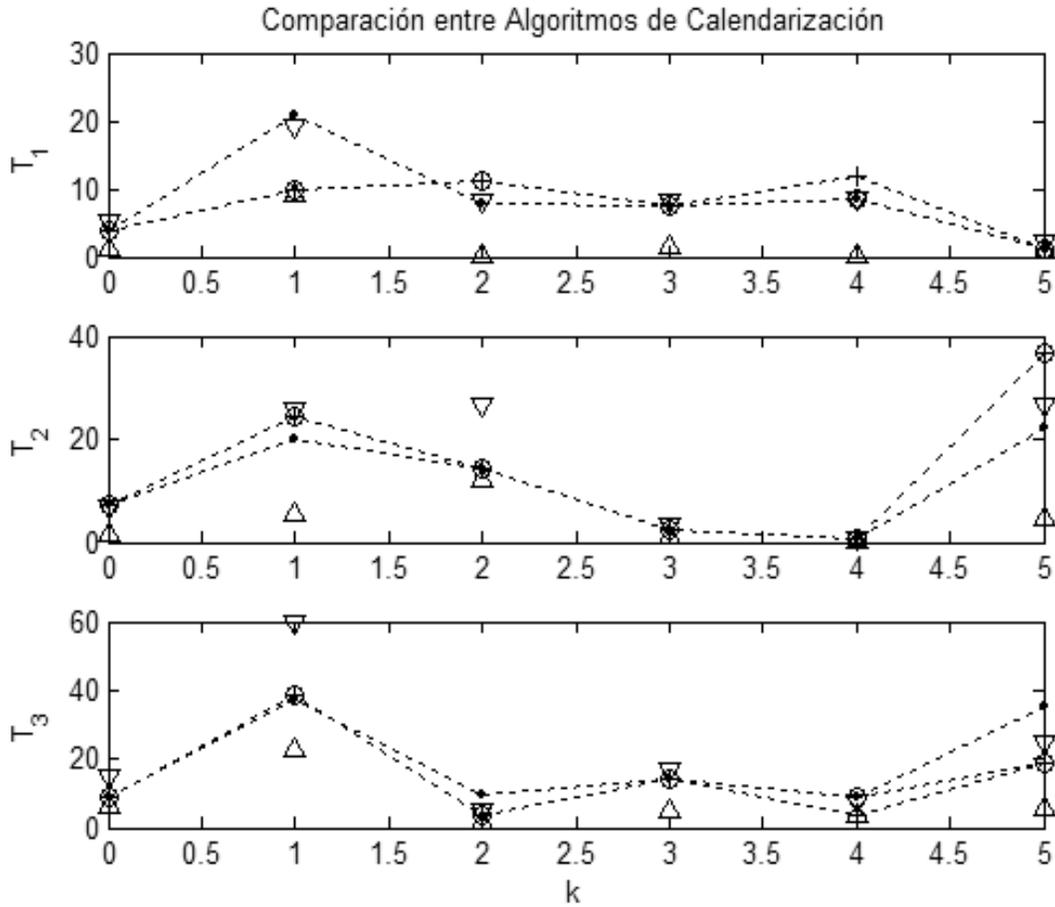
En la Figura 5.6, se muestran las trayectorias correspondientes de la simulación de los tiempos de arribo, finalizado y plazo, del conjunto  $S$ , del ejemplo 5.2. Como se puede obser-



**Figura 5.5. Simulación 2: Asignación de prioridades.**

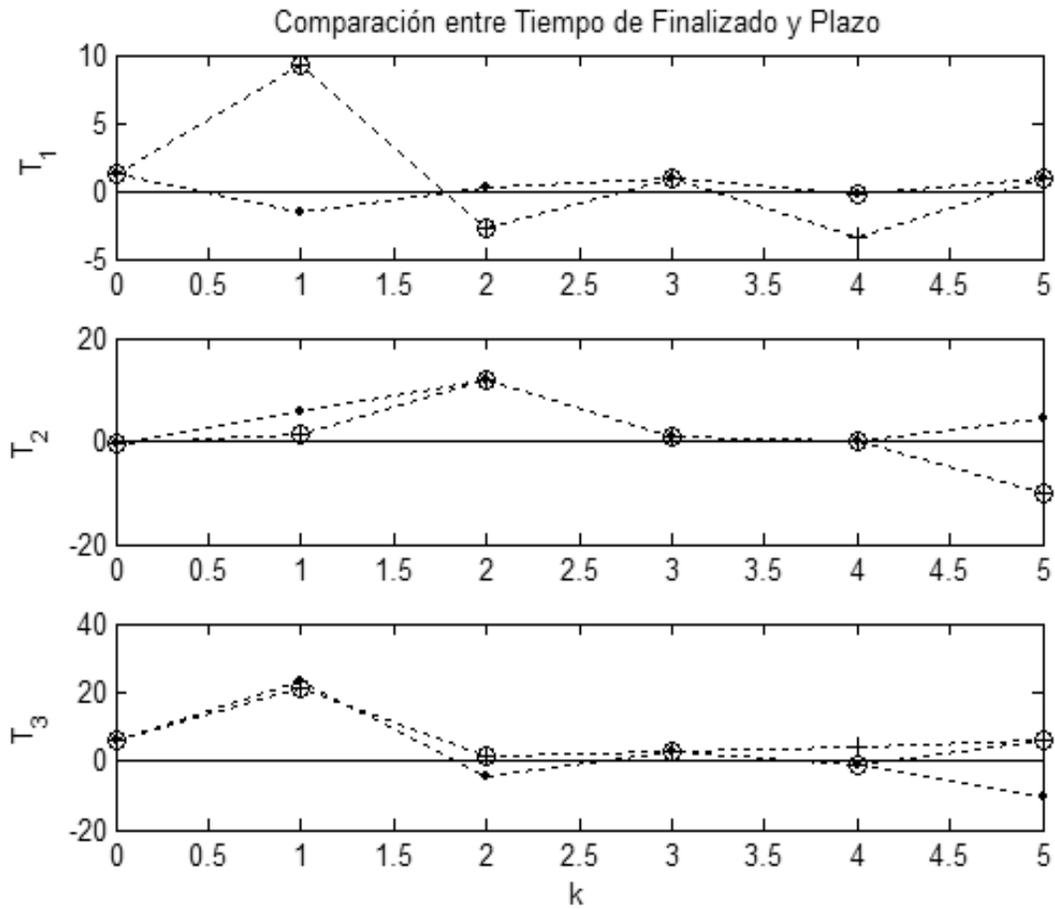
var, bajo condiciones críticas, varias tareas exceden el plazo. Es decir son no planificables. La diferencia entre las políticas de calendarización determinará los excesos de tiempo generados.

En algunas ocasiones, existen situaciones en donde las tareas no pueden ser planificadas por su incongruencia entre los tiempos de ejecución y el plazo marcado. Cuando esto sucede, el procesador busca cumplir su objetivo estadísticamente. En la Figura 5.6 se muestra una simulación donde ocurre esta situación; con el fin de apreciar las diferencias entre los plazos y los tiempos de finalizado en cada instante de la simulación.



**Figura 5.6. Simulación 2: Calendarización de Tareas.**

Cada tarea  $\tau_i$  asegura su tiempo de finalizado de acuerdo con la restricción dada por la definición 3.14. La figura muestra el comparativo entre las diferentes opciones de calendarización que se pueden tomar bajo condiciones críticas. En el tiempo  $\lambda = 2$ , de la Figura 5.7, como ejemplo de análisis, se observan los tiempos de finalizado de los algoritmos comparados. Aquí, los algoritmos EDF y PEP ejecutan la tarea  $\tau_1$  en primer lugar; ejecutando a la tarea  $\tau_2$  en tercer lugar. Esto, cuando el algoritmo RM la planifica la tarea  $\tau_2$  en primer lugar. En este punto el algoritmo RM permite que la tarea  $\tau_1$  exceda su plazo de ejecución.



**Figura 5.7. Simulación 2: Diferencia entre plazos y tiempos de finalizado.**

Aunque este caso crítico es poco probable que suceda en la realidad, permite obtener algunos resultados a partir de los ejemplos 5.1 y 5.2 analizados. En otros casos, el número de tareas es la variable crítica con tiempos de relajación mayores para cada tarea.

- Siendo el algoritmo EDF óptimo en tiempo, el algoritmo PEP también es óptimo.
- El algoritmo PEP asegura una calendarización óptima en un sentido fuerte.

- En condiciones críticas, el algoritmo RM no opera para tareas con restricciones estocásticas.

### 5.3. Resumen del Capítulo

En el presente capítulo se estudio la respuesta del algoritmo generado a través de la simulación de dos ejemplos de análisis. Para este algoritmo de calendarización se aplicó un modelo típico de tareas con restricciones temporales próximas con el fin de ver el comportamiento del algoritmo en condiciones típicas y críticas. El algoritmo, en principio, asigna prioridades en función de los tiempos de arribo de acuerdo con la medida de uniformidad formulada. Posteriormente establece, de acuerdo con la regla de viabilidad, la política de calendarización para cada tarea en un instante dado. Cuando el conjunto de tareas es planificable, el algoritmo asegura la optimalidad de la política en un sentido fuerte.

En un comparativo de desempeño se analizan tres tareas con restricciones temporales planificadas mediante las políticas de calendarización RM, EDF y PEP. Se destacan los resultados de estos métodos exponiendo un análisis de los resultados de una trayectoria típica de simulación.

# Capítulo 6

## Conclusiones

El trabajo de tesis se estructuró partiendo de las definiciones y fundamentos matemáticos requeridos para sustentar los modelos probabilísticos desarrollados. Con los cuales se realizó una aplicación del modelo de planificador propuesto a un algoritmo de calendarización. El método resultante se probó para datos teóricos de tareas estocásticas típicas.

Previo al diseño del algoritmo de planificación se dio una introducción al estado del arte en el área de calendarización de tareas en tiempo real. En el capítulo 2 se hizo referencia a los principales planificadores de acuerdo con su operación. Además se enfatizó sobre las restricciones consideradas en las características de las tareas. De esta revisión se acotó el campo de aplicación al que se refirió este trabajo de tesis.

En los capítulos 3 y 4 se definieron las variables estocásticas que conforman el modelo de asignación de prioridad; como del modelo de planificador estocástico. Se demostró en el capítulo 4 que el modelo de planificador es óptimo en un sentido fuerte; como lo establece Goossens en su trabajo [19]. Cumpliendo así con los objetivos específicos declarados en el

capítulo 1. Se comprobó, entonces, que la diferenciación de tareas mediante la medida de máxima entropía es una técnica viable para tareas con restricciones temporales estocásticas definidas como funciones de probabilidad exponenciales.

El punto principal que se aborda en el trabajo de tesis es el problema de calendarización de tareas de tiempo real del tipo suave con el enfoque de procesos estocásticos para las restricciones temporales consideradas. De los resultados obtenidos se definen los parámetros de certidumbre que satisfacen los requerimientos de tiempo real. En el capítulo 5 se presentó un acercamiento para la evaluación del desempeño del planificador comparado sus desempeño contra los métodos clásicos; RM y EDF. Para sistemas monoprocesador, la implementación resulta simple dado los resultados del modelo desarrollado. Parecido en principio al algoritmo EDF, el algoritmo PEP toma al plazo como base para su método de calendarización. Aquí la regla de viabilidad asegura la calendarización de todas las tareas; siempre y cuando estas sean planificables. Con esto se cubre el objetivo general descrito en el capítulo 1.

Se concluye ahora sobre los alcances establecidos al abordar el tema de tesis. Donde se define el objeto de estudio; delimitando el trabajo a calendarización de tareas con restricciones temporales en un procesador.

## **6.1. Conclusiones de los Alcances**

Acorde con el objetivo específico planteado, se desarrolló un modelo probabilístico para la diferenciación de las tareas de acuerdo a una jerarquía respaldada por niveles de entropía de las restricciones temporales. Permitiendo así, asignar prioridades a las tareas de manera dinámica. Para tareas con función de distribución exponencial, la máxima entropía determina la prioridad de la tarea como un valor específico para cada tarea del conjunto.

El algoritmo propuesto, "Planificador Estocástico por Plazo", se sustenta en el modelo probabilístico desarrollado a partir del análisis matemático de las funciones de distribución. Para el modelo planteado se hace un análisis de la optimalidad. Con esto se asegura que el modelo del planificador es óptimo cuando el conjunto de tareas es viable.

Se delimita el algoritmo de planificación para un servidor con el cual se desarrolla el modelo matemático. El cual tiene su fundamento en la Teoría de Líneas de Espera; como refiere Lehoczky [29]. Para sistemas multiprocesador, el modelo básico ha de incluir un modelo de asignación de procesador a través de algún criterio de selección; vea Lehoczky [30]. Del mismo modo el modelo desarrollado permite incluir restricciones como el tiempo de liberación y el tiempo de relajación, por ejemplo.

Con el marco de referencia descrito en el capítulo 1, el trabajo de tesis resuelve con un nuevo enfoque el problema de planificación de tareas en tiempo real del tipo suave. El algoritmo PEP fundamenta su operación en el modelo de planificador probabilístico de tareas desarrollado en el capítulo 4. Donde la jerarquía de las tareas se asigna de acuerdo a la máxima entropía del plazo. Limitando la planificación al conjunto de tareas planificables a través del teorema de viabilidad plantado en el mismo capítulo.

## **6.2. Trabajos Futuros**

El alcance de este trabajo es resultado de un primer análisis de tareas en tiempo real del tipo suave con restricciones temporales analizadas como proceso estocásticos diferenciadas a partir de niveles de entropía. La base matemática desarrollada fundamenta la viabilidad del trabajo en sistemas multiprocesador, como una extensión inmediata del trabajo de tesis.

Del estudio de sistemas de tiempo real como procesos continuos permite la aplicación de técnicas de control. Con lo que un modelo de calendarización adaptable es factible. En el trabajo de Bhattacharya puede verse un análisis matemático para problemas de líneas de espera  $M/GI/1$  [7]. Otra aplicación puede tomarse a partir del trabajo de Keshav [50].

Ligado a la entropía como medida de niveles de energía es posible establecer una relación con un análisis de estabilidad. A partir de la máxima entropía como una función candidata Liapunov, como lo muestra Maglaras [36] en su trabajo de tesis.

La complejidad en los sistemas digitales empuja a buscar nuevas soluciones a los problemas de calendarización. La propuesta de este trabajo ofrece una visión que se adecúa a sistemas dinámicos no determinísticos.

### 6.3. Publicaciones

Revistas:

- Toledo, M. and Medel, J., *Maximum Entropy Stochastic Tasks Classification*, Journal of Advances in Computer Science and Engineering, vol. 23, pp 53-62, 2006.

Trabajo que introduce la clasificación de tareas en función de la máxima entropía del tiempo de arribo. En él se proporciona las definiciones básicas para un modelo diferenciador de tareas.

Congresos:

- Toledo, M. and Medel, J., *Priority Tasks Allocation through the Maximum Entropy Principle*, The 8th WSEAS Int. Conf. on Automation and Information, June, 2007.
- Toledo, M. and Medel, J., *Stochastic Task Scheduling based on the Maximum Entropy Principle*, IEEE 4th CERMA Conference, September, 2007.

En estos trabajos se presenta y desarrolla los modelos de asignación de prioridad de tareas estocásticas. Así como el algoritmo de planificación estocástico por plazo. Que son resultado del trabajo expuesto en la tesis.

Carteles:

- Toledo, M. and Medel J., *Planificador Estadístico de Tareas Estacionarias*, XXII Semana Interdisciplinaria UPICSA, IPN, 2005.

# Referencias

- [1] S. Abdelzaher, T. and C. Lu. A utilization bound for aperiodic tasks and priority driven scheduling. *Journal of IEEE Transaccion on Computer*, 53(3), 2004.
- [2] L. Abeni and G. Butazzo. Qos guarantee using probabilistic deadlines. *In Proceedings of the 11th Euromicro Conference of Real-Time Systems*, pages 242–249, 1999.
- [3] A. Atlas and A. Bestavros. Statistical rate monotonic scheduling. *In Proceedings of the 19th IEEE Real-Time System Symposium*, pages 242–249, 1999.
- [4] H. Aydin and R. Melhem. Optimal reward-based scheduling for periodic real-time tasks. *Journal of IEEE Transaccion on Computer*, 50(2):111–130, February 2001.
- [5] R. L. Baruah, S.K. and R. Howell. Algorithms and complexity concerning the preemptive scheduling of periodic, real-time tasks on one processor. *Journal of Real-Time Systems*, 4(2):301–324, 1990.
- [6] C. S. Beccari, G. A technique for adaptive scheduling of soft real-time tasks. *Journal of Real-Time Systems*, 30(3):187–215, May 2005.
- [7] P. Bhattacharya and L. Georgiadis. Problems of adaptive optimization in multiclass m/gi/1 queues with bernoulli feedback. *in Proceedings of the 29th Allerton Conference on Communication, Control and Computing*, October 1991.
- [8] B. G. Bini, E. Measuring the performance of schedulability tests. *Journal of Real-Time Systems*, 30(1-2):129–154, May 2005.
- [9] M. Bunge. *La Ciencia, su Método y Filosofía*. Ed. Siglo XX, Argentina, 1977.

- [10] G. Buttazzo. *Hard Real time Computing Systems*. Kluwer Academic Publishers, Norwell, MC, USA, 1997.
- [11] G. Buttazzo. Rate monotonic vs. edf: judgment day. *Journal of Real-Time Systems*, (29):5–26, 2005.
- [12] V. Campenhout and M. Cover. Maximum entropy and conditional probability. *Journal of IEEE Transactions on Information Theory*, 4(27):483–389, 1981.
- [13] R. Chandra and X. Liu. On the scheduling of flexible and reliable real-time control systems. *Journal of Real-Time Systems*, (24):153–169, 2003.
- [14] E. J. Coffman. *Computer and job-shop scheduling theory*. John Wiley & Sons, USA, 1976.
- [15] E. J. Coffman. A characterization of waiting time performance realizable by single server queues. *Journal of Operations Research*, (28):810–821, 1980.
- [16] M. Gardner. Analyzing stochastic fixed-priority real-time systems. *In proceedings of the 15th International Conference on Tools and Algorithms for the Construction and Analysis of Systems*, 1579:44–58, 1999.
- [17] M. Gardner. Performance of algorithms for scheduling real-time systems with overrun and overload. *In proceedings of the 11th Euromicro Conference on Real-Time Systems*, June 1999.
- [18] K. Glazebrook. Scheduling tasks with exponential service time on parallel processors. *Journal of Applied Probability*, (16):685–689, 1979.
- [19] J. Goossens. *Scheduling of Hard Real-Time Periodic Systems with Various Kinds of Deadline and Offset Constraints*. PhD. Thesis, Université Libre de Bruxelles, Faculté des Sciences, Belgium, 2000.
- [20] J. Goossens and S. Funk. Priority-driven scheduling of periodic task systems on multiprocessors. *Journal of Real-Time Systems*, 25:187–205, October 2003.
- [21] K. Gopalan and T. Chiueh. Probabilistic delay guarantees using delay distribution measurement. *in Proceedings of the 12th annual ACM international conference on Multimedia*, pages 900–907, October 2004.

- [22] R. Ha and J. Liu. Validating timing constraints in multiprocessor and distributed real-time systems. *in Proceedings of the 14th IEEE International Conference on distributed Computing Systems*, 1994.
- [23] E. T. Jaynes. Information theory and statistical mechanics. *Journal of The Physical Review*, 4(106):620–630, May 1957.
- [24] E. T. Jaynes. On the rationale of maximum-entropy methods. *in Proceedings of the IEEE*, 49(70):939–952, September 1982.
- [25] K. Jeffay and S. Goddard. A theory of rate-based execution. *in Proceedings of the 20th IEEE Real-Time Systems Symposium*, 1999.
- [26] S. M. Johansson, M. Resource allocation under uncertainty using the maximum entropy principle. *Journal of IEEE Transactions on Information Theory*, 51(12):4103–4117, December 2005.
- [27] M. Joseph and P. Pandya. Finding response times in a real-time system. *The Computer Journal*, 29(5):390–395, October 1986.
- [28] L. Kleinrock and R. Muntz. Processor sharing queueing models of mixed scheduling disciplines for time shared systems. *Journal of ACM*, 19(3):464–482, October 1972.
- [29] J. Lehoczky. Real-time queueing theory. *in Proceedings of the 17th Real-Time System Symposium*, pages 186–195, December 1996.
- [30] J. Lehoczky. Real-time queueing network theory. *in Proceedings of the 18th Real-Time System Symposium*, pages 58–67, December 1997.
- [31] J. Lehoczky. Using real-time queueing theory to control lateness in real-time systems. *Journal of ACM*, pages 158–168, 1997.
- [32] J. Lehoczky and L. Sha. The rate monotonic scheduling algorithm: Exact characterization and average case behavior. *in Proceedings of the IEEE Real-Time Systems Symposium*, December 1989.
- [33] T. H. Lin and W. Tarn. Scheduling periodic and aperiodic tasks in hard real-time computing systems. *Performance Evaluation Review*, 19(1):31–38, May 1991.

- [34] C. L. Liu and J. W. Layland. Scheduling algorithms for multiprogramming in a hard-real-time environment. *Journal of ACM*, 20(1), 1973.
- [35] S. J. Lu, C. and G. Tao. Design and evaluation of a feedback control edf algorithm. *Real-Time Systems Symposium*, December 1999.
- [36] C. Maglaras. *Dynamic control of stochastic processing networks: A fluid model approach*. PhD. Thesis, Stanford University, USA, 1998.
- [37] Y. Manabe. A feasibility decision algorithm for rate monotonic and deadline monotonic scheduling. *Journal of Real-Time Systems*, (14):171–181, 1999.
- [38] S. Manolache. Schedulability analysis of applications with stochastic task execution times. *Journal of ACM Transactions on Embedded Computing Systems*, 3(4):706–735, November 2004.
- [39] A. K. Mok. *Fundamental design problems of distributed systems support for the hard real-time environment*. PhD. Thesis, MIT, Dept., MIT/LCS/TR-297, 1983.
- [40] A. K. Mok and F. Jahanian. Formal specification of real-time systems. *University of Texas at Austin*, 1888.
- [41] P.Ñain and P. Tsoucas. Interchange arguments in stochastic scheduling. *Journal of Applied Probability*, (27):815–826, 1989.
- [42] J.Ñiño-Mora. Stochastic scheduling. *Encyclopedia of Optimization*, C. A. Floudas and P. M. Pardalos, pages 367–372, 2001.
- [43] J.Ñiño-Mora. Marginal productivity index policies for scheduling a multiclass delay-loss-sensitive queue. *Queueing Systems Journal*, 54:281–312, 2006.
- [44] N.Ñissanke. *Real-time systems*. Prentice Hall, Great Britain, 1997.
- [45] D. P. and B. J. The operational analysis of queueing network model. *Journal of ACM Computing Surveys*, 10(3):225–261, September 1978.
- [46] A. Papoulis. *Probability, random variables, and stochastic processes*. McGraw-Hill, 1965.
- [47] E. Parzen. *Stochastic processes*. Holden-day, 1962.
- [48] F. Reza. *An introduction to information theory*. Dover Publications, NY, USA, 1994.

- [49] H. Ryu. Maximum entropy estimation of density and regression functions. *Journal of Econometrics*, (56):397–440, 1993.
- [50] K. S. A control-theoretic approach to flow control. *in In Proceedings of ACM SIGCOMM*, pages 3–15, September 1991.
- [51] L. Sha. Real time scheduling theory: a historical perspective. *Journal of Real-Time Systems*, (28), 2004.
- [52] L. Sha and J. Goodenough. Real-time scheduling theory and ada. *Journal of IEEE Computer*, 23(4):53–62, 1990.
- [53] C. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27, July 1948.
- [54] B. G. Spuri, M. Scheduling aperiodic tasks in dynamimc priority systems. *Journal of Real-Time Systems*, 2(10), 1996.
- [55] L. C. Stankovic, J. and G. Tao. The case for feedback control real-time scheduling. *EuroMicro Conference on Real-Time Systems*, June 1999.
- [56] R. K. Stankovic, J. What is predictability for real-time systems? *Journal of Real-Time Systems*, 2(4):247–254, November 1990.
- [57] S. M. Stankovic, J. Implications of classical scheduling results for real-time systems. *Computer Journal*, 28(6):16–25, June 1995.
- [58] T. Tia and Z. Deng. Probabilistic performance gurantee for real-time tasks with varying computation times. *In Proceedings, Real-Time Technology and Applications Symposium*, pages 164–173, May 1995.
- [59] M. Toledo and J. Medel. Maximum entropy stochastic tasks classification. *Journal of Advances in Computer Science and Engineering*, 23:53–62, 2006.
- [60] M. Toledo and J. Medel. Priority tasks allocation through the maximum entropy principle. *in The 8th WSEAS Int. Conf. on Automation and Information*, June 2007.
- [61] M. Toledo and J. Medel. Stochastic task scheduling based on the maximum entropy principle. *IEEE 4th CERMA Conference*, September 2007.

- [62] W. Wang and A. Mok. Pre-scheduling. *Journal of Real-Time Systems*, (30):83–103, 2005.
- [63] J. Xu and D. Lorge. Priority scheduling versus pre-run-time scheduling. *Journal of Time-Critical Computing Systems*, (18):7–23, 2000.