Planificación de sistemas de tiempo real crítico multiprocesador para reducir la interferencia

José María Aceituno, Ana Guasque, Patricia Balbastre, José Simó, Juan Francisco Blanes, Alfons Crespo

Instituto de Automática e Informática Industrial (ai2)

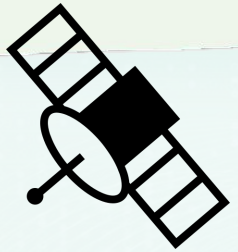
Universitat Politècnica de València

Contenido

- Introducción
- Modelo temporal
- Test de planificabilidad
- Técnicas de planificación para reducir interferencia
- Evaluación
- Conclusiones

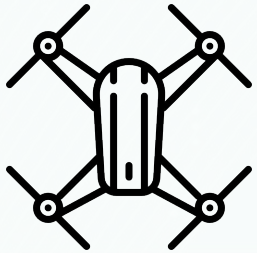
Introducción

- Sistemas críticos: Son aquellos en los que la misión se pierde en caso de fallo y este tiene consecuencias catastróficas
- Generalmente, por requisitos de certificación al estándar del sector, los recursos deben asignarse de forma estática
- Recursos temporales: qué proceso se ejecuta en qué procesador y en qué instante de tiempo



ESPACIO

AUTOMÓVIL



UAVs

TREN

PLANIFICACIÓN DE SISTEMAS DE TIEMPO REAL CRÍTICO

Planificación multiprocesador

- La planificación de los sistemas multiprocesador en tiempo real introduce más complejidad que la de los sistemas monoprocesador
- La ejecución de las tareas ya no depende sólo de su propio tiempo de cálculo o del de las tareas más prioritarias.
- Los recursos de hardware compartidos entre procesadores introducen un importante factor de interferencia
- Tres fuentes principales de interferencias
  - Memoria principal
  - Bus de memoria
  - Cache

Introducción

¿Cómo medir la interferencia?

Modelo específico

- Cálculo ad-hoc para cada tipo de recurso hw
- Sólo válido para ese hw
- Interferencia muy similar a la real

Modelo general

- Si el fabricante no da detalles del recurso hw
- Independiente del hw utilizado pero
- Valor de interferencia muy pesimista

¿Cómo se incorpora al modelo temporal?

Se añade al WCET

- La teoría de planificación existente es válida
- Muy pesimista considerar que la interferencia ocurre en cada activación

Se añade como parámetro adicional

- Es necesario definir un nuevo modelo y planificación

Introducción

¿Cómo medir la interferencia?

Modelo específico

- Cálculo ad-hoc para cada tipo de recurso hw
- Sólo válido para ese hw
- Interferencia muy similar a la real

Modelo general

- **Si el fabricante no da detalles del recurso hw**
- **Independiente del hw utilizado pero**
- **Valor de interferencia muy pesimista**

¿Cómo se incorpora al modelo temporal?

Se añade al WCET

- La teoría de planificación existente es válida
- Muy pesimista considerar que la interferencia ocurre en cada activación

Se añade como parámetro adicional

- **Es necesario definir un nuevo modelo y planificación**

Introducción

Aportaciones

- Definir modelo general que tenga en cuenta la interferencia
- Definir cómo se va a planificar
- Proporcionar tests de planificabilidad
- Proponer técnicas para reducir la interferencia
  - Actuando sobre el allocator
  - Actuando sobre la planificación

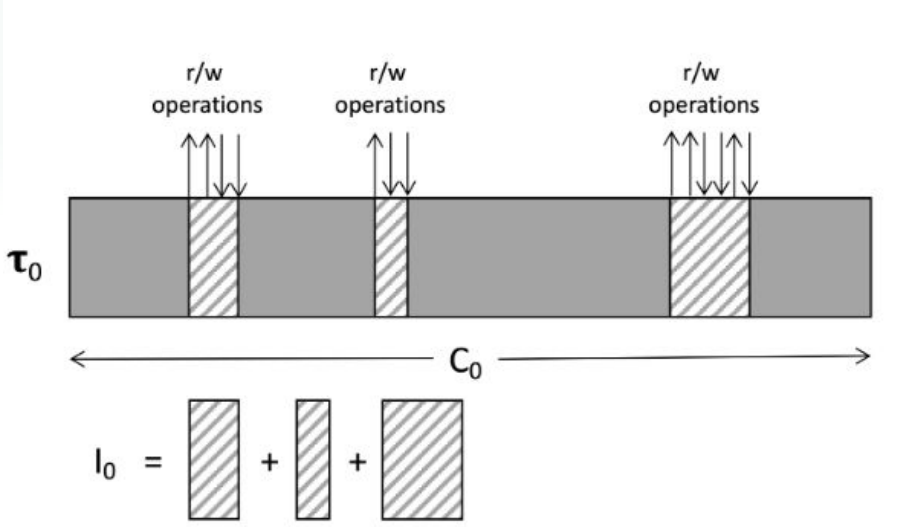
Modelo temporal

M cores: $M_1, M_2, \ldots, M_n$

N tareas: $\boldsymbol{\tau} = \{\boldsymbol{\tau}_1, \boldsymbol{\tau}_2, \ldots, \boldsymbol{\tau}_n\}$

Cada tarea se caracteriza como $\boldsymbol{\tau}_i = (C_i, D_i, T_i, I_i)$

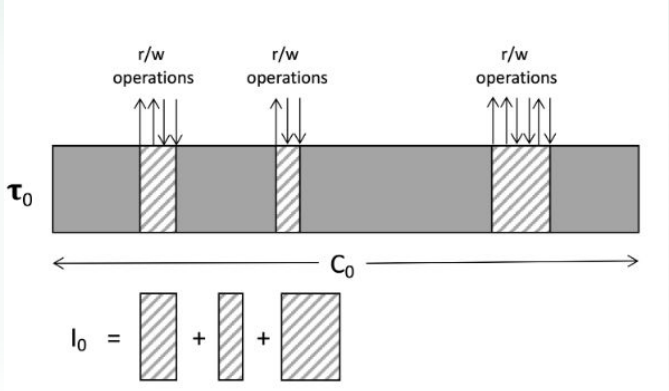
Parámetro de Interferencia I_i



- Tarea emisora ⇒ $I_i \neq 0$ por lo que provoca interferencia
- Tarea receptora ⇒ $I_i \neq 0$ por lo que recibe interferencia

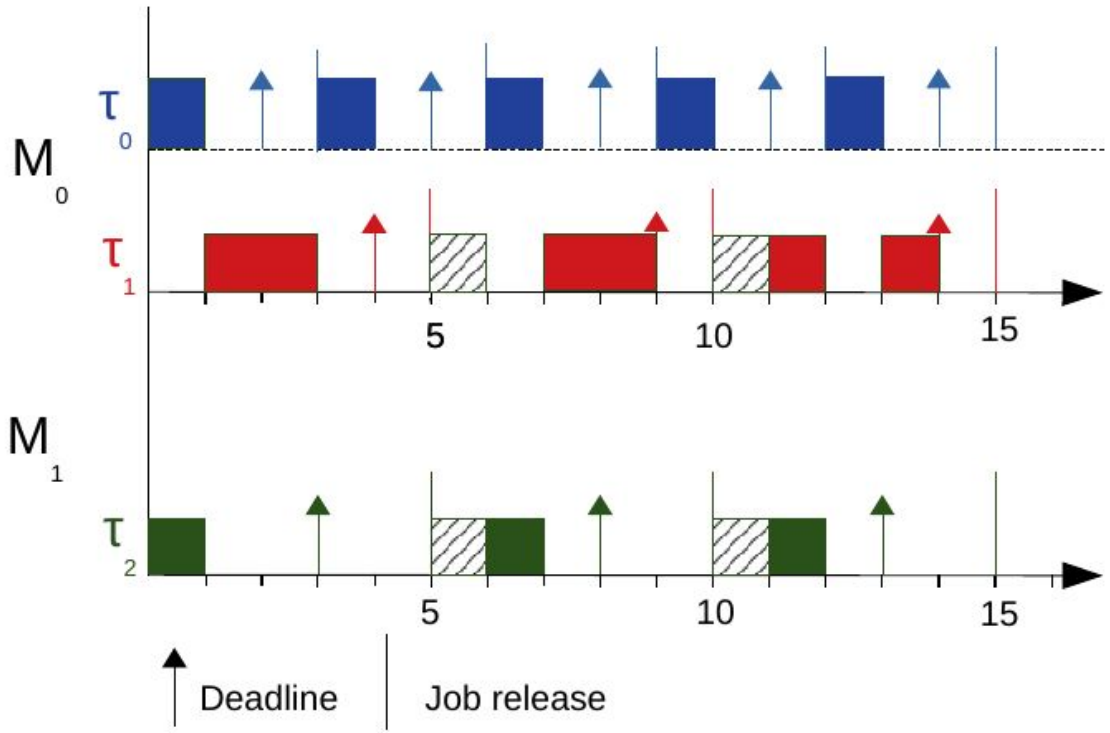
Modelo temporal

- La interferencia incrementa el tiempo de ejecución de las tareas y, por tanto, reduce la planificabilidad



$\tau_0 = (1,2,3,0)$
$\tau_1 = (2,4,5,1)$

$\tau_2 = (1,3,5,1)$



Análisis de planificabilidad

- Prioridades fijas
  - Inclusión de I_i en la fórmula del *WCRT*
  - Trabajo en progreso

- Prioridades dinámicas (EDF)
  - Inclusión de I_i en la fórmula de *dbf*
  - Resultados publicados en:

    Ana Guasque, José María Aceituno, Patricia Balbastre, José Simó, Alfons Crespo. *Schedulability analysis of dynamic priority real-time systems with contention*. The Journal of Supercomputing (2022)

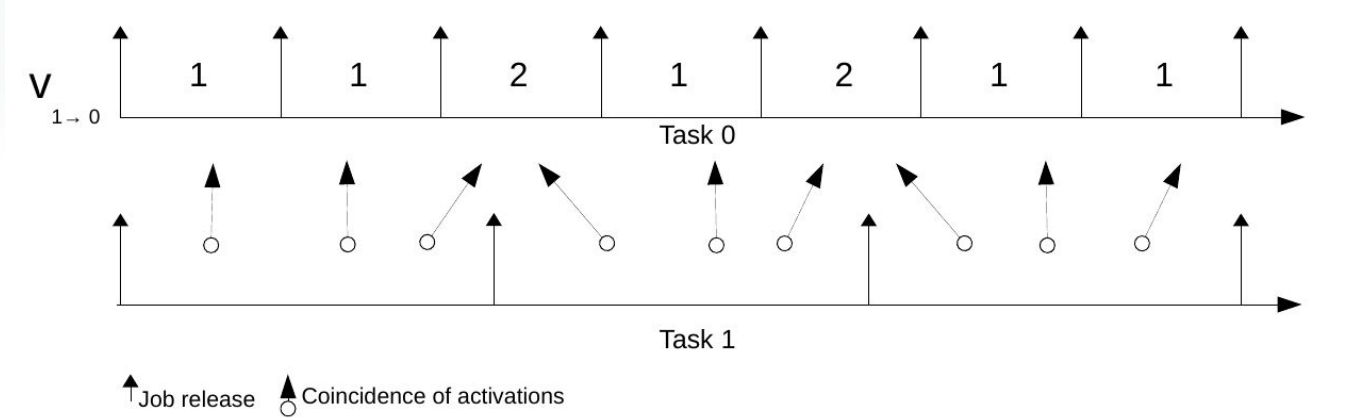
Análisis de planificabilidad

- Prioridades dinámicas (EDF)
  - Análisis conocido basado en la función de límite de demanda (dbf)
  - Queremos obtener una función equivalente a dbf aplicada al nuevo modelo temporal
  - Max. número de activaciones que τ_j provoca sobre τ_i

$$\overrightarrow{v_{j\to i}}[a] = 1 + \sum_{t=aT_i+1}^{(a+1)T_i-1} g(t)$$

$$g(t) = \begin{cases} 1 & \textit{If } t - T_j \left\lfloor \frac{t}{T_j} \right\rfloor = 0 \\ 0 & \textit{Elsewhere} \end{cases}$$

  - Ejemplo:



Análisis de planificabilidad

- Uso del vector $v_{j\to i}$ para obtener una dbf modificada que incorpore la interferencia.
- Primera aproximación: Añadir $v_{j\to i}$ al WCET $\longrightarrow$ dbf'

$$\max_{0\le a\le A_i-1} \overrightarrow{v_{j\to i}}[a]\cdot I_j$$

$$\downarrow$$

$$C_i' = C_i + \sum_{\tau_j\notin M_k} \max_{0\le a\le A_i-1} \overrightarrow{v_{j\to i}}[a]\cdot I_j$$

$$\downarrow$$

$$dbf'_{\tau_{M_k}}(t) = \sum_{\forall\tau_i\in M_k} C_i' \left\lfloor \frac{t+T_i-D_i}{T_i} \right\rfloor$$

Muy pesimista

Condición de planificabilidad

$$\longrightarrow \quad dbf'_{\tau_{M_k}}(t) \le t \quad \forall t \le L_b$$

Análisis de planificabilidad

- Uso del vector $v_{j\rightarrow i}$ para obtener una dbf modificada que incorpore la interferencia.
- Segunda aproximación: Añadir $v_{j\rightarrow i}$ a la dbf$\rightarrow$ dbf''

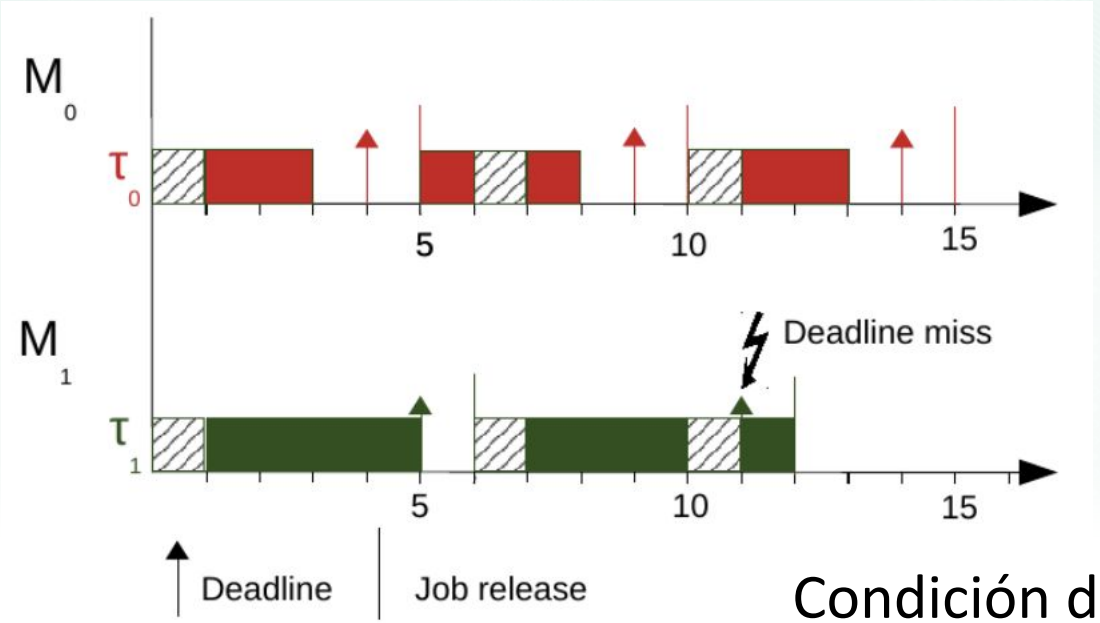
$$dbf''_{\tau_{M_k}}(t) = \sum_{\tau_i \in M_k} \left(C_i \left\lfloor \frac{t + T_i - D_i}{T_i} \right\rfloor + \sum_{\tau_j \notin M_k} \sum_{a=0}^{\left\lfloor \frac{t+T_i-D_i}{T_i} \right\rfloor - 1} \overrightarrow{v_{j\rightarrow i}}[a] \cdot I_j \right)$$

$$dbf''_{\tau_{M_k}}(t) \leq dfb'_{\tau_{M_k}}(t)$$

Menos pesimista

Análisis de planificabilidad

- Contraejemplo



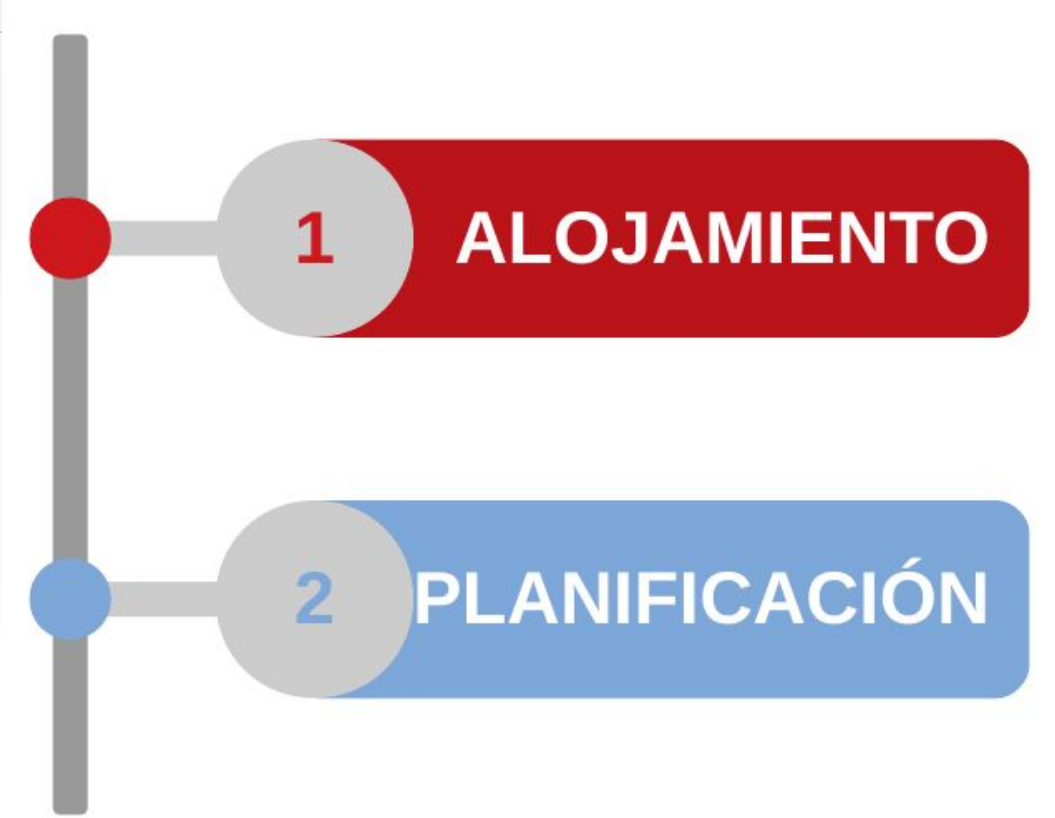
τ_0 = (2,4,5,1)
τ_1 = (4,5,6,1)

El primer intervalo ocupado no es el peor caso

Condición de planificabilidad

$$dbf''_{\tau_{M_k}}(t_1, t_2) \leq t_2 - t_1 \quad 0 \leq t_1 < t_2 \leq H$$

Técnicas de planificación para reducir interferencia



Fase de alojamiento

1 ALOJAMIENTO

- Existen técnicas bien conocidas para alojar tareas en procesadores en sistemas multiprocesador.
- Por ejemplo, técnicas de bin-packing: Worst-Fit, First-Fit…
- ¿Y si queremos un alojamiento factible pero en el que se minimice algún parámetro?

- Interferencia
- Número de cores utilizados
- Discrepancia

→ Alojamiento mediante técnicas no convencionales

Fase de alojamiento

1 ALOJAMIENTO

- Algoritmos basados en técnicas no convencionales:

Programación lineal entera (LP)

- Definir **parámetros** y **variables** (qué tarea se aloja en cada procesador…)
- Sujeto a las **restricciones** de alojamiento de sistemas multiprocesador (la utilización de un core ≤ 1, una tarea alojada en un core (no migración→sistemas críticos)…)
- Minimizar o maximizar uno o varios **objetivos** (interferencia total producida, número de procesadores utilizados…)
- **Solver** utilizado: Gurobi 9.5

Fase de alojamiento

1 ALOJAMIENTO

- Algoritmos basados en técnicas no convencionales:

UDmax, UDmin	Wmin	Imin
Basados en discrepancia: diferencia entre las utilizaciones máxima y mínima de los todos los procesadores. No tienen en cuenta la interferencia de las tareas, solo su carga (utilización)	Wij es una matriz binaria que indica en cada *t* si la tarea τi provoca (1) o no (0) interferencia en la tarea τj. Wmin aloja tareas en procesadores de forma que reduce la matriz Wij. Considera la posible interferencia producida por cada tarea.	Propone un límite superior de utilización considerando la interferencia. Imin aloja tareas en procesadores de forma que minimiza el límite superior de utilización de todas las tareas, teniendo en cuenta la interferencia.
Modelo de tareas con plazos $Di \leq Ti$ y tareas periódicas o esporádicas. Cualquier algoritmo basado en prioridades (fijas o dinámicas)		

Fase de alojamiento: UDmax y UDmin

UDmax/UDmin: MILP

SETS AND INDICES	
i	Tasks $\tau_i \in \{0, 1, 2, \ldots, n-1\}$
k	Cores $M_k \in \{0, 1, 2, \ldots, m-1\}$
PARAMETERS	
C_i	Worst case execution time of τ_i
T_i	Period of τ_i
U_i	Theoretical utilisation of τ_i
I_i	Interference factor of τ_i over other tasks
DECISION VARIABLES	
UDmin and UDmax models	
O_{ik}	Allocation matrix. 1 if τ_i allocated in M_k, 0 otherwise
U_{M_k}	Theoretical utilisation of core k
UD_τ	Utilisation discrepancy of the task set τ

Ecuación a optimizar:

$$UD_\tau = \max_{k=0}^{m-1}(U_{M_k}) - \min_{k=0}^{m-1}(U_{M_k})$$

Restricciones:

s.t:

$$\sum_{\forall k} O_{ik} = 1 \qquad \forall i$$

$$\sum_{i \in k} U_i \cdot O_{ik} = U_{M_k} \qquad \forall k$$

$$U_{M_k} \leq 1 \qquad \forall k$$

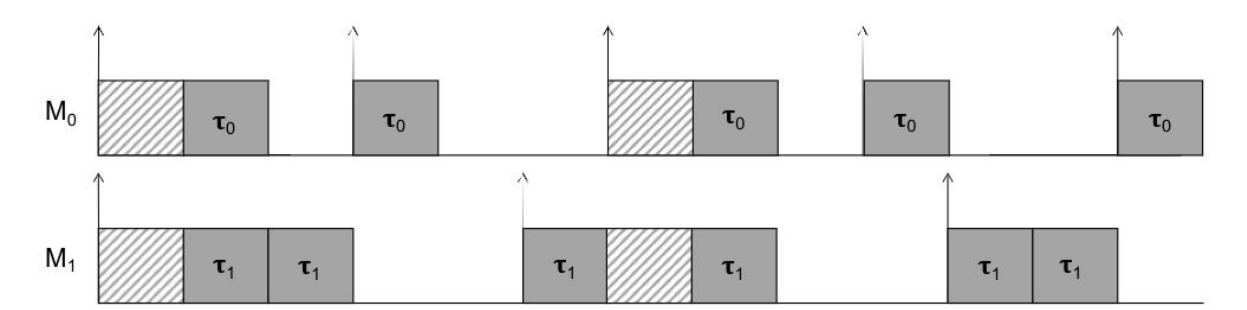
$$UD_\tau = \max_{k=0}^{m-1}(U_{M_k}) - \min_{k=0}^{m-1}(U_{M_k})$$

$$O_{ik} \in \{0, 1\}$$

$$U_{M_k} \geq 0$$

Fase de alojamiento: Wmin

- Matriz W (n x n x H): considera la posible interferencia producida
- W es una matriz binaria:
  - $W_{ij} = 1 \rightarrow \tau_i$ provoca interferencia sobre τ_j
  - $W_{ij} = 0 \rightarrow$ cualquier otro caso
- τ_i y τ_j en el mismo procesador $\rightarrow W_{ij}, W_{ji}=0$
- τ_i no es una tarea emisora $\rightarrow W_{ij}, W_{ji}=0$



t	W	Comment
0	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	τ_0 and τ_1 release
1	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	
2	$\begin{pmatrix} 0 & 0 \\ 1 & 0 \end{pmatrix}$	τ_0 finishes its first activation
3	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$	τ_1 finishes its first activation
4	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$	
5	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$	τ_1 releases, but τ_0 is not active
6	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	τ_0 releases while τ_1 is active
7	$\begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$	
8	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$	τ_0 and τ_1 finish
9->15	$\begin{pmatrix} 0 & 0 \\ 0 & 0 \end{pmatrix}$	τ_0 and τ_1 are not active at the same time

Fase de alojamiento: Wmin

Wmin: MILP

SETS AND INDICES	
i	Tasks $\tau_i \in \{0, 1, 2, \ldots, n-1\}$
k	Cores $M_k \in \{0, 1, 2, \ldots, m-1\}$
PARAMETERS	
C_i	Worst case execution time of τ_i
T_i	Period of τ_i
U_i	Theoretical utilisation of τ_i
I_i	Interference factor of τ_i over other tasks
DECISION VARIABLES	
Wmin model	
O_{ik}	Allocation matrix. 1 if τ_i is allocated in core k and 0 otherwise
U_{M_k}	Theoretical utilisation of core k
$maxW\,k$	Maximum value of the sum of all elements of W for core k
$maxW$	Maximum value of the sum of all elements of W for all cores

Ecuación a optimizar:

$$maxW = \sum_{\forall k} maxW_k = \sum_{k=1}^{m} \sum_{\substack{\tau_i \in M_k \\ I_i \neq 0}} \sum_{\tau_j \notin M_k} I_j$$

$$\text{Minimise } maxW = \sum_{\forall k} maxW_k$$

Restricciones:

s.t:

$$\sum_{\forall k} O_{ik} = 1 \qquad \forall i$$

$$\sum_{i \in k} U_i \cdot O_{ik} = U_{M_k} \qquad \forall k$$

$$U_{M_k} \leq 1 \qquad \forall k$$

$$\sum_{\substack{\tau_i \in M_k \\ I_i \neq 0}} \sum_{\tau_j \notin M_k} I_j = maxW_k \qquad \forall k$$

$$O_{ik} \in \{0, 1\}$$

$$U_{M_k}, maxW_k \geq 0$$

Fase de alojamiento: Imin

Minimiza el límite superior (ub) de la interferencia que una tarea τj provoca en una tarea τi.

$$I_{j\to i}^{Tub} = A_i \cdot A_{j\to i} \cdot I_j$$

A_i Número de activaciones de la tarea τi

$A_{j\to i}$ Máximo número de activaciones que la tarea τj "cae" dentro de la de una activación de la tarea receptora τi

I_j Factor de interferencia de la tarea τj

$$A_{j\to i} = \left\lceil \frac{T_i - 1}{T_j} \right\rceil + K$$

$$K = \begin{cases} 0 & \text{If periods } T_i \text{ and } T_j \text{ are harmonics} \\ 1 & \text{Elsewhere} \end{cases}$$

Límite superior (ub) de la utilización de una tarea receptora:

$$U_i^{ub} = U_i + U_i^{iub} = U_i + \frac{\sum_{\tau_j \notin M_{\tau_i}} I_{j\to i}^{Tub}}{T_i}$$

Fase de alojamiento: Imin

Imin: MILP

SETS AND INDICES	
i, j	Task index $i, j \in \{0, 1, 2, \ldots, n-1\}$
k	Core index $k \in \{0, 1, 2, \ldots, m-1\}$
PARAMETERS	
C_i	Worst case execution time of τ_i
T_i	Period of τ_i
U_i	Utilisation of τ_i
H	Hyperperiod of the task set τ
$I^{Tub}_{j \to i}$	Maximum interference that τ_j can cause to τ_i during $(0, T_i)$
DECISION VARIABLES	
U^{iub}_i	Upper bound of the utilisation due to interference of τ_i
U^{ub}_i	Upper bound of the total utilisation of τ_i
U_{M_k}	Utilisation of M_k
O_{ik}	Allocation matrix. 1 if τ_i is allocated to core k and 0 otherwise.

Ecuación a optimizar:

$$U^{ub}_i = U_i + U^{iub}_i = U_i + \frac{\sum_{\tau_j \notin M_{\tau_i}} I^{Tub}_{j \to i}}{T_i}$$

$$\text{Minimise} \sum_{\forall i} U^{ub}_i$$

Restricciones:

$$\sum_{\forall k} O_{ik} = 1 \qquad \forall i$$

$$U^{iub}_i = \frac{\sum_{\tau_j \notin M_{\tau_i}} I^{Tub}_{j \to i}}{T_i} \qquad \forall i$$

$$U^{ub}_i = U_i + U^{iub}_i \qquad \forall i$$

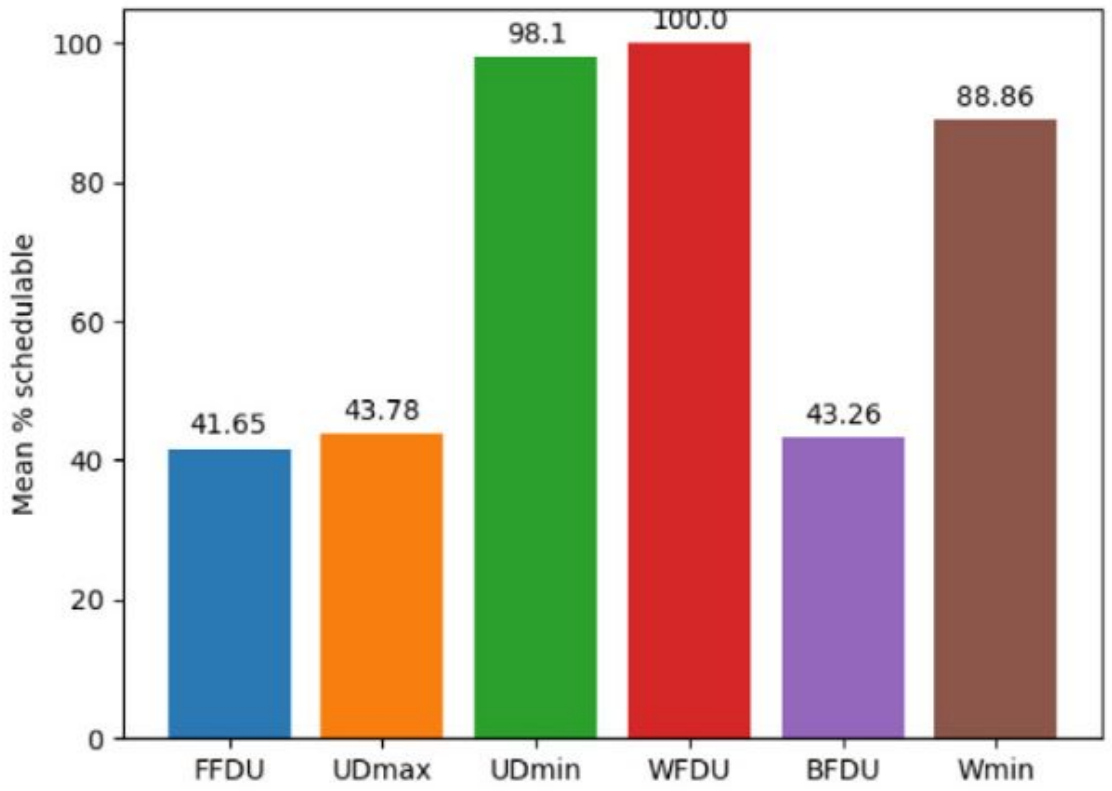
$$U_{M_k} = \sum_{\forall i} O_{ik} \cdot U_i \qquad \forall k$$

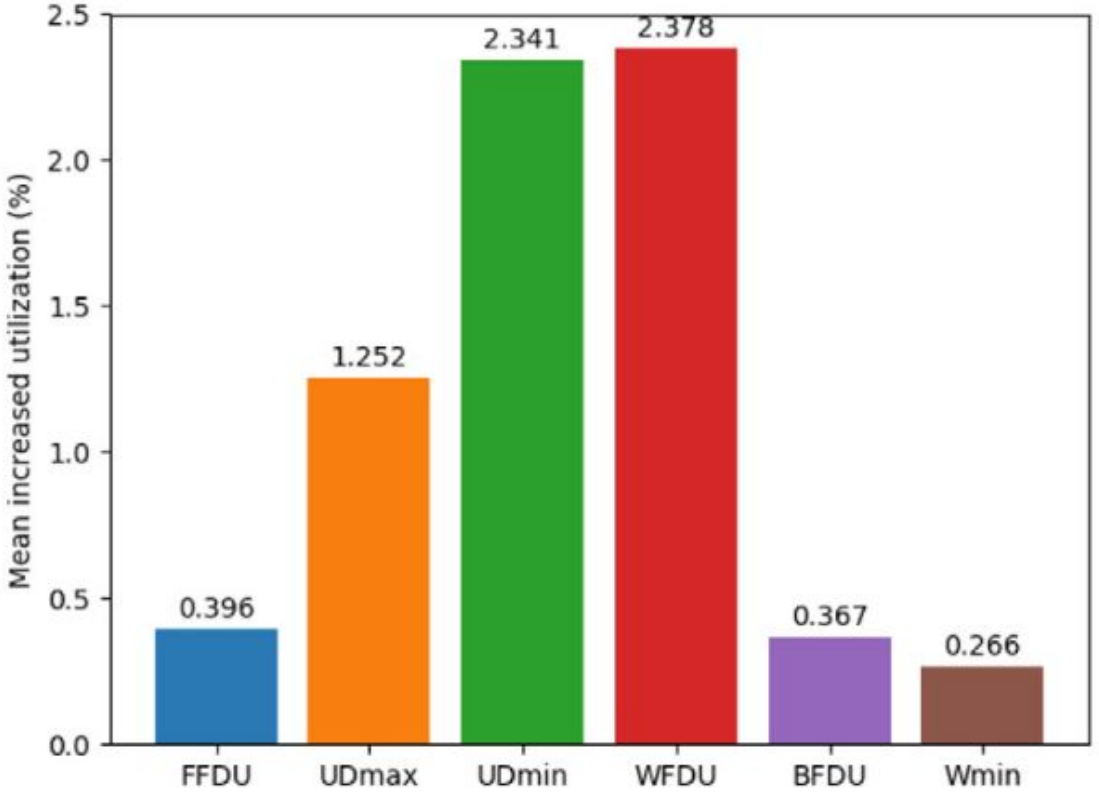
$$U_{M_k} \leq 1 \qquad \forall k$$

$$O_{ik} \in \{0, 1\}$$

$$U^{iub}_i, U^{ub}_i, U_{M_k} \geq 0$$

Conclusiones de las técnicas de alojamiento propuestas: UDmax/min y Wmin





Conclusiones de las técnicas de alojamiento propuestas: UDmax/min y Wmin

ELSEVIER

Journal of Systems Architecture
Volume 118, September 2021, 102223

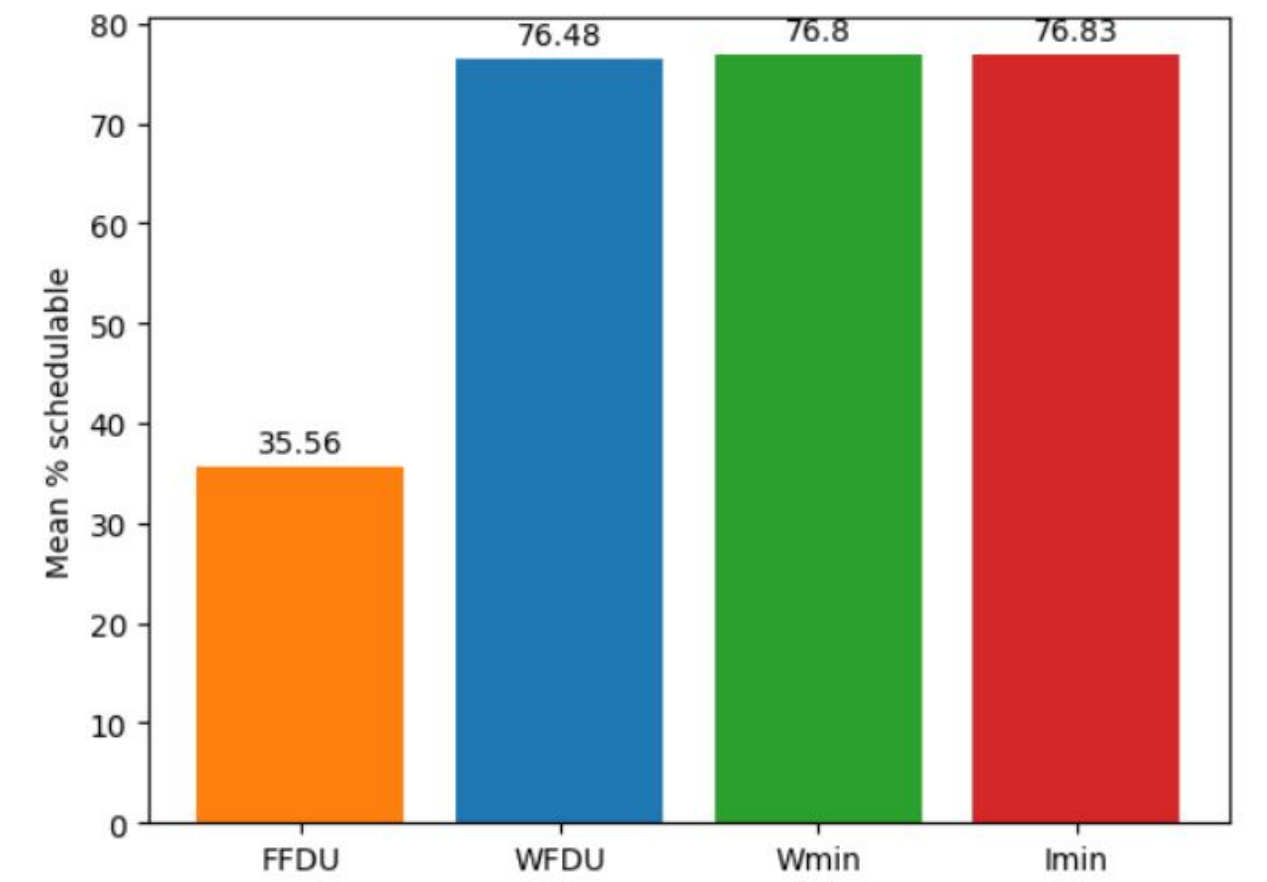
Hardware resources contention-aware scheduling of hard real-time multiprocessor systems ☆

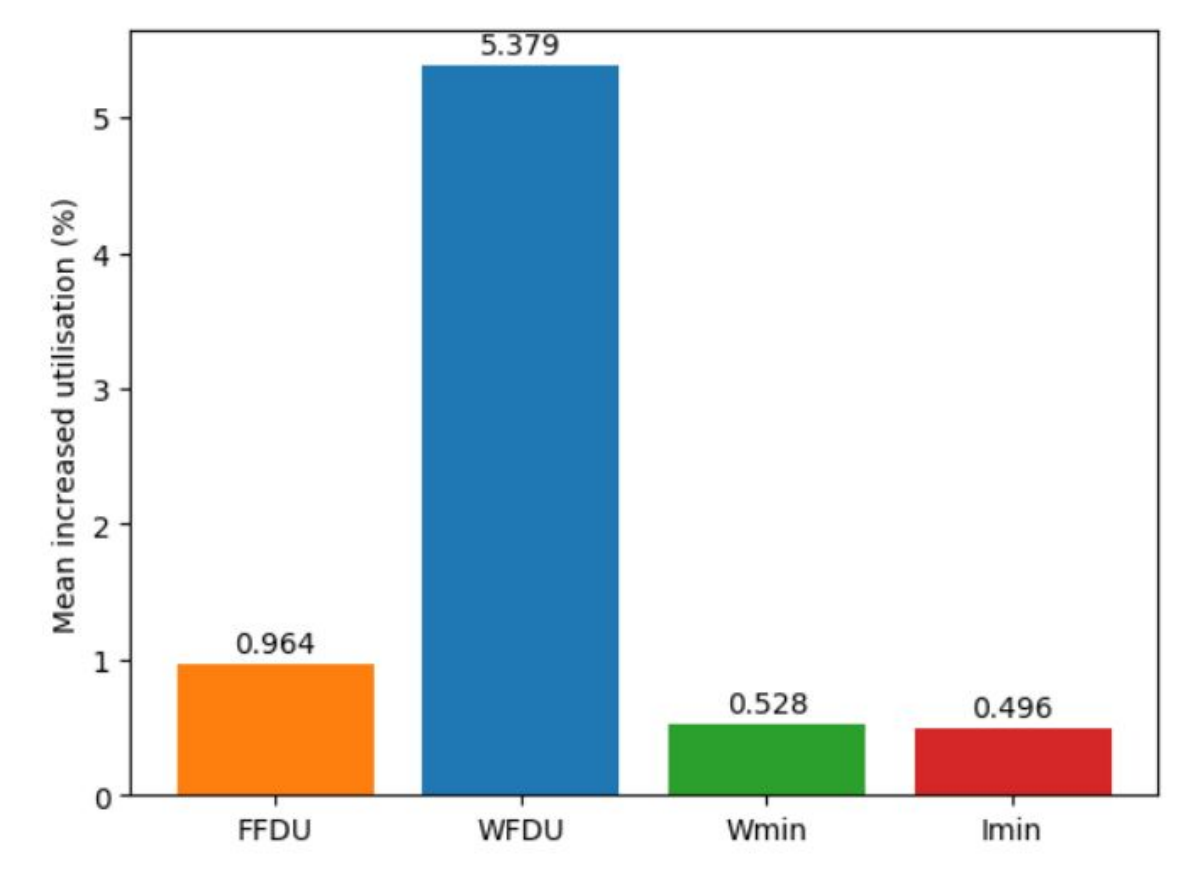
José María Aceituno, Ana Guasque, Patricia Balbastre, José Simó, Alfons Crespo

- Resultados publicados en:

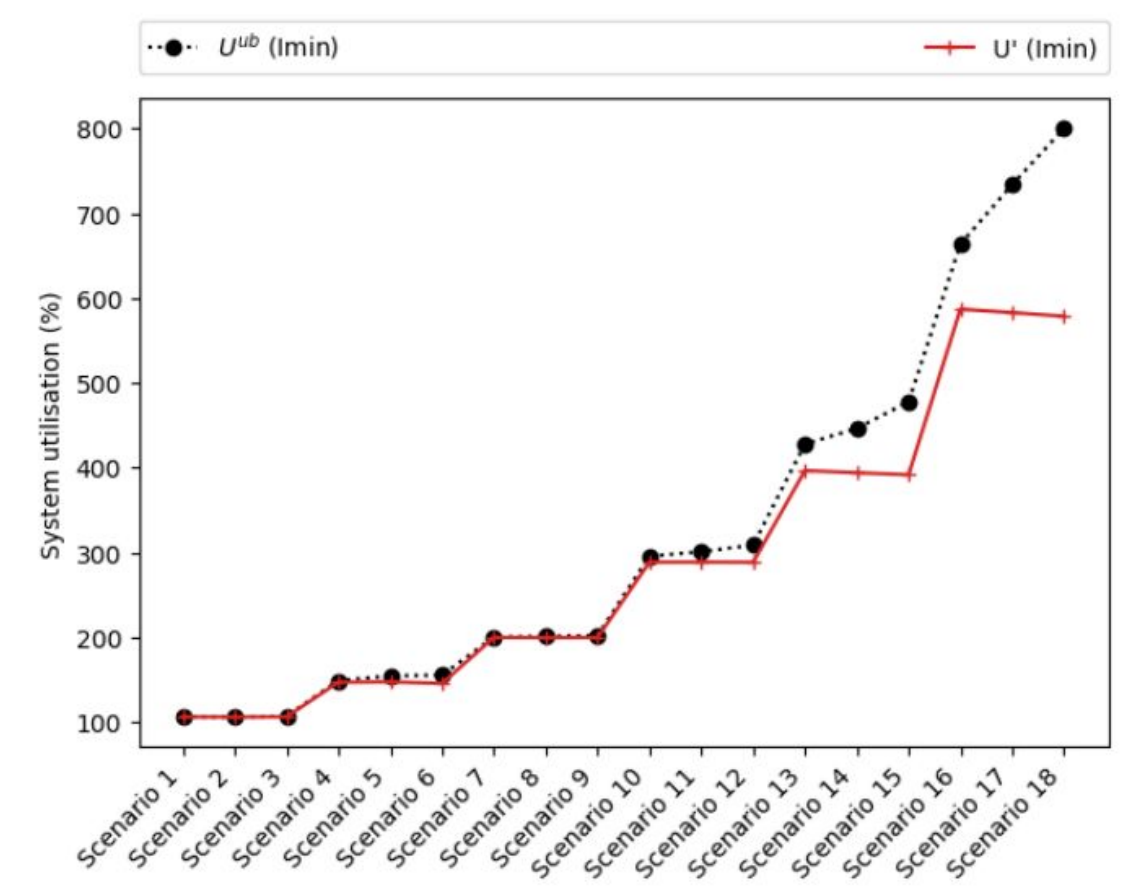
  José María Aceituno, Ana Guasque, Patricia Balbastre, José Simó, Alfons Crespo. Hardware resources contention-aware scheduling of hard *real-time multiprocessor systems*. The Journal of Systems Architecture (2021)

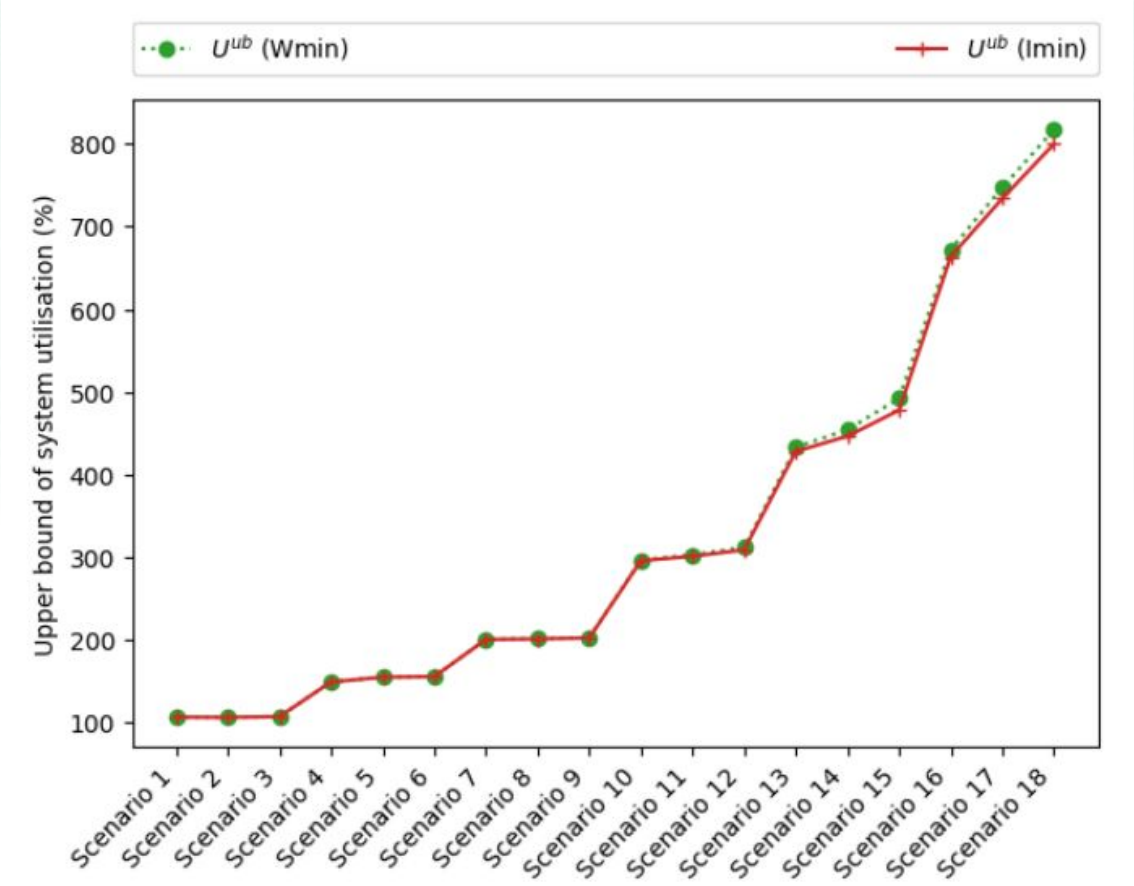
Conclusiones de las técnicas de alojamiento propuestas: Imin





Conclusiones de las técnicas de alojamiento propuestas: Imin





Conclusiones de las técnicas de alojamiento propuestas: Imin

Interference-Aware Schedulability Analysis and Task Allocation for Multicore Hard Real-Time Systems

by José María Aceituno †, Ana Guasque †, Patricia Balbastre *,†, José Simó † and Alfons Crespo †

Instituto de Automatica e Informatica Industrial (Ai2), Universitat Politècnica de València, Camino de Vera, s/n, 46022 Valencia, Spain

* Author to whom correspondence should be addressed.

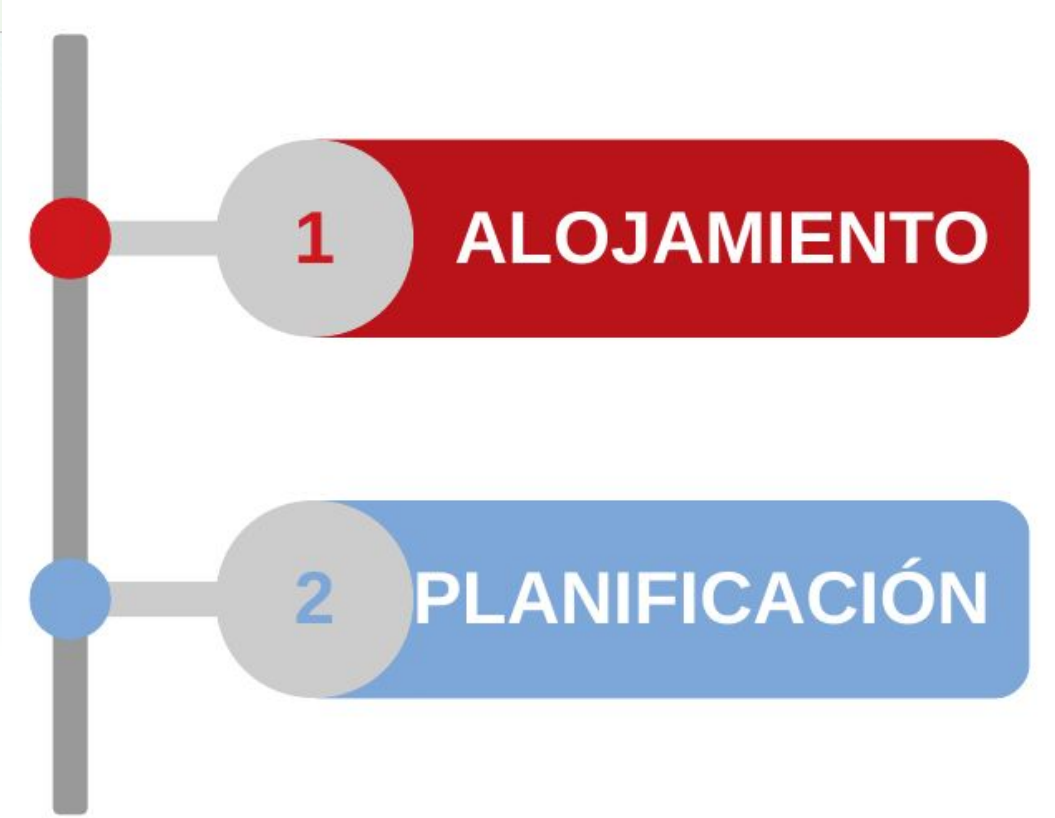
† These authors contributed equally to this work.

Electronics **2022**, *11*(9), 1313; https://doi.org/10.3390/electronics11091313

- Resultados publicados en:

  José María Aceituno, Ana Guasque, Patricia Balbastre, José Simó, Alfons Crespo. *Interference-Aware Schedulability Analysis and Task Allocation for Multicore Hard Real-Time Systems*. Electronics (2022)

Técnicas de planificación para reducir interferencia



Fase de planificación

- Existen técnicas bien conocidas para planificar sistemas multiprocesador.
- Pero se basan en proporcionar un sistema planificable: que se cumplan los plazos en cada activación de cada tarea
- ¿Y si queremos un plan factible pero en el que se minimice algún parámetro temporal?
  - Interferencia
  - Peor tiempo de respuesta
  - Cambios de contexto

→ Planificación mediante técnicas no convencionales

Fase de planificación

- Algoritmos basados en técnicas no convencionales:

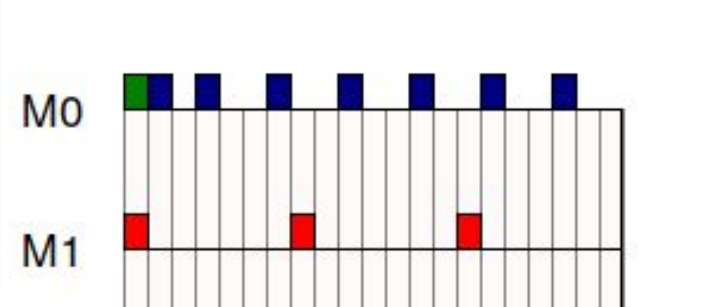
Programación lineal entera (LP)

- Definir **parámetros** y **variables** (qué tarea se ejecuta en cada instante de tiempo…)
- Sujeto a las **restricciones** de planificación de sistemas de tiempo real (cumplimiento de plazos…)
- Minimizar o maximizar uno o varios **objetivos** (tiempo de respuesta, cambios de contexto totales…)
- **Solver** utilizado: Gurobi 9.5

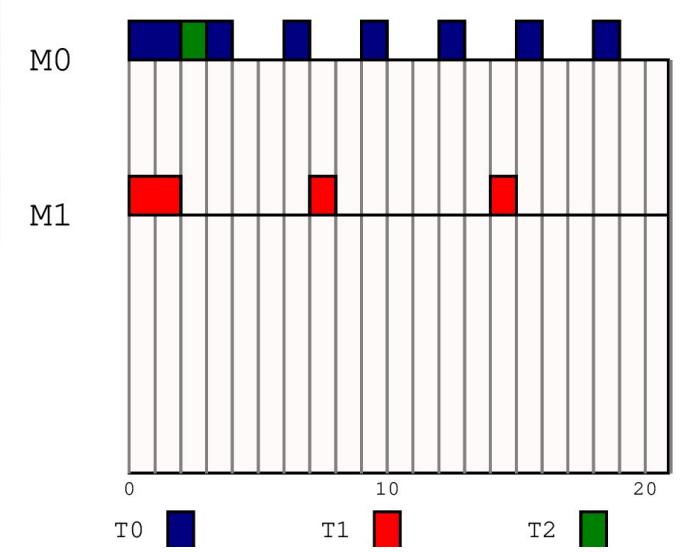
Planificación mediante técnicas no convencionales

- Ejemplo de sistema multiprocesador que minimiza interferencia usando el software de optimización Gurobi vs técnica convencional (EDF)

Gurobi



EDF



Planificación mediante técnicas no convencionales

- Inconveniente: El tiempo para conseguir la solución crece exponencialmente con el número de tareas, procesadores, utilización e hiperperiodo (m.c.m de los periodos de las tareas)
- Solución: dividir en problema en intervalos

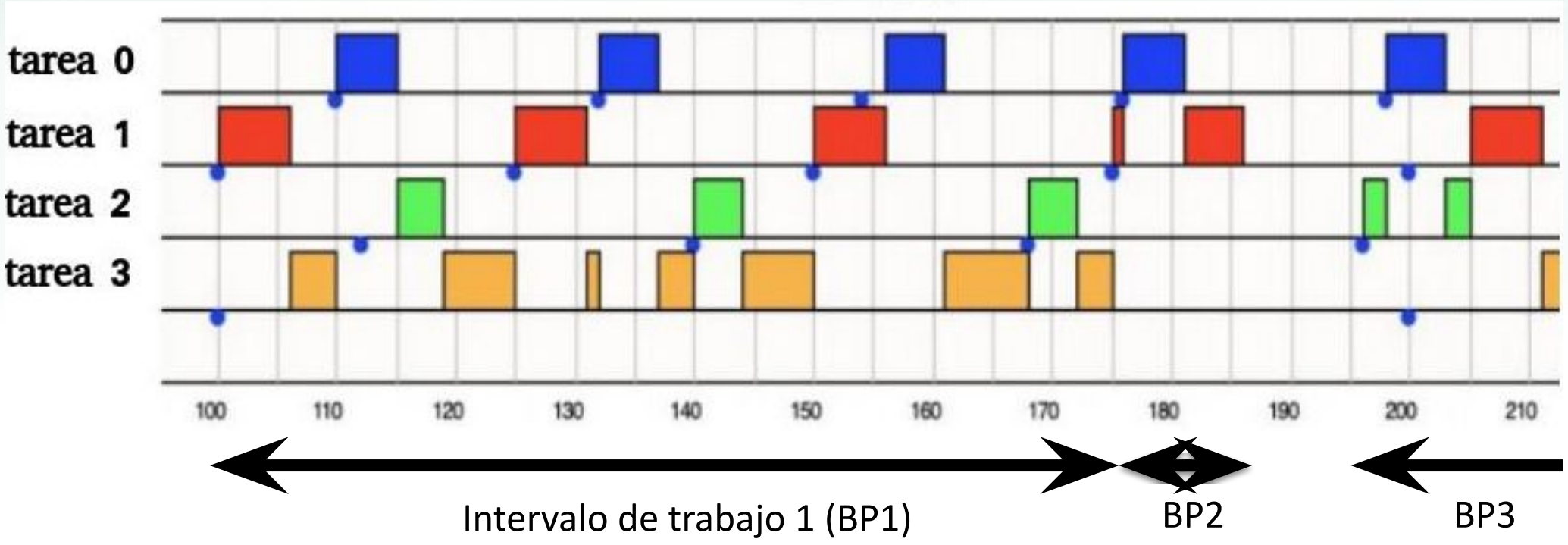
⬇

Intervalos de trabajo

- La ejecución se compone de una sucesión de intervalos de trabajo e intervalos ociosos hasta el hiperperiodo

Planificación mediante técnicas no convencionales

- Intervalos de trabajo



Planificación de Sistemas críticos

- Planificación mediante técnicas no convencionales
  - Incluso dividiendo en intervalos, sigue habiendo casos en los que no se obtiene solución (intervalos grandes)
  - En estos casos necesitamos una solución bien para:
    - Tener una planificación para el intervalo en cuestión
    - Tener una planificación que el optimizador pueda usar como solución inicial y tratar de encontrar una solución mejor
  - Esta planificación la obtenemos mediante el <u>planificador combinado</u>
    - Este planificador es capaz de aplicar una política de planificación diferente para cada intervalo de trabajo. De esta forma, se utiliza el algoritmo de planificación que obtenga menos cambios de contexto o interferencia en cada intervalo

Planificador Combinado

- El PC puede aplicar hasta 6 políticas de planificación distintas basadas en DM y EDF y sus variantes

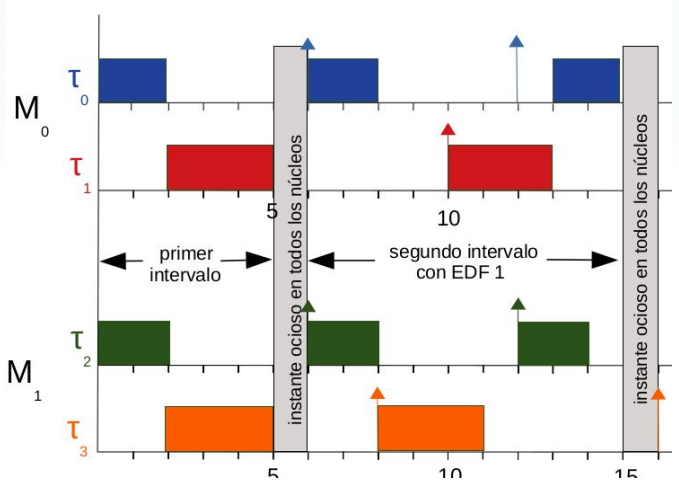
DM
Prioridad fija,
asigna mayor prioridad al plazo relativo más pequeño

EDF
Prioridad dinámica,
asigna mayor prioridad al plazo absoluto más cercano

- DM0, EDF0: algoritmos DM y EDF originales
- DM1, EDF1: algoritmos no expulsivos si el tiempo de cómputo de la tarea a expulsar es menor que la tarea más prioritaria
- DM2, EDF2: Algoritmos no expulsivos durante N unidades de tiempo

Planificador Combinado

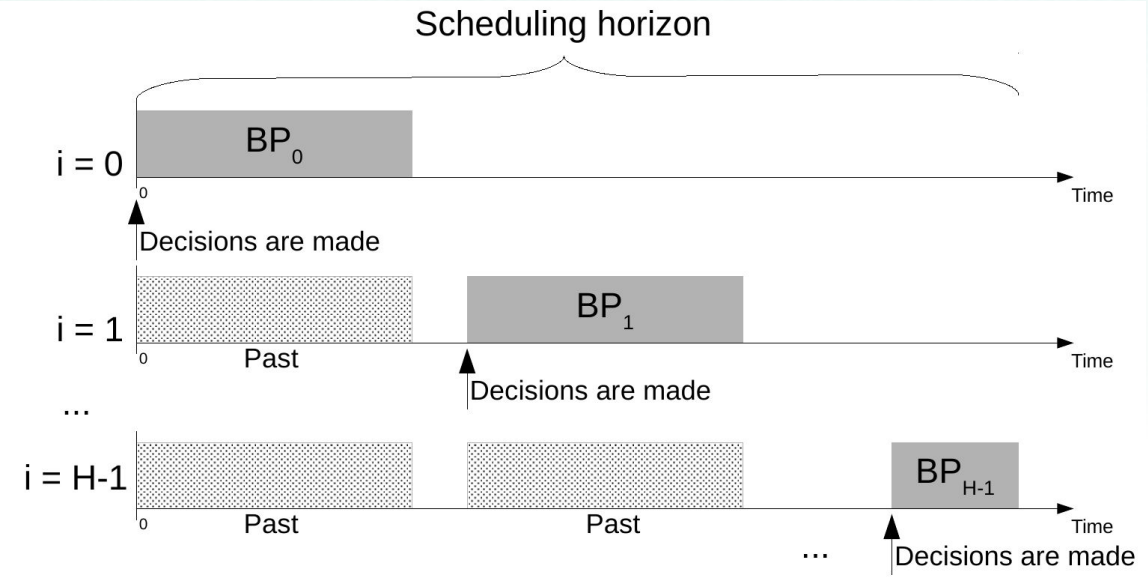
- Funcionamiento:
  1. Se planifica con las 6 variantes desde t=0 hasta llegar a un instante ocioso en todos los procesadores.
  2. Se elige la planificación que obtenga menor valor según el PC elegido. Hay dos versiones del PC según el parámetro temporal que queramos mejorar:
     - PC cambios de contexto o
     - PC interferencia.
  3. Se planifica de nuevo hasta encontrar el siguiente instante ocioso y así sucesivamente.



Planificación con técnicas no convencionales:

Rolling Horizon MILP Algorithm (RHMA)

- El concepto **Rolling Horizon** indica que el modelo se resuelve repetidamente para cada intervalo de trabajo (BP).
- Cada intervalo de trabajo representa una pequeña parte del horizonte (hiperperiodo)
- Se usa para reducir tiempo computacional para resolver grandes problemas con muchas variables.

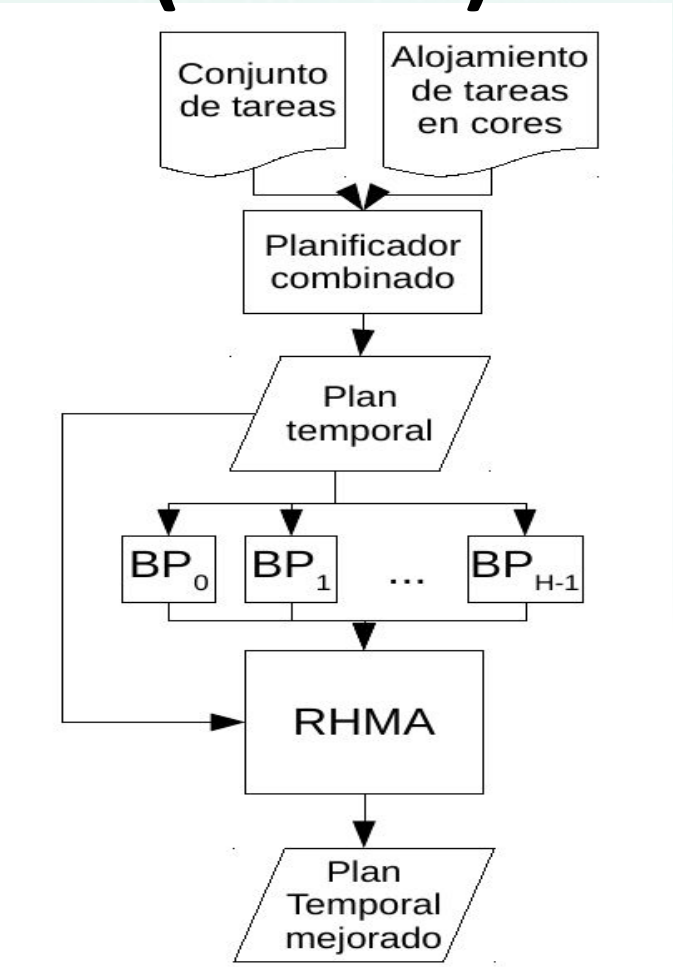


Planificación con técnicas no convencionales:

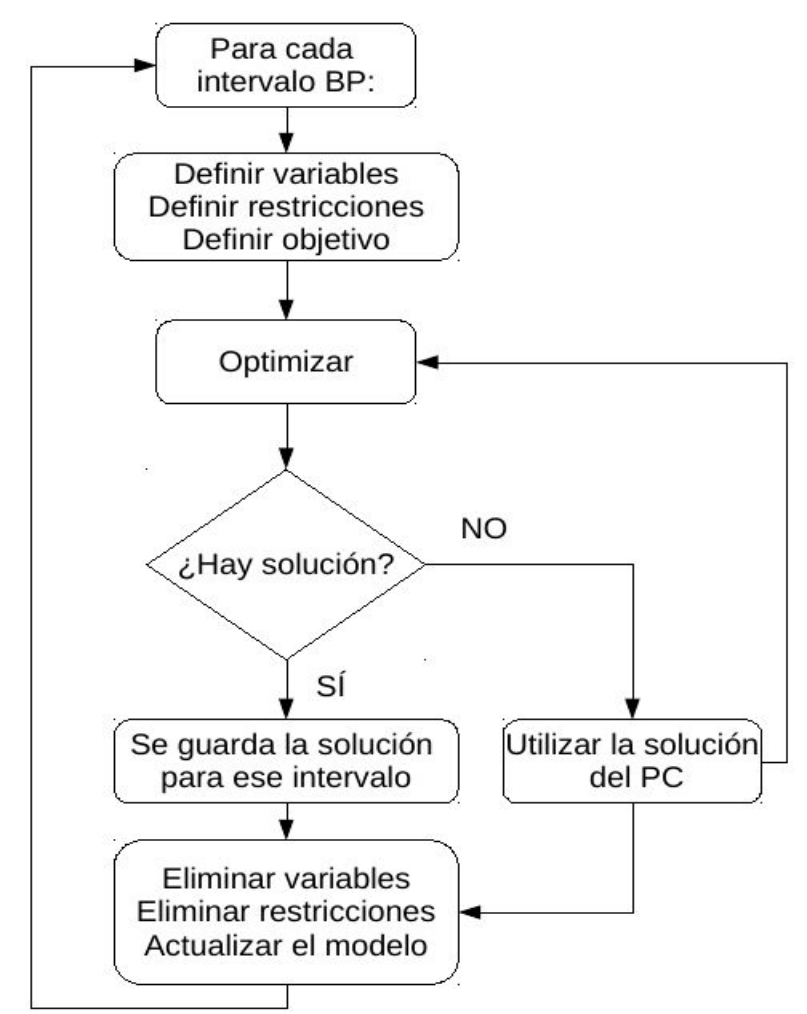
Rolling Horizon MILP Algorithm (RHMA)

Uso del Planificador Combinado en RHMA:

- Proporciona los intervalos BP
- Tiene una solución para cada intervalo en el caso de que Gurobi no sea capaz de encontrarla
- Se puede inyectar la solución inicial a Gurobi para ayudar al inicio de la búsqueda



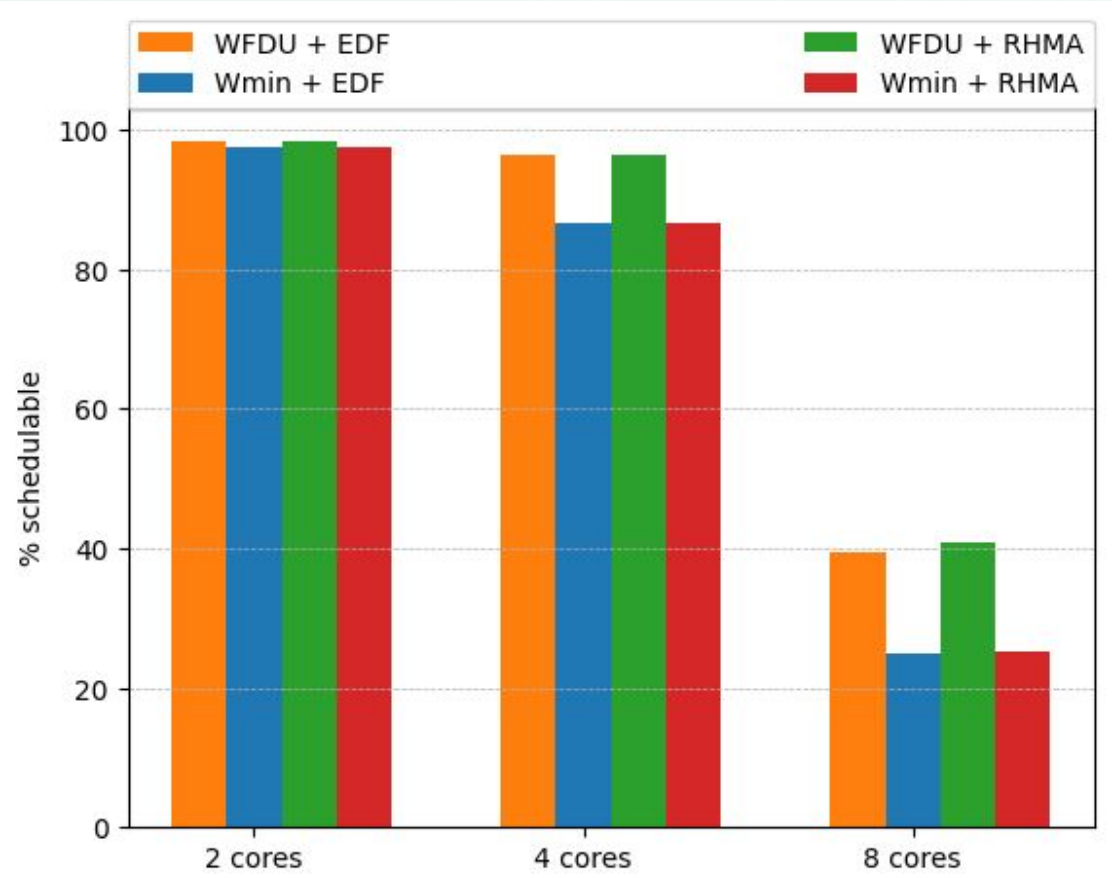
Algoritmo RHMA

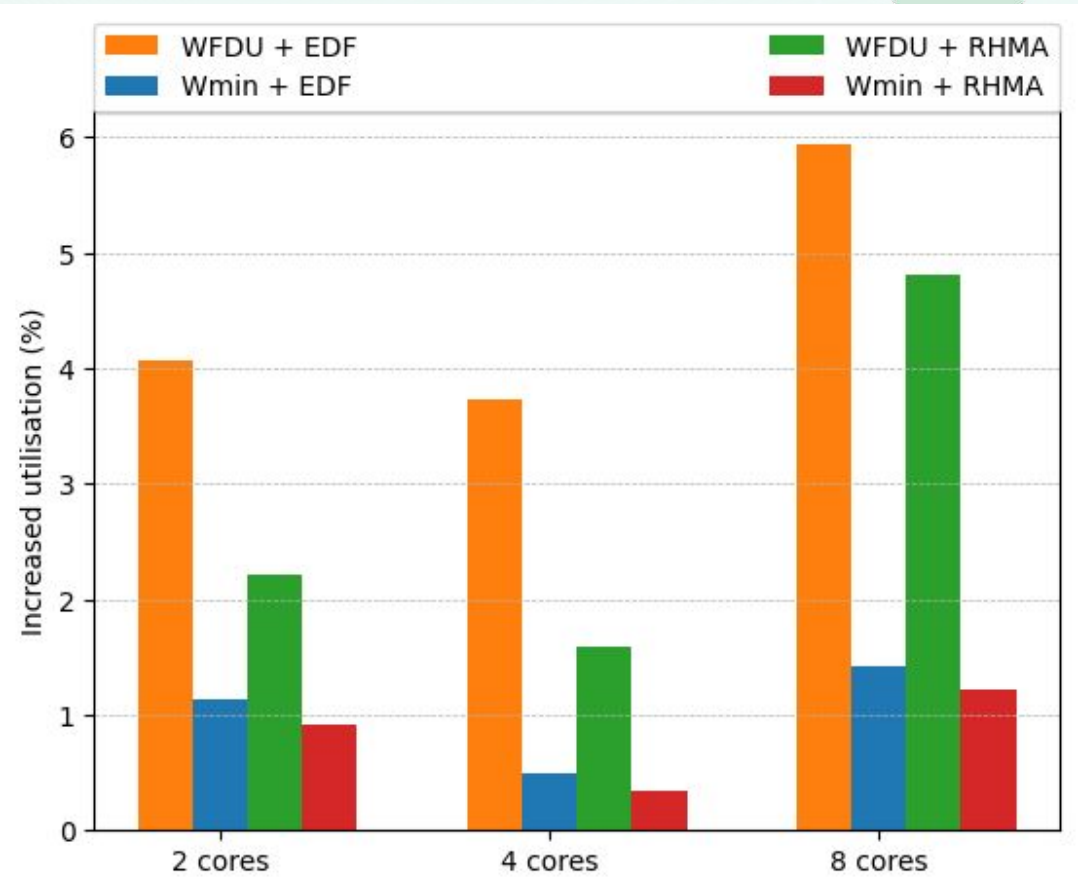


INPUT: Busy periods from Combined Scheduler, Task set and task-to-cores allocation
OUTPUT: Scheduling plan, σ
procedure ROLLING HORIZON MILP ALGORITHM
    Temporarily ordering of busy periods $\rightarrow [BP_0, ..., BP_{H-1}]$
    for $BP_h \in$ Busy Periods *[BP]* **do**
        Calculate parameters
        Define and reduce variable matrix x
        Define and reduce variable matrix m
        Define variable matrix w
        add Constraints
        set Objective and optimize
        if Optimal or Feasible **then**
            $\sigma_h = x_{iajt}$ (Save scheduling plan for BP_h)
        else
            Usage of the CS schedule in this BP
        if h $<$ H-1 **then**
            Remove constraints and variables
            Update the model

Evaluación del RHMA

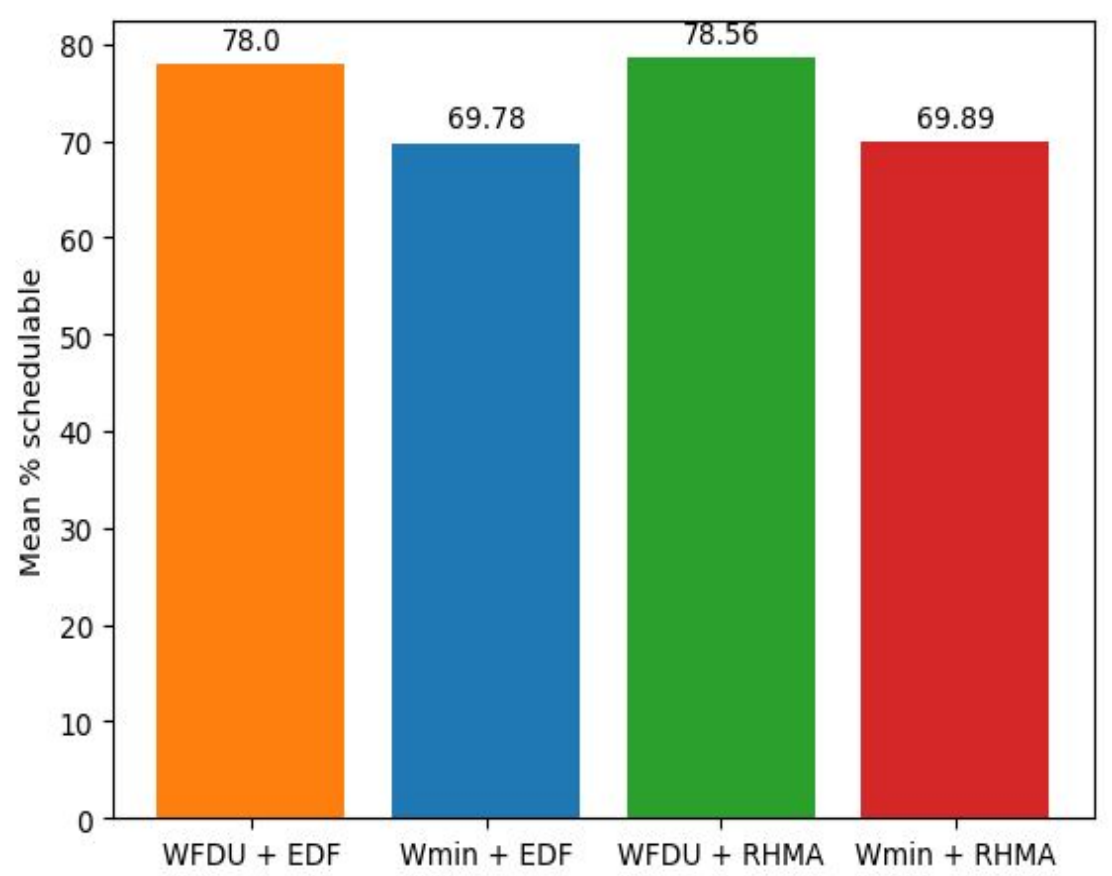
- Resultados

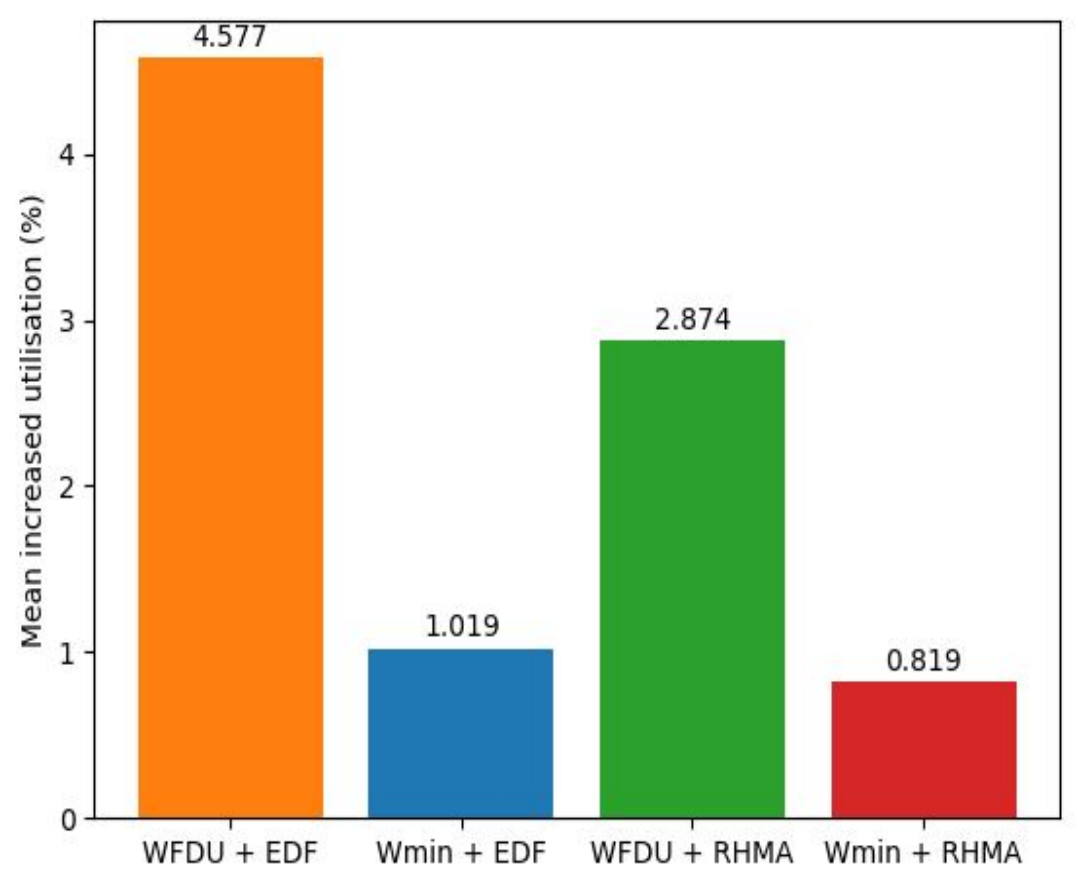




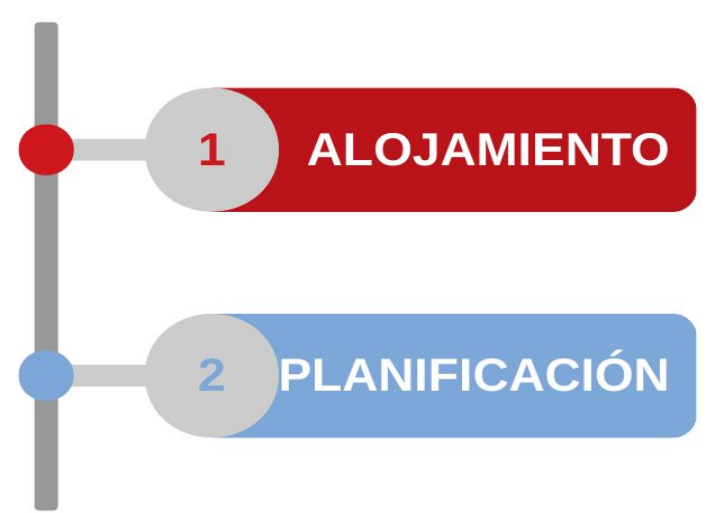
Evaluación del RHMA

- Resultados promedio





Conclusiones



OBJETIVO GENERAL

Actuar tanto en la fase de **alojar** tareas a procesadores como en la fase de **planificación** para reducir la interferencia en sistemas de tiempo real críticos multiprocesador.

Conclusiones

- Propuesta de técnicas que reducen la interferencia
  - Basadas en discrepancia: UDmax/UDmin
  - Basada en matriz de interferencia: Wmin
  - Basada en límite superior de utilización: Imin

Conclusiones

- Propuesta de test de planificabilidad para prioridades dinámicas
- Propuesta de técnicas no convencionales que reducen la interferencia:
  - Planificador combinado
    - Solución rápida que sirva como solución inicial para RHMA o como solución si RHMA no ha sido capaz de obtener una.
  - RHMA
    - Solución óptima

Trabajo futuro

- Test de planificabilidad basado en WCRT para prioridades fijas en sistemas multiprocesador con interferencia.
- Optimización combinada de otros parámetros temporales:
  - Potencia consumida
  - Cambios de contexto
- Estudiar el efecto de la longitud de los busy periods en el tiempo de respuesta del RHMA