



## ANÁLISE MULTI-CRITÉRIO NA PROGRAMAÇÃO DA PRODUÇÃO EM SISTEMAS *NO-WAIT FLOW SHOP*

**Fábio Ceron Branco**

Escola de Engenharia de São Carlos - USP  
Av. Trabalhador São-Carlense, 400 - 13566-590, São Carlos - SP  
fbranco@sc.usp.br

**Marcelo Seido Nagano**

Escola de Engenharia de São Carlos - USP  
Av. Trabalhador São-Carlense, 400 - 13566-590, São Carlos - SP  
drnagano@usp.br

**João Vitor Moccellin**

Escola de Engenharia de São Carlos - USP  
Av. Trabalhador São-Carlense, 400 - 13566-590, São Carlos - SP  
jvmoccel@sc.usp.br

### RESUMO

Este artigo trata do problema de programação de operações em um ambiente de produção *no-wait flow shop*, tendo como objetivo a minimização da soma ponderada entre a duração total da programação das tarefas e a soma dos tempos de fluxo das tarefas. A partir da análise das características do problema, um novo método heurístico é apresentado e comparado com o melhor método da literatura. Os resultados experimentais mostraram a superioridade do novo método para o conjunto de problemas tratados em relação à qualidade das soluções obtidas.

**Palavras chave:** *no-wait flow shop*, análise multi-critério, método heurístico.

**Área de classificação principal:** AGP - Administração e Gestão da Produção.

### ABSTRACT

*In this article we consider the no-wait flow shop problem with the objective of minimizing a weighted sum of makespan and total flow time. Following an investigation on the problem characteristics a new heuristic method is presented and compared with one of the best heuristic reported in the literature. Experimental results show that the new heuristic provides better solutions regarding the solution quality.*

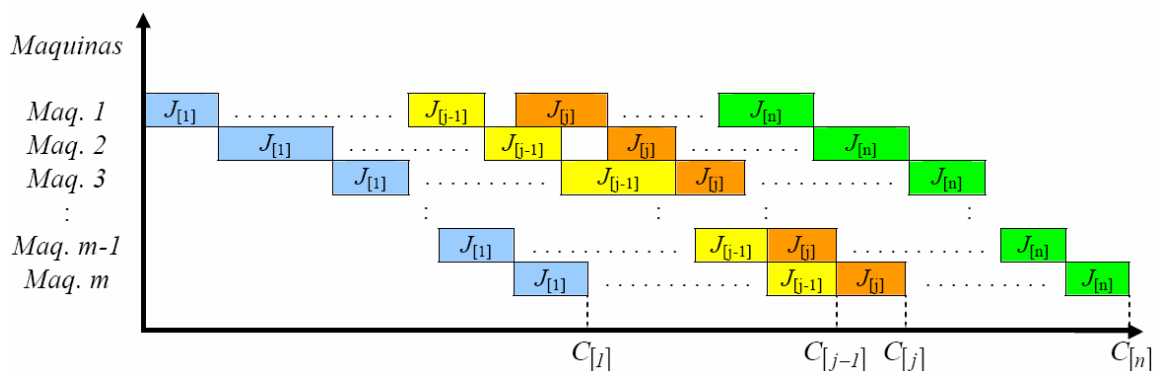
**Keywords:** *no-wait flow shop*, multi-criteria analysis, heuristic method.

**Main area:** AGP - Management & Production.

## 1. Introdução

O problema tradicional de programação *flow shop* é um problema de produção onde um conjunto de  $n$  tarefas devem ser processadas, 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 possível de programações para  $n$  tarefas é  $n!$ .

Uma situação específica e muito freqüente para este problema é a programação da produção em sistema de produção *no-wait flow shop* (*NWFS*). Nesse caso as operações de uma determinada tarefa uma vez iniciada são processadas sem interrupções nas consecutivas máquinas, conforme apresentado na figura 1.



Situações típicas para este problema são encontradas em várias indústrias, por exemplo, na produção de metais e plásticos (WISMER, 1972), indústrias químicas (RAJENDRAN, 1994), farmacêuticas (RAAYMAKERS e HOOGEVEEN, 2000) e indústrias de processamento de alimentos (HALL e SRISKANDARAYAH, 1996).

Devido a sua natureza, o problema *NWFS* é considerado *NP-Hard* para o caso de três ou mais máquinas (PAPADIMITRIOU e KANELLAKIS, 1980 e HALL e SRISKANDARAYAH, 1996).

As principais pesquisas de *NWFS* com o critério de minimização da soma dos tempos de fluxo das tarefas (*total flow time*) foram realizadas por Deman e Baker (1974), Adiri e Pohoryles (1982), Rajendran e Chaudhuri (1990), Aldowaisan e Allahverdi (1998), Bertolissi (1999), Bertolissi (2000), Allahverdi e Aldowaisan (2000, 2001) e Aldowaisan e Allahverdi (2004).

Deman e Baker (1974) foram os pioneiros a estudar o problema de *NWFS*. Em sua pesquisa eles apresentaram um algoritmo *branch and bound* para estabelecer todas as seqüências parciais, considerando a utilização de limitantes inferiores. Os autores concluíram que os resultados alcançados foram satisfatórios e que o problema podia ser solucionado tão rapidamente quanto os problemas tradicionais que utilizavam como critério a minimização da duração total da programação das tarefas (*makespan*).

Adiri e Pohoryles (1982) apresentaram propriedades para a obtenção de seqüências ótimas para o caso de duas máquinas, e teoremas para métodos polinomiais para  $m$  máquinas com séries crescentes ou decrescentes de máquinas dominantes.

Rajendran e Chaudhuri (1990) apresentaram dois métodos heurísticos, compostos de duas fases: uma primeira fase de ordenação inicial das tarefas e uma outra de construção da seqüência final avaliando seqüências parciais, estabelecendo no final a seqüência das tarefas. Os resultados da experimentação computacional mostraram que as soluções obtidas foram melhores que as soluções obtidas pelos métodos propostos por Bonney e Gundry (1976) e King e Spachis (1980).

Bertolissi (1999) apresentou um método heurístico de apenas uma fase, na qual é obtida inicialmente uma parcela da soma dos tempos de fluxo de duas tarefas adjacentes  $J_i$  e  $J_j$ , a partir do início de  $J_i$  (ou seja, no intervalo de tempo entre o início de  $J_i$  e o término de  $J_j$ ), e em seguida uma seqüência final das tarefas. As experimentações computacionais mostraram que o método proposto não obtém soluções melhores que os métodos de Rajendran e Chaudhuri (1990).

Bertolissi (2000) apresentou um novo método heurístico composto de duas fases. Na primeira fase, similar a Bertolissi (1999), e na segunda fase, foi aplicado o mesmo procedimento de construção da seqüência proposto por Rajendran e Chaudhuri (1990). Os resultados computacionais permitiram concluir que o método heurístico proposto obteve soluções melhores quando comparado aos métodos heurísticos existentes.

Allahverdi e Aldowaisan (2000, 2001) e Aldowaisan e Allahverdi (1998) analisaram o problema para duas e três máquinas com tempos de *set-up*. Foram obtidas soluções ótimas para casos especiais estabelecendo uma relação de dominância local, e fornecendo solução heurística para o caso genérico.

Aldowaisan e Allahverdi (2004) propuseram novos métodos heurísticos compostos de três fases. Conforme os resultados dos experimentos, o algoritmo PH1(p) foi o que apresentou o melhor desempenho comparado com os métodos de Rajendran e Chaudhuri (1990) e o método metaheurístico de Chen *et al.* (1996).

Para o critério de minimização da duração total da programação das tarefas as pesquisas iniciais foram realizadas por Reddi e Ramamoorthy (1972) e Wismer (1972). Posteriormente, Bonney e Gundry (1976), King e Spachis (1980), Gangadharan e Rajendran (1993) e Rajendran (1994).

Bonney e Gundry (1976) utilizaram relações geométricas e extensões de algoritmos para problemas de máquina única.

King e Spachis (1980) apresentaram métodos heurísticos com formas diferenciadas de ordenações de tarefas, e avaliaram os resultados quanto ao porte dos problemas gerados.

Gangadharan e Rajendran (1993) desenvolveram dois métodos heurísticos que se mostraram melhores que as heurísticas desenvolvidas por King e Spachis (1980). Ambos foram compostos de duas fases: uma primeira de ordenação inicial das tarefas; e uma segunda de construção da seqüência final.

Rajendran (1994) apresentou uma extensão do método de Bonney e Gundry (1976) e de King e Spachis (1980). O método foi estruturalmente semelhante aos de Gangadharan e Rajendran (1993) composto por duas fases, onde na primeira, uma ordenação inicial das tarefas é determinada por um conjunto de índices, e na segunda um procedimento de avaliação de seqüências parciais é realizado até que todas as tarefas estejam programadas.

Conforme mencionado acima, várias pesquisas têm considerado somente um dos critérios de avaliação (*makespan* ou *total flow time*). Entretanto, quando é levado em consideração ambos os critérios simultaneamente, o problema torna-se mais realístico e complexo, sendo este conhecido na literatura como análise multi-critério.

As primeiras pesquisas com análise multi-critério foram direcionadas inicialmente para os casos de máquina única (NAGAR *et al.*, 1995a).

Para o tradicional problema de *flow shop* (sem *no-wait*), Nagar *et al.*, (1995b, 1996), Sivrikaya-Serifoglu e Ulusoy (1998) e Yeh (1999, 2001) analisaram o problema com a soma ponderada entre o *makespan* e o *total flow time*. Outros estudos de multi-critério foram realizados por Rajendran (1992), Gangadharan e Rajendran (1994), Neppalli *et al.* (1996), Murata *et al.* (1996), Chakravarthy e Rajendran (1999) e Sayin e Karabati (1999).

A pesquisa mais recente para o *NWFS* com a avaliação multi-critério (*makespan* e *total flow time*) foi realizada por Allahverdi e Aldowaisan (2002), onde um novo método heurístico chamado de PAAH foi apresentado. O novo método foi avaliado com os métodos de Rajendran e Chaudhuri (1990), Gangadharan e Rajendran (1993), Rajendran (1994) e Chen *et al.* (1996), chamados respectivamente de RC, GR, RAJ-A e GA. Os resultados computacionais apresentaram a superioridade do método PAAH para o conjunto de problemas avaliados.

De acordo com a revisão da literatura efetuada na pesquisa referente a este trabalho, o método PAAH é o melhor em termos de qualidade da solução para a programação *NWFS* com a avaliação multi-critério composta pela soma ponderada entre a *makespan* e o *total flow time*. Esta função objetivo é chamada de  $W(\alpha)$ , onde  $C_{[j]}$  é o tempo de fluxo de uma tarefa na posição  $j$  na seqüência  $\sigma$  na última máquina. Logo, a expressão da função objetivo é definida por:

$$W(\alpha) = \alpha C_{max} + (1 - \alpha) TCT \quad (1)$$

onde:

$$0 \leq \alpha \leq 1$$
$$C_{max} = \max_l \{C_{[l]}\} \quad (2)$$

$$TCT = \sum_{l=1}^n C_{[l]} \quad (3)$$

O objetivo deste trabalho é apresentar um novo método heurístico para o problema comparando-o com o melhor método da literatura proposto por Allahverdi e Aldowaisan (2002) (PAAH). O novo método é avaliado verificando o seu desempenho quanto à qualidade da solução, e resgatando as características essenciais de um método heurístico: adequado equilíbrio entre a qualidade da solução e esforço computacional, simplicidade e facilidade de implementação.

## 2. Método heurístico proposto

O método heurístico proposto possui uma única fase, e será chamado de MC-FS (Multi-Critério *Flow Shop*).

O método proposto possui como base as expressões apresentadas por Lawer *et al.* (1993) e Bertolissi (2000), que tratam respectivamente do tempo da programação de pares de tarefas adjacentes e da parcela da soma dos tempos de fluxo de pares de tarefas também adjacentes.

Para o caso de  $m$  máquinas uma expressão recursiva é apresentada, onde para cada par de tarefas  $J_i$  e  $J_j$  são obtidas a duração da programação (expressão 4) e a soma dos tempos de fluxo das tarefas (expressão 5), conforme apresentado abaixo, onde o tempo de processamento da tarefa  $J_i$  na máquina  $k$  é  $p_{ki}$  ( $k = 1, 2, \dots, m; i = 1, 2, \dots, n$ ).

$$C_m(J_i, J_j) = p_{li} + \delta_m(J_i, J_j) \quad (4)$$

para  $i = 1, 2, \dots, n$  e  $j = 1, 2, \dots, n$

onde:

$$\delta_0(J_i, J_j) = 0$$

$$\delta_m(J_i, J_j) = p_{mj} + \max\left(\delta_{m-1}(J_i, J_j); \sum_{k=2}^m p_{ki}\right)$$

$$F_m(J_i, J_j) = 2p_{li} + \sum_{k=2}^m p_{ki} + \beta_m(J_i, J_j) \quad (5)$$

para  $i = 1, 2, \dots, n$  e  $j = 1, 2, \dots, n$

onde:

$$\beta_1(J_i, J_j) = p_{lj}$$

$$\beta_m(J_i, J_j) = p_{mj} + \max\left(\beta_{m-1}(J_i, J_j), \sum_{k=2}^m p_{ki}\right)$$

A seguir é apresentado o método MC-FS para solução do problema multi-critério (*makespan* e *total flow time*) na programação da produção em sistemas *no-wait flow shop*.

## 2.1. Método heurístico MC-FS

Seja  $J$  um conjunto de  $n$  tarefas ( $J = \{J_1, J_2, \dots, J_n\}$ ) a serem programadas,  $S$  o conjunto das tarefas já programadas;  $S_{[k]}$  a tarefa que ocupa a  $k$ -ésima posição da seqüência  $S$ , e  $S_p$  a melhor seqüência parcial obtida após a execução do processo de melhoria que utiliza a permutação de tarefas, e  $S_i$  a melhor seqüência parcial obtida após a execução do processo de melhoria que utiliza a inserção de tarefas.

### Passo 1

$$J = \{J_1, J_2, \dots, J_i, \dots, J_n\};$$

$$S = \emptyset;$$

$$\alpha \leftarrow \{0; 0,25; 0,50; 0,75; 1,0\}$$

$$\text{Calcule } \omega(J_i, J_j) = \alpha[C_m(J_i, J_j)] + (1 - \alpha)[F_m(J_i, J_j)]$$

Selecione o maior elemento  $\omega(J_i, J_j)$ ;

$$S = \{J_j, J_i\};$$

$$J \leftarrow J - \{J_i, J_j\};$$

$$u \leftarrow J_{[2]};$$

$$k = 3;$$

### Passo 2

Selecione o menor elemento  $\omega(J_u, J_v)$  tal que  $J_v \in J$ ;

$$S \leftarrow S \cup \{J_v\};$$

$$J \leftarrow J - \{J_v\};$$

$$k \leftarrow k + 1;$$

### Passo 3

- Examine todas as possibilidades de inserir a tarefa  $J_v$  na seqüência parcial  $S$ , e adote aquela que leva ao menor  $W$  ;
- Considerando toda a Vizinhança de Inserção da seqüência parcial com  $(k - 1)$  tarefas, constituída de  $(k - 2)^2$  seqüências, determine a seqüência  $S_i$  associada ao menor  $W$  ;
- Se  $W$  de  $S_i$  é menor que de  $S$ , atualize  $S$  com  $S_i$  e, considerando toda a Vizinhança de Permutação da seqüência parcial com  $(k - 1)$  tarefas, constituída de  $(k - 1)(k - 2)/2$  seqüências, determine a seqüência  $S_p$  associada ao menor  $W$  ;
- Se  $W$  de  $S_p$  é menor que de  $S$ , atualize  $S$  com  $S_p$  ;
- Se  $W$  de  $S_i$  não é menor que de  $S$ , considerando toda a Vizinhança de Permutação da seqüência parcial com  $(k - 1)$  tarefas, constituída de  $(k - 1)(k - 2)/2$  seqüências, determine a seqüência  $S_p$  associada ao menor  $W$  ;
- Se  $W$  de  $S_p$  é menor que de  $S$ , atualize  $S$  com  $S_p$  ;
- $u \leftarrow J_{[k]}$  de  $S$  ;
- Volte ao *Passo 2* até que todas as tarefas estejam seqüenciadas.

Em seguida serão apresentadas a experimentação computacional e a análise dos resultados do experimento.

### 3. Experimentação computacional e métodos de análise

O método heurístico proposto foi comparado com o melhor método reportado na literatura, ou seja, o PAAH desenvolvido por Allahverdi e Aldowaisan (2002). Na experimentação foi utilizada a mesma amostra constituída de 5200 problemas-teste, referente à pesquisa reportada em Nagano e Moccellini (2007) com o número de tarefas  $n \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130\}$ , o número de máquinas  $m \in \{5, 10, 15, 20\}$ , 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 IV 3.00 GHz*.

As estatísticas usadas para avaliar o desempenho dos métodos foram a Porcentagem de Sucesso, Desvio Relativo, Desvio Padrão do Desvio Relativo e Tempo de Computação.

A primeira é definida pelo quociente entre o número de problemas para os quais o método obteve o menor  $W$  e o número total de problemas resolvidos. Obviamente, quando os dois métodos obtêm o menor  $W$  para um mesmo problema, suas Porcentagens de Sucesso são simultaneamente melhoradas.

O Desvio Relativo ( $DR_{h_i}$ ) quantifica o desvio que o método  $h$  obtém em relação ao menor  $W$  para o mesmo problema  $i$ , sendo calculado conforme segue:

$$DR_{h_i} = (W_{h_i} - W_i^*) / W_i^* \quad (6)$$

onde:

$W_{h_i}$  = valor de  $W$  obtido pelo método  $h$  para o problema  $i$ ;

$W_i^*$  = menor  $W$  obtido para o problema  $i$ .

O cálculo do Desvio Padrão do Desvio Relativo para o método  $h$  ( $S_{DR_h}$ ) para uma quantidade  $q$  de problemas foi obtido pela seguinte expressão:

$$S_{DR_h} = \sqrt{\frac{\sum_{i=1}^q (DR_{hi} - \overline{DR_h})^2}{q-1}} \quad (7)$$

onde:

$DR_{hi}$  = Desvio Relativo do método  $h$  para o problema  $i$ ;

$\overline{DR_h}$  = Desvio Relativo Médio do método  $h$  para uma quantidade  $q$  de problemas  $(\sum_{i=1}^q DR_{hi} / q)$ .

O Tempo Médio de Computação de um método é obtido pela soma dos tempos de computação de cada problema dividida pelo número total de problemas resolvidos. Na experimentação computacional, o tempo médio de computação foi medido em milissegundos ( $ms$ ).

#### 4. Análise dos resultados

Os resultados principais da experimentação computacional são apresentados nas Tabelas 1 e 2.

Tabela 1 – Resultados das heurísticas para problemas de pequeno e médio porte.

Número de Tarefas	$\alpha$	Porcentagem de Sucesso (%)		Desvio Relativo Médio (%)		Desvio Padrão do Desvio Relativo		Tempo Médio de Computação (ms)	
		PAAH	MC-FS	PAAH	MC-FS	PAAH	MC-FS	PAAH	MC-FS
10	0,00	44,50	82,25	0,747	0,109	0,011	0,003	1,70	1,29
	0,25	45,50	79,75	0,749	0,146	0,012	0,004	1,58	1,33
	0,50	43,25	76,50	0,882	0,169	0,013	0,004	2,77	1,60
	0,75	44,00	75,25	0,880	0,269	0,013	0,006	1,62	1,52
	1,00	50,75	68,50	0,882	0,424	0,013	0,009	1,80	1,29
20	0,00	18,25	83,00	1,317	0,087	0,012	0,003	4,13	3,63
	0,25	13,25	87,00	1,397	0,089	0,012	0,003	4,03	3,51
	0,50	17,00	83,25	1,564	0,111	0,014	0,003	4,48	3,55
	0,75	27,00	74,00	1,318	0,244	0,014	0,006	4,09	3,55
	1,00	34,00	67,25	1,154	0,392	0,013	0,008	4,11	3,44
30	0,00	6,75	93,25	1,859	0,032	0,013	0,002	8,52	9,93
	0,25	10,50	89,50	1,740	0,043	0,013	0,002	8,55	9,97
	0,50	11,50	88,50	1,776	0,051	0,013	0,002	9,41	9,73
	0,75	18,25	81,75	1,733	0,127	0,014	0,004	8,55	10,04
	1,00	31,00	70,00	1,098	0,265	0,011	0,005	8,83	9,90
40	0,00	6,50	93,50	2,112	0,030	0,012	0,001	16,06	22,80
	0,25	5,50	94,50	2,177	0,033	0,012	0,002	16,14	22,70
	0,50	6,25	93,75	2,159	0,032	0,013	0,001	16,96	22,69
	0,75	11,50	88,50	1,989	0,073	0,014	0,002	16,37	22,58
	1,00	26,75	74,50	1,227	0,204	0,011	0,004	16,48	22,90
50	0,00	2,25	97,75	2,397	0,005	0,011	0,000	28,04	48,92
	0,25	2,25	97,75	2,352	0,007	0,012	0,000	28,01	48,99
	0,50	2,00	98,00	2,390	0,004	0,012	0,000	29,07	48,48
	0,75	8,25	91,75	2,303	0,042	0,014	0,002	28,40	48,94
	1,00	20,00	80,50	1,280	0,132	0,010	0,003	28,48	48,63

60	0,00	1,00	99,00	2,547	0,002	0,012	0,000	44,53	92,93
	0,25	1,50	98,50	2,509	0,006	0,012	0,000	44,33	92,94
	0,50	2,50	97,50	2,536	0,006	0,012	0,000	45,36	93,17
	0,75	4,00	96,00	2,462	0,014	0,013	0,001	44,65	93,21
	1,00	18,50	82,00	1,297	0,108	0,011	0,003	45,39	92,70
70	0,00	0,50	99,50	2,783	0,001	0,010	0,000	66,64	159,26
	0,25	0,50	99,50	2,655	0,005	0,011	0,000	66,32	159,26
	0,50	2,75	97,50	2,789	0,007	0,012	0,000	67,43	159,34
	0,75	3,25	96,75	2,611	0,010	0,013	0,001	66,72	159,03
	1,00	21,50	78,75	1,326	0,123	0,010	0,003	66,80	160,11

Tabela 2 – Resultados das heurísticas para problemas de grande porte.

Número de Tarefas	$\alpha$	Porcentagem de Sucesso (%)		Desvio Relativo Médio (%)		Desvio Padrão do Desvio Relativo		Tempo Médio de Computação (ms)	
		PAAH	MC-FS	PAAH	MC-FS	PAAH	MC-FS	PAAH	MC-FS
80	0,00	0,00	100,00	2,986	0,000	0,011	0,000	94,11	257,69
	0,25	1,50	98,50	2,918	0,004	0,010	0,000	94,42	257,77
	0,50	1,50	98,50	2,910	0,008	0,011	0,000	95,66	257,75
	0,75	2,75	97,25	2,810	0,012	0,012	0,001	94,64	257,49
	1,00	13,50	86,50	1,398	0,065	0,009	0,002	95,94	258,68
90	0,00	0,75	99,25	3,168	0,001	0,010	0,000	130,77	397,15
	0,25	0,25	99,75	3,151	0,001	0,010	0,000	130,67	397,07
	0,50	0,75	99,25	3,094	0,002	0,011	0,000	130,94	397,55
	0,75	2,50	97,50	2,952	0,008	0,012	0,000	131,28	397,16
	1,00	11,75	88,50	1,416	0,052	0,009	0,002	133,59	398,60
100	0,00	0,25	99,75	3,332	0,000	0,010	0,000	173,71	586,60
	0,25	0,00	100,00	3,344	0,000	0,010	0,000	173,13	586,76
	0,50	0,25	99,75	3,144	0,002	0,009	0,000	173,51	586,25
	0,75	2,00	98,00	3,164	0,006	0,011	0,000	174,14	594,34
	1,00	11,00	89,00	1,588	0,058	0,009	0,002	182,07	590,13
110	0,00	0,25	99,75	3,415	0,001	0,009	0,000	226,95	838,25
	0,25	0,25	99,75	3,355	0,000	0,009	0,000	227,22	838,48
	0,50	1,00	99,00	3,325	0,002	0,010	0,000	227,54	837,66
	0,75	0,75	99,25	3,203	0,001	0,010	0,000	227,82	841,29
	1,00	13,25	87,00	1,511	0,053	0,009	0,002	231,09	842,80
120	0,00	0,25	99,75	3,514	0,000	0,009	0,000	291,02	1172,38
	0,25	0,00	100,00	3,471	0,000	0,009	0,000	290,77	1166,18
	0,50	0,00	100,00	3,412	0,000	0,009	0,000	291,26	1167,77
	0,75	0,25	99,75	3,372	0,000	0,009	0,000	291,83	1189,06
	1,00	14,50	85,50	1,611	0,054	0,008	0,001	296,48	1170,28
130	0,00	0,25	99,75	3,602	0,000	0,009	0,000	366,17	1583,80
	0,25	0,00	100,00	3,504	0,000	0,009	0,000	366,13	1583,60
	0,50	0,25	99,75	3,487	0,001	0,009	0,000	366,76	1584,11
	0,75	1,00	99,00	3,291	0,002	0,011	0,000	367,34	1587,58
	1,00	9,25	90,75	1,600	0,045	0,008	0,001	372,00	1586,92

As porcentagens de sucesso apresentadas nas tabelas 1 e 2 mostram que os dois métodos apresentam desempenhos diferentes para problemas de pequeno, médio e grande porte, enquanto que a heurística PAAH apresenta uma tendência de crescimento da porcentagem de sucesso com o aumento de  $\alpha$  para um determinado número de tarefas, a heurística MC-FS apresenta tendência



inversa. Apesar das tendências apresentadas, a porcentagem de sucesso do MC-FS é bem superior principalmente para problemas com um maior número de tarefas.

As Tabelas 1 e 2 mostram que os desvios relativos médios para os dois métodos são diferentes, com tendência de melhor desempenho para o MC-FS. Entretanto, analisando a variação de  $\alpha$  para um determinado número de tarefas, verifica-se uma tendência inversa de desempenho das heurísticas, mais evidente para a heurística MC-FS. Para os problemas maiores, as diferenças tornam-se mais acentuadas, 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 a heurística PAAH comprovadamente obtém soluções de alta qualidade.

Observando as Tabelas 1 e 2 em relação ao Desvio Padrão do Desvio Relativo, de maneira geral, com exceção dos problemas de pequeno e médio porte, os valores do Desvio Padrão do Desvio Relativo são suficientemente pequenos para a heurística MC-FS. Deve-se ressaltar que os valores dos Desvios Padrões são pequenos uma vez que o MC-FS possui maior estabilidade de desempenho em relação ao conjunto de problemas avaliados.

Em relação aos tempos médios de computação dos métodos, verifica-se nitidamente que a heurística MC-FS apresenta maiores tempos para o número de tarefas superior a 20. Entretanto, essa diferença não é significativa levando em consideração a alta qualidade da solução obtida e estabilidade de desempenho da heurística proposta.

## 5. Considerações finais

Os resultados experimentais mostraram que o método heurístico MC-FS tem um desempenho superior tanto na porcentagem de sucesso, no desvio relativo e no desvio padrão do desvio relativo para o conjunto de problemas avaliados, em comparação com o melhor método referenciado na literatura para o problema (PAAH).

O problema clássico de seqüenciamento 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. Na literatura, as medidas de desempenho mais utilizadas têm sido a duração total da programação (*makespan*) e a soma dos tempos de fluxo (*total flow time*), com frequência maior para a primeira. 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 quando novas medidas de desempenho são combinadas ou consideradas, caso este apresentado neste trabalho.

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.

## 6. Referências

- Adiri, I e Pohoryles, D.** (1982), Flowshop/no-idle or no-wait scheduling to minimize the sum of completion times, *Naval Research Logistic Quarterly*, 29, 495-504.
- Aldowaisan, T. e Allahverdi, A.** (1998), Total flowtime in no-wait flowshops with separated setup times, *Computers & Operations Research*, 25, 757-765.
- Aldowaisan, T. e Allahverdi, A.** (2004), New heuristics for m-machine no-wait flowshop to minimize total completion time, *OMEGA – The International Journal of Management Science*, 32, 345- 352.

**Allahverdi, A. e Aldowaisan, T.** (2000), No-wait and separate setup three-machine flowshop with total completion time criterion, *International Transactions in Operational Research*, 7, 245-264.

**Allahverdi, A. e Aldowaisan, T.** (2001), Minimizing total completion time in a no-wait flowshop with sequence dependent additive changeover times, *Journal of the Operational Research Society*, 52, 449-462.

**Allahverdi, A. e Aldowaisan, T.** (2002), No-wait flowshops with bicriteria of makespan and total completion time, *Journal of the Operational Research Society*, 53, 1004-1015.

**Bertolissi, E.** (1999), A simple no-wait flowshop scheduling heuristic for the no-wait flow-shop problem, *Proceedings of the 15<sup>th</sup> International Conference on Computer-Aided Production Engineering, CAPE'99, University of Durham Publishers*, 750-755.

**Bertolissi, E.** (2000), Heuristic algorithm for scheduling in th no-wait flow-shop, *Journal of Materials Processing Technology*, 107, 459-465.

**Bonney, M. C. e Gundry, S. W.** (1976), Solutions to the constrained flowshop sequencing problem, *Operations Research Quarterly*. 24, 869-883.

**Chakravarthy, K e Rajendran, C.** (1999). A heuristic for scheduling in a flowshop with the bicriteria of makespan and maximum tardiness minimization, *Production Planning Control*, 10, 707-714.

**Chen, C., Neppalli, R.V. e Aljaber, N.** (1996), Genetic algorithms applied to the continuous flow shop problem, *Computers Industrial Engineers*, 30, 919-929.

**Demian, J. M. V. e Baker, K. R.** (1974), Minimizing mean flowtime in the flow shop with no intermediate queues, *AIIE Transactions*, 6, 28-34.

**Gangadharan, R. e Rajendran, C.** (1993), Heuristic algorithms for scheduling in the no-wait flowshop. *International Journal of Production Economics*, 32, 285-290.

**Gangadharan, R. e Rajendran, C.** (1994), A simulated annealing heuristic for scheduling in a flowshop with bicriteria, *Computers & Industrial Engineering*, 27, 473-476.

**Hall, N. G. e Sriskandarajah, C.** (1996), A survey of machine Scheduling-problems with blocking and no-wait in process, *Operations Research*, 44, 510-525.

**King, R. e Spachis, A. S.** (1980), Heuristics for flow shop scheduling. *International Journal of Production Research*, 18, 343-357.

**Lawler, E. L.; Lenstra, J. K.; Rinnooy Kan, A. H. G. e Shmoys, D. B.** (1993) *Sequencing and scheduling: algorithms and complexity*. In: Graves, S. C.; Rinnooy Kan, A. H. G.

**Murata, T., Ishibuchi, H., e Tanaka, H.** (1996), Multi-objective genetic algorithms and its applications to flowshop scheduling, *Computers & Industrial Engineering*, 30, 957-968.

**Nagano, M. S. e Moccellini, J. V.** (2007), Reducing mean flow time in permutation flow shop, *Journal of the Operational Research Society*, in press.

**Nagar, A., Haddock, J. e Heragu, S.** (1995a), Multiple and bicriteria scheduling: a literature review, *European Journal of Operational Research*, 81, 88-104.

**Nagar, A., Haddock, J. e Heragu, S.** (1995b), A branch-and-bound approach for a two-machine flowshop scheduling problem, *Journal of the Operational Research Society*, 46, 721-734.

**Nagar, A., Haddock, J. e Heragu, S.** (1996), A combined branch-and-bound and genetic algorithm based approach for a flowshop scheduling problem, *Annal Operations Research*, 46, 397-414.



- Neppalli V. R., Chen, C. L. e Gupta J. N. D.**, (1996), Genetic algorithms for the two-stage bicriteria flowshop problem, *European Journal of Operational Research*, 95, 356-373.
- Papadimitriou, C. e Kanellakis, P. C.** (1980), Flowshop scheduling with limited temporary storage. *Journal of the Association for Computing Machinery*, 27, 533-549.
- Raaymakers, W. e Hoogeveen, J.** (2000), Scheduling multipurpose batch process industries with no-wait restrictions by simulated annealing, *European Journal of Operational Research*, 126, 131-151.
- Rajendran, C.** (1994), A no-wait flowshop scheduling heuristic to minimize makespan, *Journal of the Operational Research Society*, 45, 4, 472-478.
- Rajendran, C. e Chaudhuri, D.** (1990), Heuristic algorithms for continuous flow-shop problem, *Naval Research Logistics*, 37, 695-705.
- Rajendran, C.** (1992), Two-stage flowshop scheduling problem with bicriteria, *Journal of the Operational Research Society*, 43, 871-884.
- Reddi S. S. e Ramamoorthy, C. V.** (1972), On the flowshop sequencing problems with no wait inprocess, *Operational Research Quartely*, 23, 323-331.
- Sayin, S. e Karabati, S.** (1999), A bicriteria approach to the two-machine flow shop scheduling problem. *European Journal of Operational Research*, 113, 435-449.
- Sivrikaya-Serifoglu, F. e Ulusoy, G.** (1998), A bicriteria two-machine permutation flowshop problem, *European Journal of Operational Research*, 107, 414-430.
- Wismer, D. A.** (1972), Solution of the flowshop sequencing problem with no intermediate queues, *Operations Research*, 20, 689-697.
- Yeh, W. C.** (1999), A new branch-and-bound approach for the  $n/2/\text{flowshop}/aF + bC_{max}$  flowshop scheduling problem, *Computers Operations Research*, 26, 1293-1310.
- Yeh, W. C.** (2001), An efficient branch-and-bound algorithm for the two-machine bicriteria flowshop scheduling problem, *Journal Manufacturing Systems*, 20, 113-123.