

PROGRAMAÇÃO DA PRODUÇÃO *FLOW SHOP* PERMUTACIONAL COM MINIMIZAÇÃO DO TEMPO MÉDIO DE FLUXO

Marcelo Seido Nagano

Faculdade de Economia, Administração e Contabilidade, Universidade de São Paulo
Av. dos Bandeirantes, 3900, CEP 14040-901 Ribeirão Preto-SP
drnagano@usp.br

João Vitor Moccellin

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

Luiz Antonio Nogueira Lorena

LAC/INPE
Caixa Postal 515 CEP 12201-970 São José dos Campos-SP
lorena@lac.inpe.br

Resumo

Este artigo trata do problema de Programação de Operações em um ambiente de produção Flow Shop Permutacional, tendo como objetivo minimizar o tempo médio de fluxo das tarefas, reduzindo desta forma o estoque em processamento. Um novo método heurístico é proposto para a solução do problema. Por meio de uma experimentação computacional, o método proposto é avaliado e comparado com um dos melhores métodos heurísticos reportados na literatura. Para o conjunto de problemas tratados, os resultados experimentais mostram a superioridade do novo método em relação à qualidade da solução como também quanto ao esforço computacional.

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

Os problemas de **programação de tarefas** em sistemas de produção são, tradicionalmente, classificados em função do fluxo das operações nas máquinas, conforme segue:

- **Job Shop:** cada tarefa tem sua própria seqüência de processamento no conjunto de máquinas;
- **Flow Shop:** todas as tarefas têm a mesma seqüência de processamento no conjunto de máquinas;
- **Open Shop:** não há uma seqüência específica ou preestabelecida para o processamento das tarefas nas máquinas;
- **Flow Shop Permutacional:** é um *Flow Shop* no qual em cada máquina a seqüência das tarefas é a mesma;
- **Máquina Única:** existe uma única máquina disponível para o processamento das tarefas;
- **Máquinas Paralelas:** são disponíveis diversas máquinas, geralmente idênticas, para as mesmas operações;

- **Job Shop com Máquinas Múltiplas:** é um *Job Shop* no qual existe um conjunto de máquinas paralelas em cada estágio de produção;
- **Flow Shop com Máquinas Múltiplas:** consiste em um *Flow Shop* no qual existe um conjunto de máquinas paralelas em cada estágio de produção.

A Figura 1 ilustra a relação entre as classes dos problemas de programação de operações em máquinas, conforme a classificação dada:

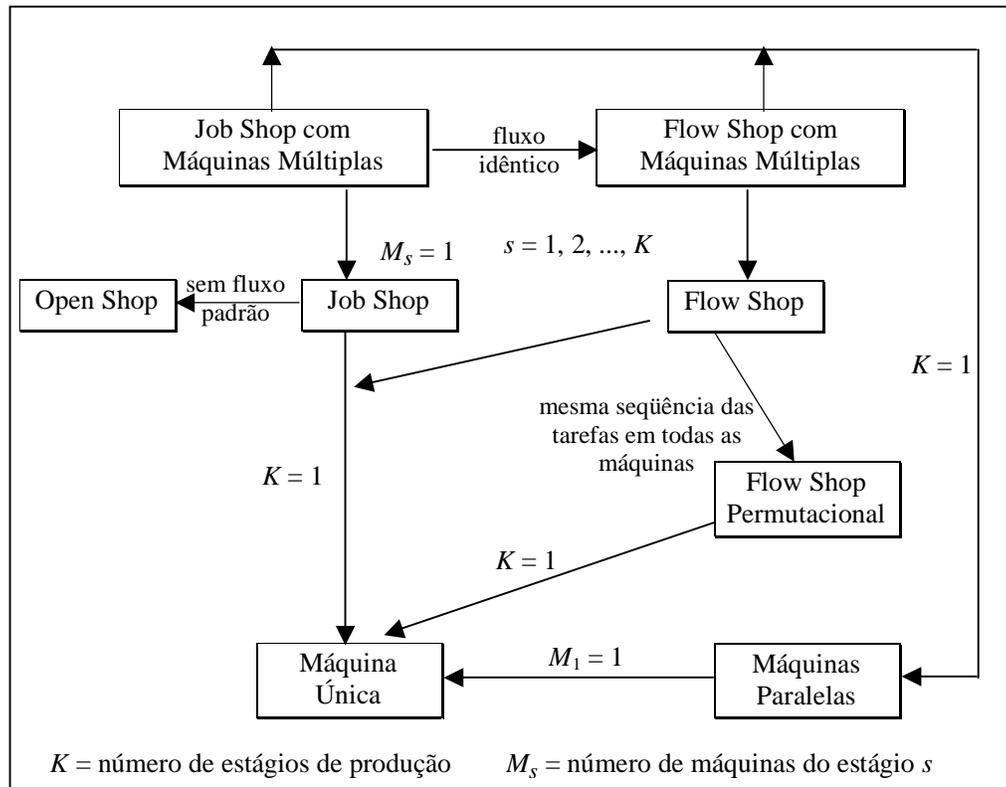


Figura 1 - Relação entre as classes de problemas de programação de operações em máquinas.

Este artigo trata do problema *Flow Shop* Permutacional, que 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.

Mais recentemente, Framinan e Leisten (2003) propuseram o método FL-IH7, inicialmente denominado *IH7-proposed*, 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.

De acordo com a revisão da literatura 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 artigo a ser brevemente publicado, Aldowaisan e Allahverdi (2004) tratam do caso *no-wait* da programação *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 a seqüência solução da 2ª etapa em S . 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 método FL-IH7 desenvolvido por Framinan e Leisten (2003). Na experimentação foi utilizada uma amostra constituída de 5200 problemas-teste, com o número de máquinas $m \in \{5, 10, 15, 20\}$ e o número de tarefas $n \in \{10, 20, 30, 40, 50, 60, 70, 80, 90, 100, 110, 120, 130\}$, correspondendo a 100 problemas para cada combinação ($m \times n$). Os problemas com até 80 tarefas foram os mesmos utilizados por Framinan e Leisten (2003) que mostraram a superioridade do FL-IH7 relativamente aos métodos anteriores, os quais foram gerados por Woo e Yim (1998). Os demais foram obtidos com as mesmas hipóteses dos outros problemas, ou seja, tempos de processamento das operações gerados aleatoriamente a partir de uma distribuição uniforme discreta no intervalo $[1, 99]$. Neste trabalho, os dois métodos heurísticos comparados foram codificados pelos autores em linguagem *Delphi* e processados conjuntamente em um mesmo microcomputador *Pentium III 550 Mhz*.

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

A primeira é definida pelo quociente entre o número 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 um mesmo problema, suas Porcentagens de Sucesso são simultaneamente melhoradas.

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.

A pesquisa relatada neste trabalho, além de buscar melhores métodos para o problema de programação da produção tratado, tem também por objetivo 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. A estrutura de três etapas do método desenvolvido atende a esses objetivos, como poderá ser notado a partir dos resultados da experimentação computacional.

Uma característica do método refere-se à primeira etapa onde as tarefas são ordenadas pela regra STPT (*Shortest Total Processing Time*), que é diretamente relacionada à tradicional ordenação SPT (*Shortest Processing Time*), que leva a soluções ótimas quanto ao tempo médio de fluxo para problemas de Máquina Única e Máquinas Paralelas.

Com o objetivo de avaliar a importância dessa ordenação inicial, todos os problemas-teste foram simultaneamente solucionados pelo SPT-flowtime e pelo método denominado *RANDOM-flowtime* cuja diferença com o primeiro ocorre somente na primeira etapa do algoritmo, no qual a ordenação inicial das tarefas é aleatória. A Figura 2 apresenta as Porcentagens de Sucesso dos dois métodos, agrupadas em função do número de tarefas.

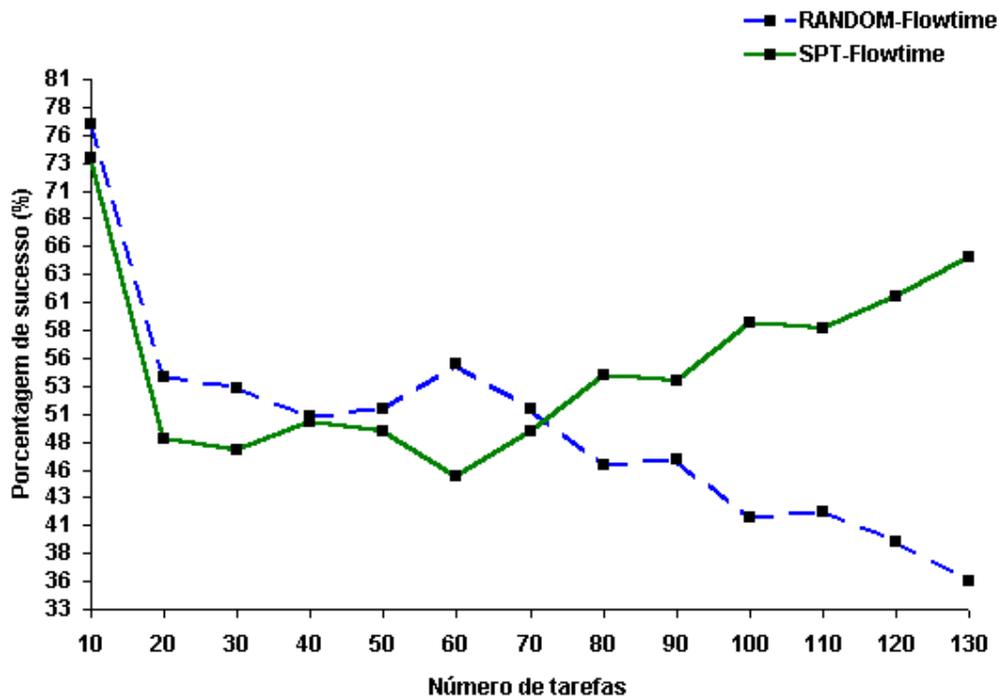


Figura 2 - Comparação de desempenho entre SPT-flowtime e RANDOM-flowtime.

Os resultados mostrados na Figura 2 indicam que:

- ✓ Para os menores problemas (10 tarefas) ocorre uma alta porcentagem de “empates” na busca pelo menor tempo médio de fluxo, porém a diferença entre as Porcentagens de Sucesso não é significativa;
- ✓ Para os problemas de 20 a 90 tarefas, os empates praticamente não mais ocorrem (quando a soma das Porcentagens de Sucesso é igual a 100%), porém as diferenças de desempenho continuam não significativas;

- ✓ A partir de 60 tarefas ocorre uma tendência consistente de incremento na Porcentagem de Sucesso do SPT-*flowtime* e de decréscimo para o *RANDOM-flowtime*. A partir de 100 tarefas as diferenças entre tais porcentagens tornam-se significativas.

Assim, pode-se concluir que a ordenação inicial pela regra STPT tem influência relevante na qualidade da solução obtida somente para os problemas de grande porte (a partir de 100 tarefas). Uma possível causa está relacionada aos “movimentos dentro do espaço de soluções” efetuados pelas etapas 2 e 3 do algoritmo proposto. Para problemas de pequeno e médio porte tais movimentos têm uma maior capacidade de busca de melhores soluções, independentemente de suas “posições” iniciais. Porém, para problemas maiores o posicionamento no espaço de soluções obtido pela ordenação inicial torna-se relevante.

A seguir, nas Tabelas 1 e 2, são apresentados os resultados da comparação entre o SPT-*flowtime* e o FL-IH7 de Framinan e Leisten (2003).

Tabela 1 - Resultados experimentais para os problemas de pequeno e médio porte

número de tarefas n	número de máquinas m	FL-IH7			SPT- <i>flowtime</i>		
		Desvio Médio Relativo (%)	Porcentagem de Sucesso	Tempo de Computação (segundos)	Desvio Médio Relativo (%)	Porcentagem de Sucesso	Tempo de Computação (segundos)
10	5	0,1932	72	0,01	0,2631	80	0,00
	10	0,1777	75	0,01	0,1416	75	0,00
	15	0,0943	76	0,04	0,1172	70	0,00
	20	0,0876	75	0,04	0,1375	74	0,01
	valores médios	0,1382	74,5	0,02	0,1648	74,7	0,002
20	5	0,2968	50	0,18	0,3212	52	0,02
	10	0,4992	40	0,30	0,2267	62	0,03
	15	0,4114	40	0,40	0,2777	61	0,05
	20	0,4042	34	0,49	0,1527	69	0,06
	valores médios	0,4028	41,0	0,34	0,2445	61,0	0,04
30	5	0,4603	34	0,80	0,2671	66	0,08
	10	0,5171	38	1,33	0,2143	62	0,16
	15	0,4571	35	1,69	0,2344	66	0,24
	20	0,4000	42	2,16	0,2095	58	0,31
	valores médios	0,4586	37,3	1,49	0,2313	63,0	0,19
40	5	0,4003	43	2,49	0,2211	57	0,23
	10	0,4913	41	3,92	0,2488	59	0,49
	15	0,6240	30	4,99	0,2190	70	0,74
	20	0,5724	31	6,30	0,1386	69	0,97
	valores médios	0,5219	36,3	4,42	0,2068	63,7	0,60
50	5	0,3728	39	5,81	0,2207	61	0,54
	10	0,5537	39	9,25	0,2603	62	1,15
	15	0,4130	39	11,92	0,2185	61	1,79
	20	0,5570	34	15,03	0,2659	66	2,34
	valores médios	0,4741	37,7	10,50	0,2413	62,5	1,45
60	5	0,2828	49	11,72	0,2370	51	1,11
	10	0,4593	37	20,06	0,2436	63	2,34
	15	0,6827	28	24,63	0,1020	72	3,57
	20	0,5259	27	29,36	0,1196	73	4,82
	valores médios	0,4876	35,3	21,44	0,1755	64,7	2,95
70	5	0,3685	34	21,49	0,1639	66	1,99
	10	0,4911	31	33,87	0,1497	69	4,24
	15	0,5268	36	46,13	0,2058	64	6,55
	20	0,5003	37	54,20	0,1751	63	8,84
	valores médios	0,4716	34,5	38,92	0,1736	65,5	5,40

Tabela 2 - Resultados experimentais para os problemas de grande porte

número de tarefas n	número de máquinas m	FL-IH7		SPT-flowtime			
		Desvio Médio Relativo (%)	Porcentagem de Sucesso	Tempo de Computação (segundos)	Desvio Médio Relativo (%)	Porcentagem de Sucesso	Tempo de Computação (segundos)
80	5	0,3170	36	35,45	0,1417	64	3,32
	10	0,4534	32	56,45	0,1567	68	7,01
	15	0,6162	32	74,19	0,1701	68	11,01
	20	0,6220	21	96,13	0,1067	79	14,88
valores médios		0,5021	30,3	65,55	0,1438	69,7	9,05
90	5	0,3160	37	49,85	0,1544	63	5,01
	10	0,4645	33	77,37	0,1602	67	8,69
	15	0,5457	27	106,71	0,1284	73	20,85
	20	0,5526	31	159,51	0,1379	69	25,45
valores médios		0,4697	32,0	98,36	0,1452	68,0	15,00
100	5	0,2524	44	67,21	0,1403	56	7,65
	10	0,4861	33	114,81	0,1476	67	17,71
	15	0,6999	23	143,18	0,1028	77	30,11
	20	0,4606	27	232,81	0,1183	73	39,32
valores médios		0,4747	31,8	139,50	0,1272	68,2	23,70
110	5	0,2433	42	88,69	0,1181	58	11,71
	10	0,4703	27	153,79	0,1229	73	29,18
	15	0,4874	28	185,73	0,1064	72	44,01
	20	0,4978	33	310,59	0,1508	67	55,48
valores médios		0,4247	32,5	184,70	0,1245	67,5	35,10
120	5	0,2565	33	115,79	0,0980	67	16,86
	10	0,4633	31	202,22	0,1270	69	44,22
	15	0,3817	33	233,18	0,1686	67	58,21
	20	0,5107	21	400,01	0,0844	79	80,71
valores médios		0,4030	29,5	237,80	0,1194	70,5	50,00
130	5	0,2320	40	156,21	0,0992	60	25,19
	10	0,4744	33	260,58	0,1344	67	62,01
	15	0,4349	29	290,31	0,1295	71	79,69
	20	0,4760	30	492,89	0,1382	70	111,09
valores médios		0,4043	33,0	300,00	0,1253	67,0	69,50

Os resultados experimentais mostram claramente a superioridade do novo método em relação ao FL-IH7 tanto em termos da qualidade da solução quanto do tempo de computação.

Somente para os menores problemas (10 tarefas) ocorre um melhor desempenho do FL-IH7 quanto ao Desvio Médio Relativo, porém não significativo. As Porcentagens Médias de Sucesso são equivalentes e indicam um número grande de empates quanto ao menor tempo médio de fluxo.

Pode-se notar, também, nas demais classes de problemas em função do número de tarefas, que a diferença de desempenho é mantida, independentemente do número de máquinas. Uma análise mais detalhada mostra que em várias classes de problemas, agrupadas em função do número de tarefas, o SPT-flowtime apresenta uma tendência de melhor desempenho para problemas com um número crescente de máquinas.

Assim, pode-se observar que os valores médios das estatísticas usadas para avaliar o desempenho dos métodos, referentes aos problemas agrupados de acordo com o número de tarefas, são suficientemente representativos para a comparação dos dois métodos heurísticos. Desta forma, com o objetivo de uma visão global dos resultados, tais valores médios são apresentados nas Figuras 3, 4 e 5.

A Figura 3 mostra, inicialmente, como já mencionado, que os métodos têm praticamente o

mesmo desempenho para os problemas com 10 tarefas, apresentando muitos empates quanto à função-objetivo. Na faixa de 20 a 80 tarefas os empates raramente ocorrem e o desempenho do SPT-*flowtime* é significativamente superior e com tendência de crescimento. A partir de 80 tarefas ocorre uma relativa estabilidade em torno de 68% de sucesso para o SPT-*flowtime* e 32% para o FL-IH7.

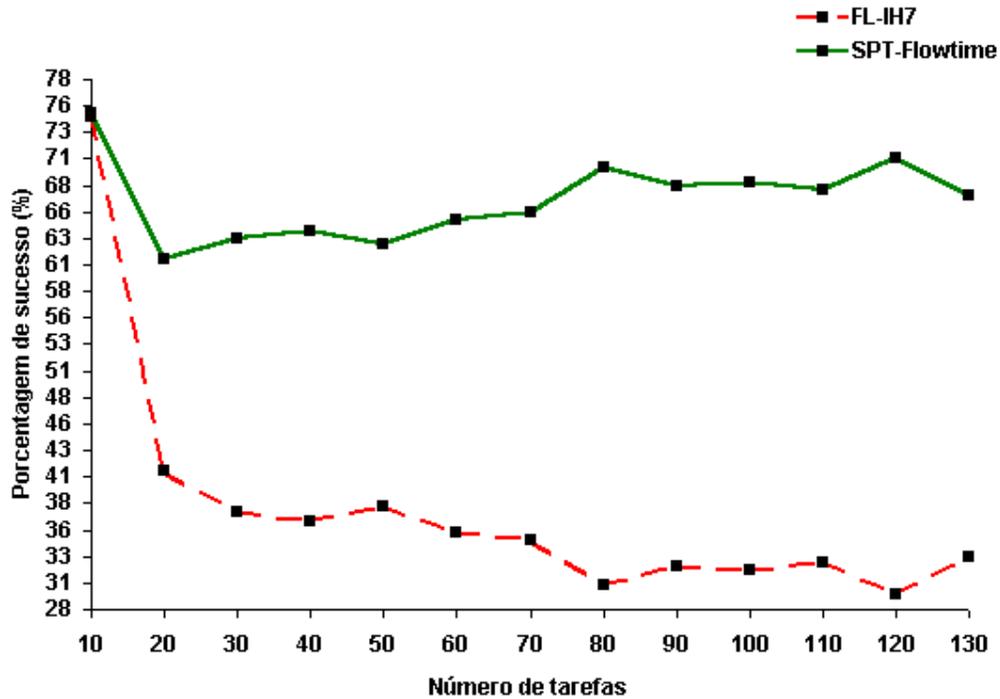


Figura 3 - Percentagens de Sucesso em função do número de tarefas.

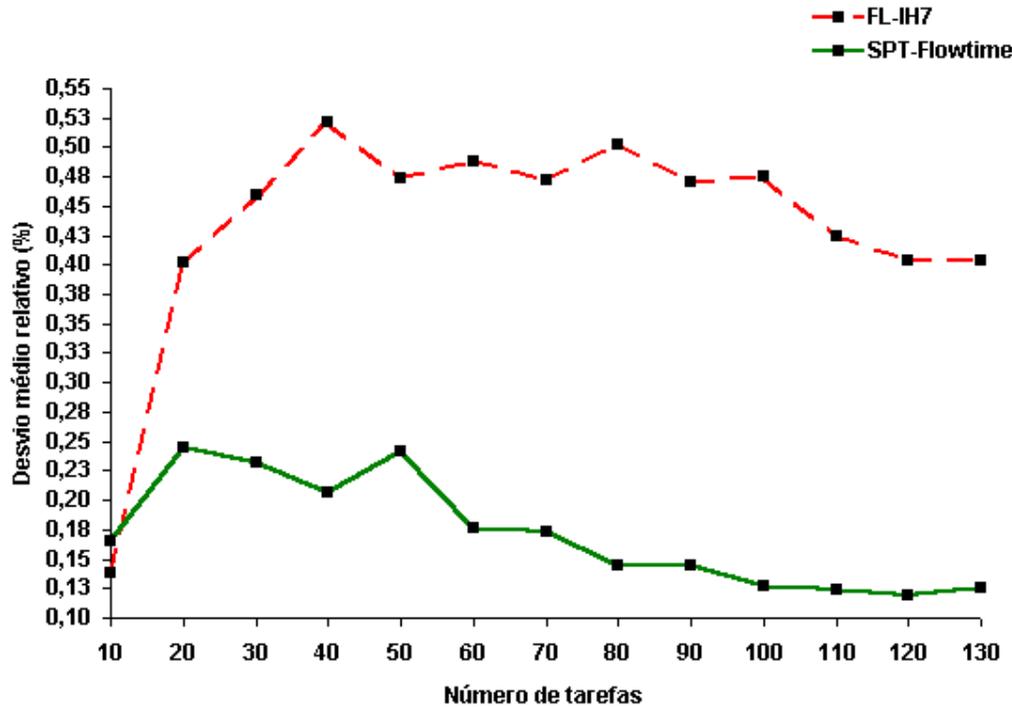


Figura 4 - Desvios Médios Relativos em função do número de tarefas.

A Figura 4 ilustra o comportamento geral dos Desvios Médios Relativos. Note-se que quanto menor o valor do Desvio Relativo, melhor o desempenho do método. Deve-se ressaltar que os valores dos desvios relativos são pequenos uma vez que o método FL-IH7 comprovadamente obtém soluções de alta qualidade. Pode-se observar que para os problemas com $n \geq 60$ tarefas as diferenças entre os Desvios Relativos obtidos pelos métodos são próximas entre si, em torno de uma média de 0,31%, valor este maior que o pior Desvio Médio Relativo do SPT-*flowtime*.

A Figura 5 mostra 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.

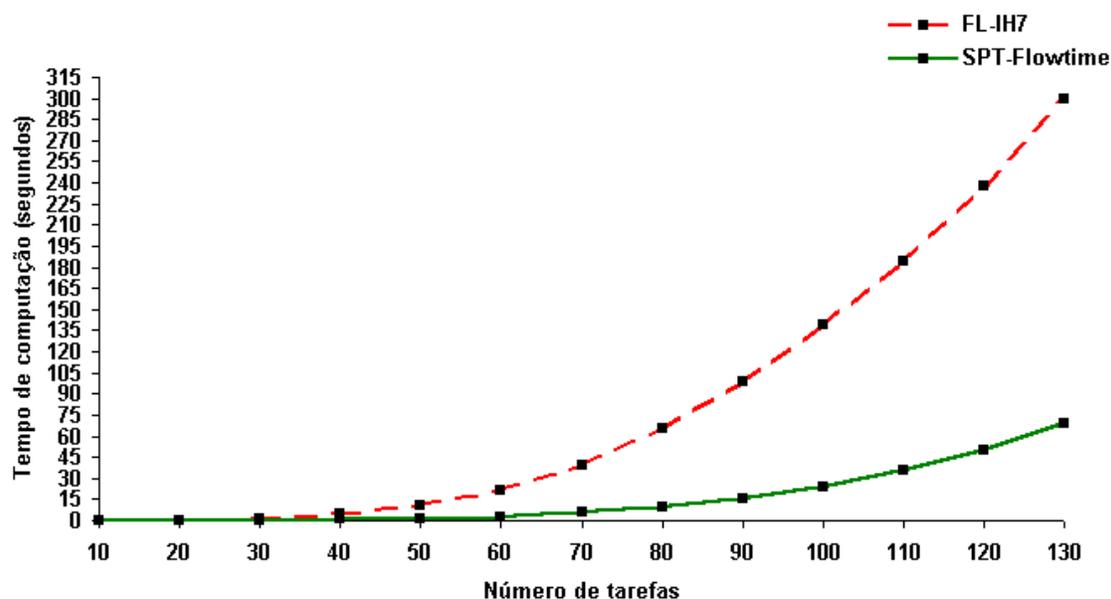


Figura 5 - Tempos de computação em função do número de tarefas.

5. Considerações Finais

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. Para fins práticos tal problema pode ser considerado já resolvido. Por outro lado, tendo em vista sua complexidade (*NP-hard*), a busca de métodos cada vez mais eficazes quanto à solução ótima 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 artigo foi motivada pelas considerações acima e, reiterando, teve também por objetivo 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.

Os resultados experimentais mostraram que o método heurístico proposto (SPT-*flowtime*) tem um desempenho superior em comparação com o método FL-IH7 desenvolvido por Framinan e Leisten (2003) para solução do problema de programação da produção em ambientes *flow shop*, tendo como medida de desempenho a minimização do tempo médio de fluxo. Além de obter soluções de melhor qualidade, o esforço computacional é bem menor.

Referências

- AHMADI, R.H.; BAGCHI, U. (1990) – Improved Lower Bounds for Minimizing the Sum of Completion Times of n Jobs over m Machines. *European Journal of Operational Research* Vol. 44, p.331-336.
- ALDOWAISAN, T. & ALLAHVERDI, A. (2004) – New Heuristics for m-machine no-wait Flowshop to Minimize Total Completion Time. *OMEGA*, article in press.
- ALLAHVERDI, A. & ALDOWAISAN, T. (2002) – New Heuristics to Minimize Total Completion Time in m-machine Flowshops. *International Journal of Production Economics* Vol. 77, p.71-83.
- CAMPBELL, H.G.; DUDEK, R.A. & SMITH, M.L. (1970) – A Heuristic Algorithm for n-job, m-machine Sequencing Problem. *Management Science* Vol. 16, p.630-637.
- FRAMINAN, J.M. & LEISTEN, R. (2003) – An Efficient Constructive Heuristic for Flowtime Minimisation in Permutation Flow Shop. *OMEGA* Vol. 31, p.311-317.
- GAREY, M.R.; JOHNSON, D.S. & SETHI, R. (1976) – The Complexity of Flowshop and Jobshop Scheduling. *Mathematics of Operations Research* Vol. 1, p.117-129.
- GUPTA, J.N.D. (1972) – Heuristic Algorithms for Multistage Flowshop Scheduling Problem. *AIIE Transactions* Vol. 4, p.11-18.
- HO, J.C. (1995) – Flowshop Sequencing with Mean Flowtime Objective. *European Journal of Operational Research* Vol. 81, p.571-578.
- MIYAZAKI, S.; NISHIYAMA, N. & HASHIMOTO, F. (1978) - An Adjacent Pairwise Approach to the Mean Flowtime Scheduling Problem. *Journal of the Operations Research Society of Japan* Vol. 21, p.287-299.
- NAWAZ, M.; ENSCORE, E.E. & HAM, I. (1983) - A Heuristic Algorithm for the m-machine, n-job Flow-shop Sequencing Problem. *OMEGA* Vol. 11, p.91-95.
- RAJENDRAN, C. & CHAUDHURI, D. (1991) – An Efficient Heuristic Approach to the Scheduling of Jobs in a Flowshop. *European Journal of Operational Research* Vol. 61, p.318-325.
- RAJENDRAN, C. & ZIEGLER, H. (1997) – An efficient Heuristic for Scheduling in a Flowshop to Minimize Total Weighted Flowtime of Jobs. *European Journal of Operational Research* Vol. 103, p.129-138.
- RAJENDRAN, C. (1993) – Heuristic Algorithm for Scheduling in a Flowshop to Minimise Total Flowtime. *International Journal Production Economics* Vol. 29, p.65-73.
- RINNOOY KAN, A.H.G. (1976) - Machine Scheduling Problems: Classification, Complexity, and Computations. *The Hague: Nijhoff*.
- WANG, C.; CHU, C. & PROTH, J.M. (1997) – heuristic Approaches for n/m/F/ ΣC_1 Scheduling Problems. *European Journal of Operational Research* Vol. 96, p.636-644.
- WOO, D.S. & YIM H.S. (1998) – A heuristic Algorithm for Mean Flowtime Objective in Flowshop Scheduling. *Computers & Operations Research* Vol. 25, p.175-182.

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