MÉTODOS HEURÍSTICOS PARA A PROGRAMAÇÃO DA PRODUÇÃO EM *FLOW SHOP* PERMUTACIONAL COM TEMPOS DE *SETUP* SEPARADOS DOS TEMPOS DE PROCESSAMENTO E INDEPENDENTES DA SEQUÊNCIA DE TAREFAS

Thays J. Perassoli Boiko
GEPPGO, DEP, FECILCAM
Campo Mourão/PR
thaysperassoli@bol.com.br

João Vitor Moccellin
DEP, EESC, USP
São Carlos/SP
jvmoccel@sc.usp.br

RESUMO

Este artigo dedica-se ao Problema de Programação da Produção em *Flow Shop* Permutacional com tempos de *setup* separados dos tempos de processamento e independentes da sequência de execução das tarefas. O objetivo é minimizar a duração total da programação (*Makespan*). Por intermédio de investigações realizadas sobre as características estruturais do Problema e sua solução, uma propriedade deste problema é apresentada. Dois novos métodos heurísticos são desenvolvidos, com base na propriedade apresentada e no procedimento de inserção de tarefas utilizado nos Métodos NEH e N&M: um construtivo, denominado BM_c, e; um melhorativo, denominado BM_m. Os métodos heurísticos propostos são comparados com os Métodos heurísticos melhorativos de Cao e; Bedworth (1992) e Rajendran e Ziegler (1997), através de um grande número de problemas gerados aleatoriamente. Os resultados da experimentação computacional mostram a qualidade do Método construtivo BM_c e a melhor performance do Método melhorativo BM_m.

PALAVRAS-CHAVE. Programação da Produção. *Flow Shop* Permutacional. Tempos de *Setup* Independente. (Programação Matemática).

ABSTRACT

This paper addresses the Permutation Flow Shop Production Scheduling problem with separated, non-batch, and sequence-independent setup times. The objective is minimizing the total time to complete the schedule (Makespan). Following an investigation of Problem structural characteristics and its solution, a property of this scheduling problem is presented. Two news heuristics methods are development, on the basis of the presented property and in the job insertion procedure used in the Methods named NEH e N&M: one constructive, denote by BM_c, and; one improvement, denote by BM_m. The proposed heuristics methods are compared with the improvement heuristics Methods of Cao; Bedworth (1992) and Rajendran; Ziegler (1997), by a large number of randomly generated problems. The results of the computational investigation show the quality of the constructive heuristic Method BM_c and that the improvement heuristic Method BM_c outperforms all others.

KEYWORDS. Scheduling Production. Permutation Flow Shop. Independent Setup Times. (Mathematical Programming).

1. Introduction

A Programação da Produção (PP) é a atividade do (BAKER, 1974; MACCARTHY e LIU, 1993; YANG e LIAO, 1999). Este é especificado Planejamento, Programação e Controle da Produção que ocorre no nível de planejamento de itens individuais. O Problema de Programação da Produção (PPP) consiste em decidir a alocação de recursos através do tempo para a realização de tarefas, para melhor satisfazer um grupo de critérios pré-definidos, em termos de sua classificação, dos critérios de desempenho adotados (MACCARTHY e LIU, 1993) e das hipóteses do problema (BOIKO e MORAIS, 2009).

Os PPP são classificados de acordo com os sistemas de produção, por tipo de posicionamento do processo de produção. Para a classificação dos PPP consultar Boiko e Morais (2009).

Os critérios de desempenho adotado no PPP caracterizam a natureza do Problema. Critérios de desempenho são listados por Maccarthy e Liu (1993) e explicados por French (1982).

As hipóteses do PPP dizem respeito à dinâmica de funcionamento do sistema de produção e as restrições envolvidas. Para hipóteses do PPP, verificar Gupta e Stafford Jr. (2006).

A pesquisa aqui relatada trata do Problema de Programação da Produção em sistemas de produção *Flow Shop* Permutacional com *m* máquinas, tendo o *Makespan* como critério de desempenho e adotando a restrição de que o tempo de *setup* é separado do tempo de processamento. O tipo de *setup* considerado é o *non-batch*, antecipatório e independente da sequência de execução.

Definições de *Flow Shop* Permutacional são apresentadas por Taillard (1993) e Moccellin (1995).

Para definições de *Makespan*, consultar Bedworth e Bailey (1987).

Enquanto que, para definições de *setup*, tempo de *setup* e classificação de *setup* ver Cao e Bedworth (1992), Allahverdi, Gupta e Aldowaisan (1999), Yang e Liao (1999), Cheng, Gupta e Wang (2000), Su e Chou (2000), Allahverdi *et al*. (2008).

Revisões de literatura sobre PP adotando a restrição de tempo de *setup* separados dos tempos de processamento são realizadas por Allahverdi, Gupta e Aldowaisan (1999), Yang e Liao (1999), Cheng, Gupta e Wang (2000) e Allahverdi *et al.* (2008).

O objetivo principal da pesquisa, aqui relatada, foi investigar o PPP em em *Flow Shop* Permutacional com *m* máquinas, com a restrição de tempos de *setup* separados dos tempos de processamento, antecipatórios e independentes da sequência de execução das tarefas, buscando propor novos métodos heurísticos para a minimização do *Makespan*.

Por intermédio de investigações realizadas sobre as características estruturais do PPP em *Flow Shop* Permutacional, tendo o *Makespan* como critério de desempenho e a restrição de tempos de *setup* separados dos tempos de processamento, antecipatórios, dependentes ou independentes da sequência de execução das tarefas e sobre a solução deste Problema, uma propriedade deste é apresentada. Tal propriedade é utilizada no desenvolvimento dos métodos heurísticos propostos, que baseiam-se também no procedimento de inserção de tarefas utilizado nos Métodos NEH (NAWAZ, ENSCORE JR. e HAM, 1983) e N&M (NAGANO e MOCCELLIN, 2002).

O método de abordagem adotado no desenvolvimento da pesquisa foi o qualitativo-quantitativo. A pesquisa classifica-se, quanto aos fins, como descritiva, explicativa e metodológica e, quanto aos meios, como bibliográfica e experimental (computacional).

O artigo está dividido em 7 seções. Após a contextualização da problemática, da apresentação do objetivo da pesquisa e da caracterização da metodologia, a revisão de literatura realizada é brevemente exposta. Na terceira seção, o PPP considerado na pesquisa é definido em termos de sua especificação e da notação utilizada. Em seguida, a propriedade estrutural do PPP desenvolvida é detalhada. Na quinta seção, os Métodos heurísticos propostos são apresentados.

Na sexta seção a experimentação computacional é relatada. Por fim, têm-se as considerações finais.

2. Trabalhos em Programação da Produção em *Flow Shop* com tempos de *setup* separados e independentes e critério de desempenho de *Makespan*

Na revisão de literatura realizada foram encontrados trabalhos e envolvendo considerações de *setup* independente em *Flow Shop* com duas, três e *m* máquinas (F*m*).

No *Flow Shop* com duas máquinas foram encontrados os trabalhos de Yoshida e Hitomi (1979), Sule (1982), Sule e Huang (1983), Szwarc (1983), Nabeshima e Maruyama (1984), Khurana e Bagga (1984), Khurana e Bagga (1985), Szwarc (1986), Logendran e Sriskandarjah (1993), Allahverdi (1995), Levner, Kogan e Maimon (1995), Allahverdi (1997), Kognan e Levner (1998), Cheng, Wang e Sriskandarjah (1999) e Glass, Shafransky e Strusevich (2000).

No *Flow Shop* com três máquinas foram encontrados os trabalhos de Sule e Huang (1983) e Nabeshima e Maruyama (1984).

No caso de *Flow Shop* com *m* máquinas aparecem os trabalhos de Gupta (1972), Szwarc (1986), Szwarc e Gupta (1987), Park e Steudel (1991), Proust, Gupta e Deschamps (1991), Cao e Bedworth (1992), Han e Dejax (1994), Rajendran e Ziegler (1997) e Brucker, Knust e Wang (2005).

Destaca-se o fato de não terem sido encontrados, no exame de literatura realizado, trabalhos em *Flow Shop* com *m* máquinas e apenas a restrição de tempos de *setup* separados dos tempos de processamento, antecipatórios e independentes da sequência, tendo o *Makespan* como critério de desempenho, que é o Problema de Programação da Produção considerado.

No entanto, os métodos heurísticos melhorativos propostos por Cao e Bedworth (1992) e Rajendran e Ziegler (1997), que, respectivamente, consideram os tempos de transferência e os tempos de remoção como separados dos tempos de processamento (assim como os tempos de *setup*), podem ser adaptados ao Problema considerado na pesquisa, fazendo-se os tempos de transferência e de remoção iguais à zero. Desta maneira, tais heurísticas são comparadas aos métodos heurísticos propostos, como o objetivo de avaliar as performances destes.

3. Definição do Problema de Programação da Produção

O Problema de Programação da Produção (PPP) considerado pode ser assim especificado:

i) **Classificação**: PPP em sistema de produção *Flow Shop* Permutacional com *m* máquinas;
ii) **Critério de desempenho**: *Makespan*;
iii) **Hipóteses**: Hipóteses do PPP em *Flow Shop* tradicional, cujos tempos de *setup* são considerados como parte dos tempos de processamento (BAKER, 1974; TAILLARD, 1993; RUIZ e MAROTO, 2005; GUPTA e STAFFORD JR., 2006) adaptadas ao caso de *Flow Shop* com tempos de *setup* separados dos tempos de processamento, antecipatórios e independentes da sequência de execução das tarefas (LOGENDRAN e SRISKANDARAJAH, 1993; YANG e LIAO, 1999) e ao caso de não existirem outras restrições.

A notação utilizada é assim definida:

n = número de tarefas a serem programadas;

m = número de máquinas do *Flow Shop*;

$\sigma = J_{[1]}, J_{[2]},..., J_{[j]},..., J_{[n]}$ = uma sequência qualquer, dentre as (n!) possíveis sequências, onde $J_{[j]}$ representa a tarefa que ocupa a j-ésima posição;

$p_{k[j]}$ = tempo de processamento na máquina M_k da tarefa $J_{[j]}$;

$s_{k[j]}$ = tempo de *setup* da máquina M_k para a execução da tarefa $J_{[j]}$;

$X_{[j]k}$ = intervalo de tempo entre o término da preparação da máquina M_k para a execução da $J_{[j]}$ e o inicio do processamento de $J_{[j]}$ nesta máquina, ou seja, é o tempo ocioso na máquina M_k entre o fim do *setup* para a tarefa $J_{[j]}$ e o início do processamento desta tarefa;

$Y_{[j]k}$ = intervalo de tempo entre o término da operação da tarefa $J_{[j]}$ na máquina M_k e o início da operação da mesma tarefa na máquina $M_{(k+1)}$, ou seja, o tempo de espera da tarefa $J_{[j]}$ após o fim de seu processamento na máquina M_k e o início de seu processamento na máquina seguinte;

LB = *Lower Bound* = Limitante Inferior;

UB = *Upper Bound* = Limitante Superior;

D = Duração total da programação = *Makespan*.

4. Uma Propriedade Estrutural do Problema

Aqui apresenta-se uma propriedade estrutural do Problema de Programação (PPP) da Produção em sistemas de produção *Flow Shop* Permutacional com *m* máquinas, tendo o *Makespan* como critério de desempenho e a restrição de tempos de *setup* separados dos tempos de processamento, antecipatórios, podendo ser dependentes ou não da sequência de execução das tarefas. Esta, denominada 'Propriedade LBY', tem sua origem em investigações realizadas sobre as características estruturais deste Problema e sua solução.

A solução deste PPP, consiste em estabelecer dentre as (n!) sequências possíveis de execução das tarefas, aquela minimiza o *Makespan*.

A partir das notações pode-se escrever:

$$D = \sum_{j=1}^{n}\left(p_{1j} + s_{1j}\right) + \sum_{k=2}^{m} p_{k[n]} + \sum_{K=1}^{m-1} Y_{[n]k} \qquad (1)$$

A expressão (1) mostra que o *Makespan* é calculado pela soma de três parcelas: i) a soma dos tempos de processamento das *n* tarefas e dos tempos de *setup* para as *n* tarefas na primeira máquina, com um valor constante e, portanto, independente da sequência σ; ii) soma dos tempos de processamento da tarefa que ocupa a última posição na sequência σ ($J_{[n]}$), a partir da máquina 2 até a máquina *m*, sendo então dependente da sequência σ que for considerada; iii) soma dos intervalos de tempo entre as operações sucessivas da tarefa $J_{[n]}$, da máquina M_1 até a máquina M_m, também dependente da sequência σ que for considerada.

Fundamentando-se no estudo efetuado por Moccellin e Nagano (2007), que pode ser considerado uma extensão do estudo anteriormente realizado por Moccellin (1992), duas tarefas adjacentes quaisquer $J_{[j]}$ e $J_{[j+1]}$, aqui denominadas J_u e J_v, podem ser consideradas. Pode-se, também, considerar a atividade de *setup* nessa condição de tarefas sucessivas (independente ou não da sequência), designada de S_{uv}, ou seja, *setup* para a tarefa J_v após o processamento da tarefa J_u ter sido feito antes. Assim, S_{uv} pode ser considerada como se fosse mais uma tarefa a ser

executada entre J_u e J_v. Além disso, após a conclusão, em uma máquina qualquer, da tarefa J_u, essa máquina já pode ser preparada para processar a tarefa J_v, não necessitando aguardar o término da operação da tarefa J_v na máquina anterior. Isto é possível uma vez que está se tratando do *setup* antecipatório. Logo não existe tempo ocioso nas máquinas entre o término de J_u e o início do *setup* S_{uv}.

Desta maneira, considera-se que J_u, S_{uv} e J_v sejam três "tarefas" sucessivas em uma dada sequência de tarefas. Assim, expressões de viabilidade entre tarefas sucessivas são aplicadas aos pares (J_u, S_{uv}) e (S_{uv}, J_v), levando a (MOCCELLIN e NAGANO, 2007):

$$X_{[u\,Suv]k} + s_{k\,uv} + Y_{[u\,Suv]k} = Y_{[Stu\,u]k} + p_{k+1,u} + X_{[u\,Suv]k+1} \quad \text{(Entre } J_u \text{ e } S_{uv}\text{)} \qquad (2)$$

$$X_{[Suv\,v]k} + p_{k\,v} + Y_{[Suv\,v]k} = Y_{[u\,Suv]k} + s_{k+1,uv} + X_{[Suv\,v]k+1} \quad \text{(Entre } S_{uv} \text{ e } J_v\text{)} \qquad (3)$$

Onde, genericamente:

$X_{[xy]k}$ = intervalo de tempo na máquina M_k entre o término de uma atividade (*setup*/processamento) da tarefa J_x e o início de uma outra atividade (processamento da tarefa J_x /*setup* da tarefa J_y), quando a tarefa J_x é precedente direta da tarefa J_y;

$s_{k\,xy}$ = tempo de *setup* da máquina M_k para a execução da tarefa J_y, quando a tarefa J_x é sua precedente direta;

$Y_{[xy]k}$ = intervalo de tempo entre o término da atividade (*setup*/processamento) da tarefa J_y na máquina M_k e o início da atividade (*setup*/processamento) da mesma tarefa na máquina $M_{(k+1)}$, com a tarefa J_x precedendo diretamente a tarefa J_y.

Na expressão (2) tem-se que $X_{[u\,Suv]k} = X_{[u\,Suv]k+1} = 0$, o que leva:

$$s_{k\,uv} + Y_{[u\,Suv]k} = Y_{[Stu\,u]k} + p_{(k+1)\,u}$$

Portanto, $Y_{[u\,Suv]k} = Y_{[Stu\,u]k} + p_{(k+1)\,u} - s_{k\,uv}$ (4)

Na expressão (4) é importante observar que $Y_{[u\,Suv]k}$ não necessariamente mantém a condição de ser positivo ou nulo. Neste caso, a data de término do *setup* S_{uv} na máquina M_k é maior que a data de término desse *setup* na máquina seguinte $M_{(k+1)}$. Isto ocorre, pois, não existem restrições de precedência entre atividades de *setup* de máquinas sucessivas para o processamento de uma tarefa qualquer, o que não elimina a validade das expressões (2) a (4).

Substituindo-se $Y_{[u\,Suv]k}$ da expressão (4) na expressão (3):

$$X_{[Suv\,v]k} + (p_{k\,v} + s_{k\,uv}) + Y_{[Suv\,v]k} = Y_{[Stu\,u]k} + (p_{(k+1)u} + s_{(k+1)\,uv}) + X_{[Suv\,v]k+1} \qquad (5)$$

A expressão (5) viabiliza, para duas tarefas sucessivas J_u e J_v e o seu *setup* S_{uv}, as relações entre tempos de máquina parada e tempos de espera de operações sucessivas de uma mesma tarefa, para o problema em que os tempos de *setup* não são incluídos nos tempos de processamento.

Como $X_{[x\,y]k}$ e $Y_{[x\,y]k}$ da expressão (5) mantêm a condição de serem sempre positivos ou nulos, é possível completar a expressão (5), com as seguintes expressões de viabilidade:

Se $X_{[Suv\,v]k+1} > 0$ então $Y_{[Suv\,v]k} = 0$, e (6)
Se $Y_{[Suv\,v]k} > 0$ então $X_{[Suv\,v]k+1} = 0$. (7)

As expressões (6) e (7), demonstram que os termos $X_{[Suv\ v]k+1}$ e $Y_{[Suv\ v]k}$ não podem ser simultaneamente positivos.

Da expressão (5), pode-se, de maneira semelhante ao desenvolvimento efetuado por Nagano (1999), desenvolver uma propriedade LBY.

Como $Y_{[stu\ u]k} \geq 0$, pode-se escrever, a partir da expressão (5):

$$X_{[Suv\ v]k+1} - Y_{[Suv\ v]k} \leq X_{[Suv\ v]k} + (p_{k\ v} + s_{k\ uv}) - (p_{(k+1)\ u} + s_{(k+1)\ uv}) \quad (8)$$

Considerando-se $UBX_{[Suv\ v]k}$ um limitante superior para $X_{[Suv\ v]k}$ (ou seja, $X_{[Suv\ v]k} \leq UBX_{[Suv\ v]k}$), pode-se reescrever a expressão (8), da seguinte maneira:

$$X_{[Suv\ v]k+1} - Y_{[Suv\ v]k} \leq UBX_{[Suv\ v]k} + (p_{k\ v} + s_{k\ uv}) - (p_{(k+1)\ u} + s_{(k+1)\ uv}) \quad (9)$$

Ou seja:

$$Y_{[Suv\ v]k} \geq (p_{(k+1)\ u} + s_{(k+1)\ uv}) - (p_{k\ v} + s_{k\ uv}) - UBX_{[Suv\ v]k} + X_{[Suv\ v]k+1} \quad (10)$$

Sendo $X_{[Suv\ v]k+1} \geq 0$, tem-se que:

$$Y_{[Suv\ v]k} \geq (p_{(k+1)\ u} + s_{(k+1)\ uv}) - (p_{k\ v} + s_{kuv}) - UBX_{[Suv\ v]k} \quad (11)$$

A partir da expressão (11) e das expressões de viabilidade (6) e (7), pode-se estabelecer a Propriedade LBY, que fornece um limitante inferior para $Y_{[Suv\ v]k}$ (ou seja, $Y_{[Suv\ v]k} \geq LBY_{[Suv\ v]k}$), dado por:

$$LBY_{[Suv\ v]k} = \text{Max}\ \{0, (p_{(k+1)\ u} + s_{(k+1)\ uv}) - (p_{k\ v} + s_{k\ uv}) - UBX_{[Suv\ v]k}\} \quad (12)$$

Sendo k = 1,2, ..., (*m*-1)

Onde, $UBX_{[Suv\ v]k}$ é um limitante superior do intervalo de tempo entre o término do *setup* e o início do processamento de J_v, na máquina M_k, quando a tarefa J_u é sua precedente direta na sequência σ, dado por (MOCCELLIN e NAGANO, 2007):

$$UBX_{[Suv\ v]k} = \text{Max}\ \{0, UBX_{[Suv\ v]k-1} + (p_{(k-1)\ v} + s_{(k-1)\ uv}) - (p_{k\ u} + s_{k\ uv})\} \quad (13)$$

Sendo k = 1, 2, ..., *m*, com $UBX_{[Suv\ v]1} = 0$.

Assim, $LBY_{[Suv\ v]k}$ é um limitante inferior do tempo de espera da operação da tarefa J_v entre as máquinas M_k e $M_{(k+1)}$, quando a tarefa J_u é sua precedente direta na sequência σ e S_{uv} o seu *setup*.

O valor de $LBY_{[Suv\ v]k}$ pode, então, ser calculado para todos os *n* (*n*-1) arranjos de pares ordenados de tarefas do conjunto de *n* tarefas $\{J_1, J_2, ..., J_j, ..., J_n\}$, tomadas duas a duas.

A somatória dos $LBY_{[Suv\ v]k}$, para k = 1, 2, ..., (*m*-1), leva a:

$$\sum_{k=1}^{m-1} LBY_{[Suv\ v]k} = LBY_{[Suv\ v]1} + LBY_{[Suv\ v]2} + ... + LBY_{[Suv\ v]m-1} \quad (14)$$

A expressão (14) fornece um limitante inferior da soma dos tempos de espera da tarefa J_v entre as *m* sucessivas máquinas do *Flow Shop*, quando a tarefa J_u for sua precedente direta e S_{uv} o seu *setup*.

5. Proposição de Novos Métodos Heurísticos

Como uma primeira aplicação da 'Propriedade LBY', foram propostos dois novos métodos heurísticos, denominados métodos BM, um construtivo (BM_c) e um melhorativo (BM_m) para o Problema de Programação da Produção considerado. Estes são descritos abaixo.

ALGORITMO DO MÉTODO BM_c

<u>Procedimento 1</u> – **Procedimento de Ordenação Inicial das Tarefas**

Passo 1:

Passo 1.1: Seja $[\sum J_v]_{nxn}$ a matriz, formada pela soma dos tempos de processamento e pela soma dos tempos de *setup* das tarefas J_v, para v = 1, 2, ..., *n*, em que os (*n*-1) elementos de uma coluna correspondem, respectivamente, à somatória dos tempos de processamento e de *setup* de uma determinada tarefa para as *m* máquinas.

Passo 1.2: Para todas as tarefas J_v, sendo v = 1, 2, ..., *n*, calcule $\sum_{k=1}^{m-1} LBY_{[Suv\ v]k}$,

Sendo: $UBX_{[Suv\ v]1} = 0$;

$UBX_{[Suv\ v]k} = \text{Max}\ \{0, UBX_{[Suv\ v]k-1} + (p_{(k-1)\ v} + s_{(k-1)\ uv}) - (p_{k\ u} + s_{k\ uv})\}$ para k = 1, 2, ..., *m*; e,

$LBY_{[Suv\ v]k} = \text{Max}\ \{0, (p_{(k+1)\ u} + s_{(k+1)\ uv}) - (p_{k\ v} + s_{k\ uv}) - UBX_{[Suv\ v]k}\}$ para k = 1, 2, ..., (*m* –1).

Passo 1.3: Seja $[\sum LBY_{Suv\ v}]_{nxn}$ a matriz $\sum_{k=1}^{m-1} LBY_{[Suv\ v]k}$.

Passo 1.4: Faça $[\sum J_v]_{nxn} - [\sum LBY_{Suv\ v}]_{nxn}$, obtendo a matriz $[\Omega]_{nxn}$, em que a posição de cada elemento linha-coluna representa respectivamente as tarefas J_u e J_v, conforme Figura 1.

Figura 1 - Matriz $[\Omega]_{nxn}$

$$[\Omega]_{nxn} = \begin{bmatrix} - & \sum_{k=1}^{m} J_2 - \sum_{k=1}^{m-1} LBY_{[s12,2]k} & \cdots & \cdots \sum_{k=1}^{m} J_n - \sum_{k=1}^{m-1} LBY_{[s1n,n]k} \\ \sum_{k=1}^{m} J_1 - \sum_{k=1}^{m-1} LBY_{[s21,1]k} & - & \cdots & \cdots \sum_{k=1}^{m} J_n - \sum_{k=1}^{m-1} LBY_{[s2n,n]k} \\ \vdots & \vdots & & \vdots \\ \sum_{k=1}^{m} J_1 - \sum_{k=1}^{m-1} LBY_{[sn1,1]k} & \sum_{k=1}^{m} J_2 - \sum_{k=1}^{m-1} LBY_{[sn2,2]k} & \cdots & \cdots - \end{bmatrix}$$

Passo 2: (Caso existam valores iguais pegue o último encontrado)

Passo 2.1: Encontre o maior elemento da matriz $[\Omega]_{nxn}$;

Passo 2.2: Identifique as tarefas adjacentes J_u e J_v, que correspondem, respectivamente, aos elementos linha-coluna da matriz. Assim, J_u ocupa a primeira posição na ordenação, enquanto que, J_v a segunda posição.

Passo 2.3: Encontre a nova tarefa J'_v, ainda não inserida na ordenação, verificando-se o maior elemento da linha da matriz correspondente a última tarefa já ordenada J_v, até que todas as tarefas sejam ordenadas.

Procedimento 2 – **Procedimento de Inserção de Tarefas**

Os Passos do Procedimento de Inserção de Tarefas são os mesmos daqueles do procedimento de inserção de tarefas utilizado nos métodos NEH e N&M, diferindo apenas no cálculo do *Makespan*.

ALGORITMO DO MÉTODO BM_m

Fase 1 – Obtenção da Sequência "Semente"

A Fase de Obtenção da Sequência "Semente" do Método BM_m corresponde ao Método BM_c, sendo que a MSEO é a sequência "semente" $\sum^s$, para ser dada como entrada para a fase de melhoramento.

Fase 2 – Melhoramento

A Fase de Melhoramento corresponde à fase de melhoramento dos métodos de Rajendran e Ziegler (1997).

6. Experimentação Computacional

Os Métodos propostos BM_c e BM_m foram avaliados em relação aos Métodos existentes de Cao e Bedworth (1992) e Rajendran e Ziegler (1997), denominados CB, RZ_1, RZ_2 e RZ_3, respectivamente.

Os métodos foram codificados em *Borland Delphi 6*, sistema operacional *Windowns XP* (Versão *2002 SP2*), processador *Pentium 4 da Intel* com 3.2 GHz de freqüência, 1 GB de memória RAM e disco rígido de 60 GB.

Os tempos de processamento e de *setup*, números inteiros, gerados aleatoriamente, foram distribuídos nos intervalos [1, 99] e [1, 49], [1, 99], [51, 149] e [101, 199], o que levou, assim como em Rajendran e Ziegler (1997), a 4 relações entre os intervalos de distribuição dos tempos de *setup* (IDs_{kj}) e o intervalo de distribuição dos tempos de processamento (IDp_{kj}), aqui denominadas Relações IDs_{kj}-IDp_{kj}.

Analisou-se um total de 3600 problemas, divididos em 30 classes de problemas, de acordo com o número de tarefas (*n*) e o número de máquinas (*m*), subdivididos em 2 grupos de problemas, constituídos, cada um, de 15 classes de problemas. Para cada classe de problemas, foram testados, objetivando a minimização do erro amostral, 120 problemas, gerados aleatoriamente. No primeiro grupo, testou-se 1800 problemas, considerados de pequeno porte, com $n \in \{4, 6, 7\}$ e $m \in \{5, 10, 15, 20, 25\}$. No segundo, com $n \in \{20, 40, 60\}$ e $m \in \{5, 10, 15, 20, 25\}$, foram testados 1800 problemas, considerados de grande porte. Os 120 problemas

testados, em cada uma destas classes, foram subdivididos em 4 tipos de 30 de problemas cada, onde os valores dos tempos *setup* e os tempos de processamento foram gerados conforme as 4 Relações IDs_{kj}-IDp_{kj}.

Os resultados obtidos na experimentação computacional foram analisados em relação à porcentagem de sucesso, desvio relativo médio (%) e tempo médio de computação (ms).

Os resultados obtidos na experimentação computacional foram agrupados em relação ao número de tarefas.

No caso da porcentagem de sucesso os resultados foram analisados em função das relações entre tempos de processamento e tempo de *setup* e em relação às classes (*n*, *m*). No primeiro caso, o objetivo foi avaliar a influência do número de máquinas *m* no desempenho de cada método. Enquanto que, no segundo caso, o objetivo foi avaliar a influência das Relações IDs_{kj}-IDp_{kj} em cada uma das classes (*n*, *m*). Também apresenta-se uma análise da porcentagem de sucesso nos problemas de pequeno e grande porte e em todos os problemas.

Quanto ao desvio relativo médio (%) e ao tempo médio de computação (ms), os resultados foram analisados apenas em função das classes (*n*, *m*).

Neste artigo, apresenta-se os resultados agrupando-se todos os problemas.

6.1 Porcentagem de Sucesso

A Tabela 1 apresenta os resultados da porcentagem de sucesso quando todos os problemas são considerados.

Tabela 1 - Porcentagem de sucesso para todos os problemas

	MÉTODOS			
	CB	**RZ_3**	**BM_c**	**BM_m**
Total de problemas	1122	2809	2375	3164
%	31,17	78,03	65,97	87,89

A porcentagem global de sucesso, apresentada na Tabela 1, comprova a superioridade do método proposto BM_m.

As análises referentes à porcentagem de sucesso mostraram que, com o aumento do número de tarefas (*n*), os métodos BM_c e BM_m melhoram muito o seu desempenho, ocorrendo o contrário com os métodos CB e RZ_3.

Com o aumento do número de tarefas (*n*) a variabilidade da porcentagem de sucesso dos métodos CB e RZ_3 aumenta. Por outro lado, a variabilidade da porcentagem de sucesso dos métodos BM_c e BM_m não são influenciadas pelo aumento do número de tarefas (*n*), o que mostra a estabilidade desses métodos.

De maneira geral, as curvas de desempenho dos métodos mantiveram o mesmo comportamento com o aumento do número de máquinas *m* e nas Relações IDs_{kj}-IDp_{kj}, o que indica que os desempenhos dos métodos em relação à porcentagem de sucesso não são influenciados por esses.

6.2 Desvio Relativo Médio (%)

Quanto ao desvio relativo médio (%), não houve, em ambos os métodos, influência do número de máquinas *m*. Além disso, o desvio relativo médio (%) não é significante, uma vez que, não ultrapassam, no pior caso, 2% em média para todos os problemas.

Nas 30 classes *n*, *m* analisadas o método proposto BM_m apresentou o menor desvio relativo em 15 classes, o método RZ_3 em 14, havendo 1 empate. Estes resultados ressaltam a qualidade do método proposto BM_m.

6.3 Tempo Médio de Computação (%)

Durante a tabulação dos dados, foi possível observar que, numa mesma classe *n*, *m*, a diferença do tempo médio de computação (ms), entre os problemas resolvidos com base nas 4 Relações IDs_{kj}-IDp_{kj}, não foi significativa, tanto no caso de problemas de pequeno porte quanto nos de grande porte.

Os tempos médios de computação (ms) de ambos os métodos são influenciados tanto o número de máquinas (*m*) quanto o número de tarefas (*n*).

Por outro lado, os valores dos tempos médios de computação do método RZ_3 são explicados pelo fato de que ele executa os métodos RZ_1 e RZ_2, ambos melhorativos, para selecionar aquele que fornece a melhor solução, usando essa solução como "semente" para aplicar a fase de melhoramento novamente.

Os resultados referentes aos tempos médios de computação enfatizam ainda mais a qualidade dos métodos propostos BM, uma vez que, estes possuem alta porcentagem de sucesso e baixo desvio relativo médio (%).

7. Considerações Finais

Nesta pesquisa existem alguns aspectos principais que podem ser destacados.

O primeiro aspecto refere-se ao fato de que, neste trabalho, os tempos de *setup* são considerados separados dos tempos de processamento.

Outro aspecto a se destacar, e com certeza, o mais relevante, é a 'Propriedade LBY', que constitui uma contribuição ao conhecimento, já existente, sobre o Problema de Programação da Produção em sistemas de produção *Flow Shop* Permutacional.

É importante destacar, também, que, neste trabalho apresenta-se uma primeira aplicação bem sucedida da 'Propriedade LBY', que corresponde aos Métodos propostos BM, que mostrara-se bastante eficientes em fornecer a solução do Problema de Programação da Produção.

Quanto aos Métodos heurísticos BM, os resultados, obtidos na experimentação computacional, evidenciam a qualidade do Método construtivo proposto BM_c e o melhor desempenho do Método melhorativo BM_m, em relação a ambos os Métodos CB e RZ_3.

Esta pesquisa reforça a qualidade do procedimento de inserção de tarefas utilizado nos Métodos NEH e N&M.

Muitas sugestões sobre o desenvolvimento de futuros trabalhos podem ser feitas. Sugere-se realizar um estudo, para avaliar a performance dos métodos BM em alcançar a solução ótima. A realização de um estudo detalhado da influência de ordenações iniciais para métodos melhorativos. O desenvolvimento de um método *Branch-and-Bound*, utilizando-se a 'Propriedade LBY', para obter um eficiente *Lower Bound* do *Makespan*. A utilização da 'Propriedade LBY' para conceber um método heurístico para o caso dos tempos de *setup* serem dependentes da sequência de execução das tarefas. Sugere-se também, a realização de adaptações da 'Propriedade LBY' para os casos em que os tempos de remoção são diferentes de zero e não podem ser considerados como parte do tempo de processamento. E, a criação de novas propriedades considerando outras medidas de desempenho, que não o *Makespan*.

Agradecimentos

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

References

Allahverdi, A. *et al.* A survey of scheduling problems with setup times or costs. *European Journal of Operational Research, Amsterdam*, 187, 3, 985-1032, 2008.

Allahverdi, A., Gupta, J. N. D. e Aldowaisan, T. A review of scheduling research involving setup considerations. Omega, *The International Journal of Management Science*, Oxford, 27, 219-239, 1999.

Allahverdi, A. Scheduling in stochastic flowshop with independent setup, processing and removal times. *Computers & Operations Research*, Oxford, 24, 10, 955-960, 1997.

Allahverdi, A. (1995). Two-stage production scheduling with separated set-up times and stochastic breakdowns. *Journal of the Operational Research Society*, 46, 7, 896-904, 1995.

Baker, K. R. *Introduction to sequencing and scheduling*. New York: John Wiley & Sons, Inc, 1974.

Bedworth, D. D. e Bailey, J. E. *Integrated Production Control Systems*: management, analysis, design. 2 ed. New York: John Wiley & Sons, Inc, 1987.

Boiko, T. J. P. e Morais, M. F. A atividade de programação da produção sobre a ótica da Pesquisa Operacional: uma abordagem teórico conceitual. In: ENCONTRO TECNOLÓGICO (VI ENTEC), 6, 2009, Campo Mourão – PR. *Anais*... Campo Mourão: 2009.

Brucker, P., Knust, S. e Wang, G. Complexity results for flow-shop problems with a single-server. European Journal of Operational Research, 165, 2, 398-407, 2005.

Cao, J. e Bedworth, D. D. Flow shop scheduling in serial multi-product process with transfer and set-up times. *International Journal of Production Research*, 30, n. 8, 1819-1830, 1992.

Cheng, T. C. E., Gupta, J. N. D. e Wang, G. A review of flowshop scheduling research with setup times. *Production and Operations Management*, 9, 3, 262-282, 2000.

Cheng, T. C. E., Wang, G. e Sriskandarjah, C. One-operator-two-machine flowsop with setup and dismouting times. *Computers and Operations Research*, 26, 715-730, 1999.

French. S. *Sequencing and scheduling: an introduction to the mathematics of the job shop.* New York: Wiley, 1982.

Glass, C. A., Shafransky, Y. M. e Strusevich, V. A. Scheduling for parallel dedicated machines with a single server. *Naval Research Logistics*, 47, 4, 304-328, 2000.

Gupta, J. N. D. e Stafford Jr., E. F. Flowshop scheduling research after five decades. *European Journal of Operational Research*, 169, 699-711, 2006.

Gupta, J. N. D. Optimal scheduling in multistage flow-shop. *AIIE Trans*, 4, 238-243, 1972.

Han, W. e Dejax, P. An Efficient Heuristic Based on Machine Workload for the Flowshop Scheduling Problem with Setup and Removal Times. *Annals of Operation Research*, 50, 263-279, 1994.

Khurana, K e Bagga, P. C. Minimizing the Makespan in a Two-Machine Flowshop with Time Lags and Setup Conditions. *Zeitschrift Operations Research*, 28, 2, 163-174, 1984.

Khurana, K e Bagga, P. C. Scheduling of Job-Block with Dedadline in *n* x 2 Flowshop Problem with Separated Setup Times. *Indian Journal of Pure Applied Mathematics*, 16, 3, 213-224, 1985.

Kogan, K. e Levner, E. A polynominal algorithm for scheduling small-scale manufacturing served by multiple robots. *Computers and Operations Research*, 25, 53-62, 1998.

Levner, E., Kogan, K. e Maimon O. Flowshop scheduling of robotic cells with jo-dependent transportation and setup effects. *Journal of the Operational Research Society*, 47, 1447-1455, 1995.

Logendran, R. e Sriskandarajah, C. Two-machine group scheduling problem with blocking and anticipatory setups. *European Journal of Operational Research*, 69, 1, 467-481, 1993.

Maccarthy, B. L. e Liu, J. Y. Adressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. *International Journal of Production Research*, 31, 1, 59-79, 1993.

Moccellin, J. V. A new heurist method for the permutation flow shop scheduling problem. *Journal of the Operational Research Society*, 46, 883-886, 1995.

Moccellin, J. V. e Nagano, M. S. Uma propriedade estrutural do problema de programação da produção *flow shop* permutacional com tempos de *setup*. *Pesquisa Operacional*, 27, 3, 487-515, set.-dez. 2007.

Moccellin, J. V. *Uma contribuição à Programação de Operações em Sistemas de Produção Intermitente 'Flow-Shop'*. Tese (Livre-Docência), Universidade de São Paulo, Escola de Engenharia de São Carlos, 1992.

Nabeshima, I. e Maruyama, S. Two- and Three-Machine Flowshop Make-span Scheduling Problems with Additional Times Separated from Processing Times. *Journal of the Operations Research Society of Japan*, 27, 4, 348-366, 1984.

Nagano, M. S. e Moccellin, J. V. A high quality solution constructive heuristic for flow shop sequencing. *Journal of the Operational Research Society*, 53, 12, 1374-1379, 2002.

Nagano, M. S. *Um novo método heurístico construtivo de alto desempenho para a programação de operações flow-shop permutacional*. Tese (Doutorado em Engenharia de Produção), Universidade de São Paulo, Escola de Engenharia de São Carlos, 1999.

Nawaz, M., Enscore Jr., E. E. e Ham, I. A heuristic algorithm for de *m*-machine, *n*-job flow-shop sequencing problem. Omega, *The International Journal of Management Science*, 11, 1, 91-95, 1983.

Proust, C., Gupta, J. N. D. e Deschamps, V. Flowshops scheduling with set-up, processing and removal times separated. *International Journal of Production Research*, 29, 3, 479-493, 1991.

Rajendran, C. e Ziegler, H. Heuristics for scheduling in a flowshop with setup, processing and removal times separated. *Production Planning & Control*, 8, 6, 568-576, 1997.

Ruiz, R. e Maroto, C. A comprehensive review and evaluation of permutation flowshop heuristics. *European Journal of Operational Research*, 165, 479-494, 2005.

Yang, Wen-Hwa e Liao, Ching-Jong. Survey of scheduling research involving setup times. *International Journal of Systems Science*, 30, 2, 143-155, 1999.

Yoshida, T. e Hitomi, K. Optimal two- stage production scheduling with setup times separated. *AIIE Transactions*, 11, 261-263, 1979.

Sule, D. R. Sequencing *n* jobs on two machines with setup, processing and removal times separated. *Naval Research Logistics Quarterly*, 29, 517-519, 1982.

Sule, D. R. e Huang, K. Y. Sequency on two and three machines with setup, processing and removal times separated. *International Journal of Production Research*, 21, 723-732, 1983.

Su, Ling-Huey e Chou, Fuh-Der. Heuristic for scheduling in a two-machine bicriteria dynamic flowshop with setup and processing times separated. *Production Planning & Control*, 11, 8, 806-819, 2000.

Szwarc, W. Flow shop problems with time lags. *Management Science*, Rhode Island, 29, 4, 477-480, 1983.

Szwarc, W. The Flowshop Problem with Time-Lags and Separated Setup Times. *Zeitschrift Operations Research*, 30, B15-B22, 1986.

Szwarc, W. e Gupta, J. N. D. A flow-Shop problem with sequence-dependent additive setup times. *Naval Research Logistics*. 34, 619-627, 1987.

Taillard, E. Benchmarks for basic scheduling problems. *European Journal of Operational Research*, 64, 278-285, 1993.