2. PROGRAMACIÓN DE TRABAJOS EN INTERVALOS

2. PROGRAMACIÓN DE TRABAJOS EN INTERVALOS

2.1 INTRODUCCIÓN

En numerosos trabajos de planificación y programación de tareas, éstas poseen restricciones de tiempo motivadas por aspectos físicos de los materiales involucrados en la fabricación, por normas de proceso, por horarios de trabajo, o por otras imposiciones en los intervalos de procesamiento. En estos casos, la capacidad del sistema, es decir, la cantidad de recursos disponibles, juega un papel fundamental.

En este tipo de sistemas, la planificación de la capacidad trata tanto de la asignación de los recursos a las tareas atendiendo a un criterio de optimización (planificación operacional) como al cálculo de la capacidad necesaria para completar todas las tareas (planificación táctica). El enfoque del problema como de una sola etapa para la realización de las tareas se encuentra en multitud de disciplinas donde existen actividades programadas de antemano, es decir, donde se parte de un conjunto de tareas planificadas previamente y el análisis se centra en gestionar los recursos necesarios convenientemente para realizar dichas tareas.

La programación de trabajos en intervalos con una sola etapa de procesamiento, puede ser definida como sigue:

Sea un conjunto de n trabajos $J_1,\dots,J_n$ *cada uno con un intervalo* $[a_i,b_i]$ *en el que debe ser procesado, un tiempo de proceso* t_i*, un peso o prioridad* w_i *y perteneciente a una determinada clase de trabajo* d_i *dentro de un conjunto de D diferentes clases de trabajos* (Figura 1). *Para la realización de los trabajos se dispone de un conjunto de máquinas* $M=\{M_1,\dots, M_m\}$ *cada una con un intervalo de disponibilidad* $[i_j, f_j]$*, un coste de utilización* u_j*, y perteneciente a una determinada clase de máquina* c_j *dentro de un conjunto de C clases diferentes de máquinas. La compatibilidad entre clases de trabajos y máquinas se expresa a través de una matriz de compatibilidad* L_{DxC} *definida como sigue:*

$$L_{DXC} = L(d,c) = \begin{cases} 1 \text{ si un trabajo de clase d puede ser procesado por una máquina de clase c} \\ 0 \text{ en otro caso} \end{cases}$$

Hay que tener en cuenta que cada máquina no puede procesar más de un trabajo al mismo tiempo y, de forma genérica, cada trabajo se procesa de forma ininterrumpida sobre alguna de las máquinas del sistema.

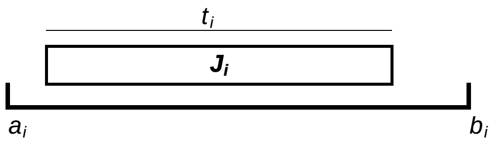


Figura 1

Siendo la programación de trabajos en intervalos un caso de programación de operaciones, y usando la notación habitualmente empleada en dicho campo, el caso de realización de tareas en una sola etapa responde a las siguientes características:

- Sistema de m máquinas en paralelo P_m: $M=\{M_1,\ldots,M_n\}$.
- Fecha de llegada al sistema o disponibilidad: a_i.
- Fecha límite de procesamiento: b_i.
- Duración del trabajo: t_i.

Como caso genérico, en la Figura 2 se presenta la asignación de trabajos a máquinas para los datos de las Tablas 1 y 2. Para el problema se dispone de 10 trabajos de 2 clases (d_i) y 3 máquinas de 2 clases (c_j), que atienden a la siguiente matriz de compatibilidad:

$$L_{2x2} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}$$

J_i	a_i	b_i	t_i	w_i	d_i
1	0	7	4	1	1
2	0	8	4	2	2
3	1	10	7	3	1
4	2	12	9	4	1
5	5	14	7	3	2
6	9	15	5	3	2
7	9	19	5	2	1
8	12	20	6	2	2
9	14	19	3	4	1
10	14	21	5	4	1

Tabla 1: Características de los Trabajos

M_j	s_j	f_j	u_j	c_j
1	0	15	3	1
2	0	20	4	1
3	2	21	5	2

Tabla 2: Características de las Máquinas

Siendo el número de máquinas una característica del problema, el objetivo a resolver sería táctico. En la asignación realizada se procesan 8 trabajos, quedando los trabajos J_5 y J_9 sin realizar. Se utilizan las 3 máquinas por lo que el coste es de 12 unidades ($u_1+u_2+u_3$). La suma de los pesos de los trabajos realizados es de 21 unidades $(w_1+w_2+w_3+w_4+w_6+w_7+w_8+w_{10})$.

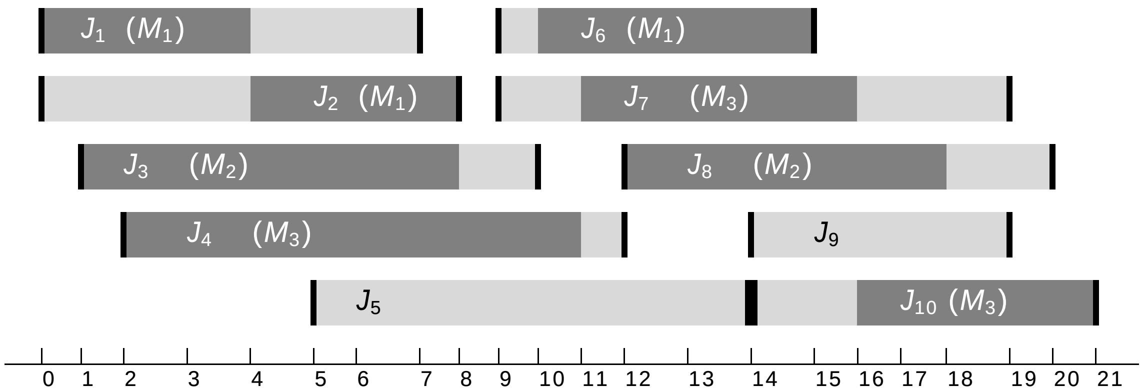


Figura 2

2.2 CLASIFICACIÓN

La programación de trabajos en intervalos aparece en la literatura con diferentes variaciones, dependiendo de los intervalos de tiempo que se tenga para el procesamiento de las tareas, del número de recursos (máquinas) de distintas clases existente, de las características que motivan las diferentes clases de trabajos y recursos y, evidentemente, del objetivo definido. La Figura 3 resume la clasificación del problema que se desarrolla más adelante:

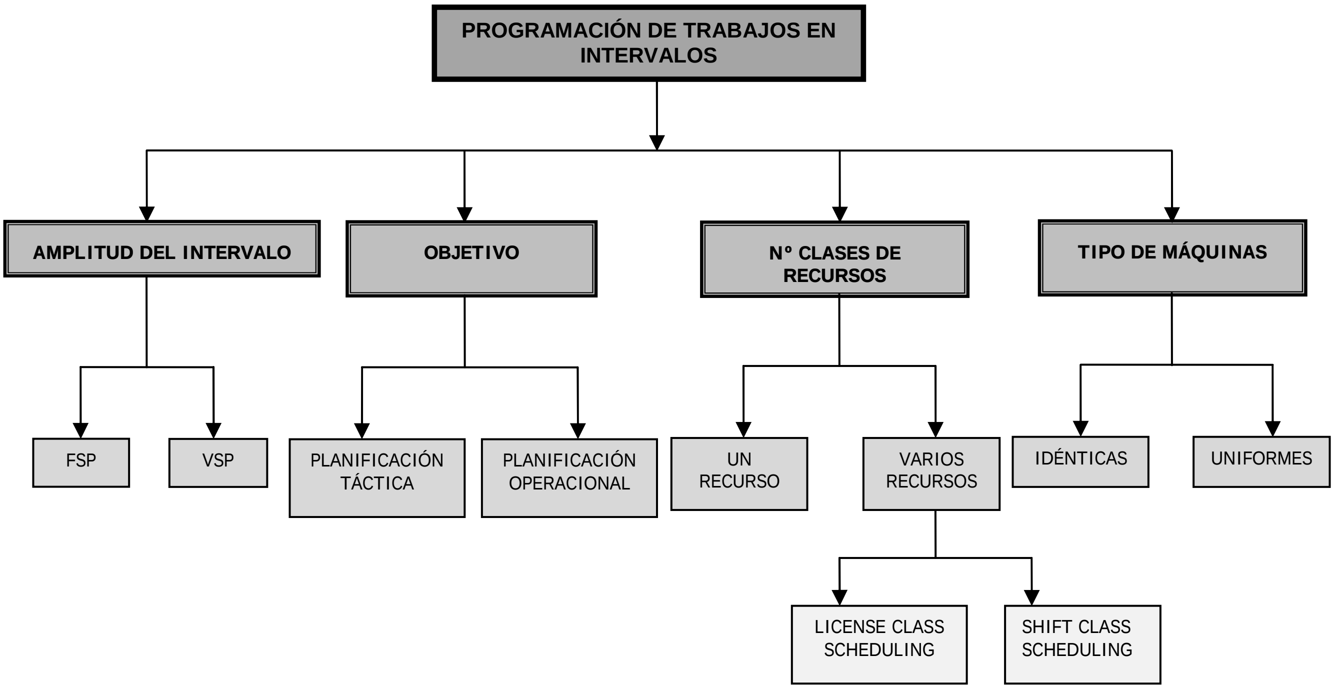


Figura 3

2.2.1 FSP FRENTE A VSP

Una primera clasificación de estos sistemas se produce según la amplitud del intervalo para el procesamiento de las tareas. Según ello, existe un primer tipo de problema donde el tiempo de proceso de las tareas coincide exactamente con la amplitud del intervalo, recibiendo este grupo el nombre de Fixed Job Scheduling Problem (FSP). Si el tiempo de proceso de alguna de las tareas es menor que la amplitud del intervalo, nos encontramos ante un Variable Job Scheduling Problem (VSP).

2.2.2 PLANIFICACIÓN TÁCTICA FRENTE A PLANIFICACIÓN OPERACIONAL

Se habla de Planificación Táctica de la capacidad cuando se pretende averiguar el número de recursos necesarios, de manera optimizada, para

atender todas las tareas. La Planificación Operacional de la capacidad es cuando, a partir de un conjunto de recursos dado, se realiza la asignación de los mismos a tareas con objeto de maximizar el peso total de las tareas completadas, pudiendo quedar tareas sin realizar.
Esta clasificación no es totalmente excluyente, y se presentan casos que se resuelven como una combinación de ambas planificaciones.

2.2.3 LICENSE CLASS SCHEDULING FRENTE A SHIFT CLASS SCHEDULING

Los primeros trabajos sobre este tema consideraron una única clase de recurso, de forma que cada tarea podía ser procesada por cualquier recurso. Sin embargo, el caso más interesante es considerar varias clases de recursos, apareciendo entonces dos variantes:
License Class Scheduling trata el caso en que los recursos están disponibles durante todo el horizonte de planificación, estando la compatibilidad entre tareas y recursos motivada por aspectos técnicos. En problemas de tipo Shift Class Scheduling, cada recurso está disponible solo durante un intervalo o conjunto de intervalos de tiempo, y podrá procesar únicamente las tareas que se produzcan dentro de dichos intervalos.

2.2.4 MÁQUINAS IDÉNTICAS FRENTE A UNIFORMES

Las máquinas pueden ser de dos tipos: idénticas o uniformes. La diferencia recae en que, en caso de que sean idénticas, tal como su nombre indica, todas las propiedades de las mismas coinciden. En caso de que sean uniformes, por el contrario, éstas presentan una velocidad de procesado distinta entre sí. Es decir, de procesar un mismo trabajo, dos máquinas idénticas lo harían en el mismo tiempo, mientras que dos máquinas uniformes lo harían en distinto tiempo.

2.2.5 VALOR DEL TRABAJO (W_i)

Es habitual asignar un valor w_i a cada trabajo que mida el beneficio derivado de completar el mismo. En ocasiones, se asigna el mismo valor a todos los trabajos, que es equivalente a asignarles a todos el valor unidad. Cuando no se consideran valores distintos para los diversos trabajos, los problemas tanto FSP como VSP con objetivo operacional pueden ser modelados como un problema de conjunto independiente de nodos en un grafo.

2.2.6 PROPIEDAD DE INTERRUMPIR EL PROCESAMIENTO (PREEMPTION)

En escenarios en que se considera admisible que el trabajo sea procesado por más de una máquina, aparece el Preemption o propiedad de interrumpir el procesamiento del trabajo en una máquina. El trabajo que esté siendo procesado por una máquina puede así saltar a otra para seguir siendo procesado por la última.

2.2.7 RESTRICCIONES ASOCIADAS AL USO DE LOS RECURSOS

Tradicionalmente se ha hablado de restricciones Spread-Time y Working-Time. Las restricciones Spread-Time imponen un límite superior al tiempo entre comienzo y finalización del procesamiento de trabajos en una máquina. Las restricciones Working-Time consideran un tiempo límite de actividad para cada máquina. Tanto esta restricción como la anterior aparecen en los casos en los que los recursos son consumibles o bien representan entidades humanas.

En este proyecto se incluyen nuevas restricciones, cercanas a las de tipo Working-Time, asociadas a la distribución de los trabajos asignados a los recursos. La restricción 5.2[1] hace referencia al número máximo de turnos de trabajo que puede realizar un trabajador en una semana. La restricción 5.4 controla que dos turnos del mismo día no los realice un mismo trabajador. Las restricciones 5.5 y 5.6 fuerzan a que si un trabajador hace un turno de día, no realice el siguiente turno de noche y vice-versa.

2.3 ENFOQUES CLÁSICOS DE LA PROGRAMACIÓN DE TRABAJOS FIJOS (FSP)

En el marco de la programación de trabajos en intervalos, el escenario que ha recibido más atención en la literatura es el de Programación de Trabajos Fijos (FSP), incluyendo variedad en las clases de recursos y trabajos, lo cual incrementa la dificultad de resolución.

Dependiendo de si contamos con una o varias clases de recursos, el FSP se plantea de distinta manera. Por otra parte, el hecho de enfrentarnos a un objetivo táctico u operacional también resulta determinante en la elección de métodos de resolución.

Cabe destacar que entre las técnicas de resolución presentes en la literatura para este tipo de casos destacan tanto métodos de búsqueda exacta de la solución como de búsqueda aproximada.

A continuación se comentan los enfoques habituales del problema FSP junto con los métodos de resolución.

2.3.1 UNA CLASE DE MÁQUINA, OBJETIVO TÁCTICO

En el problema FSP, en su versión más simple, es decir sin considerar clases de trabajos ni máquinas, se dispone de *n* trabajos con fechas de inicio y tiempos de procesamiento dados. Para un trabajo J_i, se define s_i como el instante de comienzo y f_i como el de finalización, siendo $t_i = s_i - f_i$ la duración del mismo (Figura 4).

[1] Las restricciones figuran en el Capítulo 5.

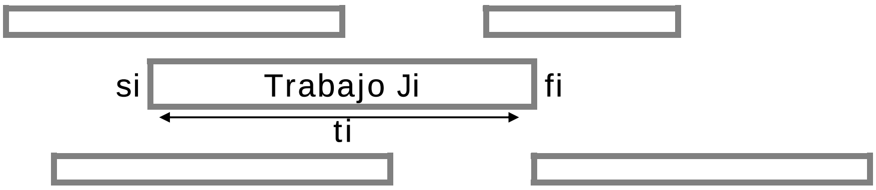


Figura 4

Los trabajos serán realizados por máquinas en paralelo (P_m). En el caso de objetivo táctico, se trata de determinar el número de máquinas para poder procesar todos los trabajos. La solución del problema se obtiene como una consecuencia directa del teorema de Dilworth[2] sobre la descomposición de conjuntos parcialmente ordenados, donde se establece que, en cualquier conjunto parcialmente ordenado, el número mínimo de cadenas disjuntas que albergan todos los elementos es igual al número mayor de elementos mutuamente no relacionados en el conjunto. El problema fue considerado en primer lugar por Danzing y Fulkerson[3], y posteriormente por Gupta et al[4] y Nakajima et al[5], mostrando algoritmos de resolución de orden $O(n \log n)$. Gertsbakh y Stern[6] también definen el FSP como un caso particular del problema de Dilworth y proponen un algoritmo basado en la construcción secuencial de programas de trabajo para máquinas, hasta agotar todos los trabajos.

El cálculo del mínimo número de máquinas se basa en el grado de solapamiento de los trabajos en el tiempo. Este último se define como el número de trabajos que se procesan en un instante t del horizonte de planificación, según la expresión $L_t=|\{J_i/s_i \leq t<f_i\}|$. El número mínimo de máquinas necesarias será el mayor valor hallado del grado de solapamiento para el horizonte de planificación, según la expresión $L = \max\{L_t \ / \ 0 \leq t \leq T\}$, con $[0,T]$ el horizonte para el cual se produce la ejecución de todos los trabajos.

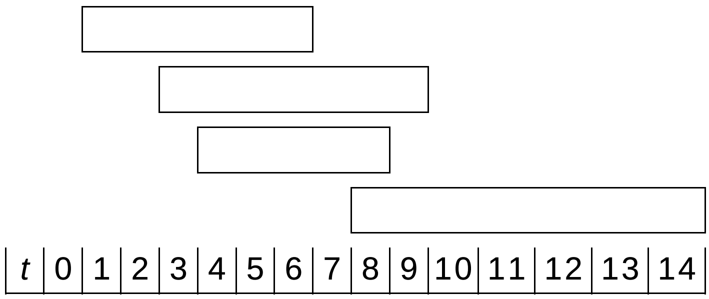
---

[2] En Dilworth, 1950.
[3] En Danzing y Fulkerson, 1954.
[4] En Gupta et al, 1979.
[5] En Nakajima et al, 1982.
[6] En Gertsbakh y Stern, 1978.

En este enfoque, la resolución del problema FSP se limita al cálculo del grado de solapamiento o simultaneidad de los trabajos en el tiempo, según el siguiente lema[7]:

<u>Lema</u>: *Una programación admisible para todos los trabajos existe si y sólo si el máximo grado de simultaneidad de los trabajos es menor o igual al número de máquinas m*.

Un ejemplo con 4 trabajos, representados mediante barras horizontes, y el cálculo del mayor grado de simultaneidad, se presenta en la Figura 5.



El cálculo del grado de simultaneidad sería el siguiente:

T	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**
L_t	0	1	1	2	3	3	3	2	3	2	1	1	1	1	1

L = máximo{0, 1, 1, 2, 3, 3, 3, 2, 3, 2, 1, 1, 1, 1, 1} = 3

Figura 5

Es claro que sólo si el número de máquinas es mayor o igual a 3 se podrían realizar todos los trabajos.

El problema también puede resolverse mediante un simple algoritmo de asignación avaricioso (Greedy) que obtiene además la asignación óptima de trabajos a máquinas[8]. Si bien dicho algoritmo asume un número de

[7] En Kroon et al, 1995.
[8] Fishetti et al, 1987.

máquinas ilimitado, esto es equivalente a tomar como cota superior el número de trabajos.

La estrategia que sigue el algoritmo consiste en recorrer el conjunto de trabajos, ordenados por fecha de inicio s_i, e ir asignando una máquina libre a cada trabajo. Las máquinas que ya hayan realizado algún trabajo se denominan máquinas activas (M_A), y son éstas las primeras que explora el algoritmo. El trabajo será asignado a cualquiera de éstas que esté libre (M_L). Si estuvieran todas ocupadas, se activaría una nueva máquina para realizar el trabajo. Inicialmente ambos conjuntos M_A y M_L están vacíos. El pseudocódigo para el algoritmo Greedy devuelve el número de máquinas utilizadas "a" (Figura 6).

```
Procedimiento GREEDY
        M_A = ∅
        M_L = ∅
        a = 0
        Para cada J_j ∈ J
          *Actualizar_Conjunto_M_L (s_i)
          Si M^L = ∅ entonces
            a = a + 1
            M_A = M_A + {M_a}
            Asigna J_j a M_a
          si no
            Selecciona una Máquina M_i ∈ M^L
            M_L = M_L - {M_i}
            Asigna P_j a M_i
          Fin Si
        Fin Para
Fin
```

Figura 6

Como una variante a la formulación general del problema FSP en este escenario, Fishetti et al[9] plantean el problema añadiendo restricciones de *Spread-Time* y *Working-Time*.

[9]En Fishetti et al, 1987, 1989, 1992.

2.3.2 UNA CLASE DE MÁQUINA, OBJETIVO OPERACIONAL

Cuando el número de máquinas es menor que el mayor grado de simultaneidad de los trabajos, el objetivo del problema pasa a ser encontrar el subconjunto de trabajos a ser procesados. Cada trabajo lleva asociado un valor o peso w_i, que representa el beneficio asociado a la realización del mismo, y ahora la selección atiende al criterio de procesar el conjunto de trabajos que de un máximo valor total según sus pesos. Encontramos por tanto un problema de secuenciación en el que las P_m máquinas en paralelo no realizan todos los trabajos, sino solamente un subconjunto de ellos, elegido en base a los pesos de los mismos.
La formulación del problema es la siguiente:

Variables:

x_j	Variable binaria {0,1}, que toma el valor de la unidad si se realiza el trabajo *i*.

Datos:

w_i	Peso asociado al trabajo *i*.
s_i	Instante de comienzo del trabajo *i*.
f_i	Instante de finalización del trabajo *i*.

Parámetros:

m	Número de máquinas del sistema.

Restricciones:

$$\sum_{\{i/s_i \le t \le f_i\}} x_i \le m \quad t \in T \qquad (2.1)$$

Imponen que para cada instante de tiempo $t \in [0...T]$ no haya más trabajos en proceso que máquinas.

Función Objetivo:

$$Max \sum_{i=1}^{n} w_i x_i \qquad (2.2)$$

Maximiza el peso total de los trabajos procesados.

Modelo:

$$Max \sum_{i=1}^{n} w_i x_i$$

$$\sum_{\{i/s_i \le t \le f_i\}} x_i \le m \quad t \in T$$

Sin embargo, si queremos obtener la asignación concreta de trabajos a máquinas, la cual no proporciona el modelo anterior, es necesario modificar dicho modelo añadiendo variables que recojan la asignación de cada trabajo a cada máquina del sistema. El modelo se ve modificado de la siguiente forma:

Variables:

x_j	Variable binaria {0,1}, que toma el valor de la unidad si se realiza el trabajo *i*.
x_{ij}	Variable binaria {0,1}, que toma el valor de la unidad si el trabajo *i* es asignado a la máquina *j.*

Datos:

w_i	Peso asociado al trabajo *i*.
s_i	Instante de comienzo del trabajo *i*.
f_i	Instante de finalización del trabajo *i*.

Parámetros:

m	Número de máquinas del sistema.

Restricciones:

$$\sum_{j=1}^{m} x_{ij} \le x_i \quad \forall i \tag{2.3}$$

Activa como procesado un trabajo que ha sido asignado a una máquina.

$$x_{ij} + x_{kj} \le 1 \qquad i = 1\ldots n-1;\; j = 1\ldots m;\; \{k > i : s_k < f_j\} \tag{2.4}$$

Controlan que dos trabajos solapados no sean procesados por la misma máquina.

La función objetivo no varía respecto al modelo anterior.

Modelo:

$$Max \sum_{i=1}^{n} w_i x_i$$

$$\sum_{j=1}^{m} x_{ij} \le x_i \quad \forall i$$

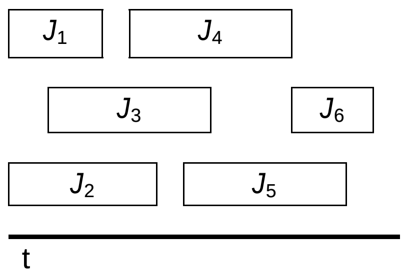
$$x_{ij} + x_{kj} \le 1 \qquad i = 1\ldots n-1;\; j = 1\ldots m;\; \{k > i : s_k < f_j\}$$

Con una única clase de máquinas y trabajos, el FSP ha sido considerado, entre otros, por Arkin y Silverberg[10], Kolen et al[11] y Gabrel[12], quienes muestran que puede ser resuelto en tiempo polinomial por un algoritmo de flujo a coste mínimo. Kroon et al[13] proponen la resolución mediante un algoritmo de flujo a coste mínimo en un grafo dirigido $G(N,A)$, construcción más directa que la propuesta por los autores anteriores. Para la formación del grafo, se define un conjunto de instantes temporales ordenado cronológicamente y formado por los instantes de comienzo y finalización de los n trabajos. Es decir, $\{t_r \,/\, r = t_0\ldots R\} = \{s_i, f_i \,/\, i = 1\ldots n\}$. El conjunto $\{t_r, r = t_0\ldots R\}$ constituye el conjunto N de nodos del grafo G. Cada trabajo i tiene asociado un arco desde el nodo que representa el instante de comienzo, s_i, hasta el nodo que representa el instante de finalización, f_i, con

[10] En Arkin y Silverberg, 1985.
[11] En Kolen et al, 1987.
[12] En Gabrel, 1995.
[13] En Kroon et al, 1995.

coste igual a $-w_i$ y una unidad de capacidad. Existe además un arco de coste cero y capacidad ilimitada desde cada nodo t_r a t_{r+1}, $r = 1...R-1$. Se inyectan m unidades de flujo en el primer nodo del grafo, las cuales serán absorbidas por el último nodo de éste. Un trabajo será procesado si en la solución óptima del problema, una unidad de flujo circula por su correspondiente arco.

En la Figura 7 se presentan los datos de los trabajos (J_i) y el grafo resultante.



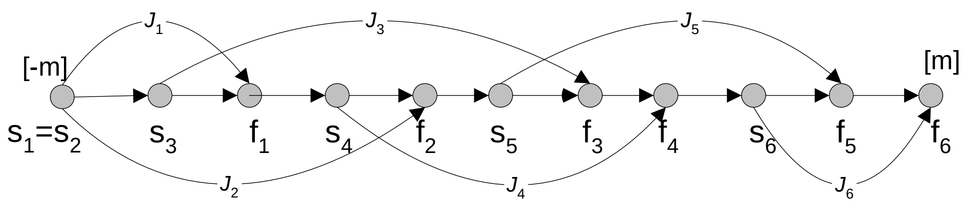


Figura 7

Kroon et al introducen también el siguiente procedimiento para reducir el grafo (Figura 8):

1. Buscar un par de nodos (i-1, i) en G tal que de i-1 no parta ningún arco o que al nodo i no llegue ningún arco, de entre los arcos que representan trabajos. Si no se encuentra ningún par, entonces FIN. En otro caso ir al paso 2.
2. Reemplazar el par de nodos (*i-1*, *i*) por un solo nodo al que lleguen y salgan los arcos que lo hacían en el nodo *i-1* e *i*. Volver al paso 1.

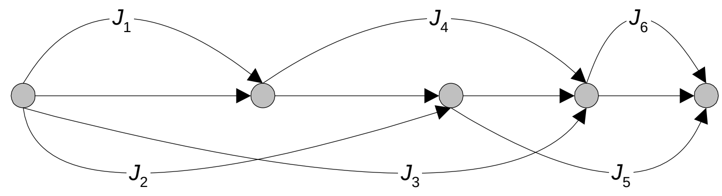


Figura 8

Este procedimiento es el que se usa en el escenario I para resolver el objetivo operacional de un problema de programación de trabajos fijos incorporando distancias entre máquinas. El mismo grafo, con un procedimiento iterativo, es usado a su vez para resolver el objetivo táctico en dicho escenario.

2.3.3 VARIAS CLASES DE MÁQUINAS, OBJETIVO TÁCTICO

En el presente caso, existe más de un tipo de máquinas, y los trabajos deben ser procesados por un cierto tipo de máquina en concreto, no ya por cualquiera de las máquinas existentes. Ya se trate un problema de tipo *License Class* o uno de tipo *Shift Class,* básicamente puede concluirse que el problema, en su objetivo táctico y operacional, es de carácter NP a partir de 3 clases diferentes de máquinas[14].

El modelo matemático que expresa el objetivo táctico del problema con varias clases de máquinas es el siguiente:

<u>Variables</u>:

x_{ij}	Variable binaria {0,1}, que toma el valor de la unidad si el trabajo *i* es asignado a la máquina *j.*
y_j	Variable binaria {0,1}, que toma el valor de la unidad si se utiliza la máquina *j.*

[14] Kolen y Kroon, 1991.

Datos:

c_j	Coste de uso de la máquina *j*.
s_i	Instante de comienzo del trabajo *i*.
f_i	Instante de finalización del trabajo *i*.

Parámetros:

$L_{D \times C}$	Matriz de compatibilidad trabajo-máquina.

Restricciones:

$$\sum_{j/L(d_i,c_j)=1} x_{ij} = 1 \quad \forall i \qquad (2.5)$$

Obligan a la realización de cada trabajo, mediante alguna de las máquinas con compatibilidad.

$$x_{ij} + x_{kj} \leq 1 \qquad i = 1 \ldots n-1; \;\; j = 1 \ldots m; \;\; \{k > i : s_k < f_j\} \qquad (2.6)$$

Controlan que dos trabajos solapados no sean procesados por la misma máquina.

Función Objetivo:

$$Min \sum_{j=1}^{m} c_j y_j \qquad (2.7)$$

Minimiza los costes totales de uso de cada máquina utilizada.

Modelo:

$$Min \sum_{j=1}^{m} c_j y_j$$

$$\sum_{j/L(d_i,c_j)=1} x_{ij} = 1 \quad \forall i$$

$$x_{ij} + x_{kj} \leq 1 \qquad i = 1 \ldots n-1; \;\; j = 1 \ldots m; \;\; \{k > i : s_k < f_j\}$$

Si suponemos el mismo coste para máquinas de la misma clase, el modelo puede reescribirse como:

Variables:

x_{jc}	Variable binaria {0,1}, que toma el valor de la unidad si el trabajo *i* es asignado a una máquina de la clase *c.*
y_c	Número de máquinas de la clase *c* utilizadas.

Datos:

c_c	Coste de uso de una máquina de la clase *c.*
s_i	Instante de comienzo del trabajo *i.*
f_i	Instante de finalización del trabajo *i.*

Parámetros:

$L_{D \times C}$	Matriz de compatibilidad trabajo-máquina.

Restricciones:

$$\sum_{c/L(d_i,c)=1} x_{ic} = 1 \quad \forall i \qquad (2.8)$$

Obligan a la realización de cada trabajo con alguna máquina de una clase compatible.

$$\sum_{\{i/s_i \le t \le f_i \wedge j \in J_c\}} x_{ic} \le y_c, c = 1 \ldots C; t \in T \qquad (2.9)$$

Definen el número de máquinas de cada clase asignadas en cada instante de tiempo.

Función Objetivo:

$$Min \sum_{c=1}^{C} c_c y_c \qquad (2.10)$$

Minimiza los costes totales de uso de cada clase de máquina.

Modelo:

$$Min \sum_{c=1}^{C} c_c y_c$$

$$\sum_{c/L(d_i,c)=1} x_{ic} = 1 \quad \forall i$$

$$\sum_{\{i/s_i \le t \le f_i \wedge j \in J_c\}} x_{ic} \le y_c, c = 1 \ldots C; t \in T$$

Las técnicas de resolución de este problema se han centrado en métodos aproximados. Generalmente se emplean algoritmos de carácter avaricioso (Greedy), que intentan aprovechar las máquinas que ya han realizado trabajos previos, aunque el uso de métodos con empleo de cotas superiores también ha manifestado buenos resultados[15].

2.3.4 VARIAS CLASES DE MÁQUINAS, OBJETIVO OPERACIONAL

El modelo que expresa el objetivo operacional es el siguiente:

Variables:

x_{jc}	Variable binaria {0,1}, que toma el valor de la unidad si el trabajo *i* es asignado a una máquina de la clase *c.*
x_i	Variable binaria {0,1}, que toma el valor de la unidad si el trabajo *i* es procesado.

Datos:

w_i	Peso del trabajo *i*.
s_i	Instante de comienzo del trabajo *i*.
f_i	Instante de finalización del trabajo *i*.
m_c	Número de máquinas de la clase *c.*

Parámetros:

$L_{D \times C}$	Matriz de compatibilidad trabajo-máquina.

[15] Kroon et al, 1997.

Restricciones:

$$\sum_{c/L(d_i,c)=1} x_{ic} = x_i \quad \forall i \tag{2.11}$$

Controlan el procesamiento de cada trabajo por una clase de máquina compatible.

$$\sum_{k/L(d_k,c)=1 \& s_k \leq t \& f_k > t} x_{kc} \leq m_c \quad \forall i, t = s_i, c = 1 \ldots C \tag{2.12}$$

Limitan, para cada instante del horizonte de planificación en el que se inicia un trabajo, el número de máquinas en uso de cada clase.

Función Objetivo:

$$Max \sum_{i=1}^{n} w_i x_i \tag{2.13}$$

Maximiza el peso total de los trabajos procesados.

Modelo:

$$Max \sum_{i=1}^{n} w_i x_i$$

$$\sum_{c/L(d_i,c)=1} x_{ic} = x_i \quad \forall i$$

$$\sum_{k/L(d_k,c)=1 \& s_k \leq t \& f_k > t} x_{kc} \leq m_c \quad \forall i, t = s_i, c = 1 \ldots C$$

Cuando el peso de los trabajos es el mismo o simplemente la unidad, y el objetivo es maximizar el número de trabajos realizados, el problema FSP con objetivo operacional e intervalos de disponibilidad para máquinas aparece también en la literatura como *k-Track Assignment Problem* (TAP), donde *k* define el número de máquinas disponibles. El TAP es considerado por Brucker y Nordmann[16], quienes proponen un método exacto de resolución de orden $O(n^{k-1}k!k^{k+1})$. El método construye un grafo que recoge todos los posibles programas admisibles del problema. La ruta máxima en el

[16] En Brucker y Nordmann, 1994.

grafo obtiene la solución óptima del problema. El método se muestra muy ineficiente con un número de máquinas superior a 4.

Para el caso genérico del problema, considerando trabajos con pesos, Arkin y Silverberg[17] proponen un método exacto con una filosofía parecida a la descrita por Brucker y Nordman, aunque con un comportamiento mejor, puesto que el orden del número de nodos y arcos del problema es, respectivamente, $O(n^m)$ y $O(n^{m+1})$.

Aproximaciones heurísticas para este problema han sido consideradas en Gabrel[18] y Kroon et al[19]. En el primer caso, modelando el problema como uno de máximo conjunto de nodos independientes (MIS), el segundo, basado en la resolución de algoritmos de flujo a coste mínimo. El problema FSP con *preemption* es abordado por Dondeti y Emmons[20], probando que puede ser resuelto en tiempo polinomial transformando el problema en un grafo de transporte.

2.4 CAMPOS DE APLICACIÓN

Los sistemas en los que las tareas poseen restricciones de tiempo y que pueden modelarse como problemas FSP son numerosos. A continuación se comentan algunos de ellos:

2.4.1 GESTIÓN DE LÍNEAS DE AUTOBÚS[21]

La variable a considerar en este caso es la planificación de los horarios de trabajo de los conductores de autobús de tal modo que se minimicen los costes asociados a dotar de servicio a todas las líneas de transporte existentes, entendiendo como costes el número de vehículos y conductores

[17] En Arkin y Silverber, 1987.
[18] En Gabrel, 1995.
[19] En Kroon et al, 1995.
[20] En Dondereti y Emmons, 1993.
[21] Tratado por Fishetti, 1987, 1989, 1992.

a utilizar. El problema se corresponde con un FSP de objetivo táctico. Las tareas son todas las jornadas predefinidas a realizar. Son trabajos fijos puesto que los horarios de salida y llegada poseen un instante fijo en el tiempo. En principio, sólo se considera el objetivo táctico, puesto que todos los viajes tienen que estar disponibles diariamente. Esta aplicación ha sido considerada con condiciones spread-time y working-time. Una tarea similar se aborda en este proyecto, donde para tener en cuenta las condiciones de trabajo de los conductores, se modela el escenario incluyendo nuevas restricciones.

2.4.2 GESTIÓN DE AEROPUERTOS: MANTENIMIENTO DE AVIONES MEDIANTE ASIGNACIÓN DE TÉCNICOS A TAREAS[22]

Los aviones que aterrizan a diario en los numerosos aeropuertos del mundo deben ser sometidos, por razones de seguridad, a operaciones de mantenimiento durante el período de tiempo que permanecen en el aeropuerto. Dado que los horarios de salidas y llegadas de los mismos son conocidos de antemano, el problema planteado consiste en minimizar el número de operarios de forma que se garantice el mantenimiento de todos los aparatos que pasen por el aeropuerto de la forma más satisfactoria posible. Este caso se puede corresponder con un problema FSP o VSP con objetivo táctico y varias clases de trabajos y máquinas.

El hecho de considerar varias clases de máquinas (o en este caso recursos) radica en la especialización de los operarios o técnicos. Generalmente, no todo el personal de mantenimiento está especializado en todas las operaciones, por lo que se hace necesario considerar clases. También se puede considerar la disponibilidad temporal de las máquinas, es decir la caracterización Shift Class.

En la bibliografía ha sido considerado al igual el objetivo operacional. En este caso, los pesos de las tareas equivalen a prioridades de procesado o importancia de la tarea. Algunas de éstas, las de menor prioridad, pueden

[22] Tratado por Kolen y Kroon, 1991 y Janson, 1994.

esperar para ser realizadas en un aterrizaje posterior del avión, por lo que se permite su no realización.

2.4.3 GESTIÓN DE AEROPUERTOS: ASIGNACIÓN DE PUERTAS A VUELOS PARA REDUCIR EL TRASLADO DE PASAJEROS HASTA LA TERMINAL[23]

El modelo consiste en asignar, a aviones de diferentes tipos, una puerta de embarque durante un intervalo de tiempo fijo. Dado que cada tipo de avión presenta unas características técnicas especiales, aparecen restricciones de compatibilidad entre las puertas de embarque y los aviones. Si no se consigue asignar una puerta a un avión, los pasajeros deben ser transportados hasta el mismo en autobús, con la consiguiente disminución de la calidad del servicio. El objetivo en este caso, consiste en encontrar la mejor solución para la asignación de puertas de embarque, para así minimizar el número de aeronaves que queden fuera de esta asignación y de este modo evitar el uso de autobuses. Obviamente, se trata de un objetivo operacional. Las puertas de embarque se corresponderían con las máquinas del sistema, mientras que los vuelos serían las tareas. Se consideran varias clases de máquinas, debido a que las puertas de embarque poseen capacidades diferentes y también porque los aviones pueden poseer características técnicas diferentes que impidan la conexión con determinadas puertas.

[23] Tratado por Kroon, 1991.

2.4.4 PLANIFICACIÓN DE SATÉLITES DE OBSERVACIÓN TERRESTRE[24]

Consideremos un satélite que describe una órbita alrededor de la tierra y cuya misión es realizar distintas operaciones, como la captura de imágenes de la superficie terrestre, tomar medidas atmosféricas o terrestres, efectuar operaciones de telecomunicaciones, etc. Dado que la demanda de operaciones por parte del satélite es numerosa, se plantea el problema de seleccionar cuál debe ser la secuencia de tareas a realizar durante un horizonte temporal dado, de manera que se maximice el número de trabajos ejecutados por cada satélite, o bien la suma de los pesos de las tareas realizadas (problemas operacionales), siendo las restricciones las siguientes:

- Un satélite solo podrá realizar tareas referidas a la parte de la superficie terrestre expuesta al sol.
- Debido al movimiento del satélite, la ejecución de cada tarea requerida sólo podrá comenzar en una ventana de posibles instantes de tiempo determinados, dentro del horizonte temporal considerado.
- El satélite sólo puede procesar una tarea en cada instante, teniendo además que dejar un tiempo de transición, supuesto constante, entre dos tareas pertenecientes a la misma secuencia.

La programación implementada procede a capturar tantas imágenes con prioridad alta como le sea posible, dentro de un período de tiempo fijo, con la ayuda de un sistema fijo de sensores incorporado al satélite. Por ejemplo, se considera que el planificador del satélite *Landsat 7* ha actuado eficientemente si se hacen al menos 250 observaciones por día. La manera de programar o planificar la realización de tareas del satélite es complicada debido al gran número de restricciones que intervienen, a destacar:

- Limitaciones de vista posterior: Los blancos de cada observación deben estar dentro de la vista del satélite. Los satélites del EOS viajan en órbitas fijas, generalmente cerca de 800 km sobre la superficie del planeta, tardando unos 100 min en recorrer una vez el

[24] Tratado por Grabel, 1995, y Wolfe y Sorensen, 2000.

perímetro de la tierra. Estas órbitas pasan sobre cualquier lugar concreto de la Tierra un número limitado, aunque predecible, de veces, por lo que hay solo algunas ventanas panorámicas para un blanco y dentro de un período de tiempo dado.

- Tiempo requerido para tomar cada imagen: La mayoría de los satélites que observan la Tierra toman una imagen unidimensional y utilizan el movimiento orbital de la nave para barrer sobre el área que será capturada. Una imagen de *Landsat*, por ejemplo, requiere 24 segundos de movimiento orbital.
- Almacenaje de datos a bordo limitado: Las imágenes se almacenan típicamente en un registrador de datos, con capacidad limitada, hasta que puedan ser enviadas a la tierra.
- Disponibilidad de la estación terrestre: Los datos almacenados a bordo se envían a la tierra cuando el satélite pasa sobre una estación de tierra.
- Ángulo de apunte: las imágenes de más alta resolución se toman cuando el blanco está directamente debajo del satélite. Otros ángulos de apunte son a veces requeridos para propósitos específicos.
- Disponibilidad de energía: La mayoría de los satélites tienen una capacidad energética muy restrictiva.
- Control térmico: Debido a que los satélites se encuentran alternativamente a la sombra de la tierra y fuera de ella, el ambiente térmico sufre cambios radicales. Esto añade restricciones al uso del sensor.
- Interposición de nubes: Algunos sensores no pueden ver a través de éstas.

Según lo descrito anteriormente, se trata de un problema operacional cuyos trabajos tienen instantes de comienzo variables, es decir VSP, y cuyos recursos no se encuentran disponibles durante todo el horizonte de planificación, es decir Shift Class.

2.4.5 ASIGNACIÓN DE AULAS DE CLASE[25]

El problema de la asignación de clases a aulas es un problema de tipo FSP, partiendo de horarios fijos de clase. El número de aulas disponibles en cualquier edificio es limitado y las clases o sesiones se encuentran planificadas previamente. Se necesita una asignación de aulas que tenga en cuenta las siguientes restricciones:

- La capacidad del aula tiene que ser suficiente para acoger al número de alumnos esperados.
- Algunas clases pueden necesitar aulas con recursos específicos, como por ejemplo audiovisuales.

También podrían plantearse restricciones que garanticen que una misma clase transcurra siempre en la misma aula durante el horizonte de planificación, o bien tener en cuenta en la función objetivo la minimización del número de cambios de aula.

Estaríamos por tanto en el caso de varias clases de máquinas y objetivo operacional. Sin embargo, es necesario atender todas las clases. Es un problema un tanto especial puesto que se trata de un problema de viabilidad, es decir, de determinar si es posible acoger todas las clases con las aulas disponibles. En caso de que no se pudiera, sería necesario ocupar otro edificio o bien replantear el programa de clases.

[25] Tratado por Carter, 1989.