

**REDUÇÃO DO ESTOQUE EM PROCESSAMENTO EM SISTEMAS DE  
PRODUÇÃO FLOW SHOP PERMUTACIONAL**

**REDUCING IN-PROCESS INVENTORY IN PERMUTATION FLOW  
SHOP PRODUCTION ENVIRONMENT**

**Marcelo Seido Nagano**

Professor Doutor

Universidade de São Paulo - Escola de Engenharia de São Carlos

Departamento de Engenharia de Produção

Avenida Trabalhador São-carlense, 400 – Centro, São Carlos-SP - Brasil - 13566-590

Telefone: (16) 3373-9428, E-mail: drnagano@usp.br

**João Vitor Mocellin**

Professor Titular

Universidade de São Paulo - Escola de Engenharia de São Carlos

Departamento de Engenharia de Produção

Avenida Trabalhador São-carlense, 400 – Centro, São Carlos-SP - Brasil - 13566-590

Telefone: (16) 3373-9428, E-mail: jvmoccel@prod.eesc.usp.br

**Luiz Antonio Nogueira Lorena**

Professor Titular

Instituto Nacional de Pesquisas Espaciais

Laboratório Associado de Computação e Matemática Aplicada

Avenida dos Astronautas, 1758 - Jardim da Granja, São José dos Campos – SP – Brasil -

12227-010

Telefone: (12) 3945-6550, E-mail: lorena@lac.inpe.br

## RESUMO

Este artigo trata do problema de Programação de Operações em um ambiente de produção *Flow Shop*, tendo como objetivo minimizar o estoque em processamento. Um novo método heurístico é proposto para solução do problema. Por meio de uma experimentação computacional, o desempenho do método proposto é avaliado e comparado com o melhor método heurístico reportado na literatura. Os resultados experimentais mostram a superioridade do novo método para o conjunto de problemas tratados.

**Palavras chave:** *Flow shop* permutacional, Estoque de processo, Métodos heurísticos.

## ABSTRACT

*This paper deals with the Permutation Flow Shop scheduling problem with the objective of minimizing mean flow time, therefore reducing in-process inventory. A new heuristic method is proposed for the scheduling problem solution. The proposed heuristic is compared with one of the best heuristics reported in the literature. Experimental results show that the new heuristic provides better solutions regarding both the solution quality and computational effort.*

**Keywords:** *Permutation flow shop, in-process inventory, heuristics.*

## 1. INTRODUÇÃO

O problema tradicional de Programação *Flow Shop* é um problema de produção onde um conjunto de  $n$  tarefas deve ser processado, na mesma seqüência, por um conjunto de  $m$  máquinas. Quando a ordem de processamento em todas as máquinas for a mesma, tem-se o ambiente de produção *Flow Shop* Permutacional, onde o número de possíveis programações para  $n$  tarefas é  $n!$ .

O problema consiste em obter uma seqüência das tarefas que otimiza uma determinada medida de desempenho. Nos modelos para solução do problema, as medidas usuais referem-se à minimização da duração total da programação (*makespan*), associada à utilização eficiente dos recursos produtivos, e à minimização do tempo médio de fluxo (*mean flow*

*time*), associado à redução do estoque em processamento, sendo esta última a medida adotada neste trabalho.

Este problema de programação da produção é *NP-hard* (GAREY *et al.*, 1976 e RINNOOY KAN, 1976), portanto a busca por uma solução ótima apresenta uma importância mais teórica do que prática, direcionando as pesquisas para o desenvolvimento de métodos heurísticos, sendo os mais recentes descritos sucintamente a seguir.

## **2. MÉTODOS HEURÍSTICOS PARA MINIMIZAÇÃO DO TEMPO MÉDIO DE FLUXO**

Rajendran e Chaudhuri (1991) desenvolveram três heurísticas que obtiveram melhor desempenho que os métodos heurísticos propostos por Gupta (1972) e Miyazaki *et al.* (1978), em termos da qualidade de solução e também quanto ao esforço computacional.

Na continuidade de suas pesquisas, Rajendran (1993) propôs um novo método que apresentou melhor desempenho do que as heurísticas acima mencionadas em termos de qualidade da solução, porém com alto esforço computacional.

Ho (1995) propôs uma heurística composta de diferentes iterações no processo de melhoria de uma solução inicial, a partir da obtenção de um ótimo local por permutação de pares de tarefas adjacentes, melhorando posteriormente a solução por movimentos de inserção de tarefas. Tal método heurístico apresentou desempenho melhor que os anteriores reportados na literatura, embora, como esperado, com desvantagem quanto ao tempo de computação, pelo fato de não ser uma heurística construtiva, apresentando semelhanças com metaheurísticas como *Simulated Annealing* e Busca Tabu.

Rajendran e Ziegler (1997) propuseram a heurística RZ, que consiste de duas fases: na primeira, uma seqüência inicial é gerada utilizando uma regra de prioridade similar ao *shortest weighted total processing time*; na segunda fase, a seqüência inicial obtida é melhorada por meio de inserções das tarefas em seqüências parciais, sucessivamente obtidas.

Wang *et al.* (1997) desenvolveram duas heurísticas denominadas LIT (*less idle time*) e SPD (*smallest process distance*). Suas heurísticas não são comparadas com as existentes, mas sim com um limitante inferior do tempo médio de fluxo, proposto por Ahmadi e Bagchi (1990).

Woo e Yim (1998) desenvolveram uma heurística denominada WY, comparando seu desempenho com a heurística de Rajendran (1993), e também com adaptações do NEH

(NAWAZ *et al.*, 1983) e CDS (CAMPBELL *et al.*, 1970) para o critério de minimização do tempo médio de fluxo. Eles verificaram que seu método apresentava desempenho superior aos outros, principalmente comparado com Rajendran (1993).

Allahverdi e Aldowaisan (2002) apresentaram sete heurísticas denominadas IHx ( $x = 1,2,\dots,7$ ). O melhor desempenho foi alcançado pela heurística IH7, a qual é constituída de três fases. Na primeira fase, uma solução inicial é obtida aplicando-se a heurística WY (WOO e YIM, 1998). Esta primeira solução é utilizada na segunda fase como seqüência inicial para a heurística RZ (RAJENDRAN e ZIEGLER, 1997). Finalmente, a segunda solução é melhorada por meio de uma busca local, com procedimentos de permutação nas posições das tarefas.

Framinan e Leisten (2003) propuseram o método denominado FL-IH7, que consiste em uma extensão do IH7 (ALLAHVERDI e ALDOWAISAN, 2002). A diferença relevante situa-se na obtenção da solução inicial do processo de três fases do IH7. No FL-IH7 tal solução inicial é obtida por um método construtivo que utiliza, inicialmente, o mesmo procedimento de obtenção de seqüências parciais do NEH (NAWAZ *et al.*, 1983), porém ordenando as tarefas de acordo com a soma dos tempos de processamento não-decrescentes. A seguir, as seqüências parciais são melhoradas por um procedimento de busca completa nas respectivas Vizinhanças de Permutação. Por meio de uma experimentação computacional constatou-se que o FL-IH7 apresenta um desempenho melhor quando comparado com IH7 original.

De acordo com a revisão da literatura efetuada na pesquisa relatada neste trabalho, o método FL-IH7 é o melhor em termos de qualidade da solução para a programação *flow shop* permutacional com minimização do tempo médio de fluxo.

### **3. O MÉTODO HEURÍSTICO PROPOSTO**

Em um artigo mais recente, Aldowaisan e Allahverdi (2004) tratam do caso *no-wait flow shop*. Os autores apresentam novos métodos heurísticos para solução do problema, os quais apresentam uma estrutura de três etapas:

**1ª etapa** - estabelecimento de índices de prioridades: as tarefas são ordenadas de acordo com critérios de prioridade para sua programação, geralmente baseados nos tempos de processamento;

**2ª etapa** – obtenção de uma solução inicial: uma solução é construída utilizando os índices estabelecidos;

**3ª etapa** - melhoria da solução: a solução inicial é melhorada, geralmente por meio de algum tipo de busca local (busca em vizinhança).

Essa estrutura é apropriada para o desenvolvimento de métodos para solução de problemas de natureza combinatorial, que é o caso do problema de programação da produção *flow shop*.

O método heurístico proposto neste trabalho apresenta a estrutura acima mencionada, sendo descrito a seguir.

**{Início 1ª etapa}**

**Passo 1** – Calcule, para cada tarefa, a soma dos tempos de processamento em todas as máquinas.

**Passo 2** – Ordene as  $n$  tarefas de acordo com os valores não-decrescentes das somas dos tempos de processamento (regra STPT – *Shortest Total Processing Time*);

**{Fim 1ª etapa}**

**{Início 2ª etapa}**

**Passo 3** – Selecione as duas primeiras tarefas da ordenação obtida no Passo 2, seqüenciando-as de maneira a minimizar o tempo médio de fluxo, considerando-se somente essas duas tarefas.

**Passo 4** – Para  $k = 3$  a  $n$ ,

Selecione a tarefa que ocupa a  $k$ -ésima posição na ordenação do Passo 2;

Examine as  $k$  possibilidades de inserir a tarefa na seqüência parcial até então obtida, adotando aquela que leva ao menor tempo médio de fluxo da programação parcial.

**{Fim 2ª etapa}**

**{Início 3ª etapa}**

**Passo 5** – Armazene em S a melhor seqüência obtida no Passo 4. No final do processo iterativo, a solução do problema será a seqüência S.

**Passo 6** – Para  $k = 3$  a  $n$ ,

Selecione as  $k$  primeiras tarefas da seqüência S e calcule seus respectivos tempos de fluxo;

Considerando toda a Vizinhança de Inserção da seqüência parcial com  $k$  tarefas, constituída de  $(k - 1)^2$  seqüências, determine a seqüência  $S'$  associada ao menor tempo médio de fluxo. Dada uma seqüência de tarefas, uma outra seqüência pertencente à sua vizinhança de inserção é obtida escolhendo-se uma das tarefas e uma das posições, inserindo nessa posição a tarefa escolhida.

A seguir, considerando toda a Vizinhança de Permutação da seqüência parcial  $S'$  com  $k$  tarefas, constituída de  $k(k - 1)/2$  seqüências, determine a seqüência  $S''$  associada ao menor tempo médio de fluxo. Dada uma seqüência de tarefas, uma outra seqüência pertencente à sua vizinhança de permutação é obtida trocando-se as posições de duas tarefas quaisquer.

Atualize, com a seqüência  $S''$ , as  $k$  primeiras posições da seqüência solução  $S$ .

**{Fim 3ª etapa}**

#### **4. EXPERIMENTAÇÃO COMPUTACIONAL**

O método heurístico proposto (denominado *SPT-Flowtime*) foi comparado com o melhor método reportado na literatura, ou seja, o FL-IH7 desenvolvido por Framinan e Leisten (2003). Na experimentação foram utilizados os mesmos problemas que mostraram a superioridade do FL-IH7 relativamente aos métodos anteriores, os quais foram gerados por Woo e Yim (1998) e constituem uma amostra de 5200 problemas com o número de máquinas  $m \in \{5, 10, 15, 20\}$  e o número de tarefas  $n \in \{5, 6, 7, 8, 9, 10, 20, 30, 40, 50, 60, 70, 80\}$ , correspondendo a 100 problemas para cada combinação ( $m \times n$ ). Os tempos de processamento das operações são números inteiros gerados aleatoriamente a partir de uma distribuição uniforme no intervalo  $[1, 99]$ . Neste trabalho, os métodos heurísticos foram codificados em linguagem *Delphi* e processados em um microcomputador *Pentium III 550 Mhz*.

As estatísticas usadas para avaliar o desempenho dos métodos foram a Porcentagem de Sucesso e o Desvio Médio Relativo.

Os resultados principais da experimentação computacional são ilustrados nas Figuras 1 a 6.

As Figuras 1 e 2 mostram as **Porcentagens de Sucesso** de cada método. Tal porcentagem é definida pelo quociente entre o número total de problemas para os quais o método obteve o melhor tempo médio de fluxo e o número total de problemas resolvidos. Obviamente, quando

os dois métodos obtêm o melhor tempo médio de fluxo para o mesmo problema, suas Percentagens de Sucesso são simultaneamente melhoradas.

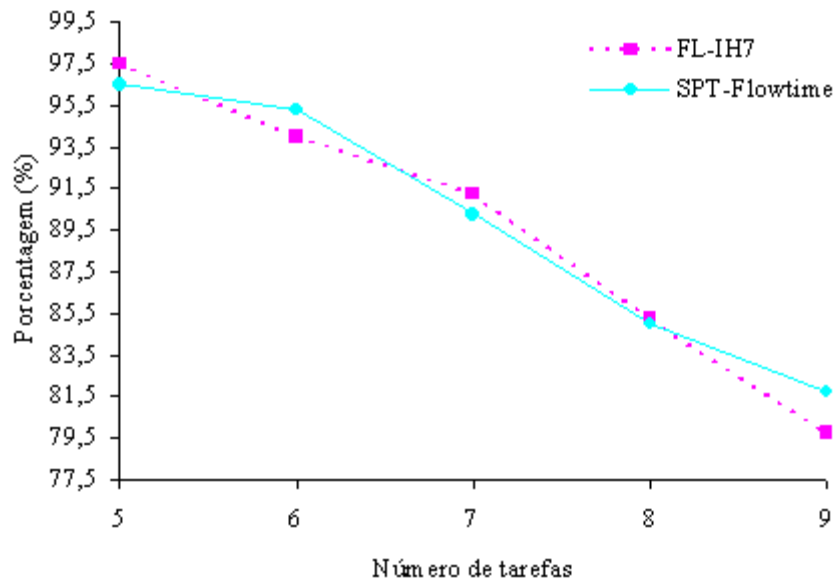


Figura 1: Percentagem de sucesso para problemas de pequeno porte.

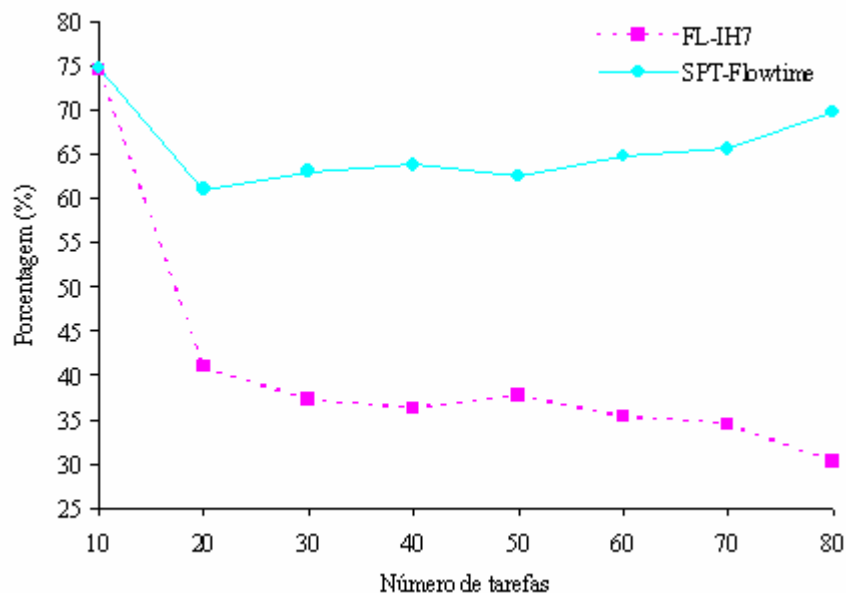


Figura 2: Percentagem de sucesso para problemas de médio e grande porte.

As percentagens de sucesso da Figura 1 mostram que os dois métodos apresentam desempenhos equivalentes para problemas pequenos, enquanto que a heurística SPT-

*Flowtime* é bem superior para problemas maiores (Figura 2), com uma tendência de crescimento da porcentagem de sucesso com o aumento do porte do problema.

As Figuras 3 e 4 apresentam os **Desvios Médios Relativos** dos dois métodos. O desvio relativo ( $DR_h$ ) quantifica o desvio que o método  $h$  obtém em relação ao melhor tempo médio de fluxo obtido para um mesmo problema, sendo calculado conforme segue:

$$DR_h = \frac{(\bar{F}_h - \bar{F}_*)}{\bar{F}_*} \quad \text{onde, } \bar{F}_h = \text{tempo médio de fluxo obtido pelo método } h;$$

$\bar{F}_*$  = melhor tempo médio de fluxo obtido pelos métodos, para um determinado problema.

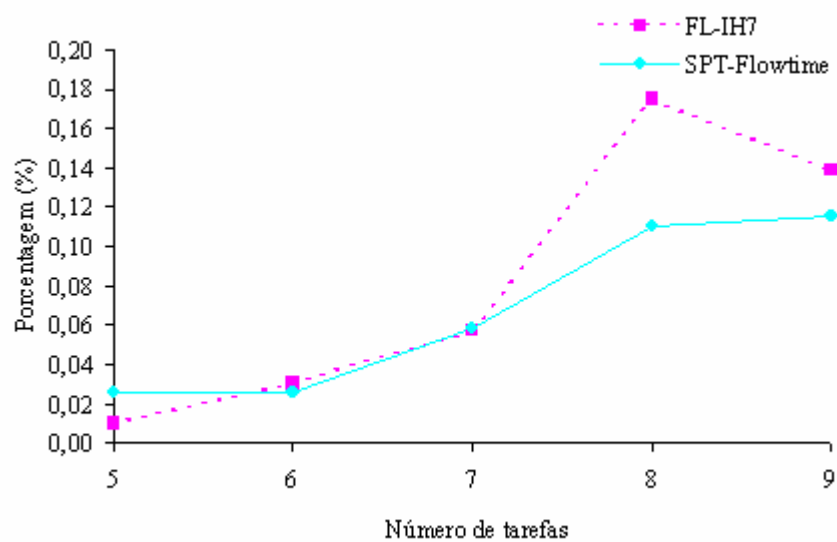


Figura 3: Desvio médio relativo para problemas de pequeno porte.



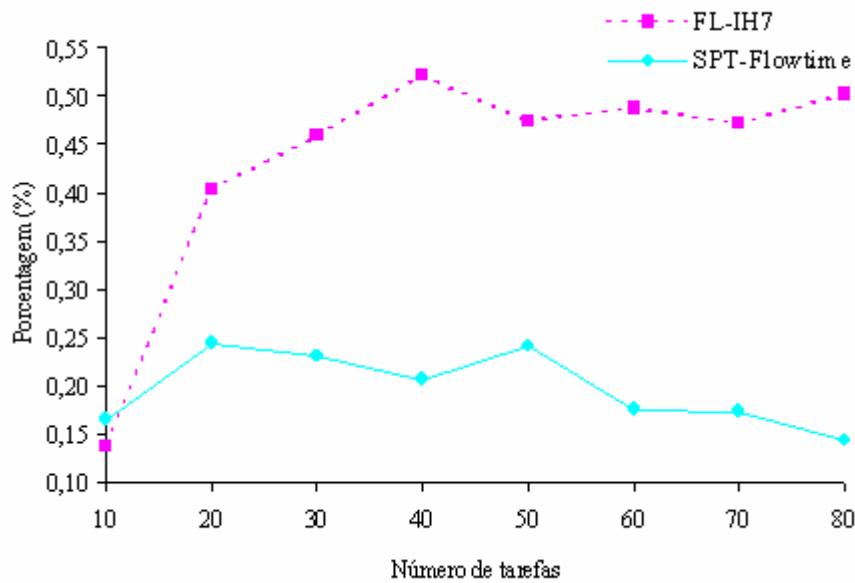


Figura 4: Desvio médio relativo para problemas de médio e grande porte.

A Figura 3 mostra que os desvios médios relativos para os dois métodos são muito próximos, com uma leve tendência de melhor desempenho para o SPT-Flowtime. Entretanto, para os problemas maiores as diferenças tornam-se mais acentuadas (Figura 4), consubstanciando a superioridade do método proposto, já mostrada pelas Porcentagens de Sucesso. Deve-se ressaltar que os valores dos desvios relativos são pequenos uma vez que o método FL-IH7 comprovadamente obtém soluções de alta qualidade.

As Figuras 5 e 6 apresentam os **Tempos Médios de Computação** em função do porte do problema.

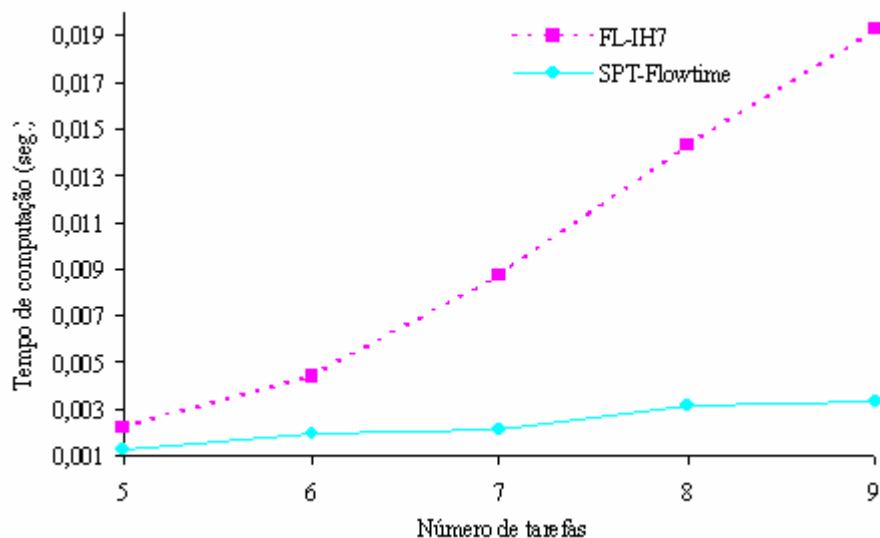


Figura 5: Tempo médio de computação para problemas de pequeno porte.

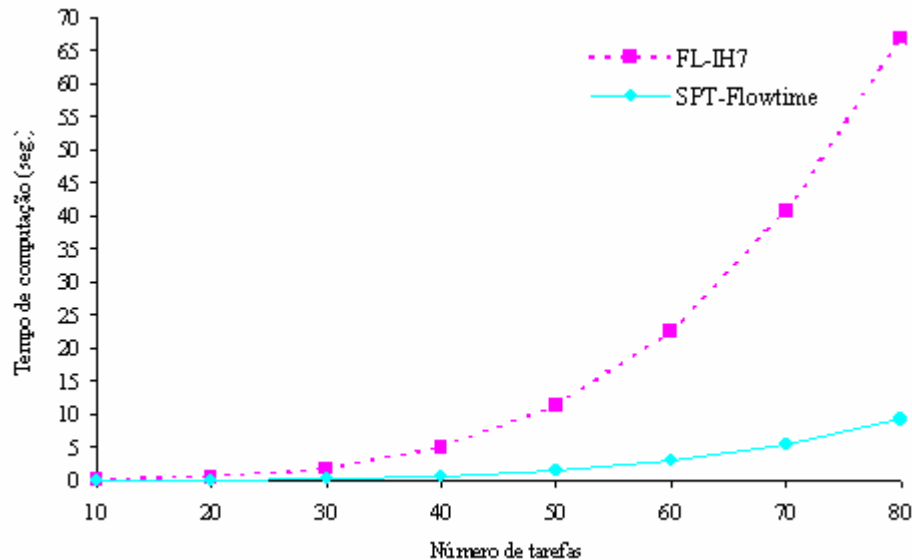


Figura 6: Tempo médio de computação para problemas de médio e grande porte.

As Figuras 5 e 6 mostram uma diferença significativa entre os tempos médios de computação dos métodos. Verifica-se nitidamente a eficiência computacional do método proposto (*SPT-Flowtime*) e a tendência de crescimento exponencial do tempo de computação do FL-IH7.

## 5. CONSIDERAÇÕES FINAIS

Os resultados experimentais mostraram que o método heurístico *SPT-Flowtime* tem um desempenho superior tanto na porcentagem de sucesso quanto no desvio médio relativo, em comparação com o melhor método referenciado na literatura para solução do problema de programação da produção em ambientes *flow shop*, tendo como medida de desempenho a redução do estoque em processamento (minimização do tempo médio de fluxo). Além de obter soluções de melhor qualidade, o esforço computacional é bem menor.

O problema clássico de sequenciamento de tarefas em um ambiente de produção *Flow Shop* tem sido objeto de intensos esforços de pesquisa nos últimos 50 anos e, para fins práticos, tal problema pode ser considerado já resolvido. Entretanto, tendo em vista sua complexidade, a busca de métodos heurísticos simples e cada vez mais eficazes quanto à qualidade da solução ainda permanece como uma direção de pesquisa. Na literatura, as medidas de desempenho

mais utilizadas têm sido a Duração Total da Programação (*makespan*) e o Tempo Médio de Fluxo (*mean flow time*), com uma frequência maior para a primeira.

A realização da pesquisa relatada neste trabalho foi motivada pelas considerações acima, procurando resgatar as características essenciais de um método heurístico, ou seja, adequado equilíbrio entre a qualidade da solução e a eficiência computacional, simplicidade e facilidade de implementação.

## REFERÊNCIAS

- AHMADI, R.H., BAGCHI, U. **Improved Lower Bounds for Minimizing the Sum of Completion Times of n Jobs over m Machines.** *European Journal of Operational Research*, 1990, v. 44, pg. 331-336.
- ALDOWAISAN, T., ALLAHVERDI, A. **New Heuristics for m-machine no-wait Flowshop to Minimize Total Completion Time.** *OMEGA*, 2004, v. 32, pg. 345-352.
- ALLAHVERDI, A., ALDOWAISAN, T. **New Heuristics to Minimize Total Completion Time in m-machine Flowshops.** *International Journal of Production Economics*, 2002, v. 77, pg. 71-83.
- CAMPBELL, H. G., DUDEK, R. A., SMITH, M. L. **A Heuristic Algorithm for n-job, m-machine Sequencing Problem.** *Management Science*, 1970. v. 16, pg. 630-637.
- FRAMINAN, J. M., LEISTEN, R. **An Efficient Constructive Heuristic for Flowtime Minimisation in Permutation Flow Shop.** *OMEGA*, 2003, v. 31, pg. 311-317.
- GAREY, M. R., JOHNSON, D. S., SETHI, R. **The Complexity of Flowshop and Jobshop Scheduling.** *Mathematics of Operations Research*, 1976, v. 1, pg. 117-129.
- GUPTA, J. N. D. **Heuristic Algorithms for Multistage Flowshop Scheduling Problem.** *AIIE Transactions*, 1972, v. 4, pg. 11-18.
- HO, J.C. **Flowshop Sequencing with Mean Flowtime Objective.** *European Journal of Operational Research*, 1995, v. 81, pg. 571-578.
- MIYAZAKI, S., NISHIYAMA, N., HASHIMOTO, F. **An Adjacent Pairwise Approach to the Mean Flowtime Scheduling Problem.** *Journal of the Operations Research Society of Japan*, 1978, v. 21, pg. 287-299.

NAWAZ, M., ENSCORE, E. E., HAM, I. **A Heuristic Algorithm for the m-machine, n-job Flow-shop Sequencing Problem.** *OMEGA*, 1983, v. 11, pg. 91-95.

RAJENDRAN, C., CHAUDHURI, D. **An Efficient Heuristic Approach to the Scheduling of Jobs in a Flowshop.** *European Journal of Operational Research*, 1991, v. 61, pg. 318-325.

RAJENDRAN, C., ZIEGLER, H. **An efficient Heuristic for Scheduling in a Flowshop to Minimize Total Weighted Flowtime of Jobs.** *European Journal of Operational Research*, 1997, v. 103, pg. 129-138.

RAJENDRAN, C. **Heuristic Algorithm for Scheduling in a Flowshop to Minimise Total Flowtime.** *International Journal Production Economics*, 1993, v. 29, pg. 65-73.

RINNOOY KAN, A.H.G. **Machine Scheduling Problems: Classification, Complexity, and Computations.** *The Hague: Nijhoff*, 1976.

WANG, C., CHU, C., PROTH, J. M. **Heuristic Approaches for n/m/F/ $\Sigma C_i$  Scheduling Problems.** *European Journal of Operational Research*, 1997, v. 96, pg. 636-644.

WOO, D. S. YIM H. S. **A heuristic Algorithm for Mean Flowtime Objective in Flowshop Scheduling.** *Computers & Operations Research*, 1998, v. 25, pg. 175-182.

---

A pesquisa relatada neste artigo teve o apoio do Conselho Nacional de Desenvolvimento Científico e Tecnológico - CNPq.

**Artigo Recebido em 15/02/2005 e aceito para publicação em 21/09/2005.**