

# Soluções de alto desempenho para a programação da produção *flow shop*

Marcelo Seido Nagano (EESC USP, SP, Brasil) – drnagano@usp.br  
• Av. Trabalhador São-carlense, 400, Centro, CEP: 13566-590, São Carlos-SP  
Fábio José Ceron Branco (EESC USP, SP, Brasil) – fbranco@sc.usp.br  
João Vítor Moccellini (EESC USP, SP, Brasil) – jvmoccel@sc.usp.br

Recebido em: 03/03/08 Aprovado em: 23/05/08

## **Resumo**

*Este trabalho é direcionado para o problema de programação de operações flow shop permutacional. Um novo método heurístico construtivo, com o objetivo de minimizar a Duração Total da Programação (makespan) é apresentado e comparado com os melhores métodos heurísticos construtivos reportados na literatura, conhecidos por NEH e NEHKK. Os resultados da experimentação computacional mostraram que o novo método heurístico obtém soluções de alta qualidade, em comparação com os métodos avaliados para o conjunto de problemas teste de Taillard.*

**Palavras-chave:** Programação da produção; flow-shop permutacional; métodos heurísticos.

## **Abstract**

*In this article we consider the permutational flow shop scheduling problem. A new constructive heuristic method is presented with the objective of minimizing makespan and it is compared with the best constructive heuristic methods reported in literature (NEH and NEHKK). Experiment results show that the new heuristic method provides high quality solutions in comparison with the evaluated methods for the set of Taillard test problems.*

**Keywords:** Scheduling, permutation flow shop, heuristic method.

## 1. INTRODUÇÃO

O problema de programação de operações *flow shop* é um problema da produção, no qual  $n$  tarefas devem ser processadas, na mesma sequência, em cada máquina de um conjunto de  $m$  máquinas distintas. Um caso específico de programação *flow shop*, denominado permutacional, é quando em cada máquina mantém-se a mesma ordem de processamento das tarefas. A solução do problema consiste em determinar dentre as  $(n!)$  sequências possíveis das tarefas, aquela que otimize uma determinada medida de desempenho, sendo uma das mais utilizadas, a duração total da programação (*makespan*). Na teoria que estuda a complexidade dos problemas de natureza combinatorial, o problema em questão é classificado como *NP-hard*, de forma que pode ser resolvido, eficientemente, de maneira ótima (solução exata), somente em casos de pequeno porte. Assim, muitos métodos heurísticos têm sido propostos para a solução do problema.

Para a modelagem deste tipo de problema, é necessário assumir algumas hipóteses:

- Cada máquina está disponível continuamente, sem interrupções;
- Cada operação tem, no máximo, uma sucessora e uma precedente (fluxo unidirecional);
- Cada operação pode ser executada por apenas uma máquina;
- Sempre que uma operação é iniciada, é finalizada, sem que haja interrupção no seu processamento;
- Os tempos de preparação das máquinas (*setup*) estão inclusos nos tempos de processamento e estes independem das tarefas precedentes e;
- Cada máquina processa apenas uma tarefa de cada vez e cada tarefa é processada por apenas uma máquina de cada vez.

Os métodos heurísticos podem ser classificados de diversas maneiras. Uma delas, classifica-os em Construtivos e outrovem Melhorativos, dependendo da forma de obtenção da solução. Os métodos construtivos se caracterizam pelo fato de gerarem apenas uma solução, que será a solução final do problema (PALMER, 1965; CAMPBELL et al., 1970; GUPTA, 1971; DANNENBRING, 1977; NAWAZ et al., 1983; KOULAMAS, 1998; DAVOUD POUR, 2001; NAGANO & MOCCELLIN, 2002; KALCZYNSKI & KAMBUROWSKI, 2007). Esta solução pode ser gerada:

- Diretamente, a partir da ordenação das tarefas, segundo índices de prioridade calculados em função dos tempos de processamento das tarefas, como por exemplo: Palmer (1965) e Koulamas (1998);
- Escolhendo-se a melhor sequência das tarefas, a partir de um conjunto de sequências também obtidas, utilizando-se índices de prioridade associados às tarefas: Campbell et al. (1970) e Hundal e Rajgopal (1988);
- Ou ainda, a partir da geração sucessiva de sequências parciais das tarefas (subsequências) até a obtenção de uma sequência completa, através de algum critério de inserção de tarefas, como por exemplo: Nawaz et al. (1983), Nagano e Moccellin (2002), Kalczyński e Kamburowski (2007).

No caso dos métodos melhorativos, obtém-se uma solução inicial e, posteriormente, através de algum procedimento iterativo (geralmente envolvendo trocas de posições das tarefas na sequência), busca-se obter uma sequência das tarefas melhor que a atual, quanto à medida de desempenho adotada, por exemplo: Osman e Potts (1989), Ogbu e Smith (1990), Widmer e Hertz (1989), Taillard (1990), Reeves (1993), Nowicki e Smutnicki (1996) e Grabowski e Wodecki (2004), que aplicaram a técnica de busca tabu (BT). Outros

métodos são o *iterated local search* (ILS) de Stützle (1998) e, mais recentemente, dois novos métodos baseados no *ant-colony optimization* (ACO), de Rajendran e Ziegler (2004) e o método baseado no procedimento *scatter search*, de Noorul Haq et al. (2007).

Dentre os métodos construtivos mais conhecidos, salienta-se o NEH, proposto por Nawaz et al. (1983), que tem sido considerado, até hoje, como o melhor método heurístico construtivo, para minimizar o *makespan* em *flow shop* permutacionais, com alta qualidade de solução e eficiência computacional. O Algoritmo NEH é bastante simples, podendo ser resumido como segue:

- Fase 1 – Ordenação inicial das tarefas, segundo os respectivos tempos totais de processamento (regra LPT – *Longest Processing Time*).
- Fase 2 – Construção da sequência, solução a partir da inserção gradativa das tarefas em sequências parciais, conforme a ordenação do Fase 1.

Nagano e Moccellini (2002) propuseram uma heurística construtiva, denominada N&M, onde há uma mudança apenas na fase de ordenação inicial, em relação ao NEH, sendo a ordenação inicial feita em dois passos:

*Passo 1* – Para cada tarefa calcular:

$$I_v = \sum TP_v - \max \sum LBY_{J_v} \quad (1)$$

onde,

$$\max \sum LBY_{J_v} = \max_{\substack{u=1 \\ u \neq v}}^n \left\{ \sum_{k=1}^{m-1} LBY_{uv}^k \right\} \quad (2)$$

$$\sum TP_v = \sum_{k=1}^m P_{kv} \quad (3)$$

$p_{ki}$  = tempo de processamento da tarefa  $i$  na máquina  $k$ ;

$n$  = número de tarefas;

$m$  = número de máquinas.

$$LBY_{uv}^k = \max (0, (p_{k+1,u} - p_{k,v}) - UBX_{uv}^k) \quad (4)$$

$$UBX_{uv}^k = \max (0, UBX_{uv}^{k-1} + (p_{k-1,v} - p_{k,u})), \quad (5)$$

sendo

$$UBX_{uv}^1 = 0 \quad (6)$$

A tarefa  $u$  é imediatamente precedente da tarefa  $v$ .

*Passo 2* – Ordenar todos os  $I_i$  de forma decrescente.

A próxima fase é idêntica ao NEH. Este método apresentou resultados superiores, com relação ao NEH (Nawaz et al., 1983).

Mais recentemente, Kalezynski e Kamburowski (2007) propuseram um novo método heurístico construtivo, denominado NEHKK. O método proposto apresentou uma modificação do método NEH, no processo de construção da sequência solução, onde em caso de empate das soluções parciais, nas subsequências obtidas, um procedimento heurístico baseada na regra de JOHNSON (Johnson, 1954), é aplicado. Os resultados obtidos, através da experimentação computacional, verificaram o melhor desempenho, quando comparados com o N&M (Nagano e Moccellin, 2002) e o NEH (Nawaz et al., 1983).

Neste trabalho, reportam-se sucintamente os resultados de uma pesquisa, com o objetivo de desenvolver um novo método heurístico construtivo, com desempenho superior ao NEH e NEHKK. Tal método é baseado no procedimento de construção da sequência solução, utilizando-se de forma combinada as vizinhanças de busca de inserção e permutação de tarefas. O novo método proposto será avaliado, verificando 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 neste trabalho apresenta a seguinte estrutura, sendo descrita a seguir.

*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-crescentes das somas dos tempos de processamento;

*Passo 3* – Selecione as duas primeiras tarefas da ordenação obtida no *Passo 2*, sequenciando-as de maneira a minimizar o *makespan*, considerando-se somente essas duas tarefas e que ocuparão as duas primeiras posições da sequência  $S$ .

*Passo 4* – Para  $k = 3$  a  $n$ ,

- Selecione a tarefa que ocupa a  $k$ -ésima posição na ordenação obtida no *Passo 2*;
- Acrescente a tarefa na  $k$ -ésima posição da sequência atual  $S$  e calcule seu respectivo *makespan* para a sequência parcial;
- Considerando toda a Vizinhança de Inserção da sequência parcial com  $k$  tarefas, constituída de  $(k-1)^2$  sequências, determine a sequência  $S_i$  associada ao menor *makespan*;
- Se o *makespan* de  $S_i$  é menor que de  $S$ , atualize  $S$  com  $S_i$  e, considerando toda a Vizinhança de Permutação da sequência parcial com  $k$  tarefas, constituída de  $k(k-1)/2$  sequências, determine a sequência  $S_p$  associada ao menor *makespan*;

- Se o *makespan* de  $S_p$  é menor que de  $S$ , atualize  $S$  com  $S_p$ ;
- Se o *makespan* de  $S_i$  não é menor que de  $S$ , considerando toda a Vizinhança de Permutação da sequência parcial com  $k$  tarefas, constituída de  $k(k-1)/2$  sequências, determine a sequência  $S_p$  associada ao menor *makespan*;
- Se o *makespan* de  $S_p$  é menor que de  $S$ , atualize  $S$  com  $S_p$ .

### 3. EXPERIMENTAÇÃO COMPUTACIONAL E MÉTODO DE ANÁLISE

Para a avaliação do desempenho do método proposto, denominado NBM-FS, foram realizadas experimentações computacionais, utilizando os problemas de Taillard (1993) (<http://ina2.eivd.ch/collaborateurs/etd/problemes.dir/ordonnancement.dir/ordonnancement.html>).

Neste trabalho, o NBM-FS foi codificado em linguagem DELPHI e processado em um microcomputador Pentium IV de 3.4 GHz, com 1 GByte de RAM.

As estatísticas usadas para avaliar o desempenho do método, foram a Porcentagem de Sucesso e o Desvio Relativo Médio. A primeira, é definida pelo quociente entre o número total de problemas, para os quais o método obteve a melhor duração total da programação (*makespan*), e o número total de problemas resolvidos. A segunda, quantifica o desvio relativo ( $DR_h$ ), que o método  $h$  obtém em relação a melhor duração total da programação, obtido para um mesmo problema, sendo calculado conforme segue:

$$DR_h = \frac{(M_h - M_*)}{M_*} \quad (7)$$

onde,

$M_h$  = duração total da programação, obtido pelo método  $h$ ;

$M_*$  = melhor duração total da programação, obtido pelo(s) método(s), para um determinado problema.

### 4. ANÁLISE DOS RESULTADOS

Conforme apresentados anteriormente, os resultados obtidos pelo método NBM-FS foram comparados com os métodos heurísticos NEH (Nawaz et al., 1983) e NEHKK (Kalezynski e Kamburowski, 2007). Os resultados principais são apresentados nas tabelas 1, 2, 3 e 4 e na figura 1.

A tabela 1 apresenta a porcentagem de sucesso e o desvio relativo médio entre os métodos avaliados. Verifica-se a superioridade do método NBM-FS para o conjunto de problemas avaliados em relação às estatísticas utilizadas. Para tarefas iguais e superiores a 50, o método NBM-FS apresentou 100% de porcentagem de sucesso e porcentagem de desvio relativo médio igual a zero.

Tabela 1 – Porcentagem de Sucesso e Porcentagem de Desvio Relativo Médio para os métodos NEH, NEHKK e NBM-FS.

n	m	Porcentagem de Sucesso (%)			Desvio Relativo Médio (%)		
		NEH	NEHKK	NBM-FS	NEH	NEHKK	NBM-FS
20	5	20	50	70	3,229	1,192	1,240
	10	0	0	100	2,374	2,220	0,000
	20	0	10	90	1,780	1,626	0,306
<b>Média</b>		<b>6,67</b>	<b>20,00</b>	<b>86,67</b>	<b>2,461</b>	<b>1,679</b>	<b>0,515</b>
50	5	10	10	100	0,538	0,491	0,000
	10	0	0	100	2,349	2,285	0,000
	20	0	0	100	2,454	2,170	0,000
<b>Média</b>		<b>3,33</b>	<b>3,33</b>	<b>100,00</b>	<b>1,780</b>	<b>1,649</b>	<b>0,000</b>
100	5	20	30	100	0,421	0,319	0,000
	10	0	0	100	1,215	0,975	0,000
	20	0	0	100	2,052	1,885	0,000
<b>Média</b>		<b>6,67</b>	<b>10,00</b>	<b>100,00</b>	<b>1,229</b>	<b>1,060</b>	<b>0,000</b>
200	10	10	10	100	0,760	0,714	0,000
	20	0	0	100	2,069	1,939	0,000
<b>Média</b>		<b>5,00</b>	<b>5,00</b>	<b>100,00</b>	<b>1,414</b>	<b>1,327</b>	<b>0,000</b>
500	20	0	0	100	1,173	1,181	0,000
<b>Média</b>		<b>0,00</b>	<b>0,00</b>	<b>100,00</b>	<b>1,173</b>	<b>1,181</b>	<b>0,000</b>

A tabela 2 apresenta o desvio relativo médio por classes de problemas para os métodos NEH, NEHKK e NBM-FS, em relação às melhores soluções obtidas para os problemas testes de Taillard. Pode-se verificar que o método NBM-FS apresentou desvio relativo médio inferior aos métodos NEH e NEHKK, para tarefas superiores a 20, consubstanciando a superioridade do método proposto, já mostrado pela tabela anterior.

Tabela 2 – Desvio Relativo Médio dos métodos NEH, NEHKK e NBM-FS, em relação às melhores soluções para os problemas testes de Taillard.

n	m	NEH	NEHKK	NBM-FS
20	5	4,564	2,485	2,543
	10	4,601	4,450	2,182
	20	3,731	3,576	2,231
<b>Média</b>		<b>4,299</b>	<b>3,503</b>	<b>2,319</b>
50	5	0,727	0,680	0,189
	10	5,073	5,006	2,666
	20	6,660	6,361	4,105
<b>Média</b>		<b>4,153</b>	<b>4,016</b>	<b>2,320</b>
100	5	0,527	0,425	0,106
	10	2,215	1,972	0,988
	20	5,345	5,172	3,227
<b>Média</b>		<b>2,696</b>	<b>2,523</b>	<b>1,440</b>
200	10	1,258	1,212	0,495
	20	4,435	4,303	2,320
<b>Média</b>		<b>2,847</b>	<b>2,758</b>	<b>1,408</b>
500	20	2,066	2,074	0,883
<b>Média</b>		<b>2,066</b>	<b>2,074</b>	<b>0,883</b>

Conforme observado na Tabela 3, o método NBM-FS apresentou soluções iguais para os problemas de Taillard, para 8 problemas testes (Ta008, Ta031, Ta033, Ta038, Ta061, Ta063, Ta067 e Ta069). Estes resultados comprovam que o método NBM-FS apresentou soluções de alta qualidade, já que os melhores resultados para os problemas testes de Taillard foram obtidos por métodos melhorativos (meta-heurísticos).

Tabela 3 – Comparação dos métodos em relação às melhores soluções para os problemas teste de Taillard.

Problema	Melhor solução	NEH	NEHKK	NBM-FS
Ta001	1278	<b>1286</b>	<b>1286</b>	1297
Ta002	1359	<b>1365</b>	<b>1365</b>	1370
Ta003	1081	1159	<b>1132</b>	<b>1132</b>
Ta004	1293	1325	1325	<b>1301</b>
Ta005	1235	1305	1305	<b>1250</b>
Ta006	1195	1228	1225	<b>1198</b>
Ta007	1234	1278	<b>1251</b>	<b>1251</b>
Ta008	1206	1223	1227	<b>1206*</b>
Ta009	1230	1291	1274	<b>1253</b>
Ta010	1108	1291	<b>1127</b>	1253
Ta011	1582	1680	1654	<b>1598</b>
Ta012	1659	1729	1776	<b>1717</b>
Ta013	1496	1557	1538	<b>1532</b>
Ta014	1377	1439	1433	<b>1406</b>
Ta015	1419	1502	1502	<b>1448</b>
Ta016	1397	1453	1433	<b>1427</b>
Ta017	1484	1562	1555	<b>1517</b>
Ta018	1538	1609	1620	<b>1583</b>
Ta019	1593	1647	1647	<b>1617</b>
Ta020	1591	1653	1656	<b>1622</b>
Ta021	2297	2410	2443	<b>2344</b>
Ta022	2099	2150	2134	<b>2133</b>
Ta023	2326	2411	2418	<b>2379</b>
Ta024	2223	2262	2263	<b>2252</b>
Ta025	2291	2397	2384	<b>2328</b>
Ta026	2226	2349	2349	<b>2253</b>
Ta027	2273	2362	2383	<b>2336</b>
Ta028	2200	2249	2249	<b>2242</b>
Ta029	2237	2320	2313	<b>2293</b>
Ta030	2178	2277	<b>2220</b>	2288
Ta031	2724	2733	2732	<b>2724*</b>
Ta032	2834	2843	2843	<b>2838</b>
Ta033	2621	2640	2650	<b>2621*</b>
Ta034	2751	<b>2782</b>	<b>2782</b>	<b>2782</b>
Ta035	2863	2868	2871	<b>2864</b>
Ta036	2829	2850	2845	<b>2835</b>
Ta037	2725	2758	2735	<b>2732</b>
Ta038	2683	2721	2695	<b>2683*</b>
Ta039	2552	2576	2607	<b>2554</b>
Ta040	2782	2790	2786	<b>2783</b>
Ta041	2991	3135	3149	<b>3126</b>
Ta042	2867	3032	3059	<b>2944</b>
Ta043	2839	2986	3000	<b>2924</b>
Ta044	3063	3198	3196	<b>3121</b>
Ta045	2976	3160	3160	<b>3094</b>
Ta046	3006	3178	3158	<b>3100</b>
Ta047	3093	3277	3252	<b>3156</b>
Ta048	3037	3123	3137	<b>3042</b>
Ta049	2897	3002	3025	<b>2964</b>
Ta050	3065	3257	3187	<b>3156</b>
Ta051	3850	4082	4093	<b>3986</b>
Ta052	3704	3921	3938	<b>3847</b>
Ta053	3640	3927	3868	<b>3795</b>
Ta054	3720	3969	3913	<b>3895</b>
Ta055	3610	3835	3874	<b>3732</b>
Ta056	3681	3914	3878	<b>3826</b>
Ta057	3704	3952	3959	<b>3875</b>
Ta058	3691	3938	3969	<b>3849</b>
Ta059	3743	3952	3951	<b>3902</b>
Ta060	3756	4079	4015	<b>3915</b>

Problema	Melhor solução	NEH	NEHKK	NBM-FS
Ta061	5493	5519	5514	5493*
Ta062	5268	5348	5284	5284
Ta063	5175	5219	5196	5175*
Ta064	5014	5023	5023	5023
Ta065	5250	5266	5266	5265
Ta066	5135	5139	5139	5139
Ta067	5246	5259	5288	5246*
Ta068	5094	5120	5129	5099
Ta069	5448	5489	5489	5448*
Ta070	5322	5341	5341	5328
Ta071	5770	5846	5886	5783
Ta072	5349	5453	5460	5388
Ta073	5676	5824	5743	5705
Ta074	5781	5929	5926	5853
Ta075	5467	5679	5647	5540
Ta076	5303	5375	5373	5331
Ta077	5595	5704	5681	5675
Ta078	5617	5760	5752	5694
Ta079	5871	6032	5993	5960
Ta080	5845	5918	5922	5903
Ta081	6202	6541	6502	6466
Ta082	6183	6523	6494	6373
Ta083	6271	6639	6534	6482
Ta084	6269	6557	6611	6460
Ta085	6314	6695	6590	6499
Ta086	6364	6664	6725	6557
Ta087	6268	6632	6631	6487
Ta088	6401	6739	6839	6648
Ta089	6275	6677	6630	6460
Ta090	6434	6677	6683	6580
Ta091	10862	10942	10986	10872
Ta092	10480	10716	10722	10559
Ta093	10922	11025	11025	10967
Ta094	10889	11057	11057	11057
Ta095	10524	10645	10575	10556
Ta096	10329	10458	10476	10345
Ta097	10854	10989	10965	10885
Ta098	10730	10829	10798	10770
Ta099	10438	10574	10594	10465
Ta100	10675	10807	10795	10758
Ta101	11195	11625	11640	11450
Ta102	11203	11675	11781	11497
Ta103	11281	11852	11789	11585
Ta104	11275	11803	11764	11562
Ta105	11259	11685	11624	11464
Ta106	11176	11629	11658	11400
Ta107	11360	11833	11798	11535
Ta108	11334	11913	11883	11595
Ta109	11192	11673	11665	11527
Ta110	11288	11869	11805	11558
Ta111	26059	26670	26743	26365
Ta112	26520	27232	27167	26817
Ta113	26371	26848	26925	26658
Ta114	26456	27055	26988	26666
Ta115	26334	26727	26788	26482
Ta116	26477	26992	27111	26698
Ta117	26389	26797	26731	26525
Ta118	26560	27138	27027	26793
Ta119	26005	26631	26627	26265
Ta120	26457	26984	26987	26685

Em relação ao tempo de computação, o método NBM-FS apresentou maior esforço computacional para a obtenção das soluções. Entretanto, o seu tempo médio de computação foi no máximo de 6,5 segundos, para os maiores problemas avaliados, sendo assim, não relevante para os aspectos práticos para fins de comparação.

Com o objetivo de verificar a existência de diferença estatisticamente significativa entre os métodos avaliados, foi aplicado o teste HSD de Tukey (Tukey Honestly Significant Difference Test), conforme apresentado na tabela 4.

Tabela 4 – Teste HSD de Tukey.

Métodos (I)	Métodos (J)	Diferenças de Médias (I-J)	Significância
NEH	NEHKK	0,284592	0,196
	NBM-FS	1,572117*	0,000
NEHKK	NEH	-0,284592	0,196
	NBM-FS	1,287525*	0,000
NBM-FS	NEH	-1,572117*	0,000
	NEHKK	-1,287525*	0,000

\* As diferenças entre as médias são significativas no nível de 5%.

Conforme o resultado apresentado na Tabela 4, verificou-se que existe uma diferença estatisticamente significativa entre os métodos NEH e NBM-FS, bem como entre os métodos NEHKK e NBM-FS. Entretanto, não existem diferenças entre os métodos NEH e NEHKK. Para observar a significância estatística entre os diferentes métodos, foi obtido o intervalo de confiança (95%), para as Porcentagens de Desvios Relativos dos métodos avaliados, conforme apresentado no Gráfico 1.

Os resultados apresentados na Gráfico 1 indicaram a presença de diferenças significativas do método NBM-FS, em comparação aos demais métodos (NEH e NEHKK).

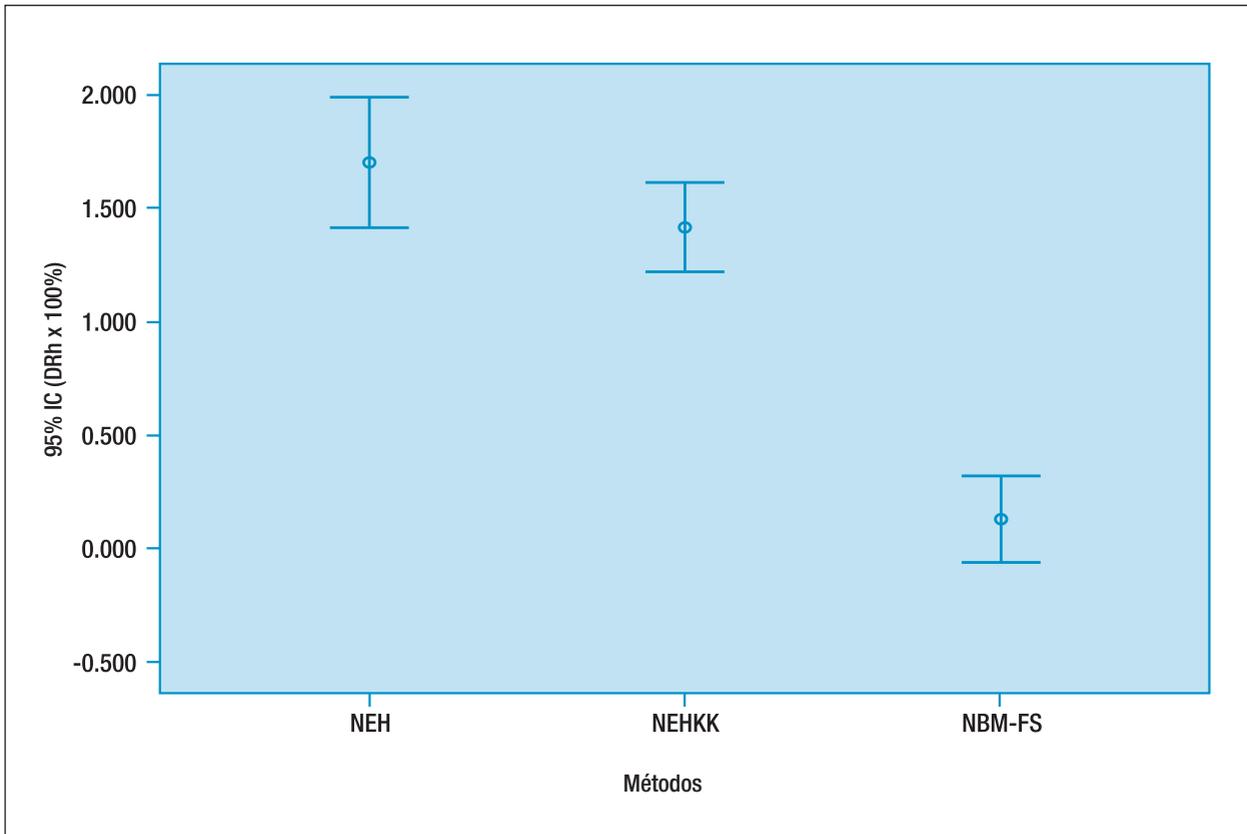


Gráfico 1 – Média e Intervalos de Confiança da Porcentagem de Desvio Relativo para os métodos avaliados.

## 5. CONSIDERAÇÕES FINAIS

Os resultados experimentais mostraram que o método heurístico NBM-FS apresentou um desempenho superior, tanto na porcentagem de sucesso e no desvio relativo médio para o conjunto de problemas avaliados, em comparação com os melhores métodos referenciados na literatura para o problema (NEH e NEHKK), além de obter soluções de altíssima qualidade.

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. Na literatura, as medidas de desempenho mais utilizadas têm sido a duração total da programação (makespan) e a soma dos tempos de fluxo (total flowtime), 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.

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 BIBLIOGRÁFICAS

- DANNENBRING, D. G. **An evaluation of flow-shop sequencing heuristics**. Management Science. v. 23, pp. 1174-1182, 1977.
- CAMPBELL, H. G.; DUDEK, R. A.; SMITH, M. L. **A heuristic algorithm for the n-job, m-machine sequencing problem**. Management Science. v. 16, pp. 630-637, 1970.
- DAVOUD POUR, H. **A new heuristic for the n-job, m-machine flow-shop problem**. Production Planning and Control. V. 12, n. 07, pp.648-653, 2001.
- GRABOWSKI, J.; WODECKI, M. **A very fast tabu search algorithm for the permutation flow shop problem with makespan criterion**. Computer & Operations Research. v. 31, pp. 1891-1909, 2004.
- GUPTA, J. N. D. **A functional heuristic algorithm for the flow-shop scheduling problem**. Operational Research Quarterly. v. 22, pp. 39-47, 1971.
- HUNDAL, T. S.; RAJGOPAL, J. **An extension of Palmer's heuristic for the flow-shop scheduling problem**. International Journal of Production Research. v. 26, pp. 1119-1124, 1988.

- JOHNSON, S. M. **Optimal two-and three-stage production schedules with setup times included**. *Naval Research Logistics Quarterly*. v. 1, pp. 61-68, 1954.
- KALCZYNSKI, P. J.; KAMBUROWSKI, J. On the NEH heuristic for minimizing the makespan in permutation flow shops. *Omega, The International Journal of Management Science*. v. 35, pp. 53-60, 2007.
- KOULAMAS, C. A new constructive heuristic for the flowshop scheduling problem. *European Journal of Operational Research*. v. 105, pp. 66-71, 1998.
- NAGANO, M. S.; MOCCELLIN, J. V. A high quality solution constructive for flow shop sequencing. *Journal of the Operational Research Society*. v. 53, pp. 1374-1379, 2002.
- NAWAZ, M.; ENSCORE JR., E.; E., HAM. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. *Omega, The International Journal of Management Science*. v. 11, pp. 91-95, 1983.
- NOORUL HAQ, A.; SARAVANAN, M.; VIVEKRAJ, A.R.; PRASAD, T. Scatter search approach for general flowshop scheduling problem. *Journal The International Journal of Advanced Manufacturing Technology*. ISSN 0268-3768. v. 31, n. 7-8, 2007.
- NOWICKI, E.; SMUTNICKI, C. A fast tabu algorithm for the permutation flow-shop problem. *Journal of Operational Research*. v. 91, pp. 160-175, 1996.
- OGBU, F.A.; SMITH, D.K. **The application of the simulated annealing algorithms to the solution of the  $n/m/C_{\max}$  flowshop problem**. *Computers & Operations Research*. v. 17, n. 3, pp. 243-253, 1990.
- OSMAN, I. H.; POTTS, C. N. Simulated annealing for permutation flow-shop scheduling. *Omega, The International Journal of Management Science*. v. 17, pp. 551-557, 1989.
- PALMER, D. S. **Sequencing jobs through a multi-stage process in the minimum total time** - A quick method of obtaining a near optimum. *Operational Research Quarterly*. v. 16, pp. 101-107, 1965.
- RAJENDRAN, C.; ZIEGLER, H. Ant-colony algorithms for permutation flowshop scheduling to minimize makespan/total flowtime of jobs. *European Journal of Operational Research*. v. 155, pp. 426-436, 2004.
- REEVES, C. R. **A genetic algorithm for flowshop sequencing**. *Computers and Operations Research*. v. 22, pp. 5-13, 1995.
- STÜTZLE, T. **Applying iterated local search to the permutation flow shop problem**. Technical report, AIDA-98-04, TU Darmstadt, FG Intellektik, 1998.
- TAILLARD, E. Some efficient heuristic methods for the flow shop sequencing problem. *European Journal of Operational Research*. v. 47, pp. 65-74, 1990.
- WIDMER, M.; HERTZ, A. A new heuristic method for the flow shop sequencing problem. *European Journal of Operational Research*. v. 41, pp. 186-193, 1989.