Problema de secuenciación para máquinas no disponibles: revisión de la literatura

Paz Pérez González, José Manuel Framiñán Torres, Pedro Luis González Rodríguez, José Miguel León Blanco, Rafael Ruiz-Usano

Dpto. de Organización. Escuela Superior de Ingenieros. Universidad de Sevilla. Camino de los Descubrimientos s/n. 41092. Sevilla. pazperez@esi.us.es, jose@esi.us.es, pedroluis@esi.us.es, miguel@esi.us.es, usano@us.es

Resumen

El problema de secuenciación de trabajos en talleres de flujo regular o flowshop ha sido ampliamente estudiado en la literatura cuando las máquinas están disponibles en todo el horizonte de planificación. Sin embargo, el interés por el caso en el que dichas máquinas no están disponibles ha sido más notable hace tan sólo una década. En esta comunicación se presenta la revisión de la literatura sobre este tipo de problemas. Además, se introduce un estudio preliminar de un caso especial de flujo regular de permutación, en el que las máquinas no están disponibles al inicio del horizonte de planificación, en el caso particular de minimización del makespan para el que se demuestra la optimalidad en el caso de dos máquinas, y la complejidad para el caso de más de dos máquinas.

Palabras clave: secuenciación, revisión de la literatura, flujo regular, máquinas no disponibles

1. Introducción

En fabricación, cuando todos los trabajos tienen que pasar por todas las máquinas, y el orden de paso por éstas es el mismo para todos los trabajos, se dice que el entorno es de flujo regular o *flowshop*. Además, el caso en el que los trabajos son secuenciados de forma que pasen en el mismo orden por todas las máquinas se denomina flujo regular de permutación o *permutation flowshop*.

En la mayor parte de los entornos de fabricación reales las máquinas no suelen estar disponibles en todo el horizonte del proceso de los trabajos, es decir, existen restricciones de disponibilidad de las máquinas. Así, definimos intervalo de no disponibilidad de la máquina i (conocido también como ventana o período de no disponibilidad) al intervalo de tiempo $[s_i, t_i]$, con $0 \leq s_i \leq t_i$, con $1 \leq i \leq m$, en el que la máquina no está disponible para realizar ningún trabajo. El caso en que $s_i=0$, es decir, el intervalo es $[0, t_i]$, denominaremos a t_i como a_i y se llamará instante de disponibilidad, al ser el momento en el que la máquina i comienza a estar disponible para procesar trabajos. Los intervalos de no disponibilidad pueden ser:

- Estocásticos, si no se conoce el momento en el que la máquina no va a estar disponible, ni cuántas veces se producirá. Este caso es producido por rotura de la máquina
- Determinista, que es el caso contrario al anterior, y el intervalo puede estar:
  - Prefijado en cualquier instante del período de planificación. Este caso se da para las tareas de mantenimiento de las máquinas, que pueden ser coordinadas con la planificación, permitiendo localizar el instante de inicio del intervalo en la planificación, o no coordinado, en caso en el que el instante de inicio está fijo.

- Prefijado al principio del período de planificación. Producido por que las máquinas están ocupadas por trabajos anteriores.

Por lo tanto, la diversidad de situaciones reales en las que se puede presentar este problema implica que es de interés su estudio. En esta comunicación se hace una revisión de la literatura para el problema de flujo regular con restricción de disponibilidad de las máquinas. Este entorno ha sido estudiado ampliamente suponiendo que las máquinas están todas disponibles de forma continua. En el caso contrario es un problema que se viene estudiando desde hace una década, aunque para otros entornos como máquinas en paralelo o una única máquina desde hace más tiempo, Schmidt (2000).

Además, se presenta un primer estudio de un problema en particular que no ha sido aún totalmente abordado, el problema de flujo regular de permutación en el caso en el que los intervalos de no disponibilidad están localizados al inicio del horizonte de planificación o tiempo cero, es decir, el caso en el que las máquinas están ocupadas con trabajos anteriores cuando llega un conjunto nuevo de trabajos. Estos no serán procesados hasta que las máquinas vayan quedándose libres. Se supondrá que la capacidad de almacenaje entre dos máquinas es ilimitada, es decir, no se produce bloqueo de la máquina. En este estudio preliminar, se demostrará la optimalidad del problema para dos máquinas, y posteriormente, la complejidad del problema para más de dos máquinas.

2. Revisión de la literatura para problemas de flujo regular con las máquinas no disponibles

Los problemas de secuenciación con restricciones de disponibilidad en las máquinas han recibido la atención de muchos investigadores a partir de la última década, Wang y Cheng (2007a). Las revisiones conocidas sobre el tema, considerando todos los medios, son de Lee et al (1997), Schmidt y Sanlaville (1998), Türkcan (1999), Schmidt (2000) y posteriormente Lee (2004). Así estos autores muestran una revisión de la literatura de medios con máquinas no disponibles: para una máquina, máquinas paralelas, y máquinas en serie (*job shop*, *flowshop y openshop*). Türkcan (1999) incluso distingue entre los casos determinísticos y estocásticos, el resto de autores sólo revisan los casos determinísticos. Al consultar estas revisiones, se observa que para flujo regular o *flowshop* no hay muchas referencias. Sin embargo, a partir del año 2000 comenzaron a aparecer bastantes estudios que abordan este problema, algunos recogidos por Lee (2004).

Por lo tanto, se presenta la revisión específica para el medio de flujo regular, en caso determinístico, y sólo para intervalos de no disponibilidad fijos en el horizonte de planificación. Primero se verá un análisis de la literatura para flujo regular con dos máquinas con restricción de disponibilidad de éstas, y a continuación para más de dos máquinas. Antes de ver las revisiones especificadas, se presentan las diferentes notaciones que se utilizará en la revisión. Para ello, en esta comunicación se denotan los problemas mediante la notación de Graham et al (1979), $\alpha \mid \beta \mid \gamma$. Para cada uno de los parámetros, se van a indicar las distintas notaciones utilizadas por las referencias citadas, para que coincidan con la usada en el documento referenciado, para que sea familiar dicha notación en caso de consultar la referencia correspondiente, y no se cree confusión.

α representa el medio, en este caso, F para denotar del medio de flujo regular, indicando también el número de máquinas. Algunos autores indican la restricción de disponibilidad de las máquinas en este campo también, junto con el medio y el número de máquinas. Podemos encontrar:

- NC_{win} para denotar el modelo arbitrario para el caso de ventanas de no disponibilidad de las máquinas (para más detalle ver Schmidt (2000));

- NC_{zz} para el caso en el que el número de máquinas disponibles cambia de m a m-1 en distintos intervalos de tiempo determinados (para más detalle ver Schmidt (2000));

- h_{ik} para representar el problema en el que se consideran i intervalos de no disponibilidad en la máquina k, Espinouse et al (1999).

β indica las características de procesamiento:

- *prmu* para indicar que el flujo regular es de permutación (también denotado por permu);

- *prmp* para representar el problema en el que se permite interrumpir un trabajo en una máquina para procesar otro trabajo (otra notación para este caso es *pmtn*). El caso particular *t-pmtn* indica que los trabajos se pueden interrumpir, pero sólo pueden ser procesados por una máquina (para más detalle ver Schmidt (2000));

- *no-wait* en flujo regular indica que todos los trabajos deben procesarse de forma continua, sin espera entre dos máquinas consecutivas;

- *setup* para los trabajos con tiempos de preparación además de tiempos de procesamiento;

- *r-a* denota la restricción de reanudación o "*resumable availability*", Lee (1997), es decir, si un trabajo no puede terminar antes del periodo de no disponibilidad de la máquina, entonces continúa cuando la máquina vuelva a estar disponible, sin ninguna pérdida de tiempo o penalización (algunos autores lo denotan por *rs*). Si no se indica nada, la no disponibilidad es para todas las máquinas que se consideren en el problema. Si es en alguna máquina en particular se indica entre paréntesis, por ejemplo r-$a(M_1)$ indicaría no disponibilidad en la primera máquina;

- *sr-a* indica la restricción de "*semiresumable availability*", Lee (1999), en el caso en el que el trabajo no puede terminar antes del periodo de no disponibilidad de la máquina, y comienza parcialmente cuando la máquina vuelva a estar disponible. La penalización, o tiempo parcial que el trabajo deberá de reprocesar se define mediante una constante $0 \leq \alpha \leq 1$ que multiplica al tiempo de procesamiento total del trabajo interrumpido. El caso α=0 es el caso anterior de "*resumable availability*", y α=0 es el caso "*nonresumable availability*" que se ve a continuación. Igual que en el caso anterior, si es en alguna máquina en particular se indica entre paréntesis;

- *nr-a* es el caso de "*nonresumable availability*", Lee (1997), es cuando el trabajo debe de empezar y procesarse completamente cuando la máquina está disponible de nuevo (todo el tiempo procesado hasta el momento es desechado). También podemos encontrar este caso con la notación *a-n* ó *sn*;

- s_i, t_i ya definidos, es la notación usada para el intervalo de no disponibilidad;

- a_i como ya se ha definido anteriormente, es el instante de disponibilidad de la máquina i.

Cabe hacer notar la diferencia que indica Breit (2006) entre los problemas con los trabajos interrumpibles y con reanudación. Interrumpible (*preemptable*) indica que un trabajo puede pararse en un instante de tiempo y continuar sin penalización, según convenga a la hora de secuenciar los trabajos, y con reanudación (*resumable*) el trabajo que corresponda debe interrumpirse en el instante dado por el intervalo de no disponibilidad de una máquina determinada, para reanudarse al término de dicho intervalo.

Por último, γ indica el objetivo a minimizar, donde nos encontraremos C_{max}, que representa el tiempo total de completitud de todos los trabajos en las máquinas (llamado *makespan*).

2.1. Problemas de secuenciación en flujo regular para dos máquinas no disponibles

El problema clásico de flujo regular para secuenciar *n* trabajos en dos máquinas con el objetivo de minimizar el makespan fue resuelto de forma óptima por Johnson (1954). Este problema para el caso que nos ocupa, con restricciones en las máquinas representadas mediante ventanas o intervalos de no disponibilidad, bajo el criterio de makespan en todos los casos, ha sido abordado por bastantes autores. En la revisión de Schmidt (2000) para todos los medios, se presentan los resultados para $F2|r\text{-}a|C_{max}$ de Lee (1997), donde demuestra que el problema para dos máquinas es NP duro tan solo con un único intervalo de no disponibilidad en una máquina; los problemas $F2,NC_{zz}|t\text{-}pmtn|C_{max}$ y $F2,NC_{win}|pmtn|C_{max}$ de Kubiak et al (1997), que demuestran que el problema es NP duro en sentido fuerte si hay un número arbitrario de intervalos en una única máquina; y $F2,NC_{win}|pmtn|C_{max}$ de Blazewicz et al (1997) y Blazewicz et al (1998), que presentan mejoras en algoritmos de resolución del problema. Estas referencias han sido incluidas en nuestro análisis, pero para las tres últimas nos hemos basado en las conclusiones de Schmidt (2000), por falta de los documentos originales.

Para resumir el análisis realizado de la revisión de la literatura, se presentan en la Tabla 1, indicando para cada autor, y por orden cronológico, el problema estudiado, si el método utilizado para resolverlo es exacto (Exac) o aproximado (Aprox), el algoritmo desarrollado para la resolución y algún comentario.

Tabla 1. Revisión para el problema con restricción de disponibilidad en dos máquinas

Autor	Problema		Algoritmo	Comentario
Lee (1997)	$F2\|r\text{-}a\|\ C_{max}$	Exac	Modelo de Programación dinámica seudo polinomial	Demuestra algunos resultados para casos concretos.
	$F2\|r\text{-}a(M_1)\|\ C_{max}$	Aprox	Algoritmo de aproximación con cota de error relativo de 3/2	Demuestra que el problema es NP con un único intervalo de no disponibilidad. Complejidad O(nlogn)
	$F2\|r\text{-}a(M_2)\|\ C_{max}$	Aprox	Algoritmo de aproximación con cota de error relativo de 4/3	Demuestra que el problema es NP con un único intervalo de no disponibilidad. Complejidad O(nlogn)
Kubiak et al (1997)**	$F2,NC_{zz}\|t\text{-}pmtn\|C_{max}$	Aprox	Heurística de tiempo no polinomial con una cota de peor caso	Solo se permiten dos intervalos de no disponibilidad
	$F2,\ NC_{win}\|pmtn\|C_{max}$	Exac	Algoritmo de branch and bound	Un único intervalo de no disponibilidad. Mejora la velocidad de solución de Lee (1997)
Blazewicz et al (1997)**	$F2,\ NC_{win}\|pmtn\|C_{max}$	Exac	Algoritmo de branch and bound paralelo	Mejora la velocidad de solución de Kubiak et al (1997)

Blazewicz et al (1998)**	$F2, NC_{win}\|pmtn\|C_{max}$	Aprox	Heurísticas constructivas y de mejora	Obtienen optimalidad en un porcentaje alto de las pruebas. Mejora además la velocidad de solución de Kubiak et al (1997)
Lee (1999)	$F2\|sr\text{-}a(M_1)\|\ C_{max}$	Exac	Algoritmo de programación dinámica	Un único intervalo de no disponibilidad. Demuestra algunos resultados relacionados al problema
		Aprox	Algoritmo de Johnson con cota de error independiente de α. Heurística con cota de error de 1 para α=0	
	$F2\|sr\text{-}a(M_2)\|\ C_{max}$	Aprox	Algoritmo de Johnson con cota de error dependiente de α. Heurística con cota de error de 1/2	Un único intervalo de no disponibilidad. Demuestra algunos resultados relacionados al problema
	$F2\|sr\text{-}a\|\ C_{max}$	Aprox	Algoritmo de Johnson	Un único intervalo de no disponibilidad. Considera unos casos especiales del problema, mostrando algunos resultados de optimalidad
Cheng y Wang (1999)	$F2\|sr\text{-}a,\ s_j = t_{\bar{j}}\ \|\ C_{max}$	Aprox	Heurística con cota de error de 2/3 para α=0 (i.e caso resumable)	$\{j, \bar{j}\} = \{1,2\}$ Un intervalo de no disponibilidad en cada máquina, consecutivos en el tiempo
Espinouse et al (1999)	$F2,h_{11}\|no\text{-}wait\|\ C_{max}$ $F2,h_{21}\|no\text{-}wait\|\ C_{max}$	Aprox	Algoritmo de tiempo polinomial con cota de error relativo de 2	Demuestra que el problema es NP-duro con un único intervalo de no disponibilidad en una máquina, y NP duro en sentido fuerte para más de un intervalo
Blazewicz et al (2000)	$F2,h_{ik}\|rs\|\ C_{max}$	Exac	Algoritmo de branch and bound paralelo	Realizan tres versiones del algoritmo y hacen una comparación computacional
Cheng y Wang (2000)	$F2\|r\text{-}a(M_1)\|\ C_{max}$	Aprox	Heurística mejorada con cota de error relativo de 1/3	Un único intervalo de no disponibilidad. Mejora el resultado de Lee (1997). Complejidad O($n$log$n$)
Espinouse et al (2001)	$F2,h_{11}\|no\text{-}wait\|\ C_{max}$ $F2,h_{21}\|no\text{-}wait\|\ C_{max}$ $F2,h_{11}\|no\text{-}wait,rs\|\ C_{max}$ $F2,h_{21}\|no\text{-}wait,rs\|\ C_{max}$	Aprox	Heurísticas basadas en el algoritmo de Gilmore y Gomory con cota de error de 2	Para los cuatro problemas demuestra que son NP-duro en sentido fuerte con la restricción de disponibilidad en una máquina. Para los dos primeros problemas se supone la restricción de "non-resumable"
Wang y Cheng (2001)	$F2\|no\text{-}wait,\ a\text{-}n(M_1)\|\ C_{max}$ $F2\|no\text{-}wait,\ a\text{-}n(M_2)\|\ C_{max}$	Aprox	Heurísticas mejoradas con cota de error relativo de 5/3	Un único intervalo de no disponibilidad. Mejora el resultado de Espinouse et al (1999)

Blazewicz et al (2001)	$F2,h_{ik}\|prmu,rs\|C_{max}$*	Aprox	Heurística basada en búsqueda local combinada con la regla de Johnson y algoritmo de Simulated Annealing	Resultados computacionales con hasta 10 intervalos de no disponibilidad. Obtienen optimalidad en un porcentaje alto de las pruebas
Kubiak et al (2002)	$F2,h_{1k}\|rs\|\ C_{max}$ $F2,h_{2k}\|rs\|\ C_{max}$	Exac	Algoritmo de branch and bound	Demuestra que el problema es NP duro en sentido fuerte con un número arbitrario de intervalos de no disponibilidad en una de las máquinas. Resultados computacionales con hasta 10 intervalos de no disponibilidad.
	$F2,h_{1k}\|rs\|\ C_{max}$	Aprox	Regla de Johnson aplicada como heurística con cota de error de 2	Complejidad O(*n*log*n*)
Cheng y Liu (2003b)	$F2,h_{11},h_{21}\|no\text{-}wait\|\ C_{max}$*	Aprox	Esquema de Aproximación en Tiempo Polinomial (PTAS)	Da las condiciones de los intervalos para poder aplicar el PTAS
Cheng y Liu (2003a)	$F2,h_{11}\ \|no\text{-}wait\|\ C_{max}$* $F2,h_{21}\ \|no\text{-}wait\|\ C_{max}$* $F2,h_{11},h_{21}\|no\text{-}wait\|\ C_{max}$*	Aprox	Algoritmos de aproximación con cota de error relativo de 3/2 para cada caso	Complejidad O(n^2log*n*). Mejora el resultado de Cheng y Liu (2003b). En el ultimo caso hay dos intervalos solapados en el tiempo
Ng y Kovalyov (2004)	$F2\|r\text{-}a(M_1)\|\ C_{max}$* denotado P-A $F2\|r\text{-}a(M_2)\|\ C_{max}$* denotado P-B	Aprox	Esquema de Aproximación en Tiempo Polinomial Completo (FPTAS)	Complejidad O(n^5/ε^4)
Breit (2004)	$F2\|r\text{-}a(M_2)\|\ C_{max}$* denotado por π	Aprox	Algoritmo de aproximación con cota de error relativo de 5/4	Mejora el resultado de Lee (1997)
Allaoui et al (2006)	$F2\|sr(M_1)\|\ C_{max}$	Exac	Regla de Johnson	Especifica las condiciones para las que la regla es óptima
	$F2\|sr(M_1)\|\ C_{max}$ $F2\|sn(M_1)\|\ C_{max}$	Aprox	Regla de Johnson aplicada como heurística con cota de error de 2	Aunque la cota es igual que la de Lee (1997), esta cota es de 1+ε con $0 \leq \varepsilon\ \leq 1$, dando cotas más ajustadas en la mayoría de casos aplicables
	$F2\|sn(M_1)\|\ C_{max}$	Aprox	Heurística de Programación dinámica	Mejora el resultado de Lee (1997)
Breit (2006)	$F2,h_{11}\|prmp\|\ C_{max}$* $F2,h_{21}\|prmp\|\ C_{max}$*	Aprox	Esquema de Aproximación en Tiempo Polinomial (PTAS)	Mejora el resultado de Cheng y Wang (2000) y Breit (2004) respect. Es alternativa a Ng y Kovalyov (2004)
Wang y Cheng (2007b)	$F2\|\ setup,r\text{-}a(M_1)\|\ C_{max}$ $F2\|\ setup,r\text{-}a(M_2)\|\ C_{max}$	Aprox	Heurísticas con cota de error de 2/3	Aunque no lo indican en la notación, el problema es de permutación. Tanto los tiempos de procesamiento como los de setup son resumables
Wang y Cheng (2007a)	$F2\|permu,setup,\ r\text{-}a(M_1)\|\ C_{max}$	Aprox	Esquema de Aproximación en Tiempo Polinomial (PTAS)	Tanto los tiempos de procesamiento como los de setup son resumables

* En el artículo no utilizan la notación indicada en el cuadro

** El análisis se ha realizado en base a las conclusiones de Schmidt (2000)

2.1.1 La regla de Johnson es óptima para $F2|prmu,a_i|C_{max}$

Como ya se ha comentado, el problema clásico de *flowshop* de secuenciar *n* trabajos en dos máquinas con criterio minimizar el makespan ($F2||C_{max}$) fue resuelto por Johnson (1954). La regla de Johnson consiste en clasificar los trabajos a secuenciar en los conjuntos $C_1 = \{j/p_{1j} < p_{2j}\}$ y $C_2 = \{j/p_{1j} > p_{2j}\}$ (si $p_{1j} = p_{2j} \Rightarrow j \in C_1 \cup C_2$ indistintamente). Los trabajos del conjunto C_1 se ordenan según la regla SPT (Short Processing Time first; se procesa primero el trabajo más corto) en función de los p_{1j}. Los trabajos del conjunto C_2 se ordenan según la regla LPT (Longest Processing Time first; se procesa primero el trabajo más largo) en función de los p_{2j}. La secuencia obtenida se denota como SPT(1)-LPT(2), y no es única. Cualquier secuencia SPT(1)-LPT(2) es óptima para $F2||C_{max}$, Pinedo (1995).

Lee (1997) estudia el caso de no disponibilidad de las máquinas en el problema de minimización del makespan en taller de flujo con dos máquinas, con un único intervalo de no disponibilidad para cada máquina (s_i, a_i), $1 \leq i \leq 2$. Lee explica que la regla de Johnson es óptima para el caso en que $s_i = 0$, i = 1, 2, es decir, para nuestro problema $F2|prmu,a_i|C_{max}$ esta regla es óptima. La demostración, que Lee no detalla, se basa en, dado un conjunto J de *n* trabajos para secuenciar en dos máquinas, crear un trabajo ficticio $n+1$ tal que $p_{1i+1} = 0$ y $p_{2j+1} = max\{0, a_2 - a_1\}$. Tomamos como instante inicial $t_0 = a_1$. Como $p_{1n+1} = p_{2n+1} \Rightarrow n+1 \in C_1$. Además como $p_{1n+1} = 0 \leq p_{1n+1} \forall j \in C_1 \Rightarrow n+1$ será el primer trabajo secuenciado según la regla SPT de los trabajos de C_1. Siguiendo la demostración de la regla de Johnson (página 98 de Pinedo (1995)) tenemos que la regla SPT(1)-LPT(2) es óptima para este problema.

Nota: Observar que sólo hay que considerar el caso en que a_2 es mayor estricto que el tiempo de procesamiento del primer trabajo secuenciado en la primera máquina cuando no se tiene en cuenta el trabajo ficticio, ya que en el caso contrario el problema se reduce a $F2| |C_{max}$.

2.2. Problemas de secuenciación en flujo regular para *m* máquinas no disponibles

El problema de con restricción de disponibilidad de las máquinas cuando hay más de dos de éstas ha sido poco estudiado. En particular, no se ha encontrado ninguna referencia en el que se trate el caso en el que las máquinas no están disponibles al principio del horizonte de planificación.

Aggoune et al (2001) estudia el problema con el objetivo de minimizar el makespan y la tardanza total ponderada para intervalos de no disponibilidad ocasionados por tareas de mantenimiento tanto no coordinado (inicio fijo en el tiempo) y coordinado (inicio flexible). Como los intervalos de no disponibilidad son arbitrarios (nunca al principio ni al final del proceso), utiliza la notación siguiente para los dos problemas: $Fm,NC_{win}||C_{max}$ y $Fm,NC_{win}||w_jT_j$, con w_j el peso del trabajo *j*, y T_j el tiempo de retraso del trabajo *j* dada la fecha de entrega de éste. Como el problema con dos máquinas y con objetivo el makespan con un número arbitrario de intervalos de no disponibilidad en una máquina es NP-duro en sentido fuerte, Kubiak et al (2002), y para la tardanza total es NP-duro, Lee (1996), concluye que los dos problemas anteriores son NP duros, Aggoune et al (2001). Para resolver el problema presentan un enfoque basado en Algoritmos Genéticos, del que no comparan resultados ante la inexistencia de algún método que resuelva este problema, por lo que realizan experimentos generando instancias de forma aleatoria. Posteriormente, Aggoune (2004) presenta otro método, basado esta vez tanto en Algoritmos Genéticos como en Búsqueda Tabú, para resolver aproximadamente el problema $Fm,NC_{win}||C_{max}$, dando los resultados de los experimentos realizados para instancias aleatorias. Al igual que el anterior no compara los resultados. Finalmente, Aggoune y Portmann (2006), basándose en un enfoque geométrico para dos trabajos, desarrollan un método en el que toman los trabajos de dos en dos para aplicar dicho enfoque combinado con una Búsqueda Tabú.

2.2.1 Complejidad del problema $Fm|prmu,a_i|C_{max}$

Como ya se ha dicho anteriormente, el problema en particular de las máquinas no disponibles en tiempo cero no ha sido estudiado. En este problema se está considerando que la secuencia es la misma para todas las máquinas, por lo que es un caso particular del problema $Fm|prmu|C_{max}$. Por lo tanto, para el problema de no disponibilidad, al igual que el problema general de *flowshop* de permutación existen *n*! posibles secuencias.

Como el problema general es NP-duro en sentido fuerte, Garey et al (1976), Pinedo (1995), se puede deducir que el problema $Fm|prmu,a_i|C_{max}$ también lo es. Esto se puede comprobar a partir de la demostración de que el problema $Fm||C_{max}$ es NP-duro en sentido fuerte de Garey et al (1976). En dicha demostración se toman $b_1, \ldots, b_{3t}$, b enteros positivos con $1/4 < b_j < 1/2$, $j = 1, \ldots, 3t$ y $\sum_{j=1}^{3t} b_j = b$. Tomando $N=4t+1$ y como tiempos de procesamiento:

$p_{10} = 0 \qquad p_{20} = b \qquad p_{30} = b$

$p_{1j} = 2b \qquad p_{2t} = b \qquad p_{3t} = 2b \qquad j = 1, \ldots, t-1$

$p_{1t} = 2b \qquad p_{2t} = b \qquad p_{3t} = 0$

$p_{1,t+j} = 0 \qquad p_{2,t+j} = b_j \qquad p_{3,t+j} = 0 \qquad j = 1, \ldots, 3t$

$z = (2t+1)b$

Añadiremos además que los instantes de disponibilidad de las máquinas como $a_1 = a_2 = 0$ y $a_3 = b$. De esta forma, la misma secuencia del problema original, $S = [0, t+1, t+2, t+3, 1, t+4, t+5, t+6, 2, \ldots, t]$ tiene como makespan el valor de z y el problema es equivalente al problema de la 3-PARTICIÓN. Así tenemos que $F3|prmu,a_i|C_{max}$ es también NP-duro en sentido fuerte. Como consecuencia $Fm|prmu,a_i|C_{max}$ es también NP-duro en sentido fuerte.

3. Conclusiones

En este trabajo se presenta la revisión de la literatura realizada para problemas de flujo regular con restricción de disponibilidad de las máquinas. Para ello se han analizando las referencias para el caso de dos máquinas, especificando el tipo de problema a resolver, si la resolución se ha llevado a cabo con método exacto o no, el tipo de algoritmo desarrollado y algunos comentarios. Además, se demuestra la optimalidad del problema con las dos máquinas no disponibles al inicio del horizonte de planificación $F2|prmu,a_i|C_{max}$, mediante la regla de Johnson (1954). Para el caso de más de dos máquinas se han resumido las conclusiones extraídas de cada una de ellas, y para el caso particular del problema $Fm|prmu,a_i|C_{max}$, se ha demostrado que es NP duro en sentido fuerte.

Agradecimientos

Querríamos agradecer a los autores de Blazewicz et al (2000) habernos proporcionado el

documento de dicha referencia.

Referencias

Aggoune, R. (2004) Minimizing the makespan for the flow shop scheduling problem with availability constraints. *European Journal of Operational Research*, Vol 153, No.3, pp. 534-543.

Aggoune, R.; Mahdi, A. H.; Portmann, M. C. (2001). Genetic algorithms for the flow shop scheduling problem with availability constraints

Aggoune, R. ; Portmann, M. C. (2006) Flow shop scheduling problem with limited machine availability: A heuristic approach. *International Journal of Production Economics*, Vol 99, No.1-2, pp. 4- 15.

Allaoui, H. et al. (2006) Scheduling of a two-machine flowshop with availability constraints on the first machine. *International Journal of Production Economics*, Vol 99, No.1-2, pp. 16- 27.

Blazewicz, J. et al. (2001) Heuristic algorithms for the two-machine flowshop with limited machine availability. *Omega*, Vol 29, No.6, pp. 599- 608.

Blazewicz, J. et al, (1998). Heuristics for two machine flow shops with limited machine availability. Discussion Paper B-9802, Fachbereich Wirtschaftswissenschaft, University of Saarland.

Blazewicz, J. et al. (2000) Parallel branch and bound algorithms for the two machine flow shop problem with limited machine availability. *Bulletin of the Polish Academy of Sciences - Technical Sciences*, Vol 48, No.1, pp. 105- 115.

Blazewicz, J. et al, (1997). A note on paralel branch and bound algorithm for the flow shop problem with limited machine availability. Working Paper, Poznan Supercomputing and Networking Center. Poznan.

Breit, J. (2004) An improved approximation algorithm for two-machine flow shop scheduling with an availability constraint. *Information Processing Letters*, Vol 90, No.6, pp. 273- 278.

Breit, J. (2006) A polynomial-time approximation scheme for the two-machine flow shop scheduling problem with an availability constraint. *Computers & Operations Research*, Vol 33, No.8, pp. 2143- 2153.

Cheng, T. C. E. ; Liu, Z. (2003a) 3/2-approximation for two-machine no-wait flowshop scheduling with availability constraints. *Information Processing Letters*, Vol 88, No.4, pp. 161-165.

Cheng, T. C. E. ; Liu, Z. (2003b) Approximability of two-machine no-wait flowshop scheduling with availability constraints. *Operations Research Letters*, Vol 31, No.4, pp. 319- 322.

Cheng, T. C. E. ; Wang, G. (2000) An improved heuristic for the two-machine flowshop scheduling with an availability constraint. *Operations Research Letters*, Vol 26, No.5, pp. 223-229.

Cheng, T. C. E. ; Wang, G. (1999) Two-machine flowshop scheduling with consecutive availability constraints. *Information Processing Letters*, Vol 71, No.2, pp. 49- 54.

Espinouse, M. L.; Formanowicz, P.; Penz, B. (2001) Complexity results and approximation algorithms for the two machine no-wait flow-shop with limited machine availability. *Journal of the Operational Research Society*, Vol 52, No.1, pp. 116- 121.

Espinouse, M. L.; Formanowicz, P.; Penz, B. (1999) Minimizing the makespan in the two-machine no-wait flow-shop with limited machine availability. *Computers & Industrial Engineering*, Vol 37, No.1-2, pp. 497- 500.

Garey, M. R.; Johnson, D. S.; Sethi, R. (1976) The Complexity of Flowshop and Jobshop Scheduling. *Mathematics of Operations Research*, Vol 1, No.2, pp. 117- 129.

Graham, R. et al. (1979) Optimization and approximation in deterministic sequencing and scheduling: A survey. *Annals of Discrete Mathematics*, Vol 5, pp. 287- 326.

Johnson, S. M. (1954) Optimal two-stages and three-stage production schedules with setup times included. *Naval Research Logistics Quaterly*, Vol Marzo.

Kubiak, W. et al. (2002) Two-machine flow shops with limited machine availability. *European Journal of Operational Research*, Vol 136, No.3, pp. 528- 540.

Kubiak, W. et al, (1997). A branch and bound algorithm for two machine flow shops with limited machine availability. Research Report RA001/97. Poznan University Technology, Institute of Computing Science, Poznan.

Lee, C. Y. (1996) Machine scheduling with an availability constraint. *Journal of Global Optimization*, Vol 9, No.4, pp. 395- 416.

Lee, C. Y. (2004). Machine Scheduling with Availability Constraints. En *Handbook of scheduling*. pp. 22.1 - 22.13. Boca Raton: CRC Press

Lee, C. Y. (1997) Minimizing the makespan in the two-machine flowshop scheduling problem with an availability constraint. *Operations Research Letters*, Vol 20, No.3, pp. 129- 139.

Lee, C. Y. (1999) Two-machine flowshop scheduling with availability constraints. *European Journal of Operational Research*, Vol 114, No.2, pp. 420- 429.

Lee, C. Y.; Lei, L.; Pinedo, M. (1997) Current trends in deterministic scheduling. *Annals of Operations Research*, Vol 70, No.0, pp. 1- 41.

Ng, C. T. ; Kovalyov, M. Y. (2004) An FPTAS for scheduling a two-machine flowshop with a one unavailability interval. *Naval Research Logistics*, Vol 51, No.3, pp. 307- 315.

Pinedo, M. (1995). *Scheduling: Theory, Algorithms, and Systems*. Prentice Hall, Englewood Cliffs, New Jersey, U.S.A.

Schmidt, G. (2000) Scheduling with limited machine availability. *European Journal of Operational Research*, Vol 121, No.1, pp. 1- 15.

Schmidt, G. ; Sanlaville, E. (1998) Machine scheduling with availability constraints. *Acta Informatica*, Vol 35, No.9, pp. 795- 811.

Türkcan, Ayten. 1999, Machine scheduling with availability constraints , Industrial Engineering Department (Bilkent University).

Wang, G. ; Cheng, T. C. E. (2001) Heuristics for two-machine no-wait flowshop scheduling with an availability constraint. *Information Processing Letters*, Vol 80, No.6, pp. 305- 309.

Wang, X. ; Cheng, T. C. E. (2007b) Heuristics for two-machine flowshop scheduling with setup times and an availability constraint. *Computers & Operations Research*, Vol 34, No.1, pp. 152-162.

Wang, X. ; Cheng, T. C. E. (2007a) An approximation scheme for two-machine flowshop scheduling with setup times and an availability constraint. *Computers & Operations Research*, Vol 34, No.10, pp. 2894- 2901.