

Avaliação de heurísticas para o problema flowshop com setup separado e independente da sequência

RESUMO

Amanda da Silva Correia

amandacorreia@alunos.utfpr.edu.br

Universidade Tecnológica Federal do Paraná (UTFPR), Ponta Grossa, PR, Brasil.

Fábio José Ceron Branco

fbranco@utfpr.edu.br

Universidade Tecnológica Federal do Paraná (UTFPR), Ponta Grossa, PR, Brasil.

O presente trabalho caracteriza-se pelo estudo comparativo de desempenho das heurísticas LPT, SPT e NEH adaptadas para a resolução de problemas de sequenciamento de tarefas *flowshop*, com tempo de setup separado e independente da sequência. Para realizar a análise de desempenho é considerado não só o tempo de processamento de cada tarefa para cada máquina disponível, mas também o tempo necessário para realizar a configuração das máquinas utilizadas. Com o intuito de identificar o melhor método de sequenciamento a ser utilizado, analisou-se em um banco de dados o desempenho dos métodos baseado nos resultados de *makespan*, de acordo com ferramentas estatísticas. Após os cálculos realizados, observou-se na primeira análise, dos métodos de ordenação SPT e LPT, o melhor desempenho do método heurístico LPT com a consideração do tempo de setup das tarefas somado aos tempos de processamento, enquanto que para o método construtivo NEH, a ordenação LPT sem considerar o tempo de setup para definição do sequenciamento das tarefas se mostrou mais efetivo. Concluiu-se também a melhor performance dos métodos construtivos NEH quando comparado aos métodos de ordenação SPT e LPT.

PALAVRAS-CHAVE: Scheduling. Setup Independente. *Flowshop*. Heurísticas. *Makespan*.

INTRODUÇÃO

O intento do planejamento e controle da produção (PCP) é conseguir que a produção seja realizada de maneira eficaz, bem como produzir os produtos e/ou realizar serviços da maneira que deve e, conseguir desempenhar esse planejamento e controle de maneira que as atividades sejam cumpridas a cada projeto com os recursos disponíveis.

Além disso, o planejamento e controle da produção possui um papel decisivo para as futuras ações e decisões a serem tomadas pela empresa, e que pode auxiliar no momento de competição no ambiente empresarial, que é resultante da constante e crescente oferta de materiais e de produtos em toda e qualquer parte do mundo.

Como primeiro nível operacional de curto prazo do PCP, define-se a programação da produção (em inglês, *scheduling*), que é encarregada de definir não só a quantidade como também quando comprar, fabricar ou montar cada item necessário para compor os produtos que são fabricados. Outra função que é abrangida neste nível é o sequenciamento de tarefas.

Dessa maneira, o *scheduling* tem como intuito alocar as tarefas, bem como máquinas e ferramentas, aos processos, sendo que estes estão sujeitos a restrições tais como tempo de liberação, datas de entrega, liberação, vencimento, e carga da máquina.

A alocação de tarefas envolve a resolução de objetivos, que geralmente se tornam contraditórios, o que dificulta a satisfação com a resolução de um problema de programação de tarefas. O objetivo é agendar as operações em cada máquina de forma a minimizar ou maximizar alguma medida de desempenho, sendo os mais utilizados, o cumprimento das datas de entrega, a minimização de atrasos, a ociosidade, do tempo total de produção, entre outros.

O sequenciamento de tarefas pode ainda se fazer necessário ou não nos sistemas de produção existentes, entre os mais conhecidos, o *jobshop*, *openshop* e *flowshop*. Esses sistemas regem por uma definição de ordem de execução das tarefas e quantidade de estágios que as tarefas precisam percorrer. Quanto ao *flowshop*, foco deste trabalho, é um problema definido com um mesmo fluxo que todas as tarefas precisam passar nas m máquinas, e se considerado como *flowshop* permutacional, existe $n!$ sequências possíveis.

Problemas de programação de *flowshop* com tempos de configurações (setup) são comuns na indústria. Dentro de situações de sequenciamento de tarefas, pode-se, ainda, analisar o tempo de setup necessário para cada tarefa em cada máquina. O setup envolve operações que podem ser ditas como não ligadas diretamente ao processo produtivo, mas que se tornam necessárias, já que incluem tanto limpeza, como substituição de ferramentas de processamento, transporte de operações de uma máquina a outra, liberação de peças para as máquinas e entre outros.

Esse tempo não é levado em consideração em sistemas de produção em que o setup é consideravelmente pequeno quando comparado ao tempo de processamento, entretanto em diversos casos dentro de um contexto industrial, não é ideal que essa informação seja simplesmente ignorada ou incluída ao

tempo de processamento das tarefas como um desvio médio. Nesta situação, trabalha-se com o setup separado.

Existem algumas classificações possíveis para esses tempos de configuração, como setup dependente e independente, quando o tempo de setup depende ou não da tarefa sucessora, bem como, e principalmente, da que está sendo processada no momento. E também, setup antecipado ou não antecipado, determinando o tempo em que pode ser realizado, se a configuração pode ser realizada durante o tempo livre da máquina mesmo que a tarefa ainda esteja em processo na máquina anterior ou não.

Os resultados de um sequenciamento de tarefas inadequado têm influência não só na motivação, como na atitude da força de trabalho e na produtividade geral, e conclui-se a necessidade de uma apropriada alocação de tarefas dentro de uma empresa. Isso torna um fator essencial não só pela demanda de trabalho e/ou pelo respeito às restrições de escalonamento das atividades de cada processo, mas também por influenciar nos custos de produção e no aproveitamento dos recursos disponíveis.

REFERENCIAL TEÓRICO

FLOWSHOP

O sistema de produção *flowshop* é uma área de pesquisa importante no ramo de produção, não só teoricamente, no campo de estudo, mas também devido a sua aplicação em problemas práticos na indústria (YENISEY; YAGMAHAN, 2014).

Como contextualização na parte histórica, Cheng, Gupta e Wang (2009), Vallada, Ruiz e Framinan (2015), e Riahi et al. (2017) enfatizam que o pioneiro na área de pesquisa da permutação *flowshop* foi Johnson (1954), para o problema de duas máquinas com o critério de minimização do tempo total de processamento (*makespan*).

Xuet al. (2017) trazem como conhecimento o fato de que muitos processos de produção, como na indústria siderúrgica, petroquímica e mecânica podem ser modelados com o problema de programação de *flowshop* permutacional.

Um problema quando considerado *flowshop* determina que exista um conjunto de máquinas que estão ordenadas, assim sendo, a primeira tarefa é executada na primeira máquina, posteriormente na segunda máquina e assim até a máquina m (GAREY; JOHNSON; SETHI, 1976).

Por conseguinte, Taillard (1990) e Buzzo e Moccellini (2000) descrevem um problema de programação *flowshop* como um problema que define um mesmo fluxo de execução para todas as tarefas nas m máquinas disponíveis, e um problema se torna permutacional quando a determinação desta sequência se encontra dentro as $n!$ sequências possíveis das tarefas. Devido a sua natureza de problema combinatorial, um problema de *flowshop* é considerado como NP-completo.

Se dentro de um processo produtivo de dez tarefas, por exemplo, 3.628.800 soluções de sequenciamento existem no espaço de soluções, e dentre elas, uma pode ser considerada como um ótima.

Ainda na mesma linha de raciocínio de Taillard (1990), pode-se elencar sete hipóteses consideradas usuais em questão de programação de tarefas *flowshop* com tempos de setup agregados aos tempos de processamento e fabricação de apenas uma unidade em cada tarefa ou lotes que não podem ser fracionados, descritas a seguir:

- a) As máquinas estão disponíveis continuamente, sem interrupções;
- b) Em cada máquina apenas uma tarefa pode ser processada por vez;
- c) As tarefas só podem ser processadas por uma máquina de cada vez;
- d) Existe predeterminação dos tempos de processamento das tarefas em todas as máquinas e estes tempos são fixos;
- e) Todas as tarefas são liberadas para processamento na primeira máquina no mesmo instante, portanto, a partir desse momento qualquer uma pode ser programada e processada;
- f) Cada operação possui um tempo de preparação para cada máquina, e estes estão incluídos em seus tempos de processamento e não dependem da sequência das operações nas máquinas;
- g) Não deve ocorrer interrupções das tarefas quando operadas nas diversas máquinas.

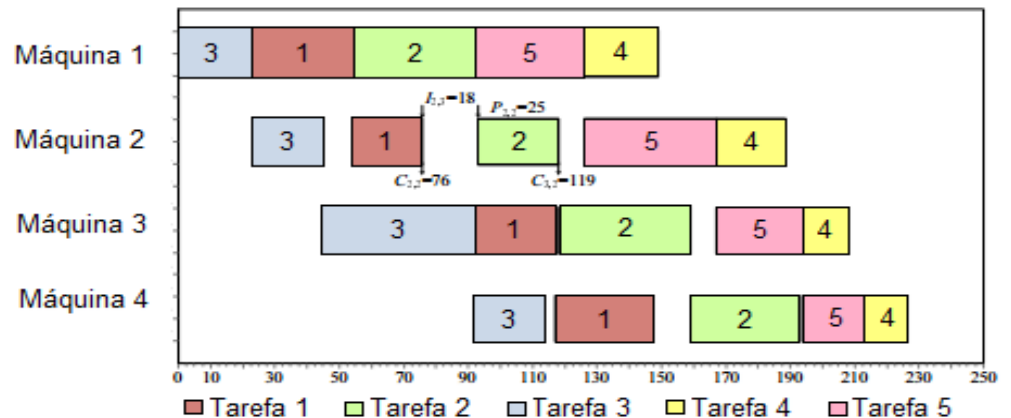
Um exemplo de Gráfico de Gantt para a programação *flowshop* pode ser visto na Figura 1. Neste caso, 5 tarefas são representadas em um fluxo de 4 máquinas, com os tempos de processamento descritos no Quadro 1.

Quadro 1 - Tempos de processamento para o exemplo de um problema de programação *Flowshop*

		Tarefas (n)				
F	Máquinas (m)	1	2	3	4	5
o	1	31	39	23	23	33
n	2	22	25	22	22	41
t	3	25	41	47	14	27
e	4	30	34	22	13	19

Fonte: Adaptado de Ruiz, Vallada e Fernández-martínez (2009)

Figura 1 – Programação de tarefas *flowshop* com 5 tarefas e 4 máquinas



Fonte: Adaptado de Ruiz, Vallada e Fernández-martínez (2009)

Ruiz, Vallada e Fernández-martínez (2009) ilustram pela Figura 1 um problema *flowshop* com base nos tempos do Quadro 1, informando também a variável $C_{i,j}$, que revela o tempo total de processamento da tarefa na posição i na máquina j , o $I_{i,j}$, tempo de espera entre a execução da tarefa na posição i e da tarefa na posição j e $P_{i,j}$, indicando o tempo de processamento da tarefa i na máquina j .

Assim como qualquer outro tipo de sequenciamento, a permutação *flowshop* tem como intuito encontrar uma ordenação que atenda o objetivo da linha de produção, de minimização ou maximização, como declaram Riahiet al. (2017).

Segundo Maccarthy e Liu (1993), a programação compreende que a estrutura de planejamento e/ou o horizonte temporal geralmente não é totalmente considerado, já que num ambiente dinâmico, existem situações de falta de mão de obra, matéria prima, falha em máquina e outros.

Arroyo e Pereira (2010) e Riahiet al. (2017) ressaltam que o *flowshop* que se concentra em um único objetivo despertou a atenção de muitos pesquisadores até a década de 1980, uma vez que há ajuda de métodos exatos e metaheurísticas.

Branco (2011) conclui que nas últimas décadas, um extenso esforço de pesquisa tem sido dedicado aos problemas de *flowshop* permutacional. Algumas técnicas de programação matemática, tais como programação linear inteiro, técnicas de enumeração do tipo *branch-and-bound* têm sido empregadas para se obter a solução ótima. Entretanto, tais técnicas não são eficientes em termos computacionais em problemas de médio e grande porte. Assim, muitos métodos heurísticos têm sido propostos, os quais, de maneira geral, podem ser classificados em dois grupos: métodos construtivos e metaheurísticos.

Vários tipos de metaheurísticas foram propostas para resolver problemas de sequenciamento de tarefas *flowshop*, como:

- Algoritmos genéticos (Reeves (1995), Ishibuchi, Yoshida e Murata (2003), Azadehet al. (2015), e Jung, Woo e Kim (2017));

- Busca tabu (Al-Anzi e Allahverdi (2006) e Gao, Chen e Deng (2013)), recozimento simulado (Al-Anzi e Allahverdi (2006));
- Enxame de partículas (Al-Anzi e Allahverdi (2009), Kennedy e Eberhart (1995), Chih et al. (2014), Chih (2015) e Hatami, Ruiz e Andrés-romano (2015));
- Colônia de formigas (Yagmahan e Yenisey (2010));
- Algoritmos iterados (Geiger (2011)) e outros.

Já para métodos heurísticos construtivos, tem-se como propostas, o método N&M introduzido por Nagano e Moccellini (2002), o algoritmo IG desenvolvido por Ruiz e Stützle (2007), a heurística construtiva NEH (Nawaz, Enscore e Ham (1983)), as baseadas na abordagem de McCormick (Leisten (1990)). A heurística PF_NEH baseada em PF a abordagem foi apresentada e melhorada pelo procedimento NEH pelos autores Liang, Pan e Chen (2010). Tasgetiren et al. (2017) propuseram a heurística PFT_NEH semelhante a contribuição dos autores Ruiz e Stützle (2007).

Algoritmo NEH

A busca por uma solução para o problema *flowshop*, com o intuito de encontrar uma sequência ótima ou quase ótima, produziu técnicas de solução exata e aproximada, concluem Nawaz, Enscore e Ham (1983). Para esses autores, as técnicas exatas resolvem o problema em princípio, mas, na maioria dos casos, o tempo de computação e a memória necessária para acompanhar os cálculos é proibitivo mesmo para problemas pequenos. Por outro lado, algoritmos heurísticos, embora não forneçam necessariamente a solução ideal para o problema, são para a maior parte, uma maneira eficiente e econômica de obter uma solução de alta qualidade para um problema.

Determinados algoritmos heurísticos propõem que as tarefas com maior tempo total do processamento recebam prioridade aos de menos tempo. Com base nessa premissa, o algoritmo NEH foi apresentado por Nawaz, Enscore e Ham (1983) garantiu sequências com boas soluções em comparação com outras heurísticas existentes.

Taillard (1990) também comparou a qualidade e a complexidade das soluções do método construtivo NEH com as heurísticas Palmer, proposta por Palmer (1965), que tem como sugestão um índice de ordem para sequenciar as tarefas em m máquinas, com base nos tempos de processamento de tarefas em máquinas diferentes.

O método Gupta sugerido pelo autor Gupta (1971) é semelhante ao de Palmer, exceto pelo fato de definir o índice de uma maneira ligeiramente diferente. Já o algoritmo CDS, proposto por Campbell, Dudek e Smith (1970), que é uma generalização heurística do algoritmo de Johnson, sendo que este procedimento gera um conjunto de $m-1$ problemas de duas máquinas artificiais a partir do problema original da m -máquina, cada um dos quais é então resolvido pelo algoritmo de Johnson, e comparado também com a heurística RA, sugerida por Dannenbring (1977).

A comparação realizada por Taillard (1990) entre essas heurísticas está representada por meio do Quadro 2. A complexidade inclui o cálculo do

makespan enquanto a qualidade das soluções é dada em porcentagem acima da média dos ótimos valores (para os problemas de 10 tarefas da máquina) ou do *makespan* obtido após 1000 iterações de heurísticas tabus.

Quadro 2 - Comparação entre heurísticas clássicas

	Complexidade	Qualidade					
		500	100	100	100	50	50
Problemas	-	500	100	100	100	50	50
Tarefas	n	9	10	20	20	40	50
Máquina	m	10	10	10	20	10	10
Gupta	$n \log(n) + nm$	13.4	12.8	19.6	18.8	18.9	17.1
Johnson	$n \log(n) + nm$	10.9	11.8	16.7	16.8	17.3	16.3
RA	$n \log(n) + nm$	8.5	9.1	12.5	13.4	13.5	11.2
Palmer	$n \log(n) + nm$	8.3	9.0	13.3	12.5	10.9	10.7
CDS	$nm^2 + mn \log(n)$	4.5	5.2	9.7	8.6	9.9	9.3
NEH	n^2m	2.1	2.2	3.9	3.8	2.6	2.1

Fonte: Adaptado de Taillard (1990)

Taillard (1990) conclui, portanto, que o algoritmo NEH parece ser a melhor heurística polinomial na prática. As heurísticas RA ou Palmer também podem ser úteis quando tempos de computação curtos são necessários. Framinan, Leisten e Rajendran (2003) também confirmaram a eficácia da regra de prioridade da heurística NEH ao testá-lo com 177 sequências iniciais diferentes com o critério *makespan*, *idletime* e *flowtime*, respectivamente.

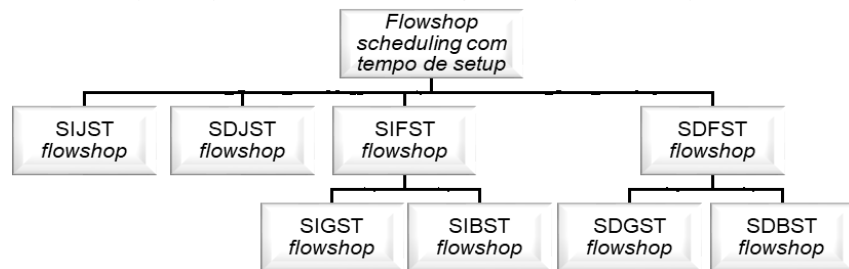
Nawaz, Enscore e Ham (1983) descreveram os passos necessários para realizá-lo:

- a) Ordenar as n tarefas em ordem decrescente de tempos de processamento nas máquinas.
- b) Selecionar as duas primeiras tarefas (k=1 e k=2) e programá-las a fim de minimizar o *makespan* parcial.
- c) Para k=3 até n fazer: Inserir a k-ésima tarefa na sequência, entre os n! possíveis, que minimiza a parcial do *makespan*.

Setup

A natureza dos tempos de setup e remoção dos problemas de *flowshop* foi discutida e sua representação em três campos dos variados problemas de sequenciamento do *flowshop* foi descrita, bem como fornecida uma hierarquia de complexidade de scheduling pelos autores Cheng, Gupta e Wang (2009). A Figura 2 ilustra estas possibilidades.

Figura 2 – Classificação de problema de scheduling com tempo de setup



Fonte: Adaptado de Cheng, Gupta e Wang (2009)

- *Flowshop* com tempos de preparação de tarefas independentes de seqüência (*SIJST flowshop*);

- *Flowshop* com tempos de configuração de tarefas dependentes da seqüência (*SDJST flowshop*);

-*Flowshop* com tempos de configuração da família de seqüência independente (*SIFST flowshop*);

-*Flowshop* com tempos de configuração de grupos independentes de seqüências (*SIGST flowshop*);

-*Flowshop* com tempos de configuração de lote independentes de seqüência (*SIBST flowshop*);

-*Flowshop* com tempos de configuração de famílias dependentes da seqüência (*SDFST flowshop*);

-*Flowshop* com tempo de configuração de grupo dependente de seqüência (*SDGST flowshop*);

-*Flowshop* com tempo de configuração de lote dependente da seqüência (*SDBST flowshop*).

Cheng, Gupta e Wang (2009) dão continuidade ao assunto ao explicitar que os problemas de *flowshop* com os tempos de setup surgem em muitas situações práticas.

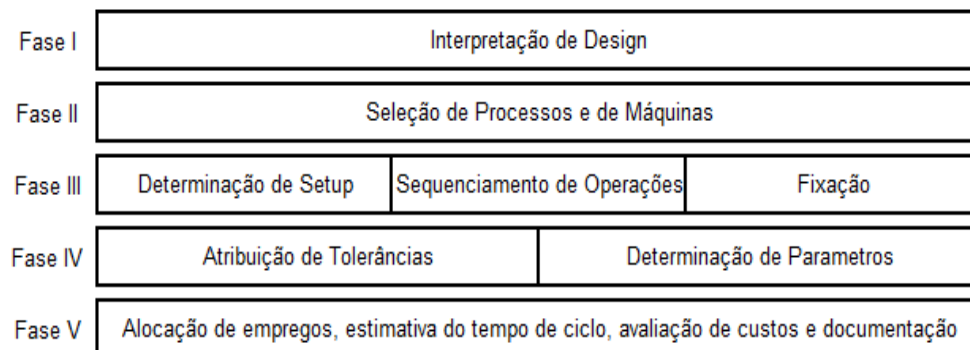
Assumir que os tempos de processamento, bem como outros parâmetros de um conjunto de operações, são valores constantes conhecidos antecipadamente é inadequado, visto que para modelar muitos processos industriais modernos uma tarefa executada sob a mesma ou quase a mesma condição possui um tempo de processamento variável. Devido a situações como perda de eficiência da máquina durante o processo, eleva-se o tempo de processamento total. Na literatura, o fenômeno é chamado de deterioração do tempo de processamento da tarefa (CHENG; SUN; HE, 2007).

O planejamento do setup é um assunto que deve ser estudado pela organização, pois pode interferir diretamente nos tempos e na programação.

Xu, Huang e Rong (2007) destacam que, de acordo com as indústrias e seu segmento de produção, o escopo do planejamento de todo o processo pode variar. Todavia, a Figura 3 ilustra um esquema básico da divisão que ocorre dentro do planejamento de processo. Cada fase que está disposta pode vir a

tomar decisões de maneira independente, emite informação para as fases inferiores e recebe resultado das fases superiores.

Figura 3 – Decomposição do Planejamento de Processo



Fonte: Adaptação realizada da figura apresentada por XU; HUANG; RONG (2007)

O termo - planejamento de instalação - amplamente utilizado na literatura de pesquisa de planejamento de processo refere-se à Fase III no esquema ilustrado. No entanto, não há um padrão bem reconhecido quanto ao seu escopo.

Na concepção dos autores Ciavotta, Minella e Ruiz (2013), o setup envolve operações que não estão ligadas diretamente ao processo produtivo, ou seja, são operações não produtivas, mas que necessitam ser realizadas nas máquinas. São atividades que incluem, mas não se limitam a limpeza, substituição de ferramentas de processamento, transporte de tarefas de uma máquina para a próxima máquina, fixação e liberação de peças para as máquinas. Normalmente, o tempo de setup não é levado em consideração, mas na maioria do contexto industrial, não é possível nem ideal que esta informação seja ignorada.

Ciavotta, Minella e Ruiz (2013) ainda contribuem ao declarar que o setup pode ser classificado em duas categorias principais, sendo setup independente e o setup dependente. Dentro da primeira classificação se encaixa o tempo de setup que depende apenas da operação que está a ser processada e pode ser apenas considerado junto com o tempo de processamento da tarefa, enquanto que na segunda classificação, o tempo de setup depende não só da tarefa sucessora, mas principalmente da que está sendo processada no momento, tornando um processo mais complexo e, portanto, a possibilidade de descrever diversos cenários operacionais.

Além disso, as configurações podem ser antecipadas ou não antecipadas. No primeiro caso, as configurações podem ser executadas assim que a máquina estiver livre e antes da próxima tarefa na sequência ser carregado, conforme afirmam Ciavotta, Minella e Ruiz (2013).

Huang, Cai e Zhang (2010) comentam que são poucas as pesquisas que consideram o tempo de configuração da máquina (setup), e quanto esse tempo é considerado, ignoram-se a disponibilidade dos trabalhadores qualificados para a realização desta atividade. Entretanto, esta informação se torna importante já que os operadores precisam ser atribuídos de uma máquina para outra, e diante dessas situações, o planejamento da produção se tornou uma atividade complexa, especialmente quando os setups das operações são dependentes da

seqüência de tarefas, além dos variados tipos de recursos que devem ser atribuídos simultaneamente.

Para demonstrar a importância do tempo de setup e o porquê de levá-lo em consideração na decisão do sequenciamento de tarefas, Huang, Cai e Zhang (2010) desenvolveram um exemplo com quatro tarefas, duas máquinas e um servidor disponível, representado no Quadro 3.

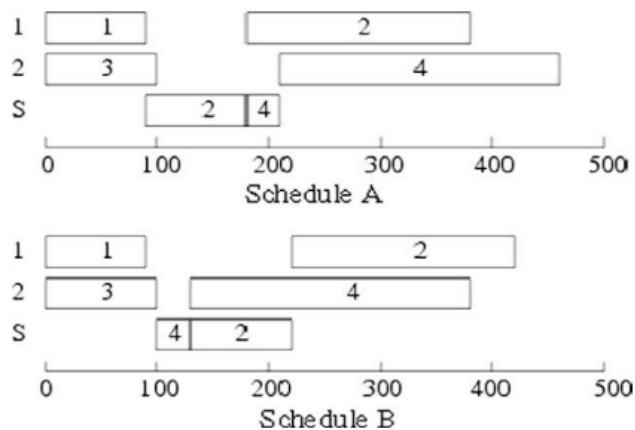
Quadro 3 - Exemplo de sequenciamento de tarefas considerando o tempo de setup

Tarefa	Máquina	Tempo de Processamento (min)
1	1	90
2	1	200
3	2	100
4	2	250

Fonte: Adaptação do exemplo proposto por Huang, Cai e Zhang (2010)

Sendo que os valores do *setup* dependendo da seqüência (*S*) são: $S_{12}=90$, $S_{34}=30$, $S_{21}=500$, $S_{43}=500$. Obviamente, seria melhor processar a tarefa 1 antes da tarefa 2 na máquina 1. Se o servidor executar a operação de configuração para tarefa 2 imediatamente após a conclusão da tarefa 1 e, em seguida, executar a operação de configuração para a tarefa 4 (agendamento A na Figura 4), o tempo total da programação do sistema é 460 unidades. No entanto, se o servidor tiver a máquina 1 aguardar e executar a operação de configuração para a tarefa 4 primeiro (agendamento B na Figura 4), a configuração do sistema é de 420 unidades.

Figura 4 - Dois diferentes sequenciamentos com o tempo de setup



Fonte: Adaptado de Huang, Cai e Zhang (2010)

Pode-se concluir, com este simples exemplo que o desempenho do sistema é afetado pela eficiência da seqüência na máquina e pela eficácia da programação de configuração das máquinas no servidor e, portanto, essas informações precisam ser explicitamente consideradas quando o problema é modelado.

PROCEDIMENTOS METODOLÓGICOS

OBJETO DE ESTUDO

A metodologia utilizada para que o estudo referente ao presente trabalho pudesse ser realizado se enquadra no método científico indutivo, definido como o levantamento de dados para concluir uma verdade ou o que se julga verdade. Sendo assim, a observação dos fenômenos acontece para que de maneira subsequente a descoberta da relação entre esses fenômenos ocorra e por fim, generalizar esta correspondência.

Além disso, a classificação quanto a natureza da pesquisa realizada se enquadra na definição de pesquisa aplicada. Pois contribui para o conhecimento já existente na literatura e apenas realiza o acúmulo de informações.

A pesquisa aplicada é fundamentalmente motivada pela necessidade de resolver problemas concretos, mais imediatos, ou não. Tem, portanto, finalidade prática, ao contrário da pesquisa pura, motivada basicamente pela curiosidade intelectual do pesquisador e situada sobretudo no nível da especulação (Vergara, 1998).

Visto que os dados utilizados no estudo são numéricos e que por meio deles busca-se confirmação das hipóteses encontradas na literatura, pode-se concluir que a pesquisa realizada possui abordagem quantitativa.

O objetivo do presente trabalho pode ser dito como exploratório, já que tem como finalidade proporcionar à literatura uma visão geral do assunto tratado, por meio de levantamento bibliográfico e documental.

PROCEDIMENTOS DE COLETA DE DADOS

O presente trabalho é baseado em um cenário fictício, descrevendo um problema de programação *flowshop* utilizando oito mil dados. Com o intuito de tornar o estudo confiável, essa amostra é resultado de combinações de máquinas (m) e tarefas (n), sendo que para cada variável, estipulou-se determinada variação de quantidade. A quantidade inicial de máquinas foi de cinco, com variação de 5 em 5 máquinas, por 4 vezes, enquanto que para as tarefas a quantidade inicial foi de 20, variando de 20 em 20, por 5 vezes. Ou seja, de forma geral, trabalhou-se com as seguintes quantidades das variáveis em questão:

$m = 5, 10, 15$ e 20 ;

$n = 20, 40, 60, 80$ e 100 .

Possibilitando, assim, vinte classes geradas da combinação de $m \times n$ para análise do estudo, equivalentes a $5 \times 20, 10 \times 20, 15 \times 20, 20 \times 20, 5 \times 40, 10 \times 40, 15 \times 40, 20 \times 40, 5 \times 60, 10 \times 60, 15 \times 60, 20 \times 60, 5 \times 80, 10 \times 80, 15 \times 80, 20 \times 80, 5 \times 100, 10 \times 100, 15 \times 100$ e 20×100 .

Para toda e qualquer combinação de máquinas e tarefas, os tempos de processamento foram gerados com intervalos de variação entre 1 e 100 unidades de tempo, sendo que para cada possível combinação, um total de 100 problemas diferentes foram armazenados, com o total de duas mil instâncias.

Enquanto que para os tempos de setup, trabalhou-se com três grupos de diferentes intervalos de variação dos tempos, o primeiro possui intervalo de variação entre 1 e 25 unidades de tempo, o segundo entre 1 e 50 unidades de tempo e por fim, o terceiro trabalha com tempos intervalados entre 1 e 100 unidades de tempo. Para cada grupo gerou-se, assim como para os tempos de processamento, 100 problemas diferentes para cada combinação de máquinas e tarefas, totalizando, para os três grupos utilizados, seis mil instâncias.

PROCEDIMENTOS DE ANÁLISE DE DADOS

Em sequência do levantamento de dados, estes puderam ser operacionalizados por meio do algoritmo implementado no software DevPascal, utilizando um computador com as seguintes especificações:

- Processador Intel® Core™ i5-3317U CPU @ 1.70 GHz;
- Memória RAM de 6 GB;
- Sistema Operacional de 64 bits, processador com base em x64.

A fim de verificar o comportamento do sistema de produção clássico em virtude da função objetivo de *makespan* para os métodos de ordenação SPT, LPT e NEH, serão utilizadas estatísticas de desempenho apresentados na próxima seção.

A medida de desempenho escolhida foi a de *makespan* por garantir maior confiabilidade nas entregas e uma utilização eficiente da linha produtiva. Em questão dos métodos heurísticos, a seleção do SPT e LPT se deve pela complexidade de um problema *flowshop* com setup separado, considerado um problema NP-Completo, além disso, apesar da simplicidade dos métodos de ordenação, colaboram com um satisfatório desempenho nos resultados para atender a FO de minimização de *makespan*.

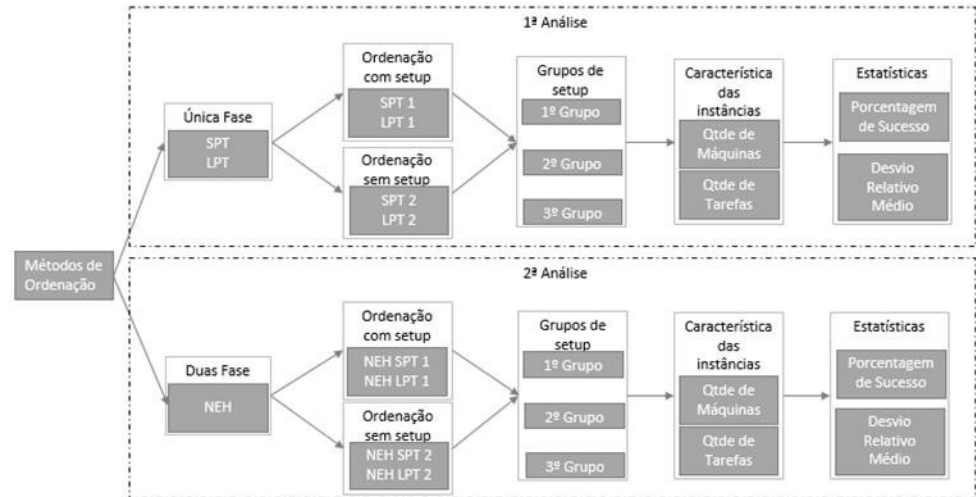
Enquanto que o método NEH está sendo analisado por ser um método construtivo de bastante referência na criação de novas heurísticas que utilizam como parâmetro as ordenações SPT e/ou LPT. Como método construtivo, possui duas fases, a primeira se baseia na ordenação dos métodos de sequenciamento SPT ou LPT e na segunda fase o NEH avança construindo a sequência à medida que a ordem das tarefas vai sendo alterada com base nos resultados obtidos da FO. Por explorar novas sequências além da ordenação inicial SPT ou LPT, este método garante, portanto, o encontro de soluções de alta qualidade.

Ademais, vale salientar também que os tempos de setup são separados dos tempos de processamento e podem ser antecipados, ou seja, a medida que determinada máquina esteja livre, a configuração para uma tarefa *n*, mesmo que esta ainda esteja sendo executada em outra máquina, pode ser realizada. Como a própria definição deste tipo de setup sugere, ocorre a antecipação da configuração, o que possibilita redução do tempo utilizado no processo produtivo.

A análise foi realizada utilizando critérios que envolvem a quantidade de fases dos métodos heurísticos, utilização, ou não, do tempo de setup na ordenação, a combinação com os grupos de tempos de setup definidos

anteriormente e a caracterização das instâncias, para serem analisado com o auxílio das estatísticas de desempenho, como ilustrado na Figura 5.

Figura 5 - Fluxograma da análise de resultados



Fonte: Autoria Própria (2019)

Como visualizado na Figura 5, inicialmente foram realizadas duas análises separadamente, seguindo o critério de configuração dos métodos heurísticos, ou seja, os métodos SPT e LPT que possuem apenas uma fase foram analisados primeiramente, e em um segundo momento apenas o método construtivo NEH foi observado por utilizar duas fases na ordenação das tarefas.

Em cada análise utilizou-se a ordenação inicial crescente (SPT) e decrescente (LPT) baseada apenas nos tempos de processamento e baseada nos tempos de processamento e tempos de setup. Cada método e sua variação foi analisado, em diferentes momentos, com cada grupo de setup que se refere ao possível intervalo de variação dos tempos. Por fim, os resultados obtidos foram agrupados de acordo com a caracterização das instâncias, primeiramente agrupou-se instâncias com a mesma quantidade de máquinas e em seguida com a mesma quantidade de tarefas para que então os resultados fossem observados por meio das estatísticas de desempenho. A análise, para cada tipo de sequenciamento de tarefas, é baseada nas seguintes estatísticas: porcentagem de sucesso (PS) e desvio relativo médio (DRM).

O conjunto de resultados dessas estatísticas foram visualizados de forma gráfica, para melhor entendimento da correlação e comparação entre os métodos heurísticos utilizados, com o auxílio da ferramenta Excel 2013.

A Porcentagem de Sucesso é definida pela quantidade de vezes que um método se mostrou mais efetivo, dividido pela quantidade total dos problemas analisados. Para determinar o sucesso de um determinado método, escolhe-se o menor valor obtido da FO, visto o caso de uma função objetivo focada em minimização de critério, e todos os resultados são comparados com este valor. Se o valor analisado equivale ao menor valor, então é considerado como um sucesso e recebe o valor 1 (um), caso contrário recebe o valor 0 (zero).

Já o Desvio Relativo analisa a variação entre os valores encontrados em um determinado método e o valor de sucesso em âmbito geral. Ou seja, o ideal para

que um método seja considerado bom é que o valor desse desvio seja o menor possível, ou até mesmo zero.

Após obtermos o cálculo e o conhecimento das estatísticas citadas para cada método heurístico, se torna possível a comparação dentre todas as instâncias os problemas analisados e por fim, a conclusão de qual método heurístico se mostra mais eficiente em cada análise realizada e, também, entre a 1ª e a 2ª análise.

RESULTADOS E DISCUSSÕES

ANÁLISE DOS MÉTODOS COM O ALGORITMO NEH

As comparações realizadas foram dos métodos de ordenação SPT, LPT e NEH. Os métodos de ordenação de apenas uma fase, ou seja, o SPT e o LPT foram utilizados de duas maneiras distintas, em um primeiro momento, as ordenações foram baseadas na somatória tanto do tempo de processamento, quanto do tempo de setup das tarefas. Assim como as ordenações também foram definidas com base apenas nos tempos de processamento.

O NEH foi analisado de quatro maneiras diferentes em sua primeira fase, ou seja, observou-se o método NEH baseado na ordenação SPT e LPT e suas variações, ao considerar o tempo de setup para definição da ordem das tarefas, ou levar em consideração apenas os tempos de processamento de cada operação nas máquinas disponíveis. Enquanto que a segunda fase deste método aconteceu sem alterações nas maneiras analisadas, construindo a sequência em cada etapa com base no menor valor obtido na FO.

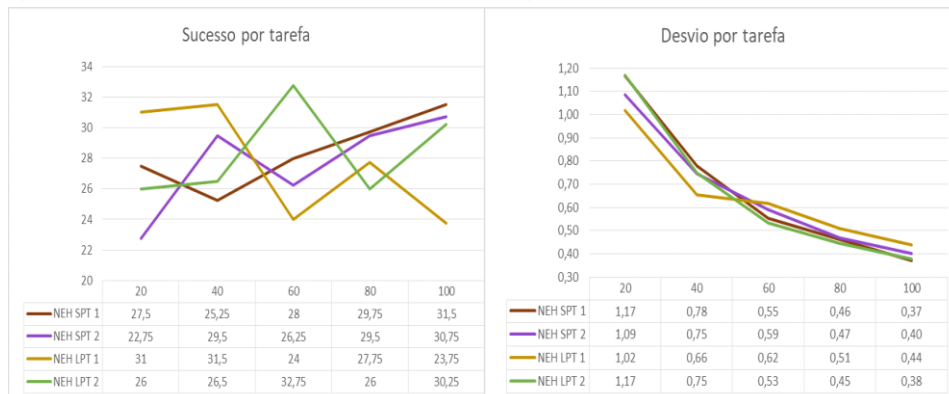
O método construtivo NEH apresenta variações, sendo que o método NEH SPT 1 tem sua ordenação inicial baseada no método SPT que considera o setup no sequenciamento das tarefas, o método NEH SPT 2 baseia-se na ordenação SPT que considera apenas os tempos de processamento no sequenciamento, enquanto que o NEH LPT 1 tem como referência o uso, durante a primeira fase, da ordenação LPT considerando, além dos tempos de processo, o setup de cada operação nas máquinas disponíveis, e por fim, o NEH LPT 2 utiliza como base a ordenação LPT sem considerar o tempo de setup.

Analisou-se por meio da porcentagem de sucesso e do desvio relativo médio, o desempenho dos métodos construtivos para duas categorias, a de instâncias com a mesma quantidade de tarefas, assim como para a de instâncias que possuem a mesma quantidade de máquinas disponíveis.

Desse modo, gerou-se gráficos (Figuras 6 até 11) para analisar cada estatística de acordo com a categoria de tarefas ou máquinas, estes são apresentados em grupos de setup, definidos pelo intervalo de variação de unidades de tempo:

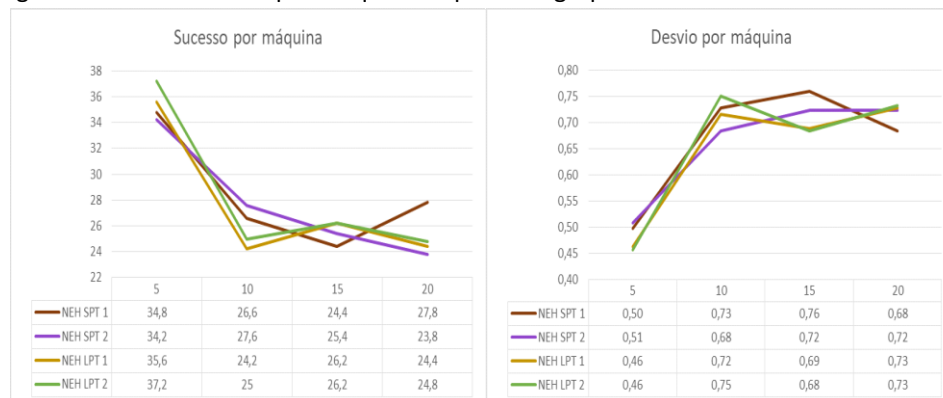
- Primeiro grupo: setup com variação de 1 a 25 unidades de tempo;
- Segundo grupo: setup com variação de 1 a 50 unidades de tempo;
- Terceiro grupo: setup com variação de 1 a 100 unidades de tempo.

Figura 6 - Sucesso e DRM por tarefa do primeiro grupo



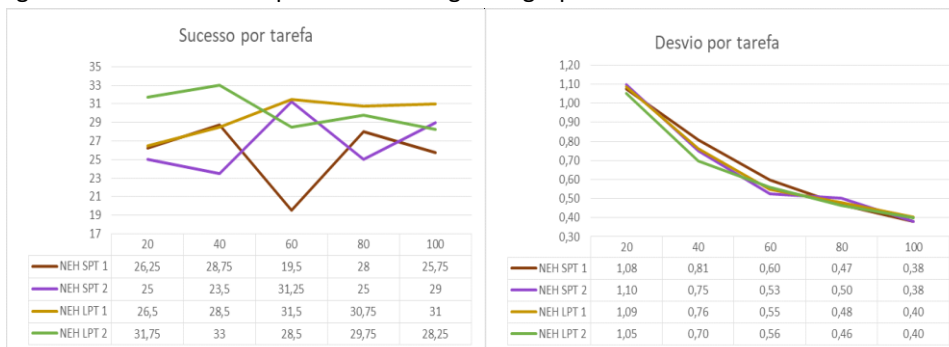
Fonte: Autoria Própria (2019)

Figura 7 - Sucesso e DRM por máquina do primeiro grupo



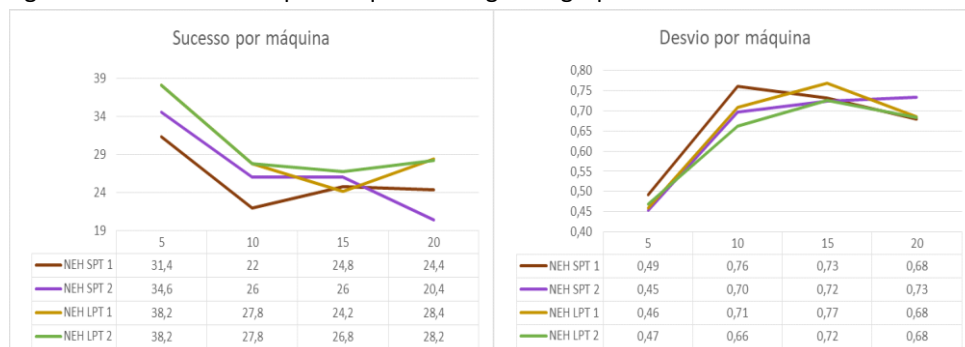
Fonte: Autoria Própria (2019)

Figura 8 - Sucesso e DRM por tarefa do segundo grupo



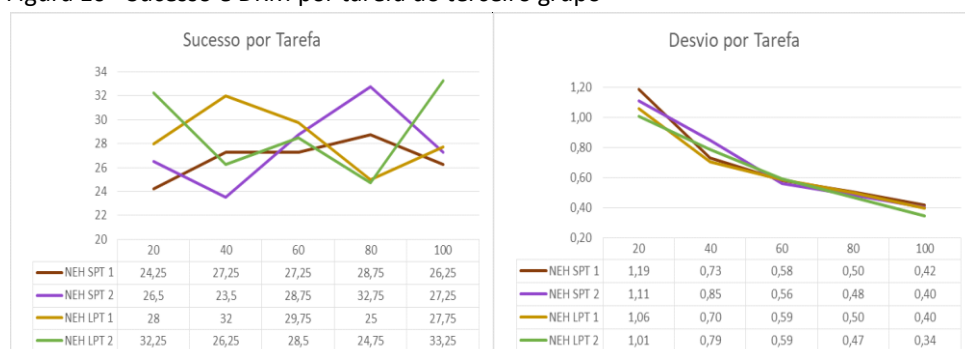
Fonte: Autoria Própria (2019)

Figura 9 - Sucesso e DRM por máquina do segundo grupo



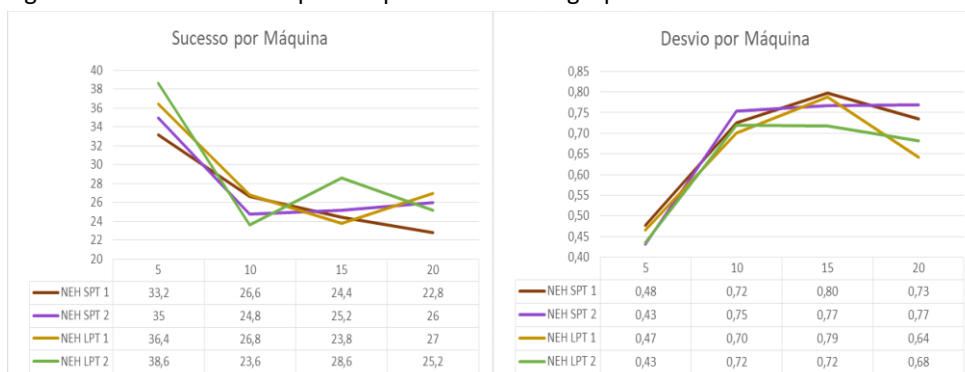
Fonte: Autoria Própria (2019)

Figura 10 - Sucesso e DRM por tarefa do terceiro grupo



Fonte: Autoria Própria (2019)

Figura 11 - Sucesso e DRM por máquina do terceiro grupo



Fonte: Autoria Própria (2019)

Na análise dos Figuras 6 e 7 realizada para o primeiro grupo, os métodos construtivos NEH que baseiam a ordenação inicial no sequenciamento definido pelo LPT são os que em média se mantêm mais próximos do melhor desempenho de sucesso.

Então, tanto para as instâncias com mesma quantidade de tarefas, quanto para instâncias com mesma quantidade de máquinas, que o método NEH SPT 1 obteve maiores resultados na estatística de porcentagem de sucesso, apesar de possuir maior DRM, sendo que o método NEH LPT 1 foi o que se manteve mais próximo dos melhores resultados.

Pode-se tomar como afirmativa, para o segundo grupo, nos Figuras 8 e 9, que os métodos que se baseiam em ordenações que consideram apenas os tempos de processamento para definição do sequenciamento de tarefas, sendo estes o NEH SPT 2 e NEH LPT 2, são os que menos se desviam dos melhores resultados de makespan, tomando como frente a ordenação inicial decrescente.

Ainda assim, de forma geral, o método que se mostrou mais eficiente nas duas estatísticas citadas, com melhores resultados de medida de desempenho bem como com valores mais próximos dos sucessos até mesmo quando não se mostrou mais eficiente, foi o NEH LPT 2, o método construtivo NEH com ordenação inicial baseada no LPT sem considerar os tempos de setup para o sequenciamento de tarefas.

Para o terceiro grupo, na análise dos Figuras 10 e 11, o NEH LPT 2 se mostra, em média, o método com índice de variação mais baixo se comparado com os valores de sucesso.

Fica evidente, também, que os métodos construtivos NEH baseados na ordenação inicial crescente, ou seja, SPT, são os métodos que demonstraram pouca eficiência para resolução de problemas cuja função objetivo é minimização do *makespan*.

De forma geral, no terceiro grupo de tempos de setup os gráficos gerados para as estatísticas de PS e DRM, tanto para as categorias de máquinas quanto para as de tarefas, refletem um mesmo cenário, em que métodos baseados no *Longest Processing Time* sem considerar o tempo de setup para definição do sequenciamento de tarefas são mais eficientes que os demais analisados.

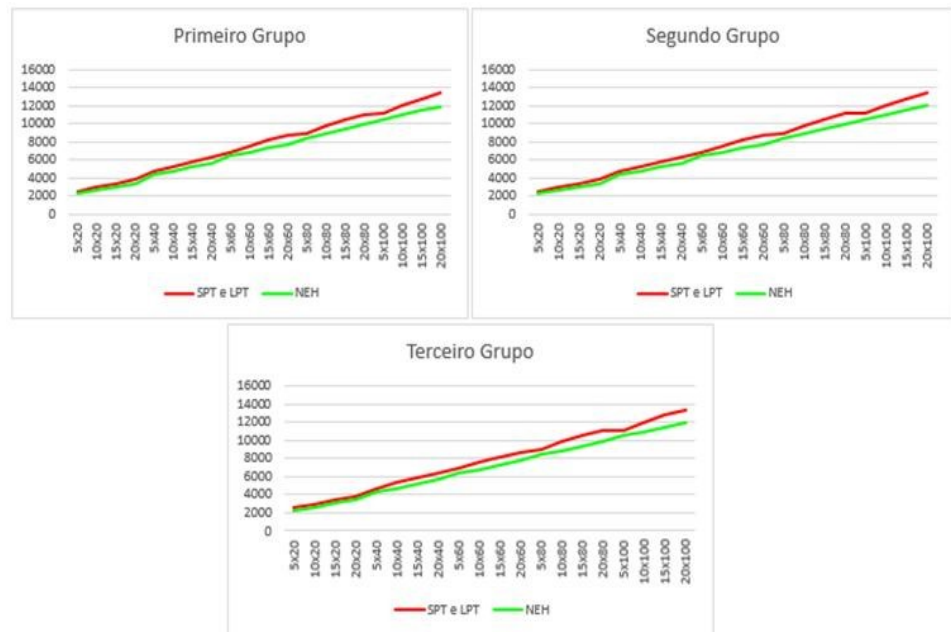
ANÁLISE COMPARATIVA DA EFICIÊNCIA DO MÉTODO NEH

A análise entre os métodos ocorreu por meio da comparação dos resultados obtidos ao executar o programa em linguagem Pascal utilizado para realização do cálculo do *makespan* de acordo com as restrições que se referem ao método de sequenciamento de tarefas inicial, SPT ou LPT, ao intervalo de variação de unidade de tempo de setup e também no uso ou não deste tempo para a definição da ordenação das tarefas.

Assim como foi feito para analisar as estatísticas, a análise entre os tipos de métodos utilizados também foi separada por grupo de tempo de setup, isto é, os menores valores encontrados em cada instância entre os métodos SPT 1, SPT 2, LPT 1 e LPT 2 foram comparados com os menores valores para todas as instâncias entre os métodos NEH SPT 1, NEH SPT 2, NEH LPT 1 e NEH LPT 2 em cada um dos três grupos de tempos de setup definidos anteriormente.

O Figura 12 retrata a média dos valores de *makespan* encontrados nos 100 problemas de cada combinação de mxn, ou seja, a comparação tem como base as 20 combinações determinadas previamente para os métodos de ordenação SPT e LPT e para o método construtivo NEH. A comparação foi realizada para o primeiro grupo, para o segundo grupo e para o terceiro grupo.

Figura 12 - Cores representativas dos métodos analisados



Fonte: Autoria Própria (2019)

Em todos os grupos de intervalos de variação de tempos de setup, o método construtivo NEH, em todas as instâncias analisadas, obteve menores valores de *makespan* quando comparado aos métodos de sequenciamento SPT e LPT.

Os resultados do método construtivo NEH surpreenderam por não definir como relevante a utilização dos tempos de setup na definição inicial do sequenciamento de tarefas, já que o NEH LPT 2 foi o que desempenhou melhor frente aos demais. Entretanto, apesar disso, resultou em valores mais baixos de *makespan* em todas as instâncias observadas em relação aos métodos SPT e LPT, portanto, não se discute a efetividade do método baseado na sua configuração de determinar a ordenação em duas fases, e isso independe das restrições dispostas.

Observa-se também que a medida que a quantidade de tarefas vai aumentando, a distância entre as linhas que representam os métodos são maiores, isto muito se deve pela maior possibilidade de variação entre os tempos tanto de processamento quanto de setup. Assim como a proporção da diferença dos resultados obtidos aumenta quando há um maior intervalo de variação dos tempos de setup, em outras palavras, a diferença média entre os valores obtidos pelos métodos SPT e LPT e pelo NEH no primeiro grupo é de 738,14 unidades de tempo, para o segundo grupo esse valor equivale a 739,82 e no terceiro grupo é de 747,32 unidades de tempo.

CONCLUSÕES

Com o intuito de averiguar e entender as reais necessidades e situações encontradas nas indústrias, o artigo buscou analisar a programação da produção ao considerar não somente os tempos de processamento, bem como os tempos de setup demandados pelas tarefas nas máquinas utilizadas nos processos produtivos.

Para entender o impacto causado por essa informação, geralmente omitida, no sequenciamento de tarefas e conseqüentemente na linha de produção, utilizou-se como base o cálculo do *makespan*, ou também definido como tempo total de um processo.

A análise foi realizada entre os métodos construtivos NEH SPT 1, NEH SPT 2, NEH LPT 1 e NEH LPT 2, e a ordenação inicial baseada na ordem decrescente obteve boa performance na maioria dos grupos, com exceção apenas do grupo de setup com intervalo de variação entre 1 e 25 unidades de tempo.

O método heurístico LPT, independentemente de quais tempos se baseou para realização da ordenação, de fato se mostrou o mais adequado para solucionar problemas de programação da produção com ênfase em minimização do tempo total do processo, como já evidenciado na revisão de literatura.

De formar a possibilitar a adequada análise de um problema de sequenciamento no ambiente produtivo, o exame de métodos de sequenciamento baseados em tempos de processamento e métodos baseados em tempos de processamento e tempo de setup, para os métodos de ordenação de apenas uma fase, evidencia a importância de considerar os tempos de configuração no processo produtivo, visto que no segundo e terceiro grupo de análise deste estudo os métodos de ordenação que englobavam os tempos de setup demonstraram melhor performance, ou seja, atenderam a especificação da função objetivo, a minimização do *makespan*.

Ademais, é observável que a medida que os tempos de setup possuem um intervalo de variação maior, estes impactam cada vez mais em um sequenciamento e em um valor de *makespan* distinto ao método de ordenação baseado na soma apenas dos tempos de processamento, pois apesar de não serem avaliados durante a ordenação de tarefas, ainda são considerados na realização das tarefas propriamente dita.

Diante dessa conclusão, é possível entender e explicar o porquê do método LPT 1 não ter sido o de melhor desempenho no primeiro grupo estudado, visto que o intervalo de variação dos tempos de setup desse grupo, entre 1 a 25 unidades de tempo, é relativamente baixo quando comparado ao intervalo de variação dos tempos de processamento, de 1 a 100 unidades de tempo.

De certa forma, os tempos de setup podem se tornar dispensáveis no auxílio do sequenciamento de tarefas devido ao fato de que o método NEH garante em suas duas fases de construção de sequência, a observação e análise, por meio dos resultados de *makespan* de cada etapa, de uma maior variedade de possibilidade de ordens de tarefas. Sendo assim, o método não fica dependente apenas dos tempos de processamento e tempos de setup, considerando apenas como base inicial um sequenciamento em ordem crescente ou decrescente da soma desses

tempos, e entende que sequências derivadas também podem se mostrar mais eficientes.

Contudo, é visto que não há uma regra de ordenação clara para todos os possíveis casos de solução para problemas *flowshop*, especialmente quando possuem o tempo de setup separado dos tempos de processamento. De certa forma, cada método heurístico desempenha-se de maneira diferente devido a limitações traçadas por suas próprias restrições, assim como por determinadas circunstâncias de cada situação em que é utilizado.

O método construtivo NEH em todas as instâncias para todos os grupos de setup analisados garantiu um melhor desempenho, ou seja, a performance do NEH comparado com os métodos LPT e SPT em todos os momentos alcançou de maneira mais satisfatória a função objetivo. Portanto, a revisão da literatura se faz verdadeira, já que o método construtivo NEH é referência para criação de outros métodos heurísticos e garante em suas duas fases identificar uma melhor sequência de tarefas.

É aconselhável testar o desempenho destas combinações em outras situações de *flowshop* com restrições, para avaliar os métodos.

Heuristics evaluation for flowshop problem with separate setup and independent of sequence

ABSTRACT

This work is characterized by the comparative performance study of LPT, SPT and NEH sequencing rules adapted for the resolution of flowshop sequencing tasks, with separate and sequence independent setup time. To carry out the performance analysis, is considered not only the processing time of each job for each machine available, but also the time needed to perform the configuration of the used machines. In order to identify the best sequencing method to be used, the performance of the methods based on the makespan results was analysed using a computer modelling software, according to statistical tools such as percentage of success and mean relative deviation. After the calculations, we found in the first analysis, of SPT and LPT heuristic methods, the best performance of LPT heuristic method with consideration of the setup time of the tasks added to the processing times. For NEH method, the LPT sequencing rules without considering the setup time for defining the sequencing of the tasks was more effective. It was also concluded the best performance of the constructive methods when compared to SPT and LPT heuristics methods, to result in 100% of problems the lowest makespan values.

PALAVRAS-CHAVE: Scheduling. Sequence independent setup. Flowshop. Heuristics. Makespan.

REFERÊNCIAS

AL-ANZI, F. S.; ALLAHVERDI, A. A Hybrid Tabu Search Heuristic for the Two-Stage Assembly Scheduling Problem. **International Journal Of Operations Research**, [s.l.], v. 3, p.109-119, maio 2006.

AL-ANZI, F. S.; ALLAHVERDI, A. Heuristics for a two-stage assembly flowshop with bicriteria of maximum lateness and makespan. **Computers & Operations Research**, [s.l.], v. 36, n. 9, p.2682-2689, set. 2009. Elsevier BV. **crossref**

ARROYO, J. E. C.; PEREIRA, A. A. S. A GRASP heuristic for the multi-objective permutation flowshop scheduling problem. **The International Journal Of Advanced Manufacturing Technology**, [s.l.], v. 55, n. 5-8, p.741-753, 23 dez. 2010. Springer Nature. **crossref**

AZADEH, A. et al. A multi-objective optimization problem for multi-state series-parallel systems: A two-stage flow-shop manufacturing system. **Reliability Engineering & System Safety**, [s.l.], v. 136, p.62-74, abr. 2015. Elsevier BV. **crossref**

BRANCO, F. J. C.. **Um novo método heurístico construtivo de alto desempenho para o problema no-idleflow shop**. 2011. 112 f. Tese (Doutorado) - Curso de Engenharia de Produção, — Escola de Engenharia de São Carlos da Universidade de São Paulo, São Carlos, 2011. Disponível em: <<http://www.teses.usp.br/teses/disponiveis/18/18140/tde-05092011-094838/pt-br.php>>. Acesso em: 18 abr. 2017.

BUZZO, W. R.; MOCCELLIN, J. V. Programação da produção em sistemas flow shop utilizando um método heurístico híbrido algoritmo genético-simulatedannealing. **Gestão & Produção**, [s.l.], v. 7, n. 3, p.364-377, dez. 2000. FapUNIFESP (SciELO). **crossref**

CHENG, M.; SUN, S.; HE, L.. Flow shop scheduling problems with deteriorating jobs on no-idle dominant machines. **European Journal Of Operational Research**, [s.l.], v. 183, n. 1, p.115-124, nov. 2007. Elsevier BV. **crossref**

CHENG, T.; GUPTA, J. N. D.; WANG, G.A review of flowshop scheduling research with setup times. **Production And Operations Management**, [s.l.], v. 9, n. 3, p.262-282, 5 jan. 2009. Wiley-Blackwell. **crossref**

CHIH, M.; LIN, C.; CHERN, M.; OU, T. Particle swarm optimization with time-varying acceleration coefficients for the multidimensional knapsack problem. **Applied Mathematical Modelling**, [s.l.], v. 38, n. 4, p.1338-1350, fev. 2014. Elsevier BV. **crossref**

CHIH, M. Self-adaptive check and repair operator-based particle swarm optimization for the multidimensional knapsack problem. **Applied Soft Computing**, [s.l.], v. 26, p.378-389, jan. 2015. Elsevier BV. [crossref](#)

CIAVOTTA, M.; MINELLA, G.; RUIZ, R. Multi-objective sequence dependent setup times permutation flowshop: A new algorithm and a comprehensive study. **European Journal Of Operational Research**, [s.l.], v. 227, n. 2, p.301-313, jun. 2013. Elsevier BV. [crossref](#)

FRAMINAN, J. M.; LEISTEN, R.; RAJENDRAN, C.. Different initial sequences for the heuristic of Nawaz, Enscore and Ham to minimize makespan, idletime or flowtime in the static permutation flowshop sequencing problem. **International Journal Of Production Research**, [s.l.], v. 41, n. 1, p.121-148, jan. 2003. Informa UK Limited. [crossref](#)

GAO, J.; CHEN, R.; DENG, W.. An efficient tabu search algorithm for the distributed permutation flowshop scheduling problem. **International Journal Of Production Research**, [s.l.], v. 51, n. 3, p.641-651, fev. 2013. Informa UK Limited. [crossref](#)

GAREY, M. R.; JOHNSON, D. S.; SETHI, R.. The Complexity of Flowshop and Jobshop Scheduling. **Mathematics Of Operations Research**, [s.l.], v. 1, n. 2, p.117-129, maio 1976. **Institute for Operations Research and the Management Sciences (INFORMS)**. [crossref](#)

GEIGER, Martin Josef. Decision support for multi-objective flow shop scheduling by the Pareto Iterated Local Search methodology. **Computers & Industrial Engineering**, [s.l.], v. 61, n. 3, p.805-812, out. 2011. Elsevier BV. [crossref](#)

HATAMI, S.; RUIZ, R.; ANDRÉS-ROMANO, C. Heuristics and metaheuristics for the distributed assembly permutation flowshop scheduling problem with sequence dependent setup times. **International Journal Of Production Economics**, [s.l.], v. 169, p.76-88, nov. 2015. Elsevier BV. [crossref](#)

HUANG, S.; CAI, L.; ZHANG, X. Parallel dedicated machine scheduling problem with sequence-dependent setups and a single server. **Computers & Industrial Engineering**, [s.l.], v. 58, n. 1, p.165-174, fev. 2010. Elsevier BV. [crossref](#)

ISHIBUCHI, H.; YOSHIDA, T.; MURATA, T.. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. **Ieee Transactions On Evolutionary Computation**, [s.l.], v. 7, n. 2, p.204-223, abr. 2003. Institute of Electrical and Electronics Engineers (IEEE). [crossref](#)

JOHNSON, S. M. Optimal two- and three-stage production schedules with setup times included. **Naval Research Logistics Quarterly**, [s.l.], v. 1, n. 1, p.61-68, mar. 1954. Wiley-Blackwell. [crossref](#)

JUNG, S.; WOO, Y.; KIM, B. Two-stage assembly scheduling problem for processing products with dynamic component-sizes and a setup time. **Computers & Industrial Engineering**, [s.l.], v. 104, p.98-113, fev. 2017. Elsevier BV. [crossref](#)

KENNEDY, J.; EBERHART, R.. Particle swarm optimization. Proceedings Of Icnnc'95 - **International Conference On Neural Networks**, [s.l.], v. 1944, p.1942-1948, jan. 1995. IEEE.

LEISTEN, R. Flowshop sequencing problems with limited buffer storage. **International Journal Of Production Research**, [s.l.], v. 28, n. 11, p.2085-2100, nov. 1990. Informa UK Limited. [crossref](#)

LIANG, J. J.; PAN, Q.; CHEN, T.A Dynamic Multi-swarm Particle Swarm Optimizer for blocking flow shop scheduling. 2010 Ieee Fifth International Conference On Bio-inspired Computing: **Theories and Applications (BIC-TA)**, [s.l.], set. 2010. IEEE. [crossref](#)

MACCARTHY, B. L.; LIU, Jiyin. Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. **International Journal Of Production Research**, [s.l.], v. 31, n. 1, p.59-79, jan. 1993. Informa UK Limited. [crossref](#)

NAGANO, M S; MOCCELLIN, J V. A high quality solution constructive heuristic for flow shop sequencing. **Journal Of The Operational Research Society**, [s.l.], v. 53, n. 12, p.1374-1379, 7 nov. 2002. Springer Nature. [crossref](#)

NAWAZ, M.; ENSCORE, E.; HAM, I. A heuristic algorithm for the m-machine, n-job flow-shop sequencing problem. **Omega**, [s.l.], v. 11, n. 1, p.91-95, jan. 1983. Elsevier BV. [crossref](#)

REEVES, C. A genetic algorithm for flowshop sequencing. **Computers & Operations Research**, [s.l.], v. 22, n. 1, p.5-13, jan. 1995. [crossref](#)

RIAHI, V. Scatter search for mixed blocking flowshop scheduling. **Expert Systems With Applications**, [s.l.], v. 79, p.20-32, ago. 2017. Elsevier BV. [crossref](#)

RUIZ, R.; STÜTZLE, T. A simple and effective iterated greedy algorithm for the permutation flowshop scheduling problem. **European Journal Of Operational Research**, [s.l.], v. 177, n. 3, p.2033-2049, mar. 2007. Elsevier BV. [crossref](#)

RUIZ, R.; VALLADA, E.; FERNÁNDEZ-MARTÍNEZ, C. Scheduling in Flowshops with No-Idle Machines. **Computational Intelligence In Flow Shop And Job Shop Scheduling**, [s.l.], p.21-51, 2009. Springer Berlin Heidelberg. [crossref](#)

TAILLARD, E. Some efficient heuristic methods for the flow shop sequencing problem. **European Journal Of Operational Research**, [s.l.], v. 47, n. 1, p.65-74, jul. 1990. Elsevier BV. [crossref](#)

TASGETIREN, M. Et al. Iterated greedy algorithms for the blocking flowshop scheduling problem with makespan criterion. **Computers & Operations Research**, [s.l.], v. 77, p.111-126, jan. 2017. Elsevier BV. [crossref](#)

VALLADA, E.; RUIZ, R.; FRAMINAN, J. New hard benchmark for flowshop scheduling problems minimising makespan. **European Journal Of Operational Research**, [s.l.], v. 240, n. 3, p.666-677, fev. 2015. Elsevier BV. [crossref](#)

VERGARA, S. **Projetos e Relatórios de Pesquisa em Administração**. 2ª ed. São Paulo: Atlas, 1998.

XU, J. et al. An iterated local search for the multi-objective permutation flowshop scheduling problem with sequence-dependent setup times. **Applied Soft Computing**, [s.l.], v. 52, p.39-47, mar. 2017. Elsevier BV. [crossref](#)

XU, N.; HUANG, S.; RONG, Y. Automatic setup planning: current state-of-the-art and future perspective. **International Journal Of Manufacturing Technology And Management**, [s.l.], v. 11, n. 2, p.193-208, 2007. Inderscience Publishers. [crossref](#)

YAGMAHAN, B.; YENISEY, M. A multi-objective ant colony system algorithm for flow shop scheduling problem. **Expert Systems With Applications**, [s.l.], v. 37, n. 2, p.1361-1368, mar. 2010. Elsevier BV. [crossref](#)

YENISEY, M.; YAGMAHAN, B. Multi-objective permutation flow shop scheduling problem: Literature review, classification and current trends. **Omega**, [s.l.], v. 45, p.119-135, jun. 2014. Elsevier BV. [crossref](#)

Recebido: 29 Mai. 2019

Aprovado: 19 Set. 2019

DOI: 10.3895/gi.v15n3.10166

Como citar:

CORREIA, A.S. et al. Avaliação de heurísticas para o problema flowshop com setup separado e independente da seqüência . **R. Gest. Industr.**, Ponta Grossa, v. 15, n. 3, p. 100-125, Jul./Set. 2019.

Disponível em: <<https://periodicos.utfpr.edu.br/revistagi>>. Acesso em:.

Correspondência:

Amanda da Silva Correia

Universidade Tecnológica Federal do Paraná (UTFPR), Ponta Grossa, PR, Brasil..

Direito autoral: Este artigo está licenciado sob os termos da Licença Creative Commons-Atribuição 4.0 Internacional.

