

PROGRAMAÇÃO DA PRODUÇÃO EM SISTEMAS POR BATELADAS: UM ESTUDO PARA A OTIMIZAÇÃO DO *MAKESPAN*

R. P. da ROCHA¹, M. de F. MORAIS¹, M. A. da S. S. RAVAGNANI², C. M. G. ANDRADE², C. M. P. R. PARAÍSO²

¹ Universidade Estadual do Paraná Campus de Campo Mourão, Colegiado de Engenharia de Produção Agroindustrial

² Universidade Estadual de Maringá, Departamento de Engenharia Química
E-mail para contato: ronypeterson_eng@hotmail.com

RESUMO – Este trabalho visa apresentar um estudo comparativo da utilização do *Excel/VBA* e do *GAMS* para um Problema de Programação da Produção em um ambiente *Flow Shop* Permutacional, cujo o objetivo é a Minimização do Tempo Total de Conclusão das Tarefas (*Makespan*). O estudo abrangeu o sequenciamento da produção em cenários de n (tarefas) $\in \{5\}$ e m (máquinas) $\in \{2,3,4, \dots, 98\}$. As experimentações foram implementadas no *Excel* e no *GAMS*, utilizando um computador com processador *Intel Corel 2 Duo*, 2,20 GHz de frequência, 1,75 GB de memória RAM. As análises foram realizadas em relação ao critério de Porcentagem de Sucesso (%S), Tempo Médio Computacional (TMC), Minimização do Tempo Total de Conclusão das Tarefas (*Makespan*) e Similaridades/Não Similaridades da Sequencia Ótima de Programação das Ordens de Processamento. O *GAMS*, quando comparado com o *Excel*, apresentou resultados superiores, referentes ao quesito porcentagem de sucesso, otimização da função objetivo e tempo médio computacional.

1. INTRODUÇÃO

Na indústria química, as necessidades de Programação da Produção (PP) são cada vez mais acentuadas, principalmente em Sistemas de Produção (SP) por Bateladas. A produção em Bateladas pode ser descrita como um SP orientado à receita, classificado conforme a quantidade de estágios envolvidos no processo, tais como: Estágio único, subdividido em unidades únicas ou paralelas; múltiplo estágios, que subdivide-se em multipropósito (*Job-Shop*) e multiprodutos (*Flow-Shop*) (MENDEZ *et al.*, 2005).

O problema de PP tratado neste trabalho é do tipo Batelada - Multiprodutos - *Flow Shop* Permutacional. *Flow-Shop*, na visão de Maccarthy e Liu (1993) é um processo onde as tarefas possuem o mesmo roteiro de processamento em todas as máquinas e o número de máquinas em cada estágio de produção é igual a um. O ambiente de PP *Flow-Shop* Permutacional ou Batelada-Multiprodutos é um tipo de SP empregado para um conjunto de produtos cuja estrutura de receita é a

mesma. Muitas indústrias de processo químico, tais como óleo e tintas, indústrias farmacêuticas e o setor de química fina se encaixam nesta categoria (POLON, 2010).

Uma forma de resolver os problemas envolvidos com a PP dos SP por Batelada é tornar esses problemas como um caso de Programação Linear Inteira Mista (PLIM). Taha (2008); Laesch e Hein (2009) e Hillier (2010) afirmam que a PLIM é uma técnica de otimização para resolver sistemas de equações lineares, de maximização ou minimização de um determinado objetivo, quando apenas algumas variáveis apresentam valores inteiros.

Assim, o presente artigo objetiva apresentar um estudo da utilização da PLIM para um problema de PP em um ambiente *Flow Shop* Permutacional ou Batelada, cujo intuito da otimização é a Minimização do Tempo Total de Conclusão das Tarefas (*Makespan*) realizadas no sequenciamento de ordens de produção em máquinas. O estudo faz um comparativo da utilização do *Excel/VBA* e do *GAMS* neste tipo de problema.

2. MATERIAIS E MÉTODOS

Para o desenvolvimento do modelo de Programação da Produção (PP) proposto neste estudo, inicialmente escolheu-se um problema hipotético envolvendo um ambiente de PP denominado *Flow Shop* Permutacional (FSP). Nesta fase, definiu-se a função objetivo, tomando o caso de PP como um problema de PLIM. O FSP estudado, diz respeito aos processos que apresentam roteiros de produção definidos como iguais para todos os produtos (Plantas Multiprodutos), isto é, cada produto processado segue um roteiro de produção igual.

Considerou-se o sequenciamento da produção de (n) produtos em (m) máquinas, por meio de um roteiro definido de produção com as seguintes restrições: (i) A armazenagem de produtos intermediários não deve estar disponível entre as unidades de processamento, isto é, caso um determinado produto esteja processado na unidade j e a unidade j+1 não esteja disponível no momento desta conclusão, o produto pronto deve ser mantido na unidade j, até a unidade j+1 estar desocupada; (ii) Ao finalizar o processamento de um produto na última unidade (equipamento), esse é imediatamente enviado ao estoque de produtos acabados, neste caso, assume-se que os tempos necessários para transferir produtos de uma unidade para outra são negligenciáveis comparado com o tempo de processamento; (iii) Todas as unidades estão inicialmente vazias no tempo zero e a manufatura de qualquer produto pode ser atrasada numa quantidade de tempo arbitraria para mantê-lo na unidade anterior e; (iv) O ordenamento das tarefas em cada processador é o mesmo.

No ambiente de Programação da Produção em Processo por Batelada caracterizado como FSP modelou-se o estudo abrangendo cenários de $n \in \{5\}$ e $m \in \{2,3,4, \dots, 98\}$, sendo que n representa o número de produtos (tarefas) e m o número máximo de máquinas analisadas. Na modelagem matemática destes cenários, foram utilizadas as Equação 1, 2, 3, 4, 5, 6, 7, 8 e 9.

A Equação 1 representa o objetivo do problema, onde N diz respeito aos número de produtos ou tarefas e M ao número de máquinas ou processadores.

Minimizar C_{NM} (1)

As Equações 2 e 3, são de ordens binárias, sendo que o subscrito i representa a tarefa a ser processada no recurso e k representa a posição dessa mesma na ordem de sequenciamento, X_{ik} é uma variável binária definida como: $X_{ik} = 1$ se a tarefa i está na posição k e $X_{ik} = 0$, caso contrário.

$$\sum_i X_{ik} = 1 \quad \forall k \quad (2)$$

$$\sum_k X_{ik} = 1 \quad \forall i \quad (3)$$

A Equação 4 representa outra restrição do problema, onde: $C_{k,j}$ = tempo de fim de processamento no processador j da tarefa ocupando a posição k na sequência; j = processadores ou estágios; k = posição na sequência; N = número de tarefas; $X_{s,k}$ = variável binária e $TP_{s,j}$ é o tempo de processamento da tarefa i no processador j .

$$C_{k,j} \geq C_{k-1,j} + \sum_{s=1}^N X_{s,k} TP_{s,j} \quad \forall j; k = 2, \dots, N \quad (4)$$

Equação 5 mostra que o *Makespan* da Tarefa de ordem k no processador j é maior ou igual ao *Makespan* da Tarefa de ordem k no processador $j-1$ mais o somatório do produto entre a variável binária $X_{s,k}$ e o tempo de processamento $TP_{s,j}$, tomando o início no processador (2) e o término no processador (3).

$$C_{k,j} \geq C_{k,j-1} + \sum_{s=1}^N X_{s,k} TP_{s,j} \quad \forall j; j = 2, \dots, M \quad (5)$$

A Equação 6 apresenta a restrição que o *Makespan* da tarefa posicionada na sequência 1 da PP no processador j é maior ou igual ao *Makespan* da tarefa posicionada na sequência 1 da PP no processador $j-1$ mais o somatório do produto da variável binária $X_{s,1}$ e o tempo de processamento $TP_{s,j}$. Nesta Equação, o processador j deve iniciar no processador 2 e terminar no processador 3, conforme o modelo proposto no início.

$$C_{1,j} \geq C_{1,j-1} + \sum_{s=1}^N X_{s,1} TP_{s,j} \quad j = 2, \dots, M \quad (6)$$

A Equação 7 mostra que o *Makespan* da tarefa sequenciada na ordem 1 no processador 1 deve ser maior ou igual ao produto da variável binária $(X_{1,1}, X_{2,1}, \dots, X_{N,1})$ e os tempos de processamentos $(TP_{1,1}, TP_{2,1}, \dots, TP_{N,1})$.

$$C_{1,1} \geq \sum_{s=1}^N X_{s,1} TP_{s,j} \quad (7)$$

A Equação 8 mostra que o *Makespan* sequenciado na ordem k no processador j deve ser maior ou igual ao *Makespan* sequenciado na ordem $k-1$ no processador $j+1$, partindo da ordem de sequência 1 no processador 1.

$$C_{k,j} \geq C_{k-1,j+1} \quad k = 1, \dots, N \quad j = 1, \dots, M - 1 \quad (8)$$

A Equação 9 diz respeito à questão de não negatividade do modelo.

$$C_{k,j} \geq 0 \quad \forall k, \forall j \quad (9)$$

Todas as equações matemáticas do modelo foram transcritas primeiramente para planilhas do *Excel* e resolvidas pelo aplicativo *Solver* e, posteriormente, resolvidas na linguagem GAMS. Na experimentação computacional do modelo pela planilha do *Excel* e pelo GAMS (Versão *Free*) foi utilizado um computador com processador *Intel Corel 2 Duo*, 2,20 GHz de frequência 1,75 GB de memória RAM. Optou-se utilizar a Versão *Free* do GAMS, por estar disponível facilmente, assim como o *Excel*. Utilizou-se o *Solver* CPLEX 12.3.0.

A análise geral dos resultados obtidos com a utilização do *Excel/VBA* e do GAMS foi realizada mediante uma comparação, em termos de critério de Porcentagem de Sucesso (%S), Tempo Médio Computacional (TMC), Minimização do Tempo Total de Conclusão das Tarefas (*Makespan*) e Similaridades/Não Similaridades da Sequencia Ótima de Programação das Ordens de Processamento do modelo.

O desempenho do modelo testado para as classes $n \in \{5\}$, em relação ao critério de desempenho Porcentagem de Sucesso (%) foi calculado pela quantidade de vezes em que a Planilha do *Excel* ou o GAMS forneceu a otimização da função objetivo C_{NM} (Equação 1) do modelo, dividido pelo total de problemas analisados por subclasse (Equação 10):

$$\%S_{PEIA/GAMS} = \frac{q_{PEIA/GAMS}}{T} \quad (10)$$

Onde: $q_{PEIA/GAMS}$: quantidade de vezes em que a Planilha do *Excel* ou o GAMS forneceu uma solução otimizada; T : total de problemas analisados na subclasse (número de produtos x número de máquinas x intervalo de tempo).

O tempo médio de computação do modelo da Planilha do *Excel* e do GAMS ($TMC_{PEIA/GAMS}$) é calculado pela soma dos tempos de computação de cada cenário de classe de problemas testado na Planilha e no GAMS, dividido pelo número total de problemas resolvidos (11).

$$TMC_{PEIA/GAMS} = \frac{\sum_{P=1}^P TC_{PEIA/GAMS}}{P} \quad (11)$$

Onde: TC_{PEIA} : tempo de computação (s) em que a Planilha do *Excel* e o GAMS levou para apresentar a resposta do *Makespan* de um determinado problema; P : total de problemas resolvidos.

O critério de Minimização do Tempo Total de Conclusão das Tarefas (*Makespan*) foi realizado de forma comparativa, isto é, foram analisados se os cenários testados no GAMS com relação aos resultados do *Makespan* eram iguais, maiores ou menores que os cenários testados no *Excel/VBA*.

A Similaridades/Não Similaridades da Sequencia Ótima de Programação das Ordens de Processamento do modelo também foi estudada. Esta diz respeito a uma análise do resultado obtido

no GAMS e no *Excel/VBA* com relação a sequência de programação que minimiza a função objetivo (*Makespan*). Sendo assim, analisou comparativamente as sequências geradas no GAMS e no *Excel/VBA*, com o intuito de verificar se essas eram as mesmas ou não.

3. RESULTADOS E DISCUSSÕES

Neste estudo, procurou-se encontrar a partir do conjunto de tarefas (n) e máquinas (m) os seus respectivos limites máximos de máquinas (m) a serem programados no *Excel/VBA* e no GAMS (Versão *Free*). Conforme testes realizados em ambos, encontrou-se o limite de até 14 problemas para a classe de 5 tarefas no GAMS (Versão *Free*) ($\{5,2\}, \dots, \{5,15\}$) e 34 no *Excel/VBA* ($\{5,2\}, \dots, \{5,35\}$).

Observando os limites apresentados nas classes de problemas $n \in \{5\}$ e $m \in \{2,3,4,5,6,7,8,9,10,15,20,25,30,40,60, \dots, 98\}$, vê-se uma redução do número de processadores (m) a partir do aumento do número de tarefas (n), tanto em relação ao *Excel/VBA* como no GAMS. Após analisar os limites de programação do GAMS e do *Excel/VBA*, realizou-se testes para analisar o desempenho do GAMS e do *Excel/VBA* nessas classes, no intervalo de tempo [1,99]. Na análise do desempenho do modelo no *Excel*, realizada para a classe 5 (Número de Tarefas (n) igual a 5), foram testadas 10 cenários para cada subclasses (n, m) apresentadas na Tabela 1, totalizando 700 problemas.

Tabela 1 - Classe $n = 5$ de problemas analisados

CLASSE (n)	SUBCLASSE (n, m)	INTERVALO DE TEMPO
5	$\{5,2\}; \{5,3\}; \{5,4\}; \{5,5\}; \{5,6\}; \{5,7\}; \{5,8\}; \{5,9\};$ $\{5,10\}; \{5,15\}; \{5,20\}; \{5,25\}; \{5,30\}; \{5,35\}.$	[1,8]; [1,12]; [1,24]; [1,45] e [1,99]

No intervalo de tempo [1,8], as subclasses $\{5,2\}, \{5,3\}, \{5,4\}, \{5,5\}, \{5,6\}$ e $\{5,10\}$, mostraram que 100% dos cenários testados no *Excel* (10 cenários/subclasses) apresentaram a otimização da função objetivo do problema de PPPB. As subclasses $\{5,7\}, \{5,8\}, \{5,9\}, \{5,15\}, \{5,20\}, \{5,25\}, \{5,30\}$ e $\{5,35\}$ apresentaram 90, 90, 70, 60, 40, 30, 10 e 0%, respectivamente.

No intervalo de tempo [1,12], as subclasses $\{5,2\}, \{5,3\}, \{5,4\}, \{5,6\}, \{5,7\}, \{5,8\}$, mostraram que 100% dos cenários testados no *Excel* apresentaram a otimização da função objetivo. As subclasses $\{5,5\}, \{5,9\}, \{5,10\}, \{5,15\}, \{5,20\}, \{5,25\}, \{5,30\}$ e $\{5,35\}$ apresentaram 90, 70, 80, 70, 20, 40 e 0%, respectivamente.

No intervalo de tempo [1,24], as subclasses $\{5,2\}, \{5,3\}, \{5,4\}, \{5,6\}$ e $\{5,9\}$, mostraram que 100% dos cenários testados no *Excel* apresentaram a otimização da função objetivo do problema, já as subclasses $\{5,5\}, \{5,7\}, \{5,8\}, \{5,10\}, \{5,15\}, \{5,20\}, \{5,25\}, \{5,30\}$ e $\{5,35\}$ apresentaram 90, 80, 90, 80, 50, 50, 40, 20 e 0%, respectivamente.

No intervalo de tempo [1,45], as subclasses $\{5,2\}, \{5,3\}, \{5,4\}$ e $\{5,6\}$, mostraram que 100% dos cenários testados no *Excel* apresentaram a otimização da função objetivo do problema, já as subclasses $\{5,5\}, \{5,7\}, \{5,8\}, \{5,9\}, \{5,10\}, \{5,15\}, \{5,20\}, \{5,25\}, \{5,30\}$ e $\{5,35\}$, apresentaram 90, 80, 70, 90, 60, 50, 30, 10, 0 e 0%, respectivamente.

No intervalo de tempo [1,99], as subclasses {5,2}, {5,3} e {5,4}, mostraram que 100% dos cenários testados no *Excel* apresentaram a otimização da função objetivo do problema, no entanto, as subclasses {5,5}, {5,6}, {5,7}, {5,8}, {5,9}, {5,10}, {5,15}, {5,20}, {5,25}, {5,30} e {5,35}, apresentaram 90, 90, 60, 50, 40, 70, 30, 20, 10, 0 e 0%, respectivamente.

Em geral, os resultados dos testes de $\%S_{PEIA}$ da classe 5, mostram que 34,28% dos cenários testados no *Excel* apresentaram 100% de respostas (otimização da função objetivo), 12,85% apresentaram 90%, 5,71% apresentaram 80%, 7,14% apresentaram 70%, 4,28% apresentaram 60%, 5,71% apresentaram 50%, 5,71% apresentaram 40%, 5,71% apresentaram 30%, 4,28% apresentaram 20%, 4,28% apresentaram 10% e 10% não apresentaram respostas. A Figura 1 ilustra a porcentagem de sucesso ($\%S_{PEIA}$) para a classe 5 de problemas.

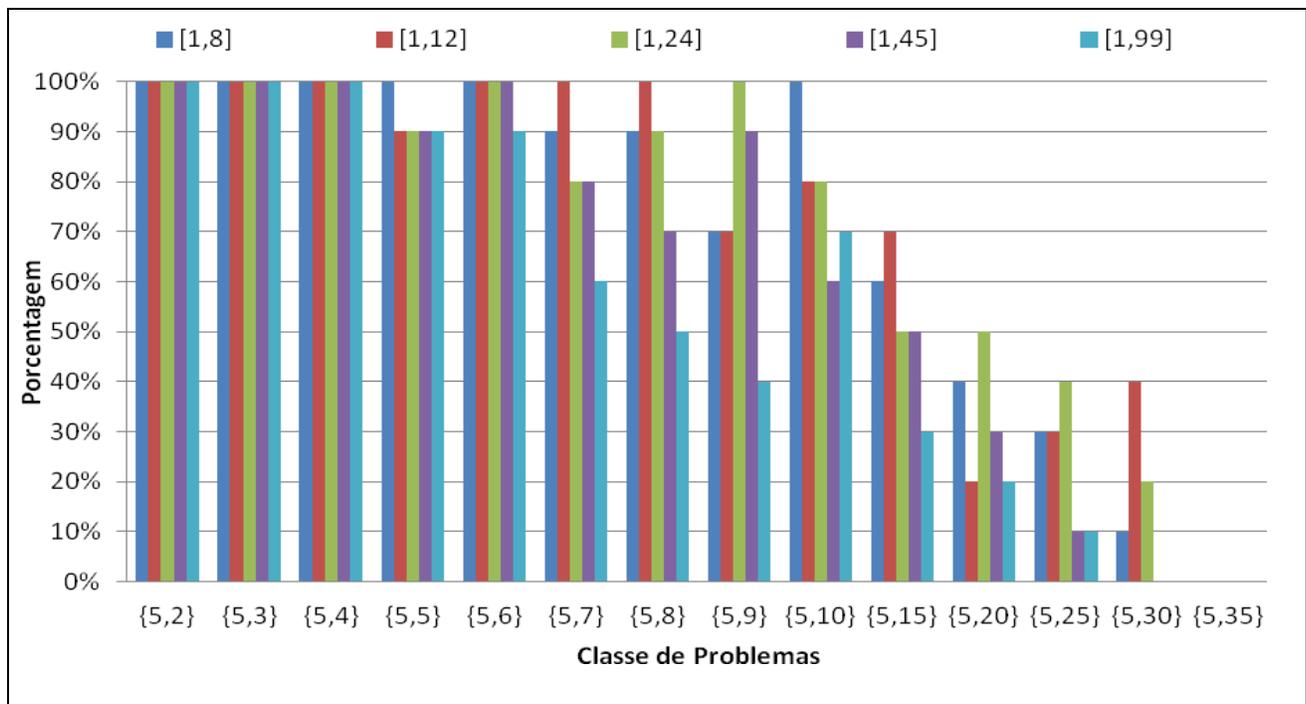


Figura 1 - Porcentagem de sucesso no *Excel* para a classe 5 de problemas.

Para a classe 5, também foi analisado o tempo médio de computação (TMC_{PEIA}) de cada subclasse. O TMC_{PEIA} de cada subclasse foi coletado a partir da equação (11). Os resultados do TMC_{PEIA} para as subclasses {5,2}, {5,3}, {5,4}, {5,5}, {5,6}, {5,7}, {5,8}, {5,9}, {5,10}, {5,15}, {5,20}, {5,25}, {5,30} e {5,35} mostraram um crescimento moderado no TMC_{PEIA} entre as classes de problemas {5,2} e {5,9} e um aumento intensificado do TMC_{PEIA} a partir da classe de problemas {5,10}. Nesta classe de problemas não houve diferença considerável de TMC_{PEIA} entre os intervalos de tempos de processamento, conforme apresentado na Figura 2.

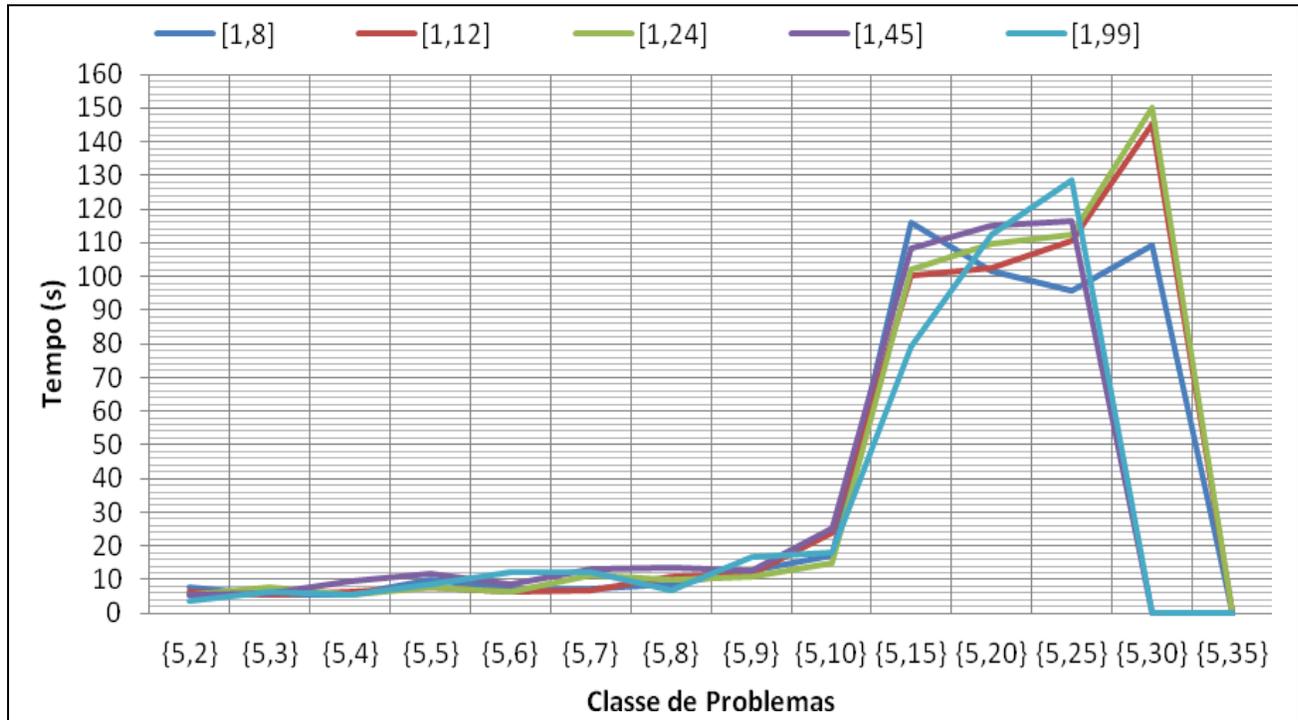


Figura 2 - Tempo médio de computação no *Excel* para a classe 5 de problemas.

Todas as equações 1, 2, 3, 4, 5, 6, 7, 8 e 9 foram implementadas na linguagem GAMS e, testados 10 cenários de problemas em cada subclasse $n \in \{5\}$ e $m \in \{2,3,4,5,6,7,8,9,10,15\}$, com intervalos de tempo [1,99], totalizando 100 problemas. A escolha de realização dos testes de cenários no GAMS em intervalos de tempo [1,99] ocorreu porque este intervalo contempla todos os intervalos $\{[1,8]; [1,12]; [1,24]; [1,45]\}$ testados no *Excel*. Salienta-se que os cenários analisados no GAMS são os mesmos que foram analisados no *Excel*. Assim, dos 700 problemas analisados no *Excel*, escolheram-se 100 problemas (10 cenários de cada subclasse de $m \in \{2,3,4,5,6,7,8,9,10,15\}$) para testes no GAMS e análise comparativa.

Os resultados para o critério de Porcentagem de Sucesso (%S) mostraram que a aplicação do modelo no GAMS obteve o mesmo percentual de sucesso que no *Excel*, nas subclasses $\{5,2\}$, $\{5,3\}$ e $\{5,4\}$, já para as classes $\{5,6\}$, $\{5,7\}$, $\{5,8\}$, $\{5,9\}$, $\{5,10\}$ e $\{5,15\}$, o GAMS obteve um desempenho superior ao do *Excel*, isto é, 100% para o GAMS e em média 61,42% para o *Excel*.

Para o Tempo Médio Computacional (TMC), um desempenho favorável foi obtido para a programação realizada no GAMS, uma vez que o TMC dos problemas analisados desempenharam um tempo médio menor que um segundo, já para os mesmos problemas programados no *Excel*, a média de tempo de resolução no computador foi de 26 segundos.

Os resultados da Minimização do Tempo Total de Conclusão das Tarefas (*Makespan*) obtidos no GAMS comparado com o resultado do *Excel* mostraram que 100% dos resultados obtidos

no GAMS são iguais as respostas do *Excel*, diferenciando apenas nas Sequencia Ótima de Programação das Ordens de Processamento do modelo.

4. CONSIDERAÇÕES FINAIS

Com os resultados encontrados, pode-se afirmar que para o objetivo de Minimização do Tempo Total de Conclusão das Tarefas (*Makespan*), no ambiente de programação FSC, descrito neste estudo, os cenários podem ser programados no *Excel/VBA* e no GAMS. O GAMS, quando comparado com o *Excel/VBA*, apresentou resultados semelhantes para a otimização da função objetivo, melhores para o quesito Porcentagem de Sucesso (%S) e Tempo Médio Computacional (TMC), no entanto, com relação ao número de classes e subclasses possíveis de programação, o *Excel/VBA* mostrou-se superior ao GAMS (Versão *Free*).

Por meio deste estudo desenvolveu-se uma Planilha Eletrônica com Interface Amigável (PEIA) para resolver o problema de sequenciamento da produção em um processo por batelada. Isto mostra que a mesma pode ser aplicada neste contexto. Em muitos quesitos o GAMS mostrou-se com um desempenho superior a PEIA, atendendo com maior eficiência ao objetivo de minimização do *Makespan* proposto neste estudo. É fundamental a extensão dos testes de comparação do GAMS com o *Excel/VBA* em demais cenários de programação, para que desta forma possa se chegar a uma maior precisão dos resultados.

5. REFERÊNCIAS BIBLIOGRÁFICAS

- HILLER, F. S. Introdução a Pesquisa Operacional. 8 ed. Porto Alegre: AMGH, 2010.
- LAESCH, C.; HEIN, N. Pesquisa Operacional: Fundamentos e modelos. São Paulo: Saraiva, 2009.
- MACCARTHY, B. L.; LIU, J. Y. Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. *Intern. Jour. of Prod. Res.*, London, v. 31, n. 1, p. 59-79, 1993.
- MÉNDEZ, C. A.; CERDÁ.; GROSSMANN, I. E. State-of-the-art Review of Optimization Methods for Short-term Scheduling of Batch Processes. *Comp. Chem. Eng.*, 30, 913, 2005.
- POLON, P. E. Otimização da Produção da Indústria de Embutidos. 115 f. Tese (Doutorado em Engenharia Química) – PEQ, Programa de Pós Graduação em Engenharia Química – UEM, Universidade Estadual de Maringá, 2010.
- TAHA, H. A. Pesquisa Operacional: Uma Visão Geral. 8 ed. São Paulo: Pearson Prentice Hall, 2008.