VIII Congreso de Ingeniería de Organización
Leganés, 9 y 10 de septiembre de 2004

Asignación de recursos a actividades fijas

José Manuel García Sánchez, Ricardo Galán de Vega

Dpto. Organización Industrial y Gestión de Empresas. Escuela Superior de Ingenieros. Universidad de Sevilla. Camino de los Descubrimientos, s/n 41092 Sevilla. jmgs@esi.us.es, rdevega@us.es

Resumen

La asignación de recursos a actividades fijas se caracteriza como el problema de programar una serie de trabajos sobre un conjunto de máquinas en paralelo. Cada trabajo posee un instante fijo de comienzo, un instante fijo de finalización, un peso y pertenece a un tipo de trabajo. Respecto a las máquinas, puede considerarse un coste asociado al uso de la máquina y, en ocasiones, uno o varios intervalos de tiempo en el que únicamente está disponible. El problema es conocido en la literatura como Fixed Job Scheduling Problem (FSP). En este trabajo se realiza una clasificación y revisión bibliográfica de todos los problemas de tipo FSP existentes en la literatura, presentando las características de cada uno de ellos, las técnicas de resolución empleadas, el tipo de problema en orden a su complejidad y las aplicaciones prácticas existentes del problema.

Palabras clave: Asignación, Trabajos fijos, Revisión bibliográfica.

1. Introducción

La asignación de recursos a actividades fijas se conoce como el problema de programar una serie de trabajos sobre un conjunto de máquinas en paralelo. Los trabajos o actividades tienen que ser realizados dentro de un intervalo temporal fijo, siendo inadmisible la realización parcial o total fuera del mismo. Las máquinas son el recurso necesario para realizar los trabajos. Cada trabajo consume un tiempo fijo de ocupación de la máquina.

La teoría de la asignación de recursos a actividades fijas, que podría ser también denominada como teoría de la programación de trabajos en intervalos fijos sobre máquinas en paralelo, abarca problemas de dos tipos. Por un lado, problemas donde el intervalo para el proceso de cada trabajo coincide con el tiempo de duración del trabajo, es decir, no existe ninguna holgura para el procesamiento del trabajo. Por otro lado, problemas donde el intervalo de tiempo para el procesamiento de cada trabajo es mayor o igual al tiempo de proceso del trabajo. El primer tipo de problemas es denominado en la literatura como *Fixed Job Scheduling Problem* (FSP) mientras que el segundo se conoce como *Variable Job Scheduling Problem* (VSP).

En este trabajo pretendemos realizar una clasificación y revisión bibliográfica de todos los problemas de tipo FSP existentes en la literatura, presentando las características de cada uno de ellos, las técnicas de resolución empleadas, el tipo de problema en orden a su complejidad y las aplicaciones prácticas existentes.

2. Definición y Clasificación del problema FSP

De la forma más genérica, el problema de la asignación de recursos a actividades o trabajos fijos(FSP) presenta las siguientes características:

- Dado un conjunto de n trabajos $N=\{J_1,..,J_i, ...J_n\}$, cada uno con un instante fijo de comienzo s_i, un instante fijo de finalización f_i, un peso o valor w_i y una clase de trabajo a_i a la que pertenece.
- Dado un conjunto de m máquinas, las cuales pertenecen a una clase de máquina b_j y llevan asociado un coste fijo c_j por su uso.
- Por otro lado se define un intervalo de tiempo $[r_j, e_j]$ para cada máquina en el que únicamente está disponible.
- Para el problema se definen A diferentes clases de trabajos y B clases de máquinas.

Las restricciones asociadas al procesamiento de los trabajos son:

- Cada máquina puede realizar un solo trabajo en un mismo instante de tiempo.
- Cada trabajo puede ser realizado sobre un subconjunto de máquinas. Se asume que cada clase de máquina puede realizar trabajos pertenecientes a un número limitado de clases de trabajos. Por ello se define una matriz de compatibilidad L entre clases de máquinas y trabajos cuyos valores pueden ser:

$$L_{A\times C} = L(fila, col) = \begin{cases} 1 & J_i \text{ es compatible con maquina } M_j \text{ si } a_i = fila \text{ y } c_j = col \\ 0 & \text{en otro caso} \end{cases}$$

La clasificación del problema se realiza, principalmente, en torno a los siguientes parámetros:

1) Objetivos del problema
2) Número de clases de máquinas y trabajos

1. Objetivos del problema. En el problema se consideran tres objetivos fundamentalmente:

1.1 Minimizar el coste total en máquinas requeridas para procesar todos los trabajos. Cuando el coste es la unidad, o no se considera, el objetivo se entiende como minimizar el número mínimo de máquinas necesarias para procesar todos los trabajos. Se supone un número suficiente de máquinas para poder realizar todos los trabajos.

1.2 Averiguar si, dado un conjunto de máquinas, es posible realizar todos los trabajos existentes (objetivo aplicado en el caso de poseer varias clases de trabajos y máquinas).

1.3 A partir de un número fijo de máquinas insuficiente para el procesamiento de todos los trabajos, maximizar el valor total de los trabajos realizados.

2. Número de clases de máquinas y trabajos

2.1 Una única clase de máquinas y trabajos: Es el caso más sencillo del problema FSP. Cada trabajo puede ser procesado por cualquier máquina. Además, las máquinas están disponible durante todo el horizonte temporal de planificación.

2.2 Varias clases de máquinas y trabajos: En este caso, cada trabajo puede únicamente ser procesado por un subconjunto de máquinas. Este escenario del problema posee dos 2 variantes:

2.2.1 *Shitf Class Design*: Se establecen un intervalo(*shift*) en el que se encuentra disponible cada máquina, de forma que cada máquina puede realizar únicamente trabajos que se procesen dentro de su intervalo.

2.2.2 *License Class Design*: Las máquinas están disponibles durante todo el horizonte temporal. La compatibilidad entre clases de trabajos y máquinas se rige por motivos técnicos.

Otros parámetros que se distinguen también en el problema son los siguientes:

- El valor del trabajo: El caso general es la asignación de un valor a cada trabajo que mide el beneficio por completar dicho trabajo. A pesar de ello, en ocasiones se asigna el mismo valor a todos los trabajos, o lo que es lo mismo, el valor de cada trabajo es la unidad.

- La propiedad de interrumpir (*preemption*) el procesamiento del trabajo en una máquina: Aunque las características del problema obligan a que no existan pausas en el procesamiento de un trabajo, puesto que el tiempo de proceso coincide con el intervalo de procesamiento, el caso de *preemption* se plantea en algunos escenarios en los que se considera admisible que el trabajo puede ser procesado por más de una máquina (el trabajo puede saltar de una máquina a otra).

En los siguientes capítulos se describen los diferentes trabajos realizados en cada uno de las diferentes clases de problemas FSP comentadas, según objetivos y tipo de máquinas y trabajos.

3. Una sola clase de trabajo y de máquina

Es la versión más simple del problema FSP. Sobre dicha versión se han aplicado los tres objetivos mencionados en la sección 2. Con una sola clase de máquina y de trabajo, los objetivos 1.1 y 1.2 son equivalentes, por lo que se describen como uno en lo que sigue.

3.1. Minimizar el número de máquinas

Puede ser éste el escenario que dio nombre al problema FSP. La solución del problema se obtiene como una consecuencia directa del teorema de Dilworth (Diworth, 1950) 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 (1954), y posteriormente por Gupta et al(1979) y Nakajima et al (1982), mostrando algoritmos de resolución de orden $O(n \log n)$. Gertsbakh y Stern(1978) también definen 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.

La idea para el cálculo del mínimo número de máquinas está basada en el cálculo del grado de solapamiento de los trabajos en el tiempo. El grado de solapamiento de los trabajos en un instante t del horizonte de planificación se define como el número de trabajos que se procesan durante el instante t. La expresión corresponde con $L_t = |\{J_i \,/\, s_i \leq t < f_i\}|$. El número mínimo de máquinas correspondería con el mayor valor del grado de solapamiento en algún instante t. El mayor grado de solapamiento se define como $L = \max\{L_t \,/\, 0 \leq t \leq T\}$, siendo $[0,T]$ el horizonte en el que se produce la ejecución de todos los trabajos.

Como una variante a la formulación general del problema FSP en este escenario, Fishetti *et al* (1987, 1989, 1992) plantean el problema añadiendo restricciones de *Spread-Time* (límite sobre el tiempo que transcurre desde que se inicia la actividad de la máquina) y *Working-Time* (límite en el tiempo de actividad de cada máquina).

3.2. Maximizar el valor total de los trabajos realizados

En el caso en el que el número mínimo de máquinas necesario para procesar todos los trabajos es mayor al número de máquinas disponibles en el problema, es necesario discriminar entre los trabajos que pueden ser realizados. El problema FSP con este objetivo puede ser formulado en términos del problema de coloreado de grafos (Golumbi, 1980). A pesar de ser ésta una forma viable de plantear el problema, la forma más eficiente de resolverlo es plantearlo como un problema de flujo a coste mínimo. Por tanto, el problema es resoluble en tiempo polinomial mediante un algoritmo de flujo a coste mínimo.

Entre las construcciones de grafos dirigidos destacan la propuesta por Arkin y Silverberg (1987) basada en un cálculo de cliques según un grafo anterior con un nodo por cada trabajo, y en el que un arco representa el solapamiento entre los trabajos de cada nodo. A partir del cálculo de los cliques máximos de este grafo, se construye un segundo grafo sobre el que si es posible aplicar un algoritmo de flujo a coste mínimo para determinar los trabajos a realizar. En este segundo grafo existe un nodo por cada clique más un nodo final. Por cada trabajo existe un arco desde el primer nodo-clique al que pertenece hasta el nodo-clique siguiente a su finalización. El coste de cada arco es $-w_i$ y la capacidad igual a uno. Además, existen arcos con infinita capacidad y coste nulo que unen los cliques consecutivos según orden cronológico.

Kroon et al (1995) definen una construcción más directa que la anterior, en la que los nodos representan los instantes de comienzo y finalización de cada uno de los trabajos. Los nodos ordenados cronológicamente se unen por arcos de capacidad infinita y coste nulo, además de un arco por cada trabajo con capacidad uno y coste $-w_i$. Plantean incluso un sencillo procedimiento de reducción del grafo con objeto de disminuir la dimensión del mismo en los casos en los que los trabajos se encuentren muy dispersos en el tiempo.

Garcia *et al* (1999) proponen una variante al grafo de Kroon *et al*(1995), que es construido igualmente de forma directa, y que reduce el número de nodos del problema. En dicho grafo existe un nodo por cada trabajo, mas un nodo final. Existe arcos que unen nodos consecutivos, de forma semejante a los grafos anteriores y un arco, por cada trabajo J_i, desde su nodo hasta el nodo que representa el primer trabajo que comienza tras la finalización de J_i. La capacidad y el coste se asignan de forma semejante a los grafos anteriores.

En todos los casos, el flujo inicial en el nodo de partida corresponde con el número de máquinas del problema. En la Figura 2 se presentan los tres grafos para el ejemplo representado en la Figura 1.

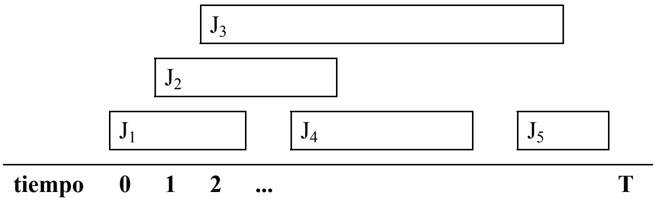

Figura 1. Ejemplo con 5 trabajos

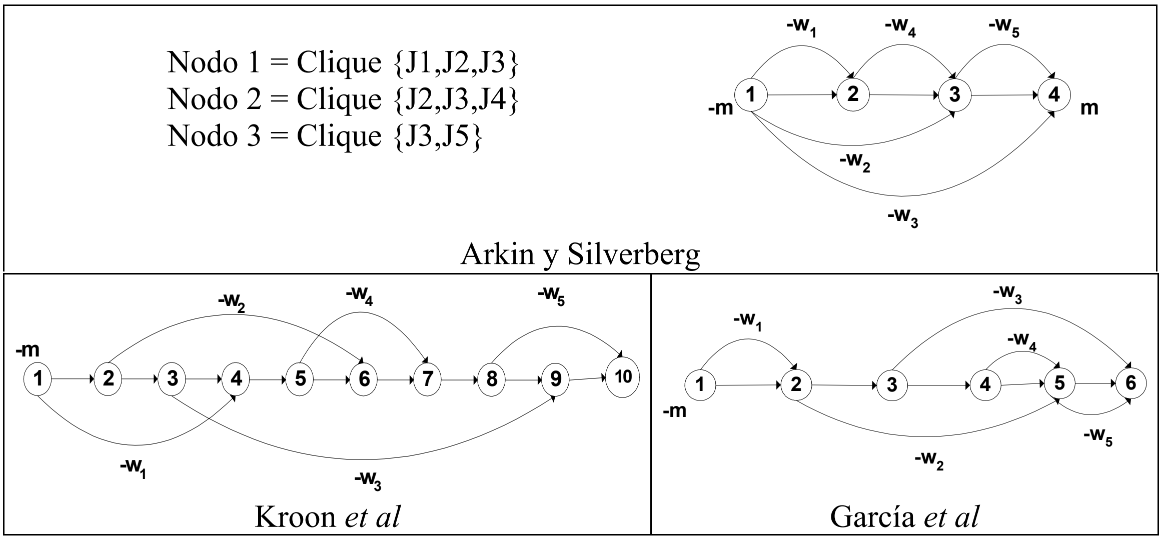

Figura 2. Grafos de Arkin y Silverberg, Kroon *et al* y Garcia *et al*

4. Varias clases de trabajos y máquinas

Sería éste el caso más complejo del problema FSP. Ahora un trabajo no puede ser procesado por cualquier máquina. El caso de varias clases de máquinas y trabajos integra las dos variantes señaladas en la clasificación realizada en la sección 2. Ambas variantes presentan las mismas características respecto a su complejidad. A pesar de ello, han sido analizados en la bibliografía de forma independiente. A continuación se describen cada una de ellas.

4.1. Shift Class Scheduling Problem (SCSP)

Kolen y Kroon (1993) estudian la complejidad del problema atendiendo al objetivo que se defina para el problema. Definen un único intervalo para cada máquina, asumiendo que dicha máquina puede únicamente procesar trabajos que se procesen por completo dentro su intervalo(*shift*). Presentan dos de los objetivos comentados del problema FSP, averiguar si, dado un conjunto de trabajos y máquinas, es posible realizar todos los trabajos (SCS), y el máximo valor total de trabajos procesados sobre un conjunto de máquinas (MSCS).

La complejidad de SCS y MSCS se define según el tipo de intervalo S definido para los trabajos. Para ello, definen un grafo no dirigido $G(S)$ con la siguiente estructura:

- Un nodo por cada intervalo
- Un arco entre cada dos intervalos que se solapen y que posean instantes de inicio y fin diferentes.

Los resultados que prueban son los siguientes:

- Si *G(S)* está formado únicamente por nodos aislados, entonces MSCS es resoluble en tiempo polinomial.
- Si *G*(*S*) es un grafo bipartito, entonces SCS es resoluble en tiempo polinomial.
- Si el máximo grado de solapamiento de los intervalos es menor o igual a 2, entonces SCS es resoluble en tiempo polinomial.
- En el resto de casos SCS es NP-Completo y MSCS es NP-Duro.

Kolen y Kroon (1994) presenta un estudio de la complejidad del problema SCSP para el primero de los objetivos del problema. Es decir, asume un coste para cada máquina y busca el conjunto de máquinas con coste mínimo capaces de procesar todos los trabajos. Establece una consideración respecto a los costes de las máquinas y divide el estudio en dos casos:

- Costes uniformes: El coste de cada máquina es independiente (SCSCU)
- Costes dependientes del intervalo: El coste depende del intervalo, no de la máquina. Dos máquinas con el mismo intervalo poseen el mismo coste (SCSCD).

Para el estudio de complejidad considera el concepto de intervalos no dominados. Un intervalo $[a,b]$ es no dominado cuando no existe ningún intervalo $[c,d]$ tal que $c \leq a < b \leq d$. Los resultados son análogos a los obtenidos para SCS y se resumen a continuación:

- SCSCU es resoluble en tiempo polinomial si el mayor grado de solapamiento de los intervalos no dominados es menor o igual a 2. En el resto de los casos es NP-Duro.
- SCSCD puede ser resuelto en tiempo polinomial en los mismos casos que SCS.

Para ambos problemas considera el caso en el que puedan romperse los trabajos (*preemptive*), mostrando que dicha variante puede ser resuelta en tiempo polinomial mediante un problema de flujo a coste mínimo.

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 intervalos para máquinas aparece también en la literatura como *k-Track Assignment Problem* (TAP). Curiosamente, el resto de características coincide con las expresadas para el problema MSCS. TAP es considerado por Brucker y Nordmann (1994) donde proponen un método exacto de resolución de orden $O(n^m k! m^m)$. El método construye un grafo que recoge todos los posibles programas admisibles del problema. La ruta máxima en el grafo obtiene la solución óptima del problema. El método se muestra muy ineficiente con un número de máquinas superior a 4.

Para un caso particular del problema presenta un algoritmo de orden $O(n^{m-1})$ basado también en el calculo de la ruta máxima. Considera también el caso en el que únicamente existen dos intervalos distintos de trabajo para las máquinas, proponiendo un algoritmo de orden $O(n)$. Este caso es contemplado también por Hsu y Tsai (1989). Faigle and Nawijin (1995) y Faigle *et al* (1999) estudian también el problema TPU tanto en su forma normal como en una versión online.

4.2. License Class Scheduling Problem (LCDP)

LCDP es la versión más general del problema FSP. Aquí la compatibilidad entre máquinas y trabajos se establece por motivos técnicos y, generalmente, la compatibilidad se expresa mediante el término "tener licencia". El problema SCSP puede ser expresado como un caso particular del problema LCDP. Por ello, los métodos de resolución para LCDP pueden ser aplicados al problema SCSP.

Kolen y Kroon (1991) (1992) presentan el estudio de la complejidad de este problema. El estudio se realiza de forma similar al realizado para el problema SCSP.

Denominando LCD al problema de determinar si, dado un conjunto de máquinas, es posible procesar todos los trabajos, MLCD al problema de maximizar el valor de los trabajos realizados con un número determinado de máquinas y, LCDC al problema de calcular el coste total mínimo necesario para procesar todos los trabajos, las conclusiones sobre la complejidad computacional del problema son:

- LCD es NP-Completo para problemas con al menos 3 clases diferentes de máquinas.
- MLCD es NP-Duro con al menos 2 clases diferentes de máquinas.
- LCDC es NP-Duro con al menos 3 clases diferentes de máquinas.

Anteriormente, Arkin y Silverberg (1987) muestra que LCD es NP-Completo cuando cada trabajo puede ser procesado por al menos 3 máquinas. A pesar de estos resultados, presenta un algoritmo exacto de orden $O(n^{m+1})$, basado en la construcción de un grafo de estados admisibles. La aplicabilidad de este algoritmo se reduce a casos con un número muy reducidos de máquinas.

El problema LCD con dos clases máquinas ha sido tratado por Dondeti y Emmons (1992), mostrando su complejidad polinomial.

Aproximaciones heurísticas para MLCD han sido consideradas en Gabrel (1995) y Kroon *et al* (1995). En el primer caso, el problema FSP es modelado como un problema de máximo conjunto de nodos independientes. Se analizan tres rocedimientos heurísticos para su resolución. El algoritmo que presenta Kroon *et al*(1995) está basado en la resolución de algoritmos de flujo a coste a mínimo sobre grafos construidos de forma análoga a la descrita en la sección 3.2. Va calculando cotas superiores e inferiores del problema hasta un número máximo de iteraciones o una cota de la diferencia.

El problema LCD, permitiendo rotura de trabajos es considerado en Dondeti y Emmons (1993) probando que puede ser resuelto en tiempo polinomial transformando el problema en un grafo de transporte.

Jansen (1994) considera una generalización de SCS y LCD, considerando costes para las máquinas. Propone una aproximación algorítmica que asigna un trabajo a una máquina en cada iteración, pero no realiza experimentos computacionales.

5. Aplicaciones prácticas del problema

Problemas prácticos que implican la resolución de instancias del problema FSP aparecen en múltiples áreas pertenecientes a la optimización de recursos. Las más importantes que han sido estudiadas han sido:

- Procesos de mantenimiento de aviones en aeropuertos mediante la asignación de ingenieros a tareas: Kolen y Kroon (1991)(1992)(1993)(1994); Janson (1994)
- Asignación de puertas a vuelos en aeropuertos con objeto de reducir el número de vuelos cuyos pasajeros son transportados a la terminal en bus: Kroon et al (1991)
- Planificación de la captura de imágenes desde satélite: Gabrel (1995)
- Asignación de conductores en líneas de autobuses: Fishetti et al (1992)

El problema de la asignación de aulas de clase (Carter, 1989) es un problema bastante próximo a FSP, aunque no posee exactamente las mismas características.
También puede ser aplicado al control del tráfico aéreo, la planificación de salas de operaciones en hospitales o la asignación de habitaciones en hoteles, entre otros.

Referencias

Arkin, E.M., and Silverberg, E.L (1987) Scheduling jobs with fixed starting and finishing times. Discrete Applied Mathematics 18, pp. 1-8.
Brucker, P. and Nordmann, L. (1994). The k-Track Assignment Problem. *Computing*, 52, pp. 97-122.
Carter, M.W. (1989). A lagrangean relaxation approach to the classroom assignment problem, INFOR 27, 230-246.
Dantzig G.B. and Fulkerson, D.R. (1954). Minimizing the number of tankers to meet a fixed schedule. *Naval Res. Logist. Quart.* 1, pp. 217-222.
Dilworth R.P. (1950). A descomposition theorem for partially ordered sets. *Annals Mathematics*, 51, pp. 161-166.
Dondeti, V., and Emmons, H. (1992). Interval scheduling with processors of two types. *Operations Research*, 40, pp. 76-85.
Dondeti, V.R., and Emmons, H. (1993). Algorithms for preemptive scheduling of different classes of processors to do jobs with fixed times. *European Journal of Operational Research*, 70, pp. 316-326.
Faigle, U., and Nawijn, W. M. (1995). Note on scheduling intervals on-line. Discrete Applied Mathematics 58, pp. 13-17.
Faigle, U., Kern, W. and Nawijn, M. (1999). A Greedy On-Line Algorithm for the k-Track Assignment Problem. Journal of Algorithms 31, pp. 196-210.
Fischetti, M., Martello, S. and Toth, P. (1987). The fixed job schedule problem with spread time constraints. Operations Research, 6, pp. 849-858.
Fischetti, M., Martello, S. and Toth, P. (1989). The fixed job schedule problem with working time constraints. Operations Research, 3, pp. 395-403.
Fischetti, M., Martello, S. and Toth, P. (1992). Approximation algorithms for fixed job schedule problems. Operations Research, 40, pp. 96-108.
Gabrel, V. (1995). Scheduling job within time windows on identical parallel machines: New model and algorithms. *European Journal of Operacional Research*, 83, pp. 320-329.

García, J.M., Lozano, S., Guerrero, F., Calle, M. and Smith, K. (2001). Production and vehicle scheduling for ready-mix operations. *Proceedings of the 29th International Conference on Computers & Industrial Engineering* (Olivier, C.Gharbi, A. eds.) pp. 70-76.
Gertsbakh I. and Stern H. (1978). Minimal Resources for Fixed and Variable Job Schedules. *Operations Research*, 26 (1), pp. 68-85.
Golumbi, M.C. (1980). Algorithmic Graph Theory and Perfect Graphs. Academic Press, New York.
Gupta, U.L., Lee, D.T., and Leung, J.Y.T (1979). An optimal solution to the channel assignment problem. *IEEE Trans. Comp.* 28, pp. 807-810.
Hsu, M. L. and Tsai K.H. (1989). A linear time algorithm for the two-track assignment problem. Proceedings of the 27th Allerton Conference on Communication, Controls and Computing, pp. 291-300.
Kolen, A.W.J., and Kroon, L.G. (1991). On the computational complexity of (maximum) class scheduling. *European Journal of Operational Research*, 54, pp. 23-38.
Kolen, A.W.J., and Kroon, L.G. (1992). License class design: complexity and algorithms. *European Journal of Operacional Research*, 63, pp. 432-444.
Kolen, A.W.J., and Kroon, L.G. (1993). On the computational complexity of (Maximum) Shift Class Scheduling. *European Journal of Operacional Research*, 64, pp. 138-151.
Kolen, A.W.J., and Kroon, L.G. (1994). An analysis of shift class design problems. *European Journal of Operacional Research*, 79, pp. 417-430.
Kroon L.G., Salomon M. and Van Wassenhove L. N. (1995). Exact and approximation algorithms for the operational fixed interval scheduling problem. *European Journal of Operational Research*, 82, 190-205
Jansen, K., (1994). An approximation algorithm for the license and shift class design problem. *European Journal of Operational Research*, 73, pp. 127-131.
Nakajima, K., Hakimi, S.L. and Lenstra, J.K. (1982). Complexity results for scheduling tasks in Fixed Intervals on two types of machines. *SIAM J. Comput*, 11, pp. 512-520.