

Métodos de alto rendimento e baixa complexidade em *flowshop*

Methods of high performance and low complexity in flowshop

Fábio José Ceron Branco¹ - Univ. Tecnológica Federal do Paraná - Dep. de Engenharia de Produção - Campus Ponta Grossa
Helio Yochihiro Fuchigami² - Universidade Federal de Goiás

RESUMO

As empresas procuram a cada dia aperfeiçoar o seu processo produtivo e reduzir os custos de produção para aumentar a sua eficiência. Este trabalho trata do problema de programação de operações *flowshop* permutacional, tratando tanto de programação de operações em um ambiente de produção clássico, no-idle e no-wait, tendo como objetivo a minimização do makespan e flowtime. A partir da combinação de procedimentos de ordenação e re-sequenciamento das tarefas, são obtidos novos métodos heurísticos construtivos. Neste trabalho, o desempenho dos principais métodos de ordenação inicial é mostrado, como forma de definir vantagens e desvantagens na sua utilização como iniciador de procedimentos de métodos construtivos e metaheurísticos. No que se refere a qualidade da solução, o método BN se mostrou mais interessante, independentemente do ambiente *flowshop* testado. Os métodos são avaliados mediante uma experimentação computacional e mostra soluções de alta qualidade quando comparados com outros métodos de destaque da literatura.

Palavras-chave: Flowshop Permutacional. No-wait. No-idle. Métodos Heurísticos. Programação da Produção. Makespan. Flow Time.

ABSTRACT

Every day, companies seek to improve their production process and reduce production costs, thus increasing their efficiency. This work addresses the problem of flowshop scheduling operations programming, programming of operations in flowshop problems, no-idle and no-wait, to minimization makespan and flowtime. From the combination of sorting procedures and re-sequencing, new constructive heuristic methods are obtained. In this study, the performance of the main initial ordering methods is shown as a way of defining advantages and disadvantages in their use as the initiator of constructives and metaheuristics methods. In terms of the quality of the solution, the BN method proved more interesting, independent of the flowshop environment tested. The methods are evaluated using computational experimentation and show high quality solutions when compared with other methods presented in the literature.

Keywords: Flowshop Permutacional. No-wait. No-idle. Heuristics. Production program. Makespan. Flow Time.

1. Avendia Monteiro Lobato, s/n, Km 4, CEP 84016-210, Ponta Grossa, Paran[a, Brasil, e-mail: fbranco@hotmail.com / fbranco@utfpr.edu.br;
2. heliofuchigami@yahoo.com.br

BRANCO, F. J. C.; FUCHIGAMI, H. Y. Métodos de alto rendimento e baixa complexidade em *flowshop*. **GEPROS. Gestão da Produção, Operações e Sistemas**, Bauru, Ano 12, nº 4, out-dez/2017, p. 32-56.

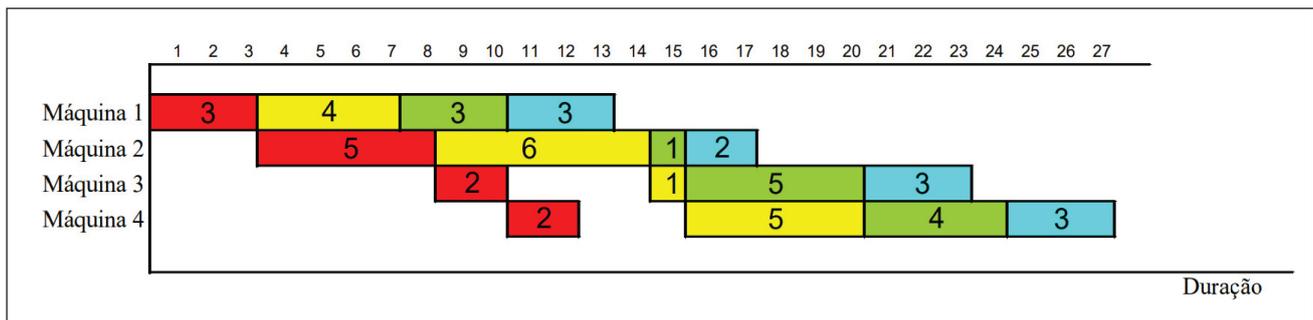
DOI: 10.15675/gepros.v12i4.1737

1. INTRODUÇÃO

Com o desenvolvimento das indústrias manufatureiras, os métodos para *Just-In-Time* (JIT) e inventário zero para a redução dos custos da produção ganharam importância, e o problema da programação da produção passou a igualmente se destacar na comunidade científica, principalmente em sistema de produção *flowshop*.

O problema tradicional de programação em sistema de produção *flowshop* ocorre em um conjunto de n tarefas que devem ser processadas, na mesma seqüência, em m máquinas. Quando a ordem de processamento em todas as máquinas for a mesma, tem-se o ambiente de produção *flowshop* permutacional, no qual o número de possíveis programações para n tarefas é $n!$. A Figura 1 mostra um exemplo do gráfico de Gantt para o problema *flowshop* clássico, com 4 tarefas e 4 máquinas, com os tempos de processamento de cada etapa dentro de cada lacuna.

Figura 1 - *Flowshop* com 4 máquinas e 4 tarefas.



Fonte: Elaboração dos autores.

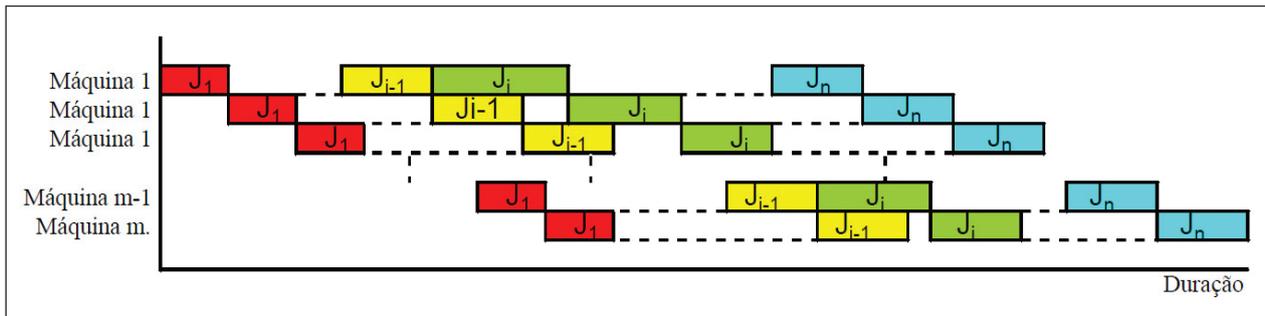
O problema consiste em obter uma seqüência das tarefas que otimize uma determinada medida de desempenho. Nos modelos para solução do problema, as medidas de desempenho usuais referem-se à minimização do tempo médio de fluxo (*mean flow time*), que reduz o estoque em processamento, e à minimização da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

Uma situação específica muito freqüente para este caso é a programação da produção em sistema de produção *flowshop* sem interrupção na execução das tarefas que, geralmente, ocorre em um ambiente caracterizado pela tecnologia do processo, por exemplo, uma variação de temperatura pode influenciar na degeneração do material, ou pela falta de capacidade de armazenamento entre as máquinas, problema encontrado nas indústrias químicas, metalúrgicas, alimentícias, serviços e farmacêuticas.

Em uma indústria de metais como alumínio e aço, a laminação de placas passa por uma série de processos, onde, primeiro a placa é pré-aquecida e depois conduzida para uma série de prensas, que diminui progressivamente a espessura até atingir sua especificação final. Nesse processo, a placa tem que passar de uma máquina a outra, na seqüência, sem ficar esperando pela disponibilidade da próxima. Nessa situação, qualquer espera pode causar o resfriamento do metal, gerando problemas na laminação e, conseqüentemente, perda de material.

No caso de indústria química, não deverá ocorrer interrupção na execução das tarefas no envasamento de compostos químicos. Um composto químico, depois de passar por um misturador, tem que ser envasado logo em seguida, pois a exposição ao ar pode deteriorá-lo. Processo semelhante também ocorre na indústria alimentícia, pois um alimento tem que ser enlatado após seu cozimento para garantir sua qualidade. Uma outra ocorrência do problema encontra-se no setor de serviços, devido ao fato do cliente ter uma alta intolerância à espera.

Em tais ambientes apresentados, procura-se reduzir custos com inventários e planejar a produção de forma a atender a demanda, sem que haja excesso ou falta do produto para que, assim, possa maximizar a utilização dos recursos e baratear o processo de acordo com a necessidade do mercado, sistema chamado *no-wait* (NW), conforme exemplificado na Figura 2.

Figura 2 - *No-Wait Flowshop* com m máquinas e n tarefas.

Fonte: Elaboração dos autores.

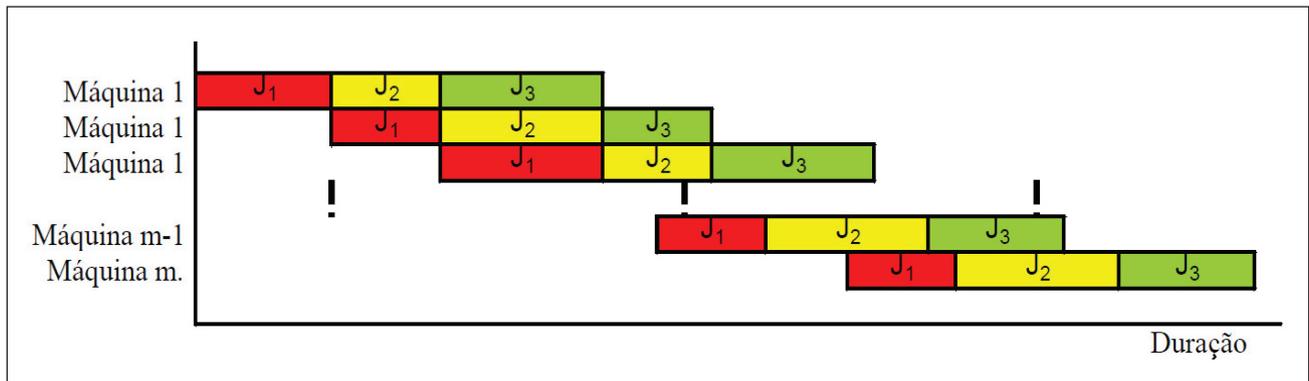
A Figura 2 apresenta a programação de um conjunto de n tarefas ($J = \{ J_1, J_{i-1}, \dots, J_i, \dots, J_n \}$) que devem ser processadas, na mesma seqüência, por um conjunto de m máquinas distintas, onde o tempo de processamento da tarefa J_i , na máquina k , é P_{ki} ($i = 1, 2, \dots, n; k = 1, 2, \dots, m$).

Uma situação específica e muito freqüente para este problema é a programação da produção em sistema de produção *flowshop* sem interrupção na execução das tarefas, que ocorre geralmente em um ambiente caracterizado pela tecnologia do processo. Algumas aplicações típicas para este problema são encontradas em indústrias de produção de metais e plásticos (WISMER, 1972), indústrias químicas (RAJENDRAN, 1994), farmacêuticas (RAAYMAKERS; HOOGEVEEN, 2000) e indústrias de processamento de alimentos (HALL; SRISKANDARAYAH, 1996).

A principal característica do NWFS é a necessidade de que a operação $g + 1$, de uma determinada tarefa J_i , tem que ser processada logo após o término da operação g ($1 \leq g \leq m - 1$), não permitindo que ocorra tempo de espera no processamento de uma tarefa de uma máquina para a outra. O único tempo de espera permitido é no início do processamento da tarefa que ocupa a primeira posição na seqüência na primeira máquina.

Outra situação específica e freqüente para este problema é a programação da produção em sistema de produção *no-idle* (NI), conforme apresentado na Figura 3.

Figura 3 - *No-Idle Flowshop* com m máquinas e $n = 3$ tarefas.



Fonte: Elaboração dos autores.

Nesse caso as operações de uma determinada máquina uma vez iniciadas são processadas sem interrupções, não havendo *Gaps*, ou seja, lacunas entre as tarefas.

Neste trabalho, reportam-se os resultados de uma pesquisa com o objetivo de desenvolver um novo método heurístico com desempenhos superiores ao LPT, NEH e SPT, nos diferentes sistemas de produção. Tal método é baseado no procedimento de construção da seqüência 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, ou seja, adequado equilíbrio entre a qualidade da solução e esforço computacional, simplicidade e facilidade de implementação. No próximo item são apresentados os principais trabalhos da literatura para cada restrição citada.

2. REVISÃO DA LITERATURA

Nesta etapa do trabalho é apresentada uma breve revisão da literatura, separada em 3 tópicos conforme o ambiente de produção tratado.

2.1. *Flowshop*

O problema de programação de operações *flowshop* é um problema da produção no qual n tarefas devem ser processadas, em cada máquina de um conjunto de m máquinas distintas. Um caso específico de programação *flowshop*, denominado permutacional, acontece quando em todas as máquinas mantém-se a mesma ordem de processamento das tarefas.

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. Devido à sua natureza, o problema NWFS é considerado como *NP-hard* para o caso de três ou mais máquinas (CHEN et al., 1996).

Os métodos heurísticos podem ser classificados de diversas maneiras. Uma delas classifica-os em Construtivos ou Melhorativos, dependendo da forma de obtenção da solução. Os métodos construtivos, foco deste trabalho, 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).

Dentre os métodos construtivos mais conhecidos, salienta-se o NEH (iniciais dos autores) proposto por Nawaz et al. (1983), que tem sido considerado até hoje como o melhor método heurístico construtivo para minimizar o *makespan* em *flowshop* permutacionais, com alta qualidade de solução e eficiência computacional.

Outros métodos surgiram durante as décadas seguintes, entre eles destaca-se Nagano e Moccellin (2002) que propuseram uma heurística construtiva denominada N&M em que há uma mudança apenas na fase de ordenação inicial em relação ao NEH.

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 seqüência solução, na qual em caso de empate das soluções parciais nas subseqüências obtidas, um procedimento heurístico foi criado baseado na “Regra de Johnson” (JOHNSON,1954).

2.2. *No-wait flowshop* (NWFS)

Devido a sua natureza, o problema *no-wait flowshop* (NWFS) é considerado *NP-Hard* para o caso de três ou mais máquinas (PAPADIMITRIOU; KANELLAKIS, 1980; HALL; SRISKANDARAYAH, 1996).

Os primeiros trabalhos para minimização do *makespan* para o problema NWFS começaram a partir de Deman e Baker (1974). Em sua pesquisa os autores 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*).

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

Posteriormente, 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 pelo método proposto por Bonney e Gundry (1976).

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_j , 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 *setup*. Foram obtidas soluções ótimas para casos especiais estabelecendo uma relação de dominância local, e fornecendo solução heurística para o caso genérico.

Mais adiante, Aldowaisan e Allahverdi (2004) propuseram métodos heurísticos compostos de três fases. Conforme os resultados dos experimentos, um algoritmo denominado PHI(p) 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 do *makespan*, Reddi e Ramamoorthy (1972) e Wismer (1972) foram os primeiros a estudar o problema. 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, isto é, uma primeira de ordenação inicial das tarefas, e uma segunda de construção da seqüência final.

Rajendran (1994) apresentou uma evolução de Bonney e Gundry (1976) e de King e Spachis (1980). O método é estruturalmente semelhante aos de Gangadharan e Rajendran (1993), pois também é composto por duas fases. Na primeira, a ordenação inicial é determinada por um índice calculado para cada tarefa; E a segunda utiliza um procedimento que avalia seqüências parciais até que todas as tarefas estejam programadas. O método construtivo de Rajendran (1994) é o melhor encontrado na literatura para o problema NWFS com critério de minimização do *makespan*.

Nas últimas três décadas, um extenso esforço de pesquisa tem sido dedicado para o problema, alguns métodos propostos tem-se baseado em métodos já existentes como, por exemplo, o método heurístico NEH (NAWAZ et al., 1983), devido sua eficiência e eficácia (qualidade da solução e tempo de computação para a obtenção da solução), vários pesquisadores tem proposto sua diversificação e adaptação em problemas de *flowshop* com restrições adicionais ou com critérios de avaliações diferentes (FRAMINAN et al., 2003).

Nagano et al. (2006) abordam a solução do NWFS com a redução de estoque em processamento (*flowtime*), com o desenvolvimento de um método heurístico denominado S&N, que demonstrou, através da experimentação computacional, terem uma superioridade de qualidade de solução quando comparado a outros métodos destacados da literatura.

Mais recentemente, Araujo e Nagano (2011) trataram do problema *No-wait*, porém considerando o *setup* separado dos tempos de processamento e dependentes da sequencia.

Sapkal e Laha (2013) apresentaram uma heurística para a minimização do *flowtime* baseado em uma sequência inicial dada pela soma dos tempos de processamento das tarefas, e adicionalmente, consideraram no sistema de produção máquinas-chave, consideradas gargalo pelo sistema produtivo.

Várias pesquisas têm considerado somente um dos critérios de avaliação (*makespan* ou *total flow time*). Entretanto, quando são levados em consideração ambos os critérios simultaneamente, o problema torna-se mais realístico e complexo, sendo este conhecido na literatura como análise multicritério. Esta análise pode ser tratada em outros trabalhos.

2.3. *No-idle flowshop* (NIFS)

Uma situação específica e freqüente para este problema é a programação da produção em sistema de produção *no-idle flowshop* (NIFS). Nesse caso as operações de uma determinada máquina uma vez iniciadas são processadas sem interrupções, não havendo *Gaps*, ou seja, lacunas entre as tarefas.

Conforme Baraz e Mosheiov (2007), situações típicas para este problema são encontradas em várias indústrias, onde o tempo de preparação (*setup*) ou custo de funcionamento são relativamente altos, implicando assim que desligá-la ou prepará-la mais vezes que o necessário provoca um processo bastante

oneroso. Mesmo sendo um problema comum na prática, considerado como do tipo *NP-Hard* por Tanaev et al. (1994), os estudos neste tópico não atraíram muita atenção dos pesquisadores.

Em um dos primeiros trabalhos neste ambiente, Adiri e Pohoryles (1982) propuseram algumas propriedades para minimização da soma dos tempos de fluxo em problemas NIFS para duas máquinas, demonstrando numericamente os teoremas propostos. Woollam (1986) propôs um método do tipo *branch-and-bound* para minimização do *makespan* em problemas NIFS que se mostrou eficiente para experimentações de até 10 tarefas, numa comparação feita com os outros 4 métodos de destaque na época adaptados ao ambiente *no-idle*.

Saadani et al. (2005) estudaram o problema de m-máquina NIFS permutacional para minimização do *makespan*. Os autores mostraram que o problema pode ser representado como um problema do Caixeiro-Viajante (TSP), e propuseram um método heurístico baseado neste tipo de estrutura. A experimentação computacional (limitada aos exemplos de até 30 máquinas e de 25 tarefas) indicou que a heurística produz soluções de alta qualidade, muito próximas das soluções ótimas.

Baraz e Mosheiov (2007) propuseram um novo método heurístico (denotado por IG) para o problema NIFS buscando a minimização do *makespan*. Esse algoritmo consiste em duas etapas: (1) um procedimento de ordenação inicial de tarefas chamado *Greedy*, baseado na adição de uma única tarefa a cada iteração, seguida (2) por um procedimento de melhoria baseado na troca de pares de tarefas. Esse método foi comparado com o procedimento proposto por Saadani et al. (2005) e com o do Caixeiro-Viajante tradicional. A experimentação realizada mostrou que o método IG possui um *makespan* médio inferior aos demais comparados, para problemas de 4 e 8 máquinas, e as tarefas variando de 10 a 400. De acordo com a revisão da literatura efetuada na pesquisa referente a este trabalho, o método IG é o melhor em termos de qualidade da solução para a programação NIFS que trata da minimização do *makespan*.

Narain e Bagga (2005) também propuseram um método baseado na técnica *branch-and-bound* para problemas NIFS, porém com duas máquinas e tratando da minimização do tempo médio de fluxo, para minimização do (*flow time*) não foram encontrados outros trabalhos.

Mais recentemente, Tasgetiren et al. (2010) propôs um algoritmo chamado vpsDE para minimização de tarefas atrasadas (tardiness) para o problema NIFS. Trata-se de uma rotina metaheurística do Simulated Annealing inserida no método DE, de Storn e Price (1997), baseado em funções não-lineares contínuas. O método apresentou resultados satisfatórios quando comparado com o método genético RKGGA, de Tasgetiren et al. (2009), e o banco de dados utilizado foi o de Taillard (1993) composto por 120 problemas.

Ainda é possível citar a metaheurística aplicada ao problema NIFS apresentada por Nagano e Januario (2013). Foi utilizado a técnica de *cluster search* em uma metodologia híbrida e o método mostrou qualidade superior quando comparado com algoritmos construtivos.

3. MÉTODO

Para a avaliação do desempenho do método proposto, denominado BN-FS, foram realizadas experimentações computacionais utilizando os problemas de Taillard (1993). Neste trabalho, o método BN-FS foi codificado em linguagem DELPHI e processado em um microcomputador AMD Athlon(tm) 64x2 Dual Core Processador 4000 + 2.11 GHz, com 1 GByte de RAM.

As estatísticas usadas para análise de desempenho do método foram à porcentagem de sucesso, o desvio relativo médio e o tempo computacional. 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:

$$DR_h = \frac{(M_h - M^*)}{M^*},$$

Onde M_h : função-objetivo do método *h* e M^* : melhor função-objetivo obtido pelo(s) método(s), para um determinado problema.

As estatísticas descritas são adaptadas ao tempo de fluxo para as análises correspondentes.

E a terceira, o tempo médio de computação de um método é calculado pela soma dos tempos de computação de cada problema dividida pelo número total de problemas resolvidos (média aritmética dos tempos de computação). Na experimentação computacional, o tempo médio de computação por problema foi medido em milissegundo (ms).

Os tempos de processamento das tarefas foram gerados aleatoriamente variando no intervalo de 1 a 120, com distribuição uniforme. Cabe lembrar que, o tempo de computação torna-se maior com o aumento do número de tarefas.

Serão discutidos posteriormente os resultados obtidos pelo método BN-FS com o critério de minimização do *makespan* e *flow time*, comparados com os métodos heurísticos NEH (NAWAZ et al., 1983), SPT e LPT, em relação aos dados dos problemas testes de Taillard. É apropriado notar que, quanto menor o valor do desvio relativo médio, maior é a qualidade da solução obtida pelo método analisado.

4. HEURÍSTICAS

Neste item, os métodos de ordenação inicial implementados são abordados conforme seus passos:

- LPT (*Longest Processing Time*): este algoritmo obedece uma ordem decrescente da soma dos tempos de processamento.
- SPT (*Shortest Processing Time*): ordenação crescente da soma dos tempos de processamento.
- NEH: proposto por Nawaz et al. (1983), que tem sido considerado até hoje como o melhor método heurístico construtivo para minimizar o *makespan* em *flowshop* 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* para *makespan* e SPT - *Shortest Processing Time* para *flow time*).
 - Fase 2: Construção da seqüência solução a partir da inserção das tarefas em todas as posições possíveis nas seqüências parciais, conforme a ordenação da Fase 1.

Adota-se aquela que leva a uma menor duração total da programação parcial.

- BN: proposto por Branco et al. (2006), e adaptado ao problema de restrições (*no-wait* e *no-idle*) tratados neste trabalho. É possível, por meio da propriedade já existente e apresentada por Lawler et al. (1993), criar novos métodos de soluções, obtendo desempenhos superiores aos já existentes. O método heurístico proposto 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 seqüência 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 seqüência atual S , e calcule seu respectivo *makespan* para a seqüência parcial;
 - > Considerando toda a Vizinhança de Inserção da seqüência parcial com k tarefas, constituída de $(k-1)^2$ seqüências, determine a seqüência 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 seqüência parcial com k tarefas, constituída de $k(k-1)/2$ seqüências, determine a seqüência 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 seqüência parcial com k tarefas, constituída de $k(k-1)/2$ seqüências, determine a seqüência S_p associada ao menor *makespan*;
 - > Se o *makespan* de S_p é menor que de S , atualize S com S_p .

Para o critério de minimização do tempo médio de programação o procedimento é o mesmo descrito anteriormente, porém associado ao *flow time*.

5. DISCUSSÃO

São apresentados na Tabela 1 a comparação dos resultados dos métodos heurísticos NEH (NAWAZ et al., 1983), SPT e LPT no ambiente clássico com critério de minimização do *makespan* e *flowtime*. As comparações são dadas nas tabelas pelos critérios já descritos: PS (porcentagem de sucesso), DRM (desvio relativo médio) e T (tempo de CPU).

Tabela 1 – Clássico: *Makespan* e *flowtime*.

n	m		LPT	NEH	SPT	BN	LPT	NEH	SPT	BN
			MAKESPAN				FLOWTIME			
20	5	PS	0	20	0	90	0	10	0	90
		DRM	2,491	0,218	2,107	0,022	4,683	0,376	1,401	0
		T	78	125	94	157	94	109	93	157
	10	PS	0	0	0	100	0	0	0	100
		DRM	2,618	0,275	2,414	0	3,669	0,29	1,297	0
		T	109	109	109	172	93	94	110	171
	20	PS	0	0	0	100	0	0	0	100
		DRM	1,926	0,181	2,102	0	2280	0,307	0,938	0
		T	125	109	110	296	110	109	94	219
Média	PS	0	6,67	0	96,67	0	3,33	0	96,67	
	DRM	2,345	0,225	2,208	0,007	3,544	0,324	1,212	0	
	T	104	114,3	104,3	208,3	99	104	99	182,3	

n	m		LPT	NEH	SPT	BN	LPT	NEH	SPT	BN
			MAKESPAN				FLOWTIME			
50	5	PS	0	10	0	100	0	0	0	100
		DRM	1,617	0,047	1,762	0	4,875	0,293	1,493	0
		T	250	250	218	1360	250	250	265	1578
	10	PS	0	0	0	100	0	0	0	100
		DRM	2,585	0,262	2,536	0	3,956	0,38	1,763	0
		T	234	282	266	2281	234	281	235	2657
	20	PS	0	0	0	100	0	0	0	100
		DRM	2,466	0,293	2,427	0	3,157	0,307	1,418	0
		T	234	312	234	4109	220	312	234	4625
Média	PS	0	3,33	0	100	0	0	0	100	
	DRM	2,223	0,201	2,242	0	3,996	0,327	1,558	0	
	T	239,3	281,3	239,3	2583,3	234,7	281	244,7	2953,3	
100	5	PS	0	30	0	90	0	0	0	100
		DRM	1,183	0,032	1,447	0,001	4,728	0,245	1,522	0
		T	485	547	438	13890	468	594	484	18485
	10	PS	0	0	0	100	0	0	0	100
		DRM	1,809	0,135	1,774	0	3,888	0,32	1,509	0
		T	453	687	485	27891	422	750	437	35999
	20	PS	0	0	0	100	0	0	0	100
		DRM	1,996	0,233	2,089	0	2,802	0,359	1,523	0
		T	469	922	438	54172	438	1031	485	68969
Média	PS	0	10	0	96,67	0	0	0	100	
	DRM	1,663	0,133	1,77	0	3,806	0,308	1,518	0	
	T	469	718,7	453,7	31984,3	442,7	791,7	468,7	41151	

n	m		LPT	NEH	SPT	BN	LPT	NEH	SPT	BN
			MAKESPAN				FLOWTIME			
200	10	PS	0	0	0	100	0	0	0	100
		DRM	1,526	0,087	1,449	0	3,708	0,199	1,474	0
		T	922	2562	906	386297	890	2953	922	512359
	20	PS	0	0	0	100	0	0	0	100
		DRM	1,918	0,254	1,95	0	2,924	0,333	1,506	0
		T	890	4203	906	738219	906	5172	890	1005109
Média	PS	0	0	0	100	0	0	0	100	
	DRM	1,722	0,171	1,699	0	3,316	0,266	1,49	0	
	T	906	3382,5	906	562258	898	4062,5	906	758734	
500	20	PS	0	0	0	100	0	0	0	100
		DRM	1,548	0,134	1,519	0	2,833	0,163	1,375	0
		T	2344	46219	2266	24343796	2282	60922	2297	33920438
	Média	PS	0	0	0	100	0	0	0	100
		DRM	1,548	0,134	1,519	0	2,833	0,163	1,375	0
		T	2344	46219	2266	24343796	2282	60922	2297	33920438

Fonte: Elaboração dos autores.

A Tabela 2 mostra a comparação dos resultados dos métodos para o ambiente *no-wait* com critério de minimização do *makespan* e *flowtime*.

Tabela 2 - *No-Wait: Makespan e flowtime.*

n	m		LPT	NEH	SPT	BN	LPT	NEH	SPT	BN
			MAKESPAN				FLOWTIME			
20	5	PS	0	0	0	100	0	10	0	90
		DRM	2,516	0,204	2,313	0	4,619	0,155	1,667	0,016
		T	94	93	94	125	94	94	125	110
	10	PS	0	10	0	90	0	0	0	100
		DRM	2,58	0,214	2,722	0,001	3,81	0,228	1,811	0
		T	109	125	110	125	94	109	94	125
	20	PS	0	0	0	100	0	10	0	90
		DRM	2,228	0,182	2,376	0	2,626	0,178	1,39	0,005
		T	110	94	109	125	109	94	109	125
Média	PS	0	3,33	0	96,67	0	6,67	0	93,33	
	DRM	2,441	0,2	2,47	0	3,685	0,187	1,623	0,007	
	T	104,3	104	104,3	125	99	99	109,3	120	
50	5	PS	0	0	0	100	0	0	0	100
		DRM	2,929	0,454	2,894	0	5,616	0,352	2,56	0
		T	234	250	266	703	266	250	235	907
	10	PS	0	0	0	100	0	0	0	100
		DRM	3,102	0,309	3,048	0	4,533	0,203	2,562	0
		T	218	235	250	718	234	218	234	890
	20	PS	0	0	0	100	0	0	0	100
		DRM	2,976	0,256	3,059	0	3,885	0,297	2,535	0
		T	266	265	234	719	234	266	266	938
Média	PS	0	0	0	100	0	0	0	100	
	DRM	3,002	0,34	3	0	4,678	0,284	2,552	0	
	T	239,3	250	250	713,3	244,7	244,7	245	911,7	

n	m		LPT	NEH	SPT	BN	LPT	NEH	SPT	BN
			MAKESPAN				FLOWTIME			
100	5	PS	0	0	0	100	0	0	0	100
		DRM	3,125	0,463	3,161	0	5,949	0,459	3,042	0
		T	437	516	468	6703	469	515	453	9515
	10	PS	0	0	0	100	0	0	0	100
		DRM	3,477	0,349	3,404	0	4,984	0,266	3,123	0
		T	500	531	468	6719	438	547	469	9500
	20	PS	0	0	0	100	0	0	0	100
		DRM	3,441	0,212	3,537	0	4,22	0,278	3,242	0
		T	500	547	516	6750	500	531	515	9531
Média	PS	0	0	0	100	0	0	0	100	
	DRM	3,348	0,341	3,367	0	5,051	0,334	3,136	0	
	T	479	531,3	484	6724	469	531	479	9515,3	
200	10	PS	0	0	0	100	0	0	0	100
		DRM	3,765	0,339	3,79	0	5,478	0,375	3,786	0
		T	1000	1469	985	90374	984	1453	984	137203
	20	PS	0	0	0	100	0	0	0	100
		DRM	4,154	0,306	4,185	0	5,002	0,298	4,102	0
		T	1078	1469	1109	90485	1094	1469	1047	137454
	Média	PS	0	0	0	100	0	0	0	100
		DRM	3,96	0,322	3,987	0	5,24	0,336	3,944	0
		T	1039	1469	1047	90429,5	1039	1461	1015,5	137328,5
500	20	PS	0	0	0	100	0	0	0	100
		DRM	4,649	0,275	4,611	0	5,658	0,314	4,725	0
		T	3422	9781	3281	3425968	3343	9719	3343	5440125
	Média	PS	0	0	0	100	0	0	0	100
		DRM	4,649	0,275	4,611	0	5,658	0,314	4,725	0
		T	3422	9781	3281	3425968	3343	9719	3343	5440125

Fonte: Elaboração dos autores.

É apresentada na Tabela 3 a comparação dos resultados dos métodos heurísticos BN, NEH (NAWAZ et al., 1983), SPT e LPT no ambiente *no-idle* com critério de minimização do *makespan* e *flowtime*.

Tabela 3 - *No-Idle: Makespan e Flowtime.*

n	m		LPT	NEH	SPT	BN	LPT	NEH	SPT	BN
			MAKESPAN				FLOWTIME			
20	5	PS	0	10	0	90	0	0	0	100
		DRM	2,426	0,221	2,164	0,009	4,054	0,71	2,626	0
		T	94	94	93	141	125	93	125	141
	10	PS	0	0	0	100	0	0	0	100
		DRM	2,764	0,518	2,746	0	3,676	0,709	2,994	0
		T	109	110	94	172	78	110	94	187
	20	PS	0	10	0	90	0	0	0	100
		DRM	2,492	0,431	2,196	0,003	2,82	0,457	2,314	0
		T	141	109	110	265	109	125	109	266
Média	PS	0	6,67	0	93,33	0	0	0	100	
	DRM	2,561	0,39	2,369	0,004	3,517	0,626	2,645	0	
	T	114,7	104,3	99	192,7	104	109,3	109,3	198	
50	5	PS	0	10	0	100	0	0	0	100
		DRM	1,653	0,067	1,676	0	4,889	0,649	2,812	0
		T	234	266	249	1703	202	249	234	1765
	10	PS	0	0	0	100	0	0	0	100
		DRM	3,075	0,463	2,949	0	5,145	0,898	3,895	0
		T	219	265	220	2907	173	267	235	3001
	20	PS	0	0	0	100	0	0	0	100
		DRM	3,094	0,467	2,853	0	4,127	0,814	3,512	0
		T	203	359	234	5406	202	312	265	5547
Média	PS	0	3,33	0	100	0	0	0	100	
	DRM	2,607	0,332	2,493	0	4,721	0,787	3,406	0	
	T	218,7	296,7	234,3	3338,7	192,3	276	244,7	3437,7	

n	m		LPT	NEH	SPT	BN	LPT	NEH	SPT	BN
			MAKESPAN				FLOWTIME			
100	5	PS	0	20	0	90	0	0	0	100
		DRM	1,243	0,048	1,492	0	4,729	0,638	2,328	0
		T	468	625	469	20734	485	625	437	22969
	10	PS	0	0	0	100	0	0	0	100
		DRM	2,017	0,136	2,031	0	4,574	0,728	3,338	0
		T	438	797	421	40109	422	828	454	41984
	20	PS	0	0	0	100	0	0	0	100
		DRM	3,118	0,366	2,976	0	4,768	0,876	4,046	0
		T	484	1156	437	78672	468	1172	468	80688
Média	PS	0	6,67	0	96,67	0	0	0	100	
	DRM	2,126	0,183	2,166	0	4,69	0,747	3,237	0	
	T	463,3	859,3	442,3	46505	458,3	875	453	48547	
200	10	PS	0	0	0	100	0	0	0	100
		DRM	2,028	0,152	1,99	0	4,81	0,617	3,093	0
		T	891	3484	891	606110	907	3594	907	638610
	20	PS	0	0	0	100	0	0	0	100
		DRM	2,807	0,269	2,681	0	4,862	0,738	3,989	0
		T	843	6157	891	1204515	922	6312	906	1235703
	Média	PS	0	0	0	100	0	0	0	100
		DRM	2,418	0,21	2,335	0	4,836	0,677	3,541	0
		T	867	4820,5	891	905312,5	914,5	4953	906,5	937156,5
500	20	PS	0	0	0	100	0	0	0	100
		DRM	2,499	0,178	2,467	0	5,105	0,623	4,116	0
		T	2266	83046	2219	46131828	2281	84313	2266	47878141
	Média	PS	0	0	0	100	0	0	0	100
		DRM	2,499	0,178	2,467	0	5,105	0,623	4,116	0
		T	2266	83046	2219	46131828	2281	84313	2266	47878141

Fonte: Elaboração dos autores.

Observa-se nas Tabelas 1, 2 e 3, a superioridade do método proposto para todos os casos analisados. Para os problemas de médio e grande porte tratados neste trabalho, em todas as situações os métodos BN possuem maiores porcentagens de sucesso, e conseqüentemente, índices próximos de zero para desvios relativos médios. Os resultados tendem a melhorar a medida que a complexidade da classe de problemas aumenta.

Como era esperado, o tempo de computação torna-se maior com o aumento do número de tarefas. O método BN tem o crescimento mais acentuado, o que pode ser explicado pelo maior número de testes em seqüências parciais para obtenção da melhor solução.

No próximo item deste trabalho, são feitas as considerações finais e a análise global da experimentação computacional.

5. CONCLUSÃO

Neste trabalho foi apresentado um novo método heurístico construtivo, denominado BN, seu desempenho foi comparado com os melhores métodos referenciados na literatura para o problema, (NEH, LPT e SPT), os resultados experimentais mostraram que o método heurístico BN apresentou um desempenho superior tanto na porcentagem de sucesso e no desvio relativo médio para o conjunto de problemas avaliados, tanto no ambiente clássico, como para *no-idle* e *no-wait*.

Os índices de desempenho mais utilizados têm sido a duração total da programação (*makespan*) e a soma dos tempos de fluxo (*total flowtime*), com frequência maior para a primeira. Os tempos computacionais médios de BN foram superiores aos concorrentes, porém aceitáveis diante de tamanha superioridade nas soluções geradas.

Entretanto, a busca de métodos heurísticos cada vez mais eficazes ainda permanece, buscando adequado equilíbrio entre a qualidade da solução, eficiência computacional, simplicidade e facilidade de implementação.

REFERÊNCIAS

- ADIRI, I.; POHORYLES, D. Flowshop/no-idle or no-wait scheduling to minimize the sum of completion times. **Naval Research Logistic Quartely**, v. 29, p.495-504, 1982.
- ALDOWAISAN, T.; ALLAHVERDI, A. Total flowtime in no-wait flowshops with separated setup times. **Computers and Operations Research**, v. 25, p. 757-765, 1998.
- ALDOWAISAN, T.; ALLAHVERDI, A. New heuristics for m-machine no-wait flowshop to minimize total completion time. **Omega**, v. 32, p. 345-352, 2004.
- ALLAHVERDI, A.; ALDOWAISAN, T. No-wait and separate setup three-machine flowshop with total completion time criterion. **International Transactions in Operational Research**, v. 7, p. 245-264, 2000.
- ALLAHVERDI, A.; ALDOWAISAN, T. Minimizing total completion time in a no-wait flowshop with sequence dependent additive changeover times. **Journal of the Operational Research Society**, v. 52, p. 449-462, 2001.
- ARAUJO D.; NAGANO M. A new effective heuristic method for the no-wait flowshop with sequence-dependent setup times problem. **Int J Ind Eng Comput**, v. 2, n. 1, p. 155-166, 2011.
- BARAZ, D.; MOSHEIOV, G. A note on a greedy heuristic for flow-shop makespan minimization with no machine idle-time. **European Journal of Operational Research**, IN PRESS. 2007.
- BERTOLISSI, E. A simple no-wait flowshop scheduling heuristic for the no-wait flow-shop problem. *In: INTERNATIONAL CONFERENCE ON COMPUTER-AIDED PRODUCTION ENGINEERING*. 15. 1999. **Anais...** University of Durham Publishers, 1999.
- BERTOLISSI, E. Heuristic algorithm for scheduling in th no-wait flow-shop. 107. ed. **Journal of Materials Processing Technology**, v. 107, p. 459-465, 2000.
- BONNEY, M. C.; GUNDRY, S. W. Solutions to the constrained flowshop sequencing problem. **Operations Research Quarterly**, v. 24, p. 869-883, 1976.

BRANCO, F.; NAGANO, M.; MOCCELLIN, J. Avaliação de métodos heurísticos para o problema no-wait flowshop com o critério de minimização da duração total da programação. *In: SIMPÓSIO BRASILEIRO PESQUISA OPERACIONAL*, 37, p. 71-81, 2006. **Anais...** São Paulo, SP, 2006.

CAMPBELL, H. G.; DUDEK, R. A.; SMITH, M. L. A heuristic algorithm for the n-job, m machine sequencing problem. **Management Science**, v. 16, p. 630-637, 1970.

CHEN, C.; NEPPALLI, R. V.; ALJABER, N. Genetic algorithms applied to the continuous flowshop problem. **Computers Industrial Engineers**, v. 30, p. 919-929, 1996.

DANNENBRING, D. G. An evaluation of flow-shop sequencing heuristics. **Management Science**, v. 23, p. 1174-1182, 1977.

DAVOUD POUR, H. A new heuristic for the n-job, m-machine flow-shop problem. **Production Planning and Control**, v. 12, n. 7, p. 648-653, 2001.

DEMAN, J. M. V.; BAKER, K. R. Minimizing mean flowtime in the flowshop with no intermediate queues. **AIIE Transactions**, v. 6, p. 28-34, 1974.

FRAMINAN, J. M.; LEISTEN, R.; RAJENDRAN, C. Different initial sequences for the heuristic of Nawaz, Ensore and Ham to minimize makespan, idletime or flowtime in the static permutation flowshop sequencing problem. **International Journal of Production Research**, v. 41, n. 1, p. 121-148, 2003.

GANGADHARAN, R.; RAJENDRAN, C. Heuristic algorithms for scheduling in the no-wait flowshop. **International Journal of Production Economics**, v. 32, p. 285-290, 1993.

GUPTA, J. N. D. A functional heuristic algorithm for the flow-shop scheduling problem. **Operational Research Quarterly**, v. 22, p. 39-47, 1971.

HALL, N. G.; SRISKANDARAJAH, C. A survey of machine Scheduling-problems with blocking and no-wait in process. **Operations Research**, v. 44, p. 510-525, 1996.

JOHNSON, S. M. Optimal two-and three-stage production schedules with setup times included. **Naval Research Logistics Quarterly**, v. 1, p. 61-38, 1954.

- KALCZYNSKI, P. J.; KAMBUROWSKI, J. On the NEH heuristic for minimizing the makespan in permutation flowshops. **Omega**, v. 35, p. 53-60, 2007.
- KING, R.; SPACHIS, A. S. Heuristics for flowshop scheduling. **International Journal of Production Research**, v. 18, p. 343-357, 1980.
- KOULAMAS, C. A new constructive heuristic for the flowshop scheduling problem. **European Journal of Operational Research**, v. 105, p. 66-71, 1998.
- LAWLER, E. L.; LENSTRA, J. K.; RINNOOY KAN, A. H. G. E SHMOYS, D. B. **Sequencing and scheduling: Algorithms and complexity**. In: GRAVES, S. C.; Rinnooy Kan, A. H. G, 1993.
- NAGANO, M.; JANUARIO, J. Evolutionary heuristic for makespan minimization in no-idle flowshop production systems. **Acta Scientiarum**, v. 35, n. 2, p. 271-278, 2013.
- NAGANO, M. S.; MOCCELLIN, J. V. A high quality solution constructive for flowshop sequencing. **Journal of the Operational Research Society**, v. 53, p. 1374-1379, 2002.
- NAGANO, M. S.; SCARDOELLI, L. Y.; MOCCELLIN, J. V. Programação da Produção em Sistema No-Wait Flow Shop com Minimização do Tempo Médio de Fluxo. **Gestão da Produção, Operações e Sistemas**, v. 3, n. 2, p. 91-99, 2008.
- NARAIN, L.; BAGGA, P. C. Flowshop/no-idle scheduling to minimise the mean flowtime. **Anziam Journal**, v. 47, p. 265-272, 2005.
- NAWAZ, M.; ENSCORE JR., E. E.; HAM, I. A heuristic algorithm for the m-machine, njob flow-shop sequencing problem. **Omega**, v. 11, p. 91-95, 1983.
- 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, p. 101-107, 1965.
- PAPADIMITRIOU, C.; KANELLAKIS, P. C. Flowshop scheduling with limited temporary storage. **Journal of the Association for Computing Machinery**, v. 27, p. 533-549, 1980.
- RAJENDRAN, C.; CHAUDHURI, D. Heuristic algorithms for continuous flow-shop problem. **Naval Research Logistics**, v. 37, p. 695-705, 1990.

- RAJENDRAN, C. A no-wait flowshop scheduling heuristic to minimize makespan. **Journal of the Operational Research Society**, v. 45, p. 472-478, 1994.
- RAAYMAKERS, W.; HOOGEVEEN, J. Scheduling multipurpose batch process industries with no-wait restrictions by simulated annealing. **European Journal of Operational Research**, v. 126, p. 131-151, 2000.
- REDDI, S. S.; RAMAMOORTHY, C. V. On the flowshop sequencing problems with no wait inprocess. **Operational Research Quarterly**, v. 23, p. 323-331, 1972.
- SAADANI, N. H.; GUINET, A.; MOALLA, M. A traveling salesman approach to solve the F/no-idle/Cmax problem. **European Journal of Operational Research**, v. 161, p. 11-20, 2005.
- SAPKAL S.; LAHA. D. A heuristic for no-wait flow shop scheduling. **The International Journal of Advanced Manufacturing Technology**, v. 68, pp. 1327-1338, 2013.
- STORN, R.; PRICE, K. Differential evolution – A simple and efficient adaptive scheme for global optimization over continuous spaces. **Journal of Global Optimization**, v. 11, n. 4, p. 341-359, 1997.
- TAILLARD, E. Some efficient heuristic methods for the flowshop sequencing problem. **European Journal of Operational Research**, v. 47, p. 65-74, 1990.
- TANAEV, V. S.; SOTSKOV, Y. N.; STRUSEVITCH, V. A. Scheduling Theory Multi-stage System. **Kluwer Academic Publishers**, ISBN 0-7923-2854-X., 1994.
- TASGETIREN, M.; PAN, Q.; SUGANTHAN, P.; LIANG, Y. **A variable parameter search based differential evolution algorithm for real-parameter continuous function optimization**. Proceedings of the 2009 congress on evolutionary computation (CEC 2009), p. 1247-1255. Disponível em: <<http://ieeexplore.ieee.org/document/4983088/?reload=true>>. Acesso em: 02 maio 2016.
- TASGETIREN, M.; PAN, Q.; SUGANTHAN, P.; CHUA, T. A differential evolution algorithm for the no-idle flowshop scheduling problem with total tardiness criterion. **International Journal of Production Research**, v. 1, p. 1-18, 2010.
- WISMER, D. A. Solution of the flowshop sequencing problem with no intermediate queues. **Operations Research**, v. 20, p. 689-697, 1972.
- WOOLLAM, C. R. Flowshop with No Idle Machine Time Allowed. **Computer & Industrial Engineering**, v. 10, p. 69-76, 1986.