



ANÁLISE DE DESEMPENHO DE REGRAS DE PRIORIDADE PARA PROGRAMAÇÃO EM SISTEMAS *FLOW SHOP* HÍBRIDO E TEMPOS DE *SETUP* DEPENDENTES DA SEQUÊNCIA

CANTIERE, Patricia Castoldi, IC, Fecilcam, CNPq, Engenharia de Produção Agroindustrial, Fecilcam, patriciacantieri@hotmail.com
MORAIS, Márcia de Fátima (OR), Fecilcam, marciafmorais@yahoo.com.br
BOIKO, Thays J. P. (CO-OR), Fecilcam, thaysperassoli@bol.com.br

INTRODUÇÃO

Esta pesquisa integra uma linha de pesquisa do Grupo de Estudos e Pesquisa em Processos e Gestão de Operações (GEPPGO), do Departamento de Engenharia de Produção (DEP), da Faculdade Estadual de Ciências e Letras de Campo Mourão (FECILCAM), denominada “Pesquisa Operacional e Sub-linha, Pesquisa Operacional Aplicada à Programação da Produção”.

Financiada pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), a presente pesquisa tem por objetivo o estudo de Regras de Prioridade que poderão ser utilizadas para o desenvolvimento de novos métodos de solução para o problema de Programação da Produção em Sistemas *Flow Shop* com Máquinas Múltiplas (também conhecido como *Flow Shop* Híbrido), com tempos de *setup* (preparação) dependentes da sequência e critérios de desempenho orientados ao *Mean Flow Time* (Tempo Médio de Fluxo).

O artigo aqui apresentado está estruturado em cinco seções. A primeira seção contextualiza o trabalho, apresenta os objetivos e a justificativa; o referencial teórico acerca da programação da produção está contido na segunda seção; na terceira seção é apresentada a metodologia utilizada; a quarta seção traz os resultados e discussões da pesquisa e; na seção cinco são apresentadas as considerações finais, e as referências utilizadas.

REFERENCIAL TEÓRICO

DEFINIÇÃO DE PROGRAMAÇÃO DA PRODUÇÃO E PROBLEMA DE PROGRAMAÇÃO DA PRODUÇÃO

A Programação da Produção é de responsabilidade do departamento de Planejamento, Programação e Controle da Produção (PPCP), e tem como função decidir em



que ordem as tarefas estabelecidas nas Ordens de Produção vão ser executadas nas máquinas que estão disponíveis, cada tarefa possui um conjunto de operações que precisam ser seguidas. Segundo Santos (1998), cada operação possui seu tempo de *setup* e utiliza uma determinada máquina por um período de tempo, sendo que o lote deve passar por todas as operações requeridas, durante o período de tempo estabelecido, para satisfazer ao Plano Mestre de Produção.

Um problema de programação segundo Moccellin (2005) pode ser definido como um problema de n tarefas $\{J_1, J_2, \dots, J_j, \dots, J_n\}$ que devem ser processadas em m máquinas $\{M_1, M_2, \dots, M_k, \dots, M_m\}$ que estão disponíveis. O processamento de uma tarefa J_j em uma máquina M_k é denominado operação (op_{kj}). Existe um tempo de processamento p_{kj} associado a cada operação. Cada tarefa J_j possui uma data de liberação l_j (*release date*), a partir da qual a tarefa pode ser executada, e uma data de entrega d_j (*due date*), referente à data em que a tarefa deve estar concluída.

Os problemas de programação de operações são classificados de acordo com Maccarthy e Liu (1993) *apud* Gilio (2007), em: Máquina Única; Máquinas Paralelas; *Flow Shop* Tradicional; *Flow Shop* Permutacional; *Flow Shop* Híbrido ou com Máquinas Múltiplas; *Job Shop* Tradicional; *Job Shop* Híbrido ou com Máquinas Múltiplas; *Open Shop*; e Por Projeto.

As principais medidas ou critérios de desempenho utilizados nos métodos de solução para programação da produção, são explicadas e descritas matematicamente em French (1982), Bedworth e Bailey (1987) e Lustosa *et al.* (2008), são elas: Tempo de Espera; Tempo Total de Espera; Data de término ou Completion Time (Cj); Tempo de fluxo ou Flow Time; Pontualidade ou Lateness (Lj); Atraso ou Tardiness (Tj).

Entre as variáveis que podem ser consideradas nos problemas de programação, está o tempo de *setup*. O *setup* é o tempo que leva para se preparar uma máquina para processar a tarefa seguinte da sequência, tal como, reunir as ferramentas necessárias, fazer a limpeza da máquina, entre outras situações (Gilio, 2007). Os tempos de *setup* podem ser dependentes ou independentes da sequência de execução das tarefas. Quando o *setup* depende apenas da tarefa que espera por processamento, ele é considerado independente, quando o *setup* também depende da tarefa que foi processada anteriormente na máquina, é então considerado dependente.

As regras de prioridade estabelecem a ordem de execução das tarefas que aguardam o processamento em uma máquina ou centro de produção. Santos (1998), explica que a atividade de sequenciamento de chão-de-fábrica é composta por duas fases distintas, a primeira consiste em atribuir uma lista de tarefas para cada máquina, e a segunda consiste em decidir em que ordem essas tarefas serão processadas na máquina.



Segundo Zaccarelli (1987) *apud* Morais *et al.* (2009), cada uma das sequências possíveis implica em diferenças: nas datas de término; no tempo ocioso; no custo de preparação; e no investimento em materiais em processamento.

Os métodos de solução para problemas de Programação da produção, segundo Morais (2008) são classificados em duas grandes categorias: Métodos de solução ótima; e Métodos Heurísticos. Sendo que os métodos de solução ótima encontram a melhor solução possível para o critério de desempenho almejado, já os métodos de solução heurísticos encontram, por meio de critérios racionais, uma boa solução, mas nem sempre a ótima, sem precisar testar todas as possibilidades possíveis até chegar a solução ótima.

METODOLOGIA

A presente pesquisa classifica-se quanto aos fins, como descritiva, explicativa e exploratória, e, quanto aos meios, como bibliográfica.

Inicialmente foi efetuado um estudo para definir a teoria de base norteadora desta pesquisa. Em seguida foi realizado um levantamento bibliográfico acerca dos trabalhos para a programação em ambientes de Máquinas Paralelas, *Flow Shop* Tradicional, *Flow Shop* Permutacional e *Flow Shop* Híbrido, com critérios de desempenho orientados ao tempo de fluxo.

O levantamento bibliográfico foi realizado em nível nacional e internacional, e englobou trabalhos com tempos de *setup* incluídos e separados dos tempos de processamento. Com base nos métodos reportados na literatura, os trabalhos foram analisados para identificar quais as Regras de Prioridade foram utilizadas nos algoritmos para solução dos problemas de programação. Em seguida, analisaram-se os desempenhos de cada regra utilizada nos algoritmos reportados na revisão de literatura em relação aos critérios de tempo de fluxo das tarefas.

As regras que apresentaram melhores desempenhos serviram de base para a elaboração de um novo método de solução para o problema de programação de operações em máquinas em ambientes *Flow Shop* Híbrido e tempos de preparação das máquinas dependentes da sequência de execução das tarefas. O método heurístico *MM-FlowTime_x* proposto por Morais (2008), foi adotado como base para o desenvolvimento do novo método de solução.



RESULTADOS E DISCUSSÕES

TRABALHOS ENCONTRADOS PARA PROGRAMAÇÃO DA PRODUÇÃO COM CRITÉRIOS DE DESEMPENHO ORIENTADOS AOS TEMPOS DE FLUXO

Foram levantados trabalhos com tempos de *setup* incluídos nos tempos de processamento e trabalhos com tempos de *setup* separados dos tempos de processamento e dependentes da sequência de execução das tarefas.

Métodos de solução para Programação da Produção (PP) em Máquinas Paralelas foram encontrados em Webster (1993), Weng, Lu e Ren (2001) e Gupta e Torres (2005).

Para PP em sistemas de produção *Flow Shop* Tradicional foi encontrado apenas o trabalho de Eren e Güner (2007).

Métodos para PP em *Flow Shop* Permutacional foram encontrados em Rajendran e Chaudhuri (1991), Ho (1995), Wang, Chu e Proth (1997), Woo e Yim (1998), Liu e Reeves (2001), Framinan e Leisten (2003), Rajendran e Ziegler (2003), Framinan *et al.* (2005), Nagano e Moccellini (2007) e Laha e Sarin (2009).

Para PP em sistemas *Flow Shop* Híbrido foram encontrados métodos em Hunsukcker e Sah (1991), Rajendran e Chaudhuri (1992), Azizoglu *et al.* (2001), Allahverdi e Al-Anzi (2006) e Morais (2008).

ANÁLISE DOS RESULTADOS

O Quadro 1 a seguir apresenta as porcentagens de trabalhos encontrados, com *setup* incluídos nos tempos de processamento e *setup* separados dos tempos de processamento, e dependentes da sequência de execução das tarefas, para sistemas de Máquinas Paralelas, *Flow Shop* Tradicional, *Flow Shop* Permutacional e *Flow Shop* Híbrido. Foram encontrados um total de 19 trabalhos, sendo 3 para Máquinas Paralelas, 1 para *Flow Shop* Tradicional, 10 para *Flow Shop* Permutacional e 5 trabalhos para *Flow Shop* Híbrido.

Quadro 1 – Porcentagem dos tipos de tratamentos dos tempos de *setup* nos trabalhos investigados.

Sistema de Produção/ <i>Setup</i>	<i>Setup</i> Incluído nos Tempos de Processamento	<i>Setup</i> Dependente da Sequência	Total
Máquinas Paralelas	66,7%	33,3%	15,8%
<i>Flow Shop</i> Tradicional	100%	0%	5,3%
<i>Flow Shop</i> Permutacional	90%	10%	52,6%
<i>Flow Shop</i> Híbrido	80%	20%	26,3%
Total	84,2%	15,8%	100%/100%



No Quadro 1 pode-se observar que do total de 19 trabalhos encontrados com critério de desempenho de tempo de fluxo, 15,8% são de Máquinas Paralelas e desses trabalhos apenas 33,33% consideram os tempos de *setup* dependentes da sequência. Apenas 5,3% dos trabalhos encontrados focam a programação em *Flow Shop* Tradicional e dentre esses trabalhos nenhum consideram o tempo de *setup* separado dos tempos de processamento. Dos trabalhos encontrados 52,6% e 26,3% são de *Flow Shop* Permutacional e *Flow Shop* Híbrido, respectivamente, e dentre esses trabalhos apenas 10% consideram o *setup* dependente da sequência para *Flow Shop* Permutacional e 20% para *Flow Shop* Híbrido. Chama-se atenção aqui, para o pequeno número de trabalhos encontrados para o ambiente, *Flow Shop* Híbrido, estudado neste trabalho.

No geral, dos 19 trabalhos encontrados 84,2% consideram os tempos de *setup* incluídos nos tempos de processamento, e apenas 15,8% consideram os tempos de *setup* separados dos tempos de processamento e dependentes da sequência de execução das tarefas.

O Quadro 2 a seguir apresenta as regras de prioridade mais utilizadas nas heurísticas dos trabalhos encontrados na literatura, divididas por Sistemas de Produção. São apresentadas apenas as regras que foram utilizadas pelo menos em mais de um dos trabalhos.

Quadro 2 – Porcentagem de utilização das regras de prioridade nos trabalhos investigados.

Regras/Sistema de Produção	Máquinas Paralelas	Flow Shop Tradicional	Flow Shop Permutacional	Flow Shop Híbrido
WSPT/ W_{\max} (Menor Tempo de Processamento Ponderado/Maior Peso)	33,33%			
MSPT (Menor Tempo de Processamento Médio)	33,33%	-	-	-
SPT (Menor Tempo de Processamento)	66,67%	-	-	50%
TSPT (Menor Tempo de Processamento Total)	-	-	40%	25%
EDD (Data de Entrega Mais Cedida)	33,33%	-	10%	-
LPT(Maior Tempo de Processamento)	-	100%	-	25%
TSST (Menor Tempo Total de Começo)	-	-	20%	-



WTMIT (Menor Tempo Total Ocioso de Máquina Ponderado)	-	-	20%	-
ATFT (Menor Tempo de Fluxo Total Artificial)	-	-	20%	-

Pode-se observar no Quadro 2 que a regra SPT foi utilizada em 66,67% dos trabalhos para programação em Máquinas Paralelas e em 50% dos trabalhos em *Flow Shop* Híbrido. As variações da regra SPT também foram utilizadas em um número expressivo de trabalhos, por exemplo, a regra WSPT e a MSPT, foram utilizadas cada um em 33,33% dos trabalhos em Máquinas Paralelas, e a TSPT em 40% e 25% dos trabalhos em *Flow Shop* Permutacional e *Flow Shop* Híbrido, respectivamente. A regra EDD foi utilizada em 33,33% dos trabalhos em Máquinas Paralelas, sendo que estes trabalhos buscavam além da minimização do tempo de fluxo a minimização do número de tarefas atrasadas. A regra LPT foi utilizada no único trabalho encontrado para *Flow Shop* Tradicional e em 25% dos trabalhos em *Flow Shop* Híbrido. As demais regras foram utilizadas em uma porcentagem pequena de trabalhos.

Com a análise dos desempenhos das regras de prioridade utilizadas nas heurísticas, identificaram-se as regras que obtiveram os melhores desempenhos de acordo com os resultados encontrados na literatura, essas regras são apresentadas no Quadro 3 a seguir.

Quadro 3 – Porcentagem de trabalhos que as regras de prioridade apresentaram o melhor desempenho.

Regras	Máquinas Paralelas	<i>Flow Shop</i> Tradicional	<i>Flow Shop</i> Permutacional	<i>Flow Shop</i> Híbrido
WSPT/ W_{\max}	33,33%	-	-	-
MSPT	33,33%	-	-	-
TSPT	-	-	30%	25%
SPT	-	-	20%	50%

Pode-se observar no Quadro 3 que as regras que apresentaram os melhores desempenhos para a maioria dos trabalhos foi a regra SPT e suas variações.

HEURÍSTICA PROPOSTA

O método heurístico MM-*FlowTime_x* proposto por Morais (2008), foi adotado como base para o desenvolvimento de um novo método de solução, uma vez que o mesmo, até o momento foi o único método reportado na literatura especializada, que trata de problemas



de programação em sistemas *Flow Shop* com Máquinas Múltiplas e tempos de *setup* dependentes da sequência, com critérios de desempenho orientados aos tempos de fluxo.

Para a proposição da nova heurística o problema objeto deste trabalho foi definido como:

- 1) *Flow shop* híbrido composto por múltiplos estágios de produção, ou seja, $K \geq 2$;
- 2) Em cada estágio k , existem M_k máquinas paralelas idênticas, onde $M_k \geq 2$;
- 3) Os tempos de preparação das máquinas são assimétricos e dependentes da sequência de execução das tarefas.

As principais hipóteses consideradas no problema são as seguintes:

- 1) Os tempos de processamento das tarefas nas diversas máquinas são determinados e fixos;
- 2) As tarefas têm a mesma data de liberação, a partir da qual qualquer uma pode ser programada e executada. Esta data de liberação pode ser considerada igual a zero, no primeiro estágio, sem perda de generalidade;
- 3) Uma vez iniciadas, as operações de cada tarefa não podem ser interrompidas nem subdivididas em sub-operações simultâneas;
- 4) Uma tarefa só pode começar a ser executada em uma máquina após a execução completa da sua operação no estágio anterior e desde que a máquina já esteja preparada;
- 5) O *setup* de uma máquina, para determinada tarefa, pode ser executado antes da operação dessa tarefa estar concluída no estágio anterior e considera-se que o *setup* da primeira operação em cada máquina já esteja realizado (no problema, o tempo de *setup* dessas operações é considerado igual a zero).

O problema consiste em programar um conjunto de n tarefas, definido como $J = \{J_1, \dots, J_j, \dots, J_n\}$, onde cada tarefa possui necessariamente uma única operação em cada estágio de produção. O objetivo do problema é minimizar o tempo médio de fluxo como medida de desempenho.

A heurística- base para a proposição do novo método utiliza duas regras de prioridade (TSPT no primeiro estágio e SRD no segundo estágio) e uma regra de alocação (SCT).

Para a nova heurística, aqui denominada *MMC-FlowTime*, sugere-se a adoção da regra de prioridade MSPT (Menor Tempo de Processamento Médio) para ordenação/seqüenciamento das tarefas no primeiro estágio de produção. A regra de prioridade MSPT, uma variação da regra SPT, consiste na ordenação das tarefas em ordem não-decrescente da média dos tempos de processamento das tarefas nos k estágios de produção.



Assim, no método *MMC-FlowTime*, o primeiro estágio de produção ($k=1$) é programado como um problema tradicional de Máquinas Paralelas idênticas, com a mesma data de liberação ($r_i = r = 0$), e os demais estágios ($k \geq 2$) são programados como problemas de Máquinas Paralelas Idênticas com diferentes datas de liberações ($r_i =$ data de término da tarefa no estágio anterior).

O método *MMC-FlowTime*, estabelece a ordenação inicial com base na regra MSPT no primeiro estágio e a alocação das tarefas às m_k máquinas em cada estágio é feita com base na regra de alocação SCT (*Shortest Completion Time*). A ordenação das tarefas nos demais estágios de produção é feita com base na regra SRD (*Shortest Release Date*).

A descrição do método *MMC-FlowTime* é apresentada a seguir.

- Método Heurístico *MMC-FlowTime*

Passo 0 – (Inicialização)

$\sigma_x = \phi$ para $x = 1, 2, \dots, M$. Faça: $J' \leftarrow J$

Passo 1 - Ordene todas as tarefas do conjunto J' de acordo com a regra de prioridade MSPT. Selecione a tarefa que ocupa a primeira posição em J' ($J'_{[1]}$). Vá para o Passo 3.

Passo 2 - Ordene todas as tarefas de J' de acordo com a regra de prioridade SRD, considerando as datas de término das tarefas no estágio anterior como as datas de liberação do estágio atual. Selecione a tarefa que ocupa a primeira posição em J' ($J'_{[1]}$). Vá para o Passo 3.

Passo 3 - Em cada uma das máquinas do estágio, analise todas as possibilidades de alocação da tarefa $J'_{[1]}$ (primeira posição de J'), e escolha a máquina x com a data de término mais cedo, ou seja, aplique a regra de alocação SCT.

Passo 4 - Atualização de conjuntos. Faça: $\sigma_x \leftarrow \sigma_x \cup \{J'_{[1]}\}$ e $J' \leftarrow J' - \{J'_{[1]}\}$

Se $J' \neq \phi$ e $k = 1$ (primeiro estágio), vá para o Passo 1.

Se $J' \neq \phi$ e $k > 1$, vá para o Passo 2.

Caso contrário, programação do estágio atual concluída.

Passo 5 – Passe para o estágio seguinte. Faça $J' \leftarrow J$ (ou seja, J' passa a conter novamente as n tarefas para programação no próximo estágio).

Se não for o último estágio, vá para o Passo 2. Caso contrário pare, programação concluída. Saída: σ_m para $m = 1, 2, \dots, M$.



CONCLUSÕES FINAIS

Como o problema de programação em *Flow Shop* Híbrido é uma combinação dos problemas de Máquinas Paralelas e *Flow Shop* Tradicional, as regras de prioridade utilizadas para o desenvolvimento de heurísticas para programação nestes ambientes também podem ser utilizadas nos casos de *Flow Shop* Híbrido.

Os resultados da pesquisa mostram que a maioria das heurísticas encontradas na literatura pesquisada, utiliza a regra de prioridade SPT (Menor Tempo de Processamento), ou alguma variação desta regra, devido à mesma, oferecer bons resultados para minimização dos tempos de fluxos das tarefas.

Foi possível perceber também, que grande parte das heurísticas estudadas originou-se da combinação de duas ou mais heurísticas, ou da combinação de partes de diferentes heurísticas. Muitas vezes, heurísticas desenvolvidas para um determinado problema podem ser adaptadas e fornecer bons resultados para um problema diferente.

As análises dos resultados demonstraram que apenas 15,8% dos métodos encontrados consideraram os tempos de *setup* separados dos tempos de processamento e dependentes da sequência de execução das tarefas.

Notou-se também que as regras de prioridade que apresentaram os melhores desempenhos para o critério de tempo de fluxo foi a regra SPT ou alguma variação desta regra.

Após as análises dos resultados um novo método de solução para programação da produção, em *Flow Shop* Híbrido com tempos de *setup* dependentes da sequência e critérios de Tempo Médio de Fluxo, foi proposto. O novo método foi desenvolvido a partir do método MM-*FlowTime_x* proposto por Morais (2008), e utiliza para ordenação inicial a regra de prioridade MSPT, encontrada no trabalho de Weng, Lu e Ren (2001). Escolheu-se essa regra devido ao ótimo desempenho apresentado pela mesma nas heurísticas propostas por Weng, Lu e Ren (2001).

Sugere-se para trabalhos futuros, a experimentação computacional do novo método proposto. E posteriormente sua aplicação na prática.

REFERÊNCIAS

ALLAHVERDI, A.; AL-ANZI, F. S. Scheduling multi-stage parallel-processor services to minimize average response time. **Journal of the Operational Research Society**, Vol. 57, p. 101–110, 2006.



AZIZOGLU, M; CAKMAK, E.; KONDAKCI, S. A flexible flowshop problem with total flow time minimization. **European Journal of Operational Research**. Amsterdam, vol. 132 p.528–538, 2001.

BEDWORTH, D. D.; BAILEY, J. E. **Integrated Production Control Systems: management, analysis, design**. 2 ed. New York: John Wiley & Sons, Inc., 1987.

EREN, T.; GÜNER, E. The tricriteria flowshop scheduling problem. **The International Journal of Advanced Manufacturing Technology**. 2008, 36, 1210-1220.

FRAMINAN, J. M.; LEISTEN, R. An efficient constructive heuristic for flowtime minimisation in permutation flow shops. **Omega – The International Journal of Management Science**. Oxford, vol. 31, p. 311 – 317, 2003.

FRAMINAN, J. M.; LEISTEN, R.; RUIZ-USANO, R. Comparison of heuristics for flowtime minimisation in permutation flowshops. **Omega – The International Journal of Management Science**. Oxford, vol. 32, p. 1237–1254, 2005.

FRENCH. S. **Sequencing and scheduling: an introduction to the mathematics of the job shop**. New York: Wiley, 1982.

GILIO, N. C. **Método heurístico para o problema de no-wait flowshop com setup separado e independente da sequência**. Monografia (TCC)- Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos, 2007.

GUPTA, J.N.D.; RUIZ-TORRES, A. J. Generating efficient schedules for identical parallel machines involving flow-time and tardy jobs. **European Journal of Operational Research**. Amsterdam, vol. 167, p. 679–695, 2005.

HO, J. C. Flowshop sequencing with mean flowtime objective. **European Journal of Operational Research**. Amsterdam, vol. 72, p. 102-114, 1995.

HUNSUCKER, J.L.; SHAH, J.R. Comparative performance analysis of priority rules in a constrained flow shop with multiple processors environment. **European Journal of Operational Research**. Amsterdam, vol. 72 p. 102-114, 1991.

LAHA, D.; SARIN, S. C. A heuristic to minimize total flow time in permutation flow shop. **The International Journal of Management Science**. 2009, 37, 734-739.

LIU, J.; REEVES, C. R. Constructive and composite heuristic solutions to the $P//\sum C_i$ scheduling problem. **European Journal of Operational Research**. Amsterdam, vol. 132, p. 439-452, 2001.

LUSTOSA, L. *et al.* **Planejamento e Controle da Produção**. Rio de Janeiro: Elsevier, 2008.
NAGANO, M. S.; MOCCELLIN, J. V. **Redução do estoque em processamento em sistemas de produção flow shop**. SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL. Fortaleza, CE. Anais do Simpósio de Brasileiro de Pesquisa Operacional. Fortaleza, CE: 2005.

MOCCELLIN, J. V. **Técnicas de Seqüenciamento e Programação de Operações em Máquinas**. 74p. Publicação Escola de Engenharia de São Carlos, Universidade de São Paulo, 2005.



MORAIS, M. F. **Métodos Heurísticos Construtivos para Redução do Estoque em Processo em Ambientes de Produção Flow Shop Híbridos com Tempos de Setup Dependentes da Sequência.** Dissertação (Mestrado) – Escola de Engenharia de São Carlos, Universidade de São Paulo, São Carlos. 2008.

MORAIS, M. F.; MENEGARDE, J. K.; CANTIERE, P. C. **Regras de prioridade e critérios de desempenho adotados em problemas de programação em ambientes flow shop.** ENCONTRO CIENTÍFICO E TECNOLÓGICO (IV EPCT), 4, 2009. Campo Mourão, PR. Anais do Encontro Científico e Tecnológico. Campo Mourão, PR: 2009.

RAJENDRAN, C.; CHAUDHURI, D. An efficient heuristic approach to the scheduling of jobs in a flowshop. **European Journal of Operational Research.** Amsterdam, vol. 61, p. 318-325, 1991.

RAJENDRAN, C.; CHAUDHURI, D. A multi-stage parallel-processor flowshop problem with minimum flowtime. **European Journal of Operational Research.** Amsterdam, vol. 57, p. 111-122, 1992.

RAJENDRAN, C.; ZIEGLER, H. Scheduling to minimize the sum of weighted flowtime and weighted tardiness of jobs in a flowshop with sequence-dependent setup times. **European Journal of Operational Research.** Amsterdam, vol. 149, p. 513-522, 2003.

SANTOS, M. A. C. dos. **Sequenciamento na indústria têxtil pode tornar-se vantagem competitiva?** Resumo de tese (Mestrado)- Universidade Federal do Rio de Janeiro, Rio de Janeiro, RJ, 1998.

WANG, C. ; CHU, C. e PROTH, J. M. Heuristic approaches for $n/m/F/\sum C_i$ scheduling problems. **European Journal of Operational Research.** Amsterdam, vol. 96, p. 636-644, 1997.

WEBSTER, S.T. A priority rule for minimizing weighted flow time in a class of parallel machine scheduling problems. **European Journal of Operational Research,** Amsterdam, vol. 70 p. 327-334, 1993.

WENG, LU e REN. Unrelated parallel machine scheduling with setup consideration and a total weighted completion time objective. **Int. J. Production Economics.** 2001, 70, 215-226.

WOO, H. S. and YIM, D. S. A heuristic algorithm for mean flowtime objective in flowshop scheduling. **Computers & Operations Research.** Oxford, Vol. 25, p. 175-182, 1998.