

Análise de métodos heurísticos construtivos para o problema flowshop permutacional na minimização do tempo total da programação

Murillo Vetroni Barros (UTFPR-PG) murillo.vetroni@gmail.com

Fabio Jose Ceron Branco (UTFPR-PG) fbranco@utfpr.edu.br

Claus Reginato Leme (UTFPR-PG) clausrleme@hotmail.com

Resumo:

Os processos industriais necessitam uma programação da produção eficiente. Essa atividade consiste na alocação dos recursos produtivos, com o objetivo de executar determinadas tarefas em um período de tempo definido. Existem diferentes tipos de recursos a serem administrados e também a quantidade de possíveis soluções aumenta exponencialmente com o crescimento da quantidade de tarefas e máquinas presentes no sistema, o que torna a tarefa de programar a produção uma das atividades mais complexas do planejamento da produção. Nesse sentido, o objetivo deste artigo é encontrar qual o melhor método heurístico a ser utilizado para minimizar o tempo total de processamento (*makespan*) e o tempo total de fluxo (*flowtime*) em um ambiente *flowshop* com restrição *no-wait*. Para isso serão analisados métodos heurísticos que possam solucionar o problema da programação da produção em ambiente *No-Wait Flowshop*. A experimentação computacional mostrou qual o melhor método heurístico a ser utilizado para a resolução deste problema. Como resultado, o trabalho gerou uma série de gráficos comparativos de desvio relativo médio para as heurísticas.

Palavras chave: Programação da produção, Métodos heurísticos, *Makespan*, *Flowshop*.

Constructive heuristic method analysis for the permutational Flowshop problem in the reduction of the total time programming

Abstract

Industrial processes need efficient production scheduling. This activity consists of the allocation of productive resources, in order to perform certain tasks in a defined period of time. There are different types of resources to be managed and also the number of possible solutions increases exponentially with the increase in the amount of tasks and machines present in the system, which makes the task of scheduling production one of the most complex production planning activities. In this sense the objective of this article is to find the best heuristic method to be used to minimize the total processing time (*makespan*) and the total flow time in a flowshop environment with no-wait constraint. For this heuristic methods that can solve the problem of production scheduling in the No-Wait Flowshop environment will be analyzed. The computational experimentation showed the best heuristic method to be used to solve this problem. As a result the article generated a series of comparative graphs of mean relative deviation for the heuristics.

Key-words: Scheduling, Heuristic methods, *Makespan*, *Flowshop*.

1. Introdução

O mercado atual está cada vez mais disputado, dessa maneira as empresas buscam a todo o momento melhorar seu processo produtivo, qualidade de produtos, reduzir os custos, reduzir os *setups* e, assim, aumentar sua eficiência para manterem-se competitivas diante de seus concorrentes.

Neste cenário se insere a programação da produção, de modo que, pode-se reduzir o tempo de produção e proporcionar benefícios à empresa, como o aumento da confiabilidade no prazo de entrega dos produtos, a melhora na eficiência da organização e a redução nos custos de produção.

Segundo Branco (2011), um problema de programação da produção frequente é o *flowshop*, em que um conjunto de n tarefas deve ser produzido na mesma sequência de um conjunto de m máquinas. Caso a ordem da produção seja a mesma em todas as máquinas tem-se o ambiente de produção *flowshop* permutacional. O objetivo do problema é obter um sequenciamento de tarefas que possa atingir uma determinada função-objetivo. As mais comuns são o tempo total da programação (chamado na literatura como *makespan*) e o tempo total de fluxo (denominado *flowtime*).

Segundo Garey e Johnson (1979), a programação de produção em ambiente *flowshop* é considerada NP-hard, pois devido à complexidade do problema, o tempo para obter uma solução ótima para problemas de porte razoável seria alto e como o tempo disponível para a tomada de decisão dentro de uma empresa geralmente é pequeno, são utilizados métodos heurísticos, que possibilitam soluções viáveis em um espaço de tempo reduzido.

Desta forma, durante o trabalho serão realizadas análises dos métodos heurísticos construtivos (aqueles que inserem tarefas a cada iteração até obter a solução final), para propor qual o melhor método para a resolução do problema, que será realizada por uma comparação entre os tempos de produção antes e depois da utilização do método.

Portanto, o objetivo do presente artigo é encontrar qual o melhor método heurístico a ser utilizado para minimizar o tempo total de processamento (*makespan*) e o tempo total de fluxo (*flowtime*) em um ambiente *flowshop* com restrição *no-wait*.

2. Referencial Teórico

2.1 Programação da Produção

Segundo Slack et al (1999), a programação da produção é uma das atividades mais complexas no gerenciamento da produção, pois os programadores precisam lidar com diversos tipos de recursos simultaneamente. As máquinas têm diferentes capacidades e o pessoal, diferentes habilidades. O número de programações possíveis cresce à medida que o número de atividades e processos aumenta. Dessa forma, para n operações há $n!$ (n fatorial) maneiras diferentes de programação dos trabalhos em um processo simples. Considerando mais de uma máquina ($m > 1$), o número de programações possíveis aumenta para $(n!)^m$, onde n é o número de tarefas e m o número de máquinas.

De acordo com Lustosa et al (2008), a programação da produção pode ser explicada como a determinação de quando e onde cada operação deve ser realizada e a determinação das datas de início e fim de cada operação que compõe um procedimento. Dessa forma, a programação da produção determina a alocação dos recursos para a realização das tarefas em um determinado período de tempo.

2.2 Programação da Produção em Máquinas

De acordo com Pinedo (1995), o problema de operações em máquinas é descrito através da notação padrão de três campos, α , β , γ , onde o campo α corresponde ao ambiente das máquinas, o campo β corresponde as características do processo e o campo γ corresponde a

função objetivo que nada mais é o critério de avaliação da programação.

Segundo Fuchigami (2010), a atividade de “*scheduling*” abrange a alocação (associação das tarefas nas máquinas), o sequenciamento (ordem das tarefas em cada máquina) e em seguida a programação (determinação dos tempos de início e término de cada operação).

A programação da produção ou *scheduling* é um processo de decisão que tem como objetivo otimizar a produção e onde o uso dos recursos está sujeito a uma série de restrições. Portanto esta programação diz respeito à alocação de recursos limitados para tarefas ao longo do tempo. As formas de recursos e tarefas podem ser diferentes, devido à variedade de indústrias e seus objetivos (KAZAMA, 2011).

2.3 Flowshop

Segundo Kazama (2011) o objetivo do flowshop é encontrar entre as $n!$ possibilidades de sequenciamento de tarefas, aquela que possa otimizar a função objetivo previamente estabelecida. Complementa o autor que o *flowshop* permutacional acontece quando o conjunto de n tarefas é processado exatamente na mesma ordem em todas as m máquinas. Para Moraes (2008), as principais funções objetivo que são otimizadas são *makespan*, que é o tempo total de produção e o *flowtime*, que é o tempo médio de fluxo.

Fuchigami (2010) apresentou um problema numérico do problema de programação da produção em ambiente *flowshop*, com 4 tarefas e 3 máquinas, mostrado na Tabela 1.

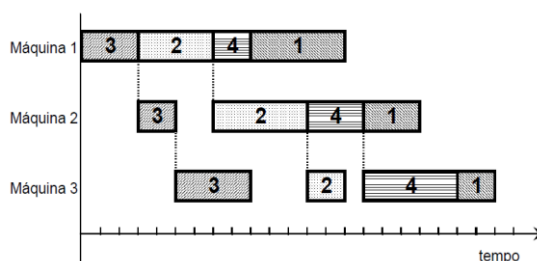
Tabela 1 - Tempo de processamento das tarefas nas máquinas

Máquinas	Tarefa 1	Tarefa 2	Tarefa 3	Tarefa 4
1	5	4	3	2
2	3	5	2	3
3	2	2	4	5

Fonte: Fuchigami (2010)

Para esse problema a sequência de programação da produção obtida foi 3,2,4,1 conforme mostra a Figura 1.

Figura 1 - Gráfico de Gantt para as tarefas 3,2,4,1



Fonte: Fuchigami (2010)

A partir do gráfico de Gantt da Figura 1 pode-se observar que o tempo total da produção (*makespan*) é de 22 unidades de tempo e o *flowtime* é de 65 unidades de tempo.

2.4 Heurísticas Propostas

Johnson (1954), foi o primeiro a trabalhar com o problema de programação da produção

flowshop, onde propôs um método de solução ótima para um problema de n tarefas sendo processadas em 2 máquinas. Palmer (1965), propôs um método que estabelecia a sequência de processamento das tarefas nas máquinas através de um índice denominado *slope index*. Esse índice era calculado de modo a organizar as tarefas em ordem não crescente de seus tempos de processamento, ou seja, as tarefas em que os tempos de processamento tendem a crescer na sequência das máquinas, possuem um índice maior e, conseqüentemente, devem ocupar as primeiras posições na ordem de execução.

Nagano e Moccellini (2007), propuseram o *SPT-Flowtime*, um algoritmo heurístico para minimizar o tempo médio de fluxo em um ambiente de produção *flowshop* permutacional. Os resultados experimentais desse algoritmo apresentaram desempenho superior quando comparados a métodos anteriores, com soluções de boa qualidade e baixo esforço computacional.

2.5 No-Wait Flowshop

Segundo Branco (2006), o problema *no-wait flowshop* tem como base um conjunto M de m máquinas e um conjunto J de n tarefas, sendo que cada tarefa possui m operações que são processadas nas m máquinas. Complementa o autor que, em um problema clássico, uma tarefa só pode ser processada em uma máquina por vez e cada máquina pode processar somente uma operação ao mesmo tempo.

Ainda Branco (2006), aponta que a restrição *no-wait* não permite que haja tempo de espera no processamento de uma tarefa de uma máquina para a próxima, sendo assim, a principal característica do problema é que a operação $i + 1$ de uma tarefa tem a obrigatoriedade de ser processada logo após o término da operação i . O único tempo de espera permitido é no início do processamento da tarefa que ocupa a primeira posição na sequência, na primeira máquina (BRANCO, 2011).

2.6 Principais Heurísticas da Literatura

Uns dos pioneiros a estudar o problema de *no-wait flowshop*, com critério de minimização do tempo total de fluxo foram Deman e Baker (1974) que propuseram um algoritmo *branch and bound* para estabelecer todas as sequenciais parciais, considerando a utilização de limites inferiores. A conclusão dos autores foi que os resultados alcançados foram satisfatórios e que a solução do problema era encontrada tão rapidamente quanto os problemas tradicionais que tinham como função objetivo a minimização do *makespan*.

3. Metodologia

3.1 Heurística Longest Processing Time (LPT)

No presente trabalho foram utilizadas as heurísticas de ordenação: *Longest Processing Time* (LPT) e *Shortest Processing Time* (SPT).

O algoritmo LPT é realizado através do somatório do tempo de processamento das tarefas, e segundo Cruz e Branco (2009), a ordenação deste algoritmo obedece a uma ordem não crescente da soma dos tempos de processamento. A fim de exemplificar o funcionamento do sequenciamento das tarefas nas máquinas, foi criada a Tabela 2, para elaboração da ordenação LPT e também para a elaboração do gráfico de Gantt.

Tabela 2 - Tempos de processamento das quatro tarefas em três máquinas

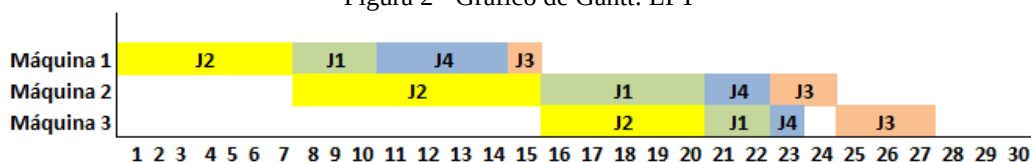
	J ₁	J ₂	J ₃	J ₄
Máquina 1	3	7	1	4
Máquina 2	5	8	2	2
Máquina 3	2	5	3	1

Fonte: Autores (2017)

Pela tabela 2, as somatórias, de J₁ a J₄, respectivamente, são: 10, 20, 6 e 7. Assim, o sequenciamento pela LPT é definido por J₂, J₁, J₄ e J₃. O “Ji” vêm da palavra *job* (em inglês), representando o conjunto de máquinas paralelas em cada estágio de produção.

A Figura 2 mostra a dinâmica da ordenação das tarefas (J₁, J₂, J₃ e J₄) nas máquinas, seguindo o princípio do funcionamento do algoritmo que é da maior soma dos tempos de processamento das tarefas para a menor soma dos tempos de processamento das tarefas sem restrição. Pode-se perceber que as tarefas 1, 3 e 4 aguardam respectivamente 5, 7 e 6 unidades de tempos para iniciar seu processamento na máquina 2, demonstrando assim a não utilização da restrição *no-wait*.

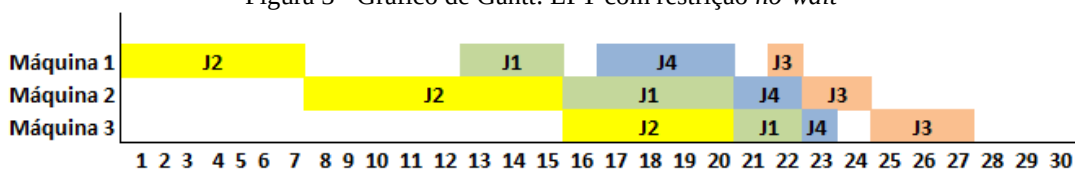
Figura 2 - Gráfico de Gantt: LPT



Fonte: Autores (2017)

O gráfico de Gantt da Figura 3 apresenta a dinâmica de ordenação da heurística LPT com a restrição *no-wait*. A implementação da restrição *no-wait* pode ser percebida pelo atraso no início das tarefas 1, 3 e 4, de modo que o fluxo de produção dessas tarefas fosse contínuo.

Figura 3 - Gráfico de Gantt: LPT com restrição *no-wait*



Fonte: Autores (2017)

Ao comparar os gráficos de Gantt das figuras 2 e 3 sobre a ordenação das tarefas, pode-se observar que o segundo gráfico da Gantt não apresenta tempo de espera das tarefas para serem processadas nas máquinas, uma vez que foi implementada a restrição *no-wait*. O Apêndice A mostra a comparação entre os métodos LPT e LPT + NEH, com variação de $\alpha = 0,0$ e $1,0$.

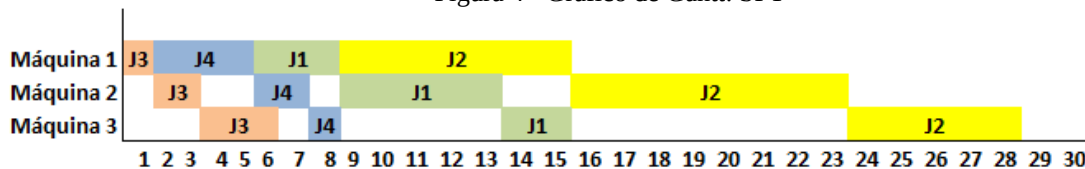
3.2 Heurística *Shortest Processing Time* (SPT)

O algoritmo SPT é realizado através da soma dos tempos de processamentos das tarefas, e segundo Cruz e Branco (2009), a ordenação obedece à ordem crescente da soma desses tempos de processamento.

Para exemplificar o funcionamento da heurística SPT foi utilizada a Tabela 2 de tempos de

processamento das tarefas em máquinas para a elaboração do gráfico de Gantt que está apresentado pelas Figuras 4 e 5 demonstram a heurística sem a restrição *no-wait* e com a restrição respectivamente.

Figura 4 - Gráfico de Gantt: SPT

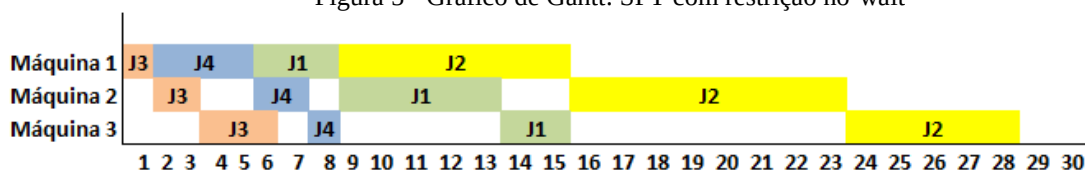


Fonte: Autores (2017)

Pode-se perceber no gráfico de Gantt da Figura 4, que todas as tarefas possuem um fluxo contínuo de produção, ou seja, não há tempo de espera das tarefas para serem processadas na máquina seguinte. O Apêndice B mostra a comparação entre os métodos SPT e SPT + NEH, com variação de $\alpha = 0,0$ e $1,0$.

Dessa forma o gráfico de Gantt, da heurística SPT sem a restrição *no-wait* será exatamente o mesmo que o gráfico de Gantt com a implementação da restrição como pode ser visto na Figura 5.

Figura 5 - Gráfico de Gantt: SPT com restrição no-wait



Fonte: Autores (2017)

3.3 Experimento Computacional

Para a verificação do melhor método heurístico será utilizado um banco de dados composto por 100 problemas para cada classe. Estas classes são determinadas pela combinação entre o número de tarefas n e o número de máquinas m , sendo que os tempos de processamento das tarefas foram gerados aleatoriamente no intervalo de 1 a 99, com distribuição uniforme.

Os problemas encontram-se descritos através da combinação ($n \times m$): 10x5, 10x10, 10x15, 10x20, 20x5, 20x10, 20x15, 20x20, 30x5, 30x10, 30x15, 30x20, 40x5, 40x10, 40x15, 40x20, 50x5, 50x10, 50x15, 50x20, 60x5, 60x10, 60x15, 60x20, 70x5, 70x10, 70x15, 70x20, 80x5, 80x10, 80x15, 80x20, 90x5, 90x10, 90x15, 90x20, 100x5, 100x10, 100x15, 100x20, 110x5, 110x10, 110x20, 120x5, 120x10, 120x15, 120x20, 130x5, 130x10, 130x15, 130x20, 140x5, 140x10, 140x15, 140x20, 150x5, 150x10, 150x15, 150x20, 160x5, 160x10, 160x15, 160x20, 170x5, 170x10, 170x15, 170x20, 180x5, 180x10, 180x15, 180x20, 190x5, 190x10, 190x15, 190x20, 200x5, 200x10, 200x15 e 200x20.

Através de um *software* gerador de dados, as combinações entre o número de tarefas e o número de máquinas descritas no parágrafo anterior, puderam ser realizadas. Os arquivos de saída do gerador de dados foram utilizados para a execução das heurísticas SPT, SPT + NEH, LPT, LPT + NEH, com restrição *no-wait*, que foram desenvolvidas por meio de programação em linguagem Pascal, utilizando o *software* Dev Pascal.

As heurísticas, cada qual seguindo seu princípio de funcionamento, utilizaram-se da função biobjetivo, representada pela equação 1, por meio da interação entre *makespan* e *flowtime*.

$$fO = \{\alpha \times Makespan + (1-\alpha) \times Flowtime\} \quad (1)$$

Para cada uma das heurísticas foi alterado o valor do parâmetro alfa de 0,0 e 1,0 com variação de 0,1. Aos onze valores do parâmetro alfa foram atribuídos os resultados de programação e o tempo médio de computação para cada heurística.

As heurísticas utilizadas foram LPT e SPT, com e sem a presença da heurística NEH. As heurísticas primeiramente foram analisadas em duplas com e sem a presença da heurística NEH e após a comprovação da eficiência da heurística NEH no melhoramento da solução inicial, foi realizada uma nova análise entre as heurísticas citadas acima com a presença do NEH para que pudesse ser avaliado qual o melhor método a ser utilizado de acordo com a variação do parâmetro alfa com base nas porcentagens de sucesso, desvio relativo médio e tempo médio de computação para cada um dos valores do parâmetro alfa e para cada uma das 20 possibilidades de tarefas (10, 20, 30, 40,..., 200). Para cada número de tarefa existem 400 instâncias sendo analisadas para gerar o valor da função biobjetivo, uma vez que existem quatro possibilidades de máquinas (5, 10, 15 e 20). Os valores do parâmetro alfa são atribuídos como: 0; 0,1; 0,2; 0,3; 0,4; 0,5; 0,6; 0,7; 0,8; 0,9 e 1,0.

Para determinar o desempenho dos algoritmos, os resultados serão avaliados pelas estatísticas de porcentagem de sucesso, desvio relativo médio e tempo médio computacional. O desvio relativo mede a variação correspondente à melhor solução obtida pelos métodos. Quando o desvio relativo da solução de um problema é igual a zero para um determinado método, significa que o tempo total da produção fornecido é o menor, ou seja, o algoritmo apresentou a melhor programação. Desta maneira, o melhor algoritmo é aquele que apresenta o menor valor de desvio relativo médio (a média aritmética dos desvios) para uma determinada classe de problemas.

O desvio relativo (DR_x) de um método x para um determinado problema pode ser calculado da seguinte maneira:

$$DR_x = \frac{D_x - D^*}{D^*} \times 100$$

Onde DR_x é o valor da função objetivo obtido pelo método, e D* é o melhor valor da função objetivo obtido pelos métodos. A estatística que determinará a escolha do melhor método heurístico a ser utilizado será a Porcentagem de Sucesso. O desvio relativo será a ela correlacionada e será usado para reiterar o desempenho quanto à estatística principal. Assim, quando um método apresentar valores superiores da porcentagem de sucesso, será estatisticamente esperado que o desvio relativo apresente valores mínimos.

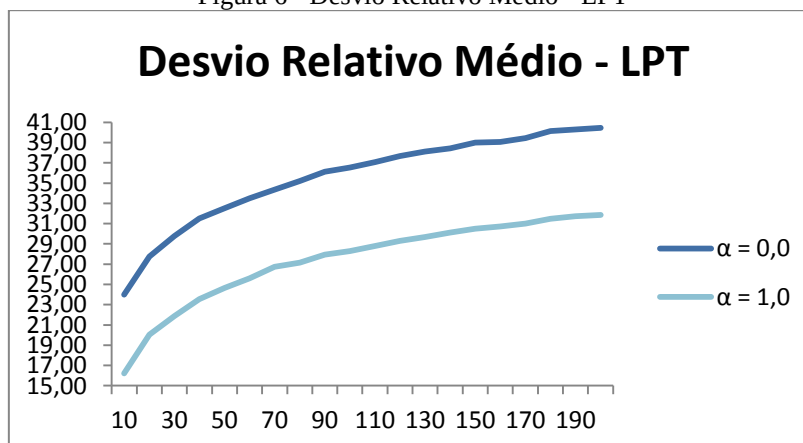
Em relação ao tempo médio de computação, tendo em vista que os métodos serão projetados para não apresentarem tempos excessivos e grandes diferenças para problemas da mesma classe (m máquinas, n tarefas), eles serão apresentados apenas com o propósito de avaliação dos esforços computacionais.

A experimentação computacional foi realizada em um microcomputador com processador Intel Core I5 5200U 2.2GHz, com memória RAM de 4 Gb e sistema operacional Windows 10.

4. Resultados

Os gráficos apresentados nesta seção demonstram os dados obtidos por meio dos cálculos descritos anteriormente para que possa ser analisada a eficiência da heurística NEH no melhoramento da solução mediante as variações do parâmetro alfa. Os valores da função biobjetivo representam os menores valores de cada n entre as heurísticas analisadas para porcentagem de sucesso, desvio relativo médio e tempo médio de computação. Inicialmente, a Figura 6 apresenta o desvio relativo médio para heurística LPT.

Figura 6 - Desvio Relativo Médio - LPT



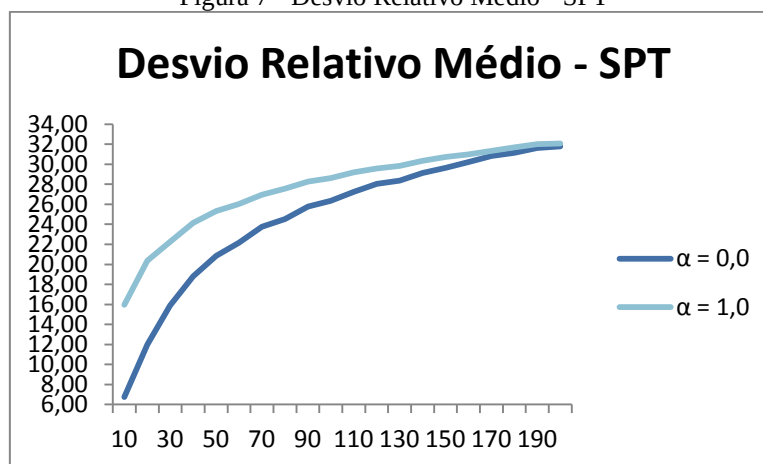
Fonte: Autores, 2017

Pode-se observar que independente da variação do parâmetro alfa, a heurística LPT + NEH apresentou uma melhoria significativa nas soluções em relação a heurística LPT, obtendo 100% de sucesso em todos os ambientes analisados, independentemente do número de tarefas e do valor do parâmetro alfa.

Analisando o gráfico da Figura 6, é possível observar que no intervalo de $\alpha = 0,0$ até $\alpha = 1,0$ a heurística LPT + NEH apresenta uma melhora significativa para *flowtime* em aproximadamente 18% a 40% quando comparada a heurística LPT. Quando comparados os valores em relação ao *makespan* ($\alpha = 1,0$) a melhora na solução varia em aproximadamente 18% a 32%, comprovando assim a eficiência da heurística LPT + NEH em relação a heurística LPT.

Em relação aos tempos médios de computação, na grande maioria dos casos a heurística LPT apresenta um tempo menor, porém não há uma grande diferença no tempo entre as duas heurísticas analisadas. A Figura 7 mostra o desvio relativo médio para heurística SPT.

Figura 7 - Desvio Relativo Médio - SPT



Fonte: Autores, 2017

Pode-se observar que mesmo com a variação do parâmetro alfa, a heurística SPT + NEH apresentou significativa melhora na solução do problema quando comparada a heurística SPT, apresentando na maioria dos casos 100% de sucesso.

Através da análise do gráfico da Figura 7 de desvio relativo médio, é possível afirmar que a heurística SPT + NEH apresenta uma melhora de aproximadamente 6,73% a 31,85% na solução envolvendo o *flowtime* com o parâmetro alfa variando de 0 a 0,9 quando comparado a heurística SPT. Em relação ao *makespan* ($\alpha = 1,0$) a heurística SPT + NEH também apresenta uma grande melhoria nos resultados, chegando a minimizar o tempo médio de produção em aproximadamente 32% quando estão sendo realizadas 200 tarefas.

Pode-se afirmar também que conforme aumenta o número de tarefas, independente do parâmetro alfa estabelecido, maior é a porcentagem de melhora na solução da heurística SPT + NEH quando comparada a heurística SPT.

Em relação ao tempo médio de computação, pode-se afirmar que na maioria dos casos a heurística SPT apresenta melhores tempos quando comparado a heurística SPT + NEH, porém a diferença entre os tempos não é significativa, o que permite dizer que devido a melhora que a heurística SPT + NEH apresenta, ela é mais eficiente e não demanda muito mais tempo a ser executada.

5. Considerações Finais

As heurísticas avaliadas no presente trabalho, SPT, SPT + NEH, LPT, LPT + NEH, foram adaptadas ao problema *no-wait* com o objetivo de analisar o comportamento das mesmas perante a uma função objetivo além de testar a eficiência da heurística NEH a partir da solução inicial gerada pelas demais heurísticas.

Este trabalho teve como objetivo atuar em um problema de programação de tarefas em máquinas em um ambiente *flowshop* com restrição *no-wait*, com base na função objetivo, $f_0 = \{\alpha \times Makespan + (1 - \alpha) \times Flowtime\}$, que busca obter uma solução por meio da combinação de *makespan* e *flowtime*. O parâmetro alfa foi inserido a fim de que fosse possível avaliar o quanto a heurística NEH melhoraria a solução inicial das demais heurísticas para *makespan* ou *flowtime* e também avaliar qual das heurísticas com o NEH em análise o parâmetro demonstra maior favorecimento da solução em relação ao *makespan* e ao *flowtime*.

O presente trabalho demonstrou que, independente da variação do parâmetro alfa, a heurística

NEH apresentou uma melhora significativa na solução inicial gerada pelas demais heurísticas analisadas, provando dessa maneira a sua eficiência em relação às demais.

A partir da análise das duas heurísticas com o NEH, pode-se observar que a heurística SPT + NEH foi a que obteve o melhor desempenho em relação a porcentagem de sucesso e desvio relativo médio na grande maioria dos casos analisados, seja para *makespan* ou para o *flowtime*, independentemente do número de tarefas que estavam sendo analisados. Os tempos médios de computação entre todas as heurísticas foram equivalentes e baixos, tornando dessa forma viável a utilização das heurísticas para a resolução do problema. As heurísticas devem prezar pela resposta de alta qualidade de solução em tempo de computação pouco custoso.

Este trabalho pode ser utilizado em aplicações futuras, utilizando estas mesmas ordenações em outros cenários, adaptando-as a sistemas de produção com diferentes funções objetivos e/ou restrições.

Referências

- BRANCO, F. J. C.** Avaliação de métodos heurísticos para o problema no-wait flowshop com o critério de minimização da duração total da programação. Dissertação (Mestrado em Engenharia de Produção). Escola de Engenharia de São Carlos. Universidade de São Paulo. São Carlos. 2006.
- BRANCO, F. J. C.** Um Novo Método Heurístico Construtivo de Alto Desempenho para o Problema No-Idle Flow Shop. Dissertação (Doutorado em Engenharia de Produção). Escola de Engenharia de São Carlos. Universidade de São Paulo. São Carlos. 2011.
- CRUZ, F. E. V; BRANCO, F. J. C.** Análise de Métodos Para Solução Problema de Programação de Operações Flowshop, Simpósio de Engenharia de Produção. 2009.
- DEMAN, J. M. V; BAKER K. R.** Minimizing Mean Flowtime in the Flow Shop With No Intermediate Queues. AIIE Transactions, v.6, n.1, p.28-34. 1974
- FUCHIGAMI, H. Y.** Flexible Flow Line com Tempos de Setup: Métodos Heurísticos. Dissertação (Doutorado em Engenharia de Produção). Escola de Engenharia de São Carlos. Universidade de São Paulo. São Carlos. 2010.
- GAREY, M. R; JOHNSON, D. S.** Computers and intractability: a guide of NP-completeness. New York: W. H. Freeman, 1979.
- JOHNSON, S. M.** Optimal two-and-three-stage production schedules with setup times included. Naval Research Logistics Quarterly. Washington, v.1, n.1, p.61-68, Mar, 1954.
- KAZAMA, E. K.** Heurísticas Construtivas Para Programação de Operações em Sistemas de Produção Flowshop Permutacional: Classificação de suas Fases Construtivas. TCC. Escola de Engenharia de São Carlos. Universidade de São Paulo. São Carlos. 2011.
- LUSTOSA, L. et al.** Planejamento e Controle da Produção. Rio de Janeiro: Elsevier, 2008. 4ª ed.
- MORAIS, M. F.** Métodos Heurísticos Construtivos para Redução do Estoque em Processo em Ambientes de Produção Flow Shop Híbridos com Tempo de Setup Dependentes da Sequência. Dissertação (Mestrado em Engenharia de Produção). Escola de Engenharia de São Carlos. Universidade de São Paulo. São Carlos. 2008.
- NAGANO, M. S.; MOCCELLIN, J. V.** Reducing Mean Flow Time in Permutacion Flowshop. Journal of the Operational Research Society, Birmingham, advance online publication. 2007.
- PALMER, D.S.** Sequencing Jobs Throught a Multi-Stage Process in the Minimum Total Time – A quick Method of Obtaining a Near Optimum. Operational Research Quaterly, London, v.16, n.1, p.101-107, Mar, 1965.
- PINEDO, M.** Scheduling: Theory, Algorithms, and Systems. New Jersey. Prentice-Hall. 1995.
- SLACK, N. et al.** Administração da produção: Edição Compacta. São Paulo: Atlas, 1999. 1ª ed.

APENDICE A - Comparação entre os métodos LPT e LPT +NEH, com variação de $\alpha = 0,0$ e $1,0$

	N	% DE SUCESSO		DESVIO RELATIVO MÉDIO		TEMPO MÉDIO	
		LPT	LPT + NEH	LPT	LPT + NEH	LPT	LPT + NEH
A = 0,0	10	0	<u>100</u>	23,98	<u>0,00</u>	<u>34,65</u>	44,88
	20	0	<u>100</u>	27,76	<u>0,00</u>	32,89	32,89
	30	0	<u>100</u>	29,76	<u>0,00</u>	32,93	<u>32,89</u>
	40	0	<u>100</u>	31,52	<u>0,00</u>	32,89	32,89
	50	0	<u>100</u>	32,52	<u>0,00</u>	32,93	<u>32,89</u>
	60	0	<u>100</u>	33,51	<u>0,00</u>	32,93	<u>32,85</u>
	70	0	<u>100</u>	34,36	<u>0,00</u>	32,89	<u>32,85</u>
	80	0	<u>100</u>	35,21	<u>0,00</u>	32,89	32,89
	90	0	<u>100</u>	36,14	<u>0,00</u>	32,89	32,89
	100	0	<u>100</u>	36,52	<u>0,00</u>	<u>32,93</u>	33,13
	110	0	<u>100</u>	37,08	<u>0,00</u>	<u>32,93</u>	33,91
	120	0	<u>100</u>	37,66	<u>0,00</u>	<u>32,93</u>	34,30
	130	0	<u>100</u>	38,10	<u>0,00</u>	<u>33,40</u>	34,38
	140	0	<u>100</u>	38,42	<u>0,00</u>	<u>33,17</u>	34,53
	150	0	<u>100</u>	39,00	<u>0,00</u>	<u>33,91</u>	34,61
	160	0	<u>100</u>	39,07	<u>0,00</u>	33,87	<u>33,48</u>
	170	0	<u>100</u>	39,43	<u>0,00</u>	<u>34,22</u>	36,92
	180	0	<u>100</u>	40,14	<u>0,00</u>	<u>34,06</u>	42,54
	190	0	<u>100</u>	40,31	<u>0,00</u>	<u>34,57</u>	48,83
	200	0	<u>100</u>	40,45	<u>0,00</u>	<u>35,39</u>	55,43
A = 1,0	10	0	<u>100</u>	16,21	<u>0,00</u>	<u>39,84</u>	41,88
	20	0	<u>100</u>	20,05	<u>0,00</u>	32,93	32,93
	30	0	<u>100</u>	21,88	<u>0,00</u>	<u>32,89</u>	32,97
	40	0	<u>100</u>	23,55	<u>0,00</u>	<u>32,89</u>	32,93
	50	0	<u>100</u>	24,65	<u>0,00</u>	32,93	32,93
	60	0	<u>100</u>	25,59	<u>0,00</u>	<u>32,89</u>	32,93
	70	0	<u>100</u>	26,74	<u>0,00</u>	<u>32,93</u>	32,97
	80	0	<u>100</u>	27,15	<u>0,00</u>	32,93	32,93
	90	0	<u>100</u>	27,92	<u>0,00</u>	32,93	32,93
	100	0	<u>100</u>	28,28	<u>0,00</u>	<u>32,89</u>	33,75
	110	0	<u>100</u>	28,80	<u>0,00</u>	<u>32,89</u>	34,02
	120	0	<u>100</u>	29,31	<u>0,00</u>	<u>32,93</u>	34,42
	130	0	<u>100</u>	29,68	<u>0,00</u>	<u>33,17</u>	35,16
	140	0	<u>100</u>	30,10	<u>0,00</u>	<u>33,40</u>	35,24
	150	0	<u>100</u>	30,48	<u>0,00</u>	<u>33,99</u>	34,65
	160	0	<u>100</u>	30,72	<u>0,00</u>	34,18	<u>33,87</u>
	170	0	<u>100</u>	31,00	<u>0,00</u>	<u>34,30</u>	36,80
	180	0	<u>100</u>	31,47	<u>0,00</u>	<u>34,30</u>	42,07
	190	0	<u>100</u>	31,72	<u>0,00</u>	<u>34,92</u>	47,46
	200	0	<u>100</u>	31,84	<u>0,00</u>	<u>34,96</u>	52,93

APÊNDICE B - Comparação entre os métodos SPT e SPT + NEH, com variação de $\alpha = 0,0$ e $1,0$

		% DE SUCESSO		DESVIO RELATIVO MÉDIO		TEMPO MÉDIO	
	N	SPT	SPT + NEH	SPT	SPT + NEH	SPT	SPT + NEH
A = 0,0	10	1,5	<u>99,25</u>	6,73	<u>0,01</u>	42,27	<u>41,95</u>
	20	0	<u>100</u>	12,00	<u>0,00</u>	<u>32,89</u>	32,93
	30	0	<u>100</u>	15,90	<u>0,00</u>	<u>32,85</u>	32,93
	40	0	<u>100</u>	18,79	<u>0,00</u>	<u>32,89</u>	32,93
	50	0	<u>100</u>	20,85	<u>0,00</u>	32,89	32,89
	60	0	<u>100</u>	22,15	<u>0,00</u>	32,93	32,93
	70	0	<u>100</u>	23,76	<u>0,00</u>	<u>32,89</u>	32,93
	80	0	<u>100</u>	24,54	<u>0,00</u>	<u>32,85</u>	32,93
	90	0	<u>100</u>	25,76	<u>0,00</u>	<u>32,89</u>	32,93
	100	0	<u>100</u>	26,34	<u>0,00</u>	32,93	<u>32,89</u>
	110	0	<u>100</u>	27,25	<u>0,00</u>	32,89	32,89
	120	0	<u>100</u>	28,05	<u>0,00</u>	32,93	32,93
	130	0	<u>100</u>	28,37	<u>0,00</u>	33,01	<u>32,93</u>
	140	0	<u>100</u>	29,14	<u>0,00</u>	33,48	<u>32,93</u>
	150	0	<u>100</u>	29,65	<u>0,00</u>	33,91	<u>32,93</u>
	160	0	<u>100</u>	30,23	<u>0,00</u>	34,06	<u>32,93</u>
	170	0	<u>100</u>	30,83	<u>0,00</u>	<u>33,79</u>	35,08
	180	0	<u>100</u>	31,14	<u>0,00</u>	<u>34,73</u>	39,88
	190	0	<u>100</u>	31,62	<u>0,00</u>	<u>34,42</u>	45,43
	200	0	<u>100</u>	31,80	<u>0,00</u>	<u>35,16</u>	51,48
A = 1,0	10	0,25	<u>100</u>	15,95	<u>0,00</u>	<u>38,60</u>	41,95
	20	0	<u>100</u>	20,35	<u>0,00</u>	32,93	32,93
	30	0	<u>100</u>	22,26	<u>0,00</u>	<u>32,93</u>	32,97
	40	0	<u>100</u>	24,14	<u>0,00</u>	32,93	32,93
	50	0	<u>100</u>	25,34	<u>0,00</u>	<u>32,93</u>	32,97
	60	0	<u>100</u>	26,04	<u>0,00</u>	<u>32,89</u>	32,93
	70	0	<u>100</u>	26,95	<u>0,00</u>	<u>32,93</u>	32,97
	80	0	<u>100</u>	27,57	<u>0,00</u>	32,93	32,93
	90	0	<u>100</u>	28,25	<u>0,00</u>	<u>32,93</u>	32,97
	100	0	<u>100</u>	28,60	<u>0,00</u>	<u>32,89</u>	32,93
	110	0	<u>100</u>	29,20	<u>0,00</u>	<u>32,93</u>	32,97
	120	0	<u>100</u>	29,57	<u>0,00</u>	32,93	<u>32,89</u>
	130	0	<u>100</u>	29,84	<u>0,00</u>	33,01	33,01
	140	0	<u>100</u>	30,35	<u>0,00</u>	33,20	<u>32,93</u>
	150	0	<u>100</u>	30,73	<u>0,00</u>	33,83	<u>32,93</u>
	160	0	<u>100</u>	30,97	<u>0,00</u>	33,75	<u>32,93</u>
	170	0	<u>100</u>	31,34	<u>0,00</u>	33,60	<u>32,93</u>
	180	0	<u>100</u>	31,69	<u>0,00</u>	<u>34,02</u>	34,92
	190	0	<u>100</u>	31,99	<u>0,00</u>	<u>34,14</u>	39,57
	200	0	<u>100</u>	32,06	<u>0,00</u>	<u>34,61</u>	44,69