Minimización del tiempo total de flujo de tareas en una sola máquina: Estado del arte

Jairo R. Montoya Torres*, Carlos D. Paternina-Arboleda*,
Yannick Frein**

Resumen

La programación de operaciones en una sola máquina es un problema clásico de la investigación de operaciones. Numerosos métodos han sido propuestos para resolver diferentes instancias del problema, dependiendo de las restricciones impuestas y del objetivo del mismo. En este artículo estamos interesados en ilustrar el estado actual de desarrollo de los métodos y algoritmos existentes en la literatura para el problema de minimización del flujo total de tareas sujetas a fechas de llegadas, tanto en problemas estáticos como dinámicos. Además, las posibilidades de trabajo y las preguntas abiertas serán igualmente expuestas.
Palabras clave: Programación de operaciones, una máquina, tiempo de flujo, fechas de llegada, estático, dinámico.

Fecha de recepción: 10 de abril del 2002

Abstract

Scheduling on a single-machine is a classic problem on operations research. Lots of methods have been proposed for solving different instances of the problem, depending on the constraints and the type of objective function. In this paper, we are interested in illustrating the state of the art for minimizing total flow time for jobs arriving over time, for both cases, static and dynamics problems. In addition, work possibilities and open questions will be exposed.
Key words: Scheduling, single-machine, flow-time, release dates, static, dynamic.

1. INTRODUCCIÓN

La programación de operaciones es una rama de la optimización combinatoria que ha desarrollado su propia metodología utilizando herramientas matemáticas y computacionales muy variadas. El objetivo es encontrar una secuencia óptima para la ejecución de una serie de tareas en las máquinas o recursos disponibles (limitados). El concepto de óptimo, obviamente, depende de la función objetivo.

* Grupo de Investigación en Sistemas Inteligentes, Universidad del Norte. *Jairo.Montoya@ensgi.inpg.fr*, *cpaterni@uninorte.edu.co*

** Laboratoire GILCO. Institut National Polytechnique de Grenoble (Francia). *frein@gilco.inpg.fr*

Dentro de este contexto, el estudio de los sistemas de una sola máquina permite representar el cuello de botella de una secuencia de máquinas en serie, o reagrupar todo un conjunto de ellas (o un taller de fabricación completo) de tal forma que su representación y análisis resulte más «sencillo». Igualmente, el tratamiento matemático es mucho menos complejo, y puede permitir el desarrollo de métodos heurísticos para otros tipos de sistemas. Por este motivo, estos problemas han recibido mucha atención por parte de los investigadores. En la literatura es posible encontrar varios modelos de solución para diferentes tipos de problemas de una máquina, donde la función objetivo y las restricciones del sistema juegan un papel importante en su complejidad (ver Pinedo [1995], Blazewicz *et al.* [1996]).

Por estas razones, en este artículo se ilustran los conceptos básicos y los métodos existentes en la actualidad para resolver el problema de secuenciación (programación) de operaciones en una sola máquina, donde el objetivo es la minimización del tiempo total (promedio) de terminación de los trabajos (equivalente a minimizar el tiempo total de flujo [Conway *et al.*, 1967] cuando las diferentes tareas llegan en el transcurso del horizonte de planeación. El análisis de este problema resulta interesante para diversas aplicaciones; por ejemplo, en la planeación del aterrizaje de aviones en las torres de control de los aeropuertos, donde el objetivo es minimizar el tiempo total de aterrizaje (contado desde la llegada del avión); o en la secuenciación de las instrucciones de compilación de las computadoras con el fin de optimizar su tiempo de compilación [Chekuri, 1998]. Algunos de estos métodos de solución consisten en algoritmos estáticos u *off-line*, para los cuales es necesario conocer toda la información desde el instante inicial de planeación. Otros métodos son de tipo dinámico (también llamados *on-line*), en los cuales cierta información no está disponible en t=0, pero llega en diferentes instantes de tiempo a medida que transcurre la ejecución de las tareas.

Desde el punto de vista industrial, los problemas dinámicos cobran sentido en cuanto a la incertidumbre con respecto a ciertos datos concernientes a las operaciones que se va a planificar. Debido a esto, en la actualidad las investigaciones en este campo se centran en el estudio de estos problemas, ya sea analizando los problemas a través de información probabilística o valiéndose de los modelos estáticos ya conocidos, para de esta manera «crear» los llamados métodos estático-iterativos.

Definiciones importantes

A lo largo de este artículo vamos a definir el horizonte de planeación $T_o=[0,T]$, donde T puede ser eventualmente infinito, como el horizonte de estudio para el cual se desea programar las tareas. Distinguiremos dos tipos de problemas: estático y dinámico. Un **problema estático** es aquel por el cual todos los parámetros de una instancia cualquiera son conocidos con antelación, por ejemplo, el número de tareas que se va a planificar, sus tiempos de operación, sus instantes de llegada o disponibilidad, etc. Éste es un problema idealista y utilizado como una simplificación de un problema, por el cual algunos datos son inciertos. En la literatura, los algoritmos de soluciones para este tipo de problemas son llamados *off-line*.

Un **problema dinámico** es un problema en el cual faltan algunas informaciones al instante de la toma de decisión. Sin embargo, se sabe que estas informaciones inciertas llegarán en algún momento durante el horizonte de planeación, y que las decisiones deben ser tomadas en cada instante sin tener conocimiento certero del futuro. Estas decisiones influenciarán la evolución del sistema y podrán ser tomadas de forma regular (o periódica) o cada vez que una nueva información sea conocida.

2. DESCRIPCIÓN DEL PROBLEMA

Una instancia del problema comprende un conjunto $N=\{J_1, J_2, \ldots, J_n\}$ de *n* tareas (u operaciones), en las cuales la tarea J_j tiene un tiempo de operación p_j y un instante de llegada (o fecha de llegada) r_j anterior al cual no es posible comenzar su operación. La secuenciación (programación) de las tareas en una sola máquina consiste en encontrar una secuencia S de tratamiento de las *n* tareas, con el fin de minimizar el tiempo total de flujo. Según la notación propuesta por Graham *et al.* [1979], un problema de programación de operaciones está definido por tres campos $\alpha|\beta|\gamma$, donde α representa la configuración del taller, β representa las restricciones y/o características propias a las tareas, y γ es la función objetivo que se va a minimizar. Por consiguiente, nuestro problema es conocido como $1|r_j, p_j|\sum F_j$; o equivalente, $1|r_j, p_j|\sum C_j$ [Conway *et al.*, 1967], donde $F_j=C_j-r_j$ representa el tiempo de flujo de la tarea J_j, y C_j es su instante de terminación. De esta manera, los parámetros que no aparecen dentro del campo son considerados como desconocidos al instante inicial de planeacion t=0, pero que se vuelven conocidos a lo largo del horizonte de estudio.

3. PROBLEMA ESTÁTICO EN UNA SOLA MÁQUINA

Para varios problemas de programación de operaciones en una máquina, para los cuales el objetivo es la minimización de tiempos de terminación, existen algoritmos polinomiales para encontrar la solución óptima. Para el problema $1|r_j = 0, p_j|\sum w_j C_j$, donde todas las tareas están disponibles en t=0 y cada una de ellas tiene un peso (o prioridad relativa) w_j representando su importancia, la solución óptima se obtiene ordenándolas por p_j/w_j creciente. Este criterio es conocido como la regla WSPT (*Weighted Shortest Processing Time rule*) de Smith [1956]. El caso especial $1|r_j = 0, p_j|\sum C_j$, donde $w_j=1$ para la tarea J_j, conduce a la regla SPT (*Shortest Processing Time*): ordenar las tareas por tiempos de operación no-decrecientes. La hipótesis más importante de esta solución es la igualdad de fechas de llegada para todas las tareas. Esto puede contrastar con la situación real vivida en las empresas, donde normalmente las tareas u operaciones no siempre son conocidas desde el principio.

De allí el interés de estudiar el problema cuando las tareas llegan en diferentes instantes de tiempo r_j. Una simple modificación a la regla SPT de Smith permite encontrar la solución óptima del problema $1\left|r_j, p_j, prmp\right|\sum C_j$ cuando la interrupción de la tarea en curso de ejecución está permitida: *Shortest Remaining Processing Time rule* (SRPT) [Blazewicz *et al.*, 1996]. Para el problema con interrupciones permitidas y con pesos w_j arbitrarios $1\left|r_j, p_j, prmp\right|\sum w_j C_j$, una extensión de la regla WSPT podría ser utilizada: ordenar las tareas por p_j^r / w_j creciente (donde p_j^r es el tiempo operatorio residual del trabajo J_j al momento de la decisión). Sin embargo, ésta no es óptima, puesto que este problema es, de hecho, NP-difícil [Labetoulle *et al.*, 1984]. Por consiguiente, algoritmos polinomiales han sido desarrollados para encontrar soluciones aproximativas (por ejemplo, Goemans *et al.* [1997], Schulz & Skutella [1999], entre otros).

El problema $1\left|r_j, p_j\right|\sum C_j$ con pesos w_j unitarios o iguales es NP-difícil [Lenstra *et al.*, 1977]. Como consecuencia, métodos de solución de tipo *branch & bound* (separación/evaluación, o simplemente B&B) han sido propuestos por Chandra [1979] y Dessouky y Deogun [1981]. Este último fue, posteriormente, mejorado por Deogun [1983]. Chu [1992b] ha propuesto igualmente un método B&B para resolver este problema. Para el caso más general $1\left|r_j, p_j\right|\sum w_j C_j$ con pesos w_j arbitrarios, Rinaldi y Sassano [1977] y Bianco y Ricciardinelli [1982] presentan algunas reglas de dominancia. Otros métodos enumerativos y programas lineales enteros mixtos han sido propuestos por Hairi y Potts [1983], Dyer y Wolsey [1990], Sousa y Wolsey [1992], Queyranne [1993], Belouadah *et al.* [1992], van den Akker *et al.* [1999].

Debido a su complejidad, varios algoritmos de aproximación han sido desarrollados [Chu, 1992a, Phillips *et al.*, 1995, Mao & Rifkin, 1995, Chekuri, 1998, Kellerer *et al.*, 1999] para resolverlos en tiempo polinomial, con un cierto factor de competitividad. El resultado más importante fue obtenido por Chekuri *et al.* [1997], quienes propusieron un método de solución para este problema basado en la regla SRPT. El algoritmo ordena las tareas aplicando esta metodología. Una vez que una fracción de su tiempo operatorio ha sido concluida, la tarea es entonces llevada a la lista de producción. La secuencia final consiste en producir las tareas según su orden en esta lista, obteniendo así un algoritmo pseudo-óptimo de competitividad 1,58.

Más tarde, Afrati *et al.* [1999] mejoraron este resultado al proponer el primer esquema aproximativo en tiempo polinomial (PTAS: *Polynomial-Time Approximation Scheme*). PTAS es un método que, para un valor cualquiera de $\varepsilon>0$, encuentra una solución con una aproximación de $(1+\varepsilon)$ veces el valor óptimo en tiempo polinomial. Este método es utilizado para resolver de esta manera diferentes ambientes de producción.

4. PROBLEMA DINÁMICO O MÉTODOS DE TIPO *ON-LINE*

Desde un punto de vista práctico, no toda la información está disponible en el instante de tiempo t=0, y además las tareas van llegando durante el horizonte de estudio. De esta manera, cada vez que la máquina está libre, un algoritmo *on-line* decide cuál es la tarea que podría ser tratada (si hay alguna).

El primero en estudiar los algoritmos *on-line* en programacion de operaciones fue Graham [1969]. El proponía un método *on-line* para resolver el problema de secuenciamiento de *n* operaciones en un multiprocesador, donde el objetivo era la minimizacion del *makespan*. Graham definió su algoritmo *on-line* como una Lista de Producción (*List Scheduling*), en la cual las tareas son programadas y cuando la máquina está libre, la primera tarea de la lista es tomada en ejecución.

Para el caso de una máquina, Phillips *et al.* [1995] fueron los primeros en proponer algoritmos de aproximación para problemas dinámicos de secuenciación donde el objetivo es la minimización de los tiempos de terminación de las tareas. La idea utilizada se basa en convertir secuencias de tratamiento con interrupción de tareas (preempt, en curso de operación) en no-preemptivas, penalizando el valor solución en un factor constante ρ. En primera medida, se encuentra una secuencia con interrupción de tareas {P} utilizando la regla SRPT. Luego, a partir de {P}, las tareas son ordenadas en orden creciente según las fechas de terminación C_j^P e introduciendo tiempos muertos en la máquina en caso de ser necesario. Una simple demostración permite probar que para el problema $1||\sum C_j$, la tarea J_j sólo puede terminarse como máximo en $2C_j^P$, implicando un algoritmo aproximativo de factor 2. Además, en su artículo, ellos demuestran que no es posible encontrar ningún algoritmo de tipo *on-line* (dinámico) para resolver este problema que pueda tener un factor de competitividad $\rho<3/2$.

Después de los trabajos de Phillips *et al.*, otros algoritmos polinomiales *on-line*, con la misma competitividad, fueron propuestos por Stougie [1995], Hoogeveen y

Vestjens [1996] y Goemans [1997]. Igualmente, es posible encontrar en la literatura otros trabajos que utilizan algoritmos *off-line* y *on-line*, algunos aleatorios, en combinación con relajaciones de programas lineales o con algunos de los métodos enumerativos [Hall *et al.*, 1997, Schulz & Skutella, 1997, Goemans *et al.*, 1999].

Posteriormente, Chekuri *et al.* [1997] utilizaron algunas de estas ideas para proponer otro método llamado BEST-α. Esta vez la garantía de competitividad encontrada fue de $\frac{e}{e-1} = 1{,}58$. Ellos utilizan el concepto de α-*points* introducido por Phillips *et al.* [1995] y Hall *et al.* [1996]. De la misma manera, esta idea ha sido tomada por otros autores [Schulz & Skutella, 1997, Skutella, 1998, Chekuri *et al.*, 1997, Goemans *et al.*, 1999] para el diseño de diversos algoritmos de aproximación de solución de problemas de programación de operaciones bajo diferentes configuraciones y tipos de restricciones.

El método BEST-α de Chekuri *et al.* es, de hecho, un algoritmo *off-line* (estático) que puede convertirse en *on-line* (dinámico) randomizado con la misma garantía de resultados. El concepto de base es la posibilidad de definir una función de densidad de probabilidad para α, lo cual produce un algoritmo $\frac{e}{e-1}$–aproximativo *on-line* randomizado. Torng y Uthaisombut [1999] demostraron que ningún otro algoritmo (no solamente un algoritmo de sub-secuencias SRPT) puede tener una garantía de competitividad mejor que $\frac{e}{e-1} = 1{,}58$, y para lograr esto es necesario mirar más allá de las sub-secuencias de tipo SRPT. De esta manera, es posible afirmar que BEST-ρ es un algoritmo «óptimo» (pseudo-óptimo) para el problema dinámico. Cabe anotar también que el límite de p<3/2 encontrado por Phillips *et al.* [1995] se aplica únicamente a instancias dinámicas del problema $1 || \sum C_j$, al igual que la aproximación del método BEST-α. Montoya Torres [2001] ha demostrado experimentalmente que esta referencia deja de ser válida para algunos problemas dinámicos donde cierta información futura es conocida (v.g., las fechas de llegada r_j). Si bien algunos métodos allí ensayados sólo mejoran el resultado «en promedio», también es cierto que el análisis del «peor de los casos» arroja buenas expectativas al respecto.

El lector interesado puede referirse a la obra de Borodin y El-Yaniv [1998] y al artículo de Sgall [1999] sobre la evolución, los resultados más recientes y las aplicaciones de los algoritmos *on-line* en programación de operaciones.

5. EJEMPLO DE APLICACIÓN

Una instancia del problema consiste en N={J_1, J_2, J_3, J_4, J_5}, 5 tareas, las cuales hay que ordenar con el objetivo de minimizar la suma total de los tiempos de terminación, $\sum C_j$. Los vectores **p**=(3, 18, 17, 21, 25) y **r**=(13, 0, 12, 15, 44) corresponden a los tiempos de operación y a las fechas de llegada, respectivamente.

a) Inicialmente, resolveremos el problema $1 \mid r_j = 0,\ p_j \mid \sum C_j$ suponiendo que todas las tareas están disponibles en t=0 (r_j=0, «J_j). Para este caso, la regla SPT nos da la solución óptima (figura 1.a). El valor solución es $\sum C_j^{SPT}$ =204

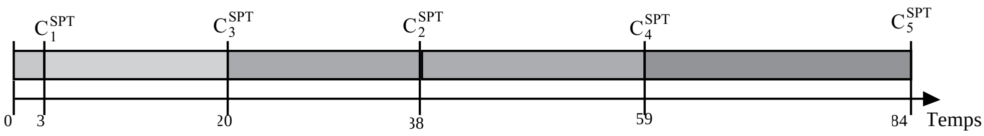

Fig. 1.a. Secuencia SPT (problema $1 \mid r_j = 0, p_j \mid \sum C_j$)

b) Ahora, el problema $1 \mid r_j, p_j, prmp \mid \sum C_j$ será resuelto utilizando la regla SRPT, con la cual obtenemos igualmente la solución óptima (figura 1.b). El vector **r**=(13, 0, 12, 15, 44) de fechas de llegada es tomado en consideración esta vez. El valor solución es, para este caso, $\sum C_j^{SRPT}$ =218.

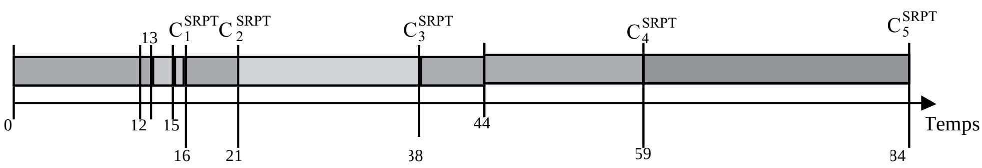

Fig. 1.b. Secuencia SRPT (problema $1 \mid r_j, p_j, prmp \mid \sum C_j$)

c) La solución óptima para el problema $1 \mid r_j, p_j \mid \sum C_j$ es ilustrada en la figura 1.c. Esta vez se utilizó un algoritmo de *branch & bound*. El valor final de $\sum C_j^{B\&B}$ =220

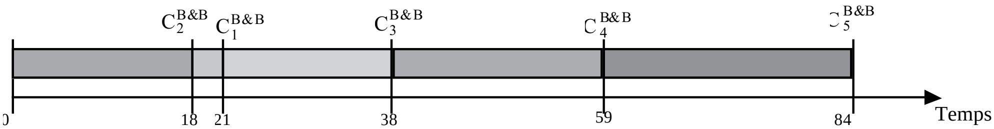

Fig. 1.c. Secuencia B&B (problema $1|r_j, p_j|\sum C_j$)

d) Finalmente, el problema dinámico $1||\sum C_j$ es resuelto utilizando el método BEST-α. La solución es ilustrada en la figura 1.d. El valor final de $\sum C_j^{Best-\alpha}$ =270.

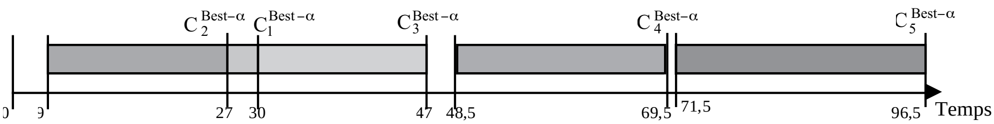

Fig. 1.d. Secuencia del problema $1||\sum C_j$

CONCLUSIONES Y PERSPECTIVAS

La minimización del tiempo total de flujo (o la suma de tiempos de terminación de tareas) en una máquina es un problema de secuenciación clásico de la literatura en optimización combinatoria. Este problema ha sido ampliamente estudiado y se han encontrado algoritmos de solución óptima en tiempo polinomial para muchas de sus variantes. Obviamente, la complejidad de un problema depende de sus características y de las restricciones impuestas.

En este artículo hemos visto que la regla WSPT nos permite obtener la solución óptima para el problema $1|r_j = 0, p_j|\sum w_j C_j$. Ahora bien, si todos los pesos de las tareas son iguales o unitarios, ésta se convierte en la conocida regla SPT. También hemos visto que una simple modificación a este método nos permite resolver en forma óptima y en tiempo polinomial el problema $1|r_j, p_j, prmp|\sum C_j$, donde las tareas tienen una fecha de llegada r_j, prioridad relativa w_j=1 y es aceptada la interrupción de la tarea en curso de operación. De otro lado, si las tareas tienen pesos w_j diferentes, éste pasa a ser un problema NP-difícil. Además, si la interrupción de la tarea en curso no es permitida, entonces los problemas de secuenciación $1|r_j, p_j|\sum C_j$ et $1|r_j, p_j|\sum w_j C_j$ se convierten también en NP-difíciles.

Para estos últimos, numerosos métodos enumerativos de tipo *branch & bound* han sido ampliamente estudiados en la literatura. Igualmente, algoritmos polinomiales *off-line* y *on-line* han sido propuestos por múltiples autores con el objetivo de encontrar soluciones aproximadas al problema.

De otra parte, si bien es cierto que el estudio dinámico de los problemas de programación de operaciones ha tenido un desarrollo bastante interesante en los últimos años, múltiples problemas todavía quedan sin resolverse. En la medida en que exista mejor conocimiento de los parámetros y restricciones del problema, el valor solución es cada vez mejor. Para diferentes problemas, las caracterísiticas de flexibilidad (restricciones sobre las operaciones) permite, igualmente, obtener mejores soluciones. Tal es el caso del problema $1 \mid r_j, p_j, \text{prmp} \mid \sum C_j$: una misma instancia del problema resuelto bajo esta situación permite un menor valor solución que la situación restringida donde no es aceptada la interrupción.

Por otro lado, el desconocimiento de algunas informaciones o parámetros conlleva a la obtención de valores solución más «elevados» que un problema con la misma instancia para el cual esa información está disponible desde el inicio de las operaciones ($1 \mid r_j, p_j \mid \sum C_j$ Vs. $1 \mid \mid \sum C_j$).

Un estudio interesante en este tema no trata solamente de comparar los valores solución, sino que el análisis de estas relaciones va mucho más allá, con el ánimo de diferenciar el (los) interés(es) de utilización de cada uno de estos enfoques. Cabe citar, por ejemplo, los trabajos realizados por Benyoucef [2000] y Benyoucef *et al.* [2000a, 2000b, 2000c] aplicados a un problema específico encontrado en la industria automotriz dentro de un contexto de «despacho sincrónico» y que ha sentado las bases para continuar estos análisis a través de problemas clásicos de la literatura [Montoya Torres, 2001]. Este último se focaliza únicamente en el manejo de información determínistica para establecer políticas de secuenciamiento conociendo algunas informaciones futuras de las tareas, dejando así abierta la posibilidad de ampliarlo utilizando estadísticamente la información del pasado (la información recibida sobre las tareas ya ejecutadas hasta un momento dado), o a través de modelización probabilística (estocástica) de los datos faltantes. De igual forma, tanto el manejo de otras funciones objetivos como la posibilidad de estudiar problemas más complejos de múltiples máquinas en parelelo o las configuraciones tipo shops se plantean como campos abiertos interesantes para este tipo de análisis.

En el caso de aplicaciones sobre problemas estocásticos, se pueden trabajar estrategias de optimización meta-heurísticas, las cuales no son objeto de estudio de este artículo. Dichas estrategias se caracterizan por elementos algorítmicos iterativos de convergencia aproximada hacia valores cercanos al óptimo. En muchos casos no se puede garantizar la optimalidad global de una solución, pero en definitiva son excelentes herramientas para discusión y comparación con los métodos exactos propuestos en este artículo y sus modificaciones hacia entornos estocásticos.

Referencias

AFRATI, F., BAMPIS, E., CHEKURI, C. *et al.* (1999). Approximation Schemes for Minimizing Average Weighted Completion Time with Release Dates. In *Report 640-1999 of the 40th Annual IEEE Symposium on Foundations of Computer Science, FOCS99*: 32-44.

VAN DEN AKKER, J.M., VAN HOESEL, C.P.M., SAVELSBERGH, M.W.P. (1999). A polyhedral approach to single-machine scheduling problems. *Math. Program., 82:* 541-572.

BELOUADAH, H., POSNER, M.E., POTTS, C.N. (1992). Scheduling with release dates on a single machine to minimize total weighted completion time. *Discrete Applied Mathematics, 36:* 213-231.

BENYOUCEF, L. (2000). *Résolution d'un problème d'ordonnancement dynamique d'un fournisseur dans un mode d'approvisionnement de type ´Livraison Synchrone[a]*. PhD Thesis, Institut National Polytechnique de Grenoble.

BENYOUCEF, L., FREIN, Y., PENZ, B. (2000a). Optimal solution for a two-product dynamic scheduling problem in a just-in-time environment. *To appear International Journal of Production Economics*.

BENYOUCEF, L., FREIN, Y., PENZ, B., WALLACE, S.W. (2000b). Résolution d'un problème d'ordonnancement dynamique dans un contexte de livraison synchrone: cas trois produits. *Journal Européen des Systèmes Automatisés*, vol. 34, Nº 10, pp.1253-1276.

BENYOUCEF, L., FREIN, Y., PENZ, B., WALLACE, S.W. (2000c). Synchronous Production or Delivery: New Dynamic Scheduling Problems. *To appear Journal of Scheduling*.

BIANCO, L., RICCIARDELLI, S. (1982). Scheduling of a Single Machine to Minimize Total Weighted Completion Time Subject to Release Dates. *Naval Research Logistic Quarter, 29:* 151-167.

BLAZEWICZ, J., ECKER, K.H., PESCH, E., SCHMIDT, G., WERGLARZ, J. (1996). *Scheduling Computer and Manufacturing Processes*. Springer.

BORODIN, A., EL-YAVIN, R. (1998). *Online Computation and Competitive Analysis*. Cambridge University Press.

CHANDRA, R. (1979). On $n/1/\bar{F}$ dynamic deterministic systems. *Naval Research Logistic Quarter, 26:* 537-544.

CHEKURI, C. (1998). *Approximation algorithms for scheduling problems*. PhD Thesis, Stanford University.

CHEKURI, C., MOTWANI, R., NATARAJAN, B., STEIN, C. (1997). Approximation techniques for average completion time scheduling. In *Proceedings of the 8th Annual ACM-SIAM Symposium on Discrete Algorithms*: 609-618. Published electronically (2001) in *SIAM J. Computing, 31* (1)*:* 146-166.

CHU, C. (1992a). Efficient heuristics to minimize total flow time with release dates. *Operations. Research Letters, 12:* 321-330.

CHU, C. (1992b). A branch and bound to minimize total flow time with unequal release dates. *Naval Research Logistic Quarter, 39:* 859-875.

CONWAY, R., MAXWELL, W., MILLER, L.W. (1967). *Theory of Scheduling*. Addison-Wesley, Reading, MA.

DEOGUN, J.S. (1983). On the Scheduling with Ready Times to Minimize Mean Flow Time. *Computer Journal*, *26* (4): 320-328.

DESSOUKY, M.I., DEOGUN, J.S. (1981). Sequencing Jobs with Unequal Ready Times to Minimize Mean Flow Time. *SIAM J. Computing*, *10* (1): 192-202.

DYER, M.E., WOLSEY, L.A. (1990). Formulating the single machine sequencing problem with release dates as a mixed integer program. *Discrete Applied Mathematics*, *26:* 255-270.

GOEMANS, M.X. (1997). Improved approximation algorithms for scheduling with release dates. In *Proceedings of the 8th ACM-SIAM Symposium on Discrete Algorithms*: 591-598.

GOEMANS, M.X., QUEYRANNE, M., SCHULZ, A., SKUTELLA, M., WANG, Y. (1999). Single machine scheduling with release dates. *Manuscript*.

GOEMANS, M.X., WEIN, J.M., WILLIAMSON, D.P. (1997). A 1,47-approximation algorithm for a preemptive single-machine scheduling problem. *Manuscript*.

GRAHAM, R.L. (1969). Bounds on multiprocessing timing anomalies. *SIAM J. Applied Mathematics*, *17:* 416-429.

GRAHAM, R.L., LAWLER, E.L., LENSTRA, J.K., RINNOOY KAN, A.H.G. (1979). Optimization and Approximation in Deterministic Sequencing and Scheduling Theory: A Survey. *Ann. Discrete Math.*, *5*: 287-326.

HAIRI, A.M.A., POTTS, C.N. (1983). An algorithm for the single machine sequencing with release dates to minimize total weighted completion time. *Discrete Applied Mathematics*, *5:* 99-109.

HALL, L.A., SCHULZ, A., SHMOYS, D.B., WEIN, J. (1997). Scheduling to minimize average completion time: off-line and on-line approximation algorithms. *Mathematics Op. Research*, *22* (3): 513-544.

HALL, L.A., SHMOYS, D.B., WEIN, J. (1996). Scheduling to minimize average completion time: off-line and on-line algorithms. In *Proceedings of the 7th ACM-SIAM Symposium on Discrete Algorithms*: 142-151.

HOOGEVEEN, J.A., VESTJENS, A.P.A. (1996). Optimal on-line algorithms for single-machine scheduling. In: W.H. Cunningham, S.T. McCornick, M. Queyranne (eds.), *Integer Programming and Combinatorial Optimization* (Proceedings of the 5th International IPCO Conference), Lecture notes in Computer Science, vol. 1048, Springer, Berlín: 404-414.

KELLERER, H., TAUTENHAHN, T., WOEGINGER, G. (1999). Approximability and Nonapproximability results for minimizing total flow time on a single machine. *SIAM J. Computing*, *28* (4): 1155-1166.

LABETOULLE, J., LAWLER, E.L., LENSTRA, J.K., RINNOOY KAN, A.H.G. (1984). Preemptive Scheduling of Uniform Machines Subject to Release Dates. In W.R. Pulleyblank (ed.), *Progress in Combinatorial Optimization* (pp. 245-261), New York, Academic Press.

LENSTRA, J.K., RINNOOY KAN, A.H.G., BRUCKER, P. (1977). Complexity of Machine Scheduling Problems. *Ann. Discrete Math.*, *1:* 343-362.

MAO, W., RIFKIN, A. (1995). On-line algorithms for a single machine scheduling problem. In S. Nash, A. Sofer (eds.) *The impact of Emerging Technologies on Computer Science and Operations Research* (pp. 157-173). Norwell, MA: Kluwer Academic Publishers.

MONTOYA TORRES, J.R. (2001). Sur une comparaison des approches statiques et dynamiques pour la minimisation des encours sur une machine. *Rapport de Stage de Recherche*, Laboratoire GILCO-INPG.

PHILLIPS, C., STEIN, C., WEIN, J. (1995). Scheduling jobs that arrive over time. In *Proceedings of the 4th Workshop on Algorithms and Data Structures*, Lecture notes on Computer Science, vol. 955, Springer, Berlín: 86-97. Journal version as: Minimizing Average Completion

Time in the Presence of Release Dates (1998). *Math. Program.*, *82:* 199-223.
PINEDO, M. (1995). *Scheduling: Theory, Algorithms and Systems*. Prentice Hall.
QUEYRANNE, M. (1993). Structure of a simple scheduling polyhedron. *Math. Program.*, *58:* 263-285.
RINALDI, G. SASSANO, A. (1977). On a job scheduling problem with different ready times: some properties and a new algorithm to determine the optimal solution. *Report R77-24, Istituto di Automatica*, Università di Roma. Cited in [Belouadah *et al.* 1992].
SCHULZ, A.S., SKUTELLA, M. (1997). Random-based scheduling: New approximations and LP lower bounds (Extended Abstract). Preprint 549/1997, Department of Mathematics, Technical University of Berlin, Germany.
SCHULZ, A.S., SKUTELLA, M. (1999). The power of α-points in preemptive single machine scheduling. Preprint 639/1999, Department of Mathematics, Technical University of Berlin, Germany.
SGALL, J. (1998). On-Line Scheduling – A Survey. In A. Fiat, G.I. Woeginger (eds.), *Proceedings of the Dagstuhl Workshop on On-Line Algorithms* (pp. 196-231), Lecture Notes in Computer Science, vol. 1442. Nueva York, Springer-Verlag.
SKUTELLA, M. (1998). *Approximation and Randomization in Scheduling*. PhD thesis, Technischen Universität Berlin.
SMITH, W.E. (1956). Various Optimizer for Single-stage Production. *Naval Research Logistic Quarter*, *3:* 606-612.
SOUSA, J.P., WOLSEY, L.A. (1992). A time indexed formulation of non-preemptive single machine scheduling problems. *Math. Program.*, *54:* 353-367.
STOUGIE, L. (1995). Personal communication, cited in Hoogeveen et Vestjens [1996].
TORNG, E., UTHAISOMBUT, P. (1999). Lower bounds for SRPT-subsequence algorithms for non-preemptive scheduling. In *Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms*, 973-974.