

## **Um Estudo Comparativo do *Excel/VBA* e do *GAMS* para Minimização do *Makespan* em um Sistema de Produção por Batelada (SPPB)**

Rony Peterson da Rocha (GEPPGO/DEP/FECILCAM) [peteronccbpr@hotmail.com](mailto:peteronccbpr@hotmail.com)

Fabício Wesley da Rocha (LPT/UNICAMP) [phabriciowesley@hotmail.com](mailto:phabriciowesley@hotmail.com)

Ruby de Mello Campos (DEP/FECILCAM) [rubyadmc@hotmail.com](mailto:rubyadmc@hotmail.com)

Luiz Mário de Matos Jorge (PEQ/UEM) [lmj@deq.uem.br](mailto:lmj@deq.uem.br)

Paulo Roberto Paraíso (PEQ/UEM) [paulo@deq.uem.br](mailto:paulo@deq.uem.br)

### **Resumo:**

A diversidade e complexidade de um Sistema de Produção por Batelada (SPPB) necessita constantemente da aplicação de técnicas de Planejamento, Programação e Controle da Produção (PPCP). A Programação da Produção (PP), fase de curto prazo do PPCP é responsável por atividades como administração de estoques, sequenciamento e emissão de ordens de produção. Neste trabalho será abordado a questão do sequenciamento de ordens de produção, ou seja, o objetivo é analisar a aplicação de uma Planilha do *Excel* com Interface Amigável (PEIA) e do software *GAMS* no sequenciamento da produção de um SPPB abordado na literatura, com o intuito de Minimizar o Tempo de Conclusão das Tarefas (*Makespan*). Os resultados mostraram que é possível utilizar a PEIA e o *GAMS* para o sequenciamento desses processos, sendo que o *GAMS* apresentou um desempenho superior ao da PEIA.

**Palavras chave:** *Excel/VBA*, *GAMS*, Programação da Produção, *Makespan*.

## **A Comparative Study of Excel / VBA and GAMS for Minimizing Makespan on a Batch Production System (BPS)**

### **Abstract**

The diversity and complexity of a Production System for Batch (PSB) requires constant application of techniques of Planning, Programming and Production Control (PPPC). The Production Scheduling (PS), short-term phase of PPCP is responsible for activities such as inventory management, scheduling and issuing production orders. This paper will address the issue of sequencing of production orders, in other words, the goal is to analyze the application of an Excel spreadsheet with Friendly Interface (PEIA) and sequencing software *GAMS* in the production of a SPPB addressed in the literature, with order to minimize the Task Completion Time (makespan). The results show that it is possible to use the tether and *GAMS* for sequencing of these processes, and the *GAMS* outperformed that of PEIA

**Key-words:** *Excel/VBA*, *GAMS*, Scheduling, *Makespan*.

### **1. Introdução**

O mercado consumidor, que impõem as empresas um processo constante de mudanças, conforme descrito por Scalabrin *et al.* (2006), provoca nas organizações uma busca por diferenciais competitivos, exigindo cada vez mais produtos de qualidade e com menor custo. Diante disso, buscar ferramentas de apoio as questões de planejamento, programação e controle da produção (PPCP) é uma alternativa bastante viável para obter tais diferenciais.

As atividades de PPCP, segundo Sá Motta *In* Machline (1972); Corrêa *et al.* (2001); Tubino (2007); Lustosa *et al.* (2008); Moreira (2010); Fernandes e Filho (2010), visa desenvolver os

planos de orientação da produção e são desenvolvidas em três níveis hierárquicos dentro da empresa. Esses níveis são caracterizados conforme o horizonte de planejamento (HP) seja de longo, de médio ou de curto prazo. No HP de curto prazo, uma importante atividade desempenhada pelo PPCP refere-se a programação da produção (PP).

Na literatura especializada, Baker (1974) aborda a PP como uma ferramenta para a tomada de decisão na alocação de recursos através do tempo. Stevenson (2001) e Andrade (2009), referem-se a PP como a determinação do momento oportuno para a utilização e alocação de determinados recursos de uma organização, de maneira atender a um determinado objetivo.

Uma forma de resolver os problemas envolvidos com a PP de muitos tipos de processos, é tornar esses problemas como um caso de programação linear inteira mista (PLIM). Prado (2003); Caixeta-Filho (2004); Taha (2008); Laesch e Hein (2009) e Hillier (2010) afirmam que a PLIM é uma técnica de resolução de sistemas de equações lineares, considerada como uma técnica de otimização para resolver problemas de maximização ou minimização de um determinado objetivo, quando apenas algumas variáveis apresentam valores inteiros.

A programação linear (PL), conforme descrita por Scalabrin *et al.* (2006) tem sido aplicada nas áreas mais diversas, tais como: formulação de alimentos, rações e adubos; blindagem de ligas metálicas e petróleo; transporte; localização industrial; carteira de ações (investimentos); alocação de recursos em fábricas, fazendas, escritórios, etc; designação de pessoas e tarefas (composição de tabelas de horários); corte de barras e chapas.

Na indústria química, as necessidades de PPCP são cada vez mais acentuadas, principalmente no SP batelada. Para Reklaitis (1995), produção em bateladas pode ser descrita como um SP orientado à receita. Mendez *et al.* (2005) classifica esse tipo de operação em: Estágio único, que subdivide-se em unidades únicas ou paralelas; múltiplo estágios, que subdivide-se em multipropósito (*Job-Shop*) e multiprodutos (*Flow-Shop*).

A diversidade e complexidade de um SP em batelada, segundo Severo (2007) recomenda a aplicação de técnicas de PPCP a fim de cumprir as exigências dos consumidores. Assim, há um crescente desenvolvimento científico na área de PP, sobretudo, referentes a esses SP.

O problema de PP tratado neste trabalho é do tipo Batelada - Multiprodutos - *Flow Shop* Permutacional. *Flow-Shop*, na visão de Maccarthy e Liu (1993) e Baker (1974) é um tipo de 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 é descrito por Baker (1974); Maccarthy e Liu (1993); Taillard (1993) e Moccellini (1995) como um tipo de SP em que a ordem de processamento das tarefas em todas as máquinas é a mesma. Segundo Polon (2010), as plantas por Batelada-Multiprodutos, em geral, são empregadas para um conjunto de produtos cuja a estrutura de receita é a mesma e as linhas de produção podem ser denominadas também de *Flow Shop*. Muitas indústrias de processo químico, tais como óleo e tintas, indústrias farmacêuticas e o setor de química fina se encaixam nesta categoria.

Assim, este trabalho visa apresentar um estudo comparativo da utilização do *Excel/VBA* e do *GAMS* para o Problema de Minimização do Tempo Total de Conclusão das Tarefas (*Makespan*) de um ambiente de PP *Flow Shop* Permutacional, também denominado de programação da produção em processo batelada (PPPB).

## 2. Metodologia

No presente estudo, considerou-se o sequenciamento da produção de (n) produtos em (m) reatores, por meio de um roteiro definido de produção com as seguintes restrições:

- 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;
- 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;
- 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;
- O ordenamento das tarefas em cada processador é o mesmo.

O estudo abrangeu cenários de  $n \in \{2,3\}$  e  $m \in \{2,3,4, \dots, 98\}$ , sendo que  $n$  representa o número de produtos (tarefas) e  $m$  o número de máquinas (reatores ou processadores). Na modelagem matemática destes cenários, foram utilizadas as Equação 1, 2, 3, 4, 5, 6, 7, 8 e 9.

$$\text{Minimizar } C_{NM} \quad (1)$$

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

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

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

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

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

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

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

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

A Equação 1 representa o objetivo central do problema a ser resolvido, onde  $N$  diz respeito aos número de produtos ou tarefas e  $M$  ao número de máquinas ou processadores (reatores).

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 seqüenciamento,  $X_{ik}$  é uma variável binária definida como:  $X_{ik} = 1$  se a tarefa  $i$  está na posição  $k$  e  $X_{ik} = 0$ , caso contrário.

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 seqüência;  $j$  = processadores ou estágios;  $k$  = posição na seqüência;  $N$  = número de tarefas;  $X_{s,k}$  = variável binária e  $TP_{s,j}$  é o tempo de processamento da tarefa  $i$  no processador  $j$ .

A 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).

A Equação 6 apresenta a restrição que o *Makespan* da tarefa posicionada na seqüência 1 da PP no processador  $j$  é maior ou igual ao *Makespan* da tarefa posicionada na seqüência 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.

A Equação 7 mostra que o *Makespan* da tarefa seqüenciada 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  $(TP_{1,1}, TP_{2,1}, \dots, TP_{N,1})$ .

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

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

### 3. Implementação do Modelo no Excel

Todas as equações matemáticas do modelo foram transcritas para planilhas do *Excel*, lançadas primeiramente de forma manual e resolvidas pelo aplicativo *Solver*. Nesta etapa, o propósito foi de resolver o modelo de forma não automatizada, para posteriormente, na etapa de criação da Interface Amigável, comparar os resultados obtidos de forma manual com os valores obtidos com a Interface Amigável. Esta comparação diz respeito exclusivamente aos resultados das equações generalizadas. O objetivo desta comparação foi de validar os resultados da Planilha do *Excel* com Interface Amigável (PEIA).

A Linguagem VBA foi utilizada para criar uma interface entre o usuário e a PEIA, desta forma, as equações matemáticas do modelo, juntamente com o aplicativo *Solver*, foram desenvolvidas de maneira a criar um modelo generalizado de programação, para este ambiente e com as restrições impostas pelo modelo aqui proposto.

A avaliação da coerência das equações generalizadas na PEIA foi realizada por meio da comparação dos resultados encontrados com as planilhas do *Excel* executada de forma manual e da desenvolvida na PEIA, pela Linguagem VBA.

Para validar a planilha automatizada em relação à veracidade das equações geradas com a expansão do modelo foram testados 10 cenários de problemas em cada subclasse  $n \in \{2\}$  e  $m \in \{2,3,4,5,6,7,8,9,10,15,20,25,30,40,60,98\}$ , com intervalos de tempo  $[1,99]$ , totalizando 160 problemas. Também foram testados 10 cenários de problemas em cada uma das subclasses  $n \in \{3\}$  e  $m \in \{2,3,4,5,6,7,8,9,10,15,20,25,30,40,62\}$ , com intervalos de tempo  $[1,99]$ , totalizando 150. Nestes testes além de observar a veracidade de cada uma das equações (1), (2), (3), (4), (5), (6), (7), (8) e (9) expandidas na PEIA, procurou encontrar os limites de células ajustáveis da planilha. Entende-se por células ajustáveis, o maior número de produtos ( $n$ ) e processadores ( $m$ ) das subclasses que apresentam a possibilidade de resolução da função objetivo (minimização do *Makespan*) na PEIA, independente da solução final.

Na experimentação computacional do modelo pela PEIA foi utilizado um computador com processador *Intel Corel 2 Duo*, 2,20 GHz de frequência, 1,75 GB de memória RAM.

### 4. Implementação do Modelo no GAMS

Todas as equações generalizadas (1), (2), (3), (4), (5), (6), (7), (8) e (9) foram implementadas na linguagem GAMS. Foram testados 10 cenários de problemas em cada subclasse  $n \in \{2\}$  e  $m \in \{2,3,4,5,6,7,8,9,10,15,20,25,30,40,42\}$ , com intervalos de tempo  $[1,99]$ , totalizando 150 problemas. Também foram testados 10 cenários de problemas em cada uma das subclasses  $n$

$n \in \{3\}$  e  $m \in \{2,3,4,5,6,7,8,9,10,15,20,25,27\}$ , com intervalos de tempo [1,99], totalizando 130. No geral, foram testados 280 cenários de problemas no GAMS.

Na experimentação computacional do modelo pelo GAMS foi utilizado um computador com processador *Intel Core 2 Duo*, 2,20 GHz de frequência, 1,75 GB de memória RAM. Utilizou-se uma versão Free do GAMS 23.7, disponível no site: <http://www.gams.com/solvers/solvers.htm#CPLEX>.

## 5. Método de Análise do Modelo

A análise geral dos resultados obtidos com a utilização da PEIA 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  $n \in \{2,3\}$ , em relação ao critério de desempenho Porcentagem de Sucesso (%) foi calculado pela quantidade de vezes em que a PEIA 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 PEIA 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 PEIA e do GAMS ( $TMC_{PEIA/GAMS}$ ) foi calculado pela soma dos tempos de computação de cada cenário de classe de problemas testado na PEIA 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 PEIA 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 na PEIA.

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 na PEIA com relação a sequencia de programação que minimiza a função objetivo (*Makespan*). Sendo assim, analisou comparativamente as sequencias geradas no GAMS e na PEIA, com o intuito de verificar se essas eram as mesmas ou não.

## 6. Resultados e Discussões

Depois de realizados todos os testes nas classes de problemas  $n \in \{2,3,4,5,6,7,8,9,10,11,12,13\}$  e  $m \in \{2,3,4,5,6,7,8,9,10,15,20,25,30,40,60$  e  $98\}$ , com intervalos de tempo [1,99], para validação da planilha automatizada em relação a veracidade das equações geradas com a expansão do modelo e estabelecimento dos limites das classes n

x m do modelo de PPPB, pode-se concluir que todas as equações (1), (2), (3.4), (3), (4), (5), (6), (7), (8) e (9) foram expandidas de forma correta em todos os 310 problemas analisados, garantindo desta forma, a veracidade dos resultados obtidos com o modelo. A Tabela 1 apresenta um resumo dos limites de programação do modelo analisado na PEIA para aplicação da ferramenta *Solver* na otimização do *Makespan*, a última subclasse de cada problema corresponde ao limite do modelo analisado.

CLASSE	SUBCLASSE
2	{2,2};{2,3};{2,4};{2,5};{2,6};{2,7};{2,8};{2,9};{2,10};{2,15};{2,20};{2,25};{2,30}; {2,40};{2,60};{2,98}
3	{3,2};{3,3};{3,4};{3,5};{3,6};{3,7};{3,8};{3,9};{3,10};{3,15};{3,20};{3,25};{3,30}; {3,40};{3,62};
4	{4,2};{4,3};{4,4};{4,5};{4,6};{4,7};{4,8};{4,9};{4,10};{4,15};{4,20};{4,25};{4,30}; {4,40};{4,46};
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};
6	{6,2};{6,3};{6,4};{6,5};{6,6};{6,7};{6,8};{6,9};{6,10};{6,15};{6,20};{6,25};{6,27};
7	{7,2};{7,3};{7,4};{7,5};{7,6};{7,7};{7,8};{7,9};{7,10};{7,15};{7,20};{7,21};
8	{8,2};{8,3};{8,4};{8,5};{8,6};{8,7};{8,8};{8,9};{8,10};{8,15};{8,17};
9	{9,2};{9,3};{9,4};{9,5};{9,6};{9,7};{9,8};{9,9};{9,10};{9,13};
10	{10,2};{10,3};{10,4};{10,5};{10,6};{10,7};{10,8};{10,9};{10,10};
11	{11,2};{11,3};{11,4};{11,5};{11,6};{11,7};
12	{12,2};{12,3};{12,4};
13	{13,2};

Tabela 1 – Limites de Programação do Modelo Analisado na PEIA

Foram realizados testes nas subclasses de problemas  $n \in \{2,3,4,5,6,7,8,9,10,11,12,13\}$  e  $m \in \{2,3,4,5,6,7,8,9,10,15,20,25,30,40,60 \text{ e } 98\}$ , com intervalos de tempo [1,99], no GAMS, para analisar os limites de programação do modelo em questão. Os resultados são mostrados na Tabela 2.

CLASSE	SUBCLASSE
2	{2,2};{2,3};{2,4};{2,5};{2,6};{2,7};{2,8};{2,9};{2,10};{2,15};{2,20};{2,25};{2,30}; {2,40};{2,42};
3	{3,2};{3,3};{3,4};{3,5};{3,6};{3,7};{3,8};{3,9};{3,10};{3,15};{3,20};{3,25};{3,27};
4	{4,2};{4,3};{4,4};{4,5};{4,6};{4,7};{4,8};{4,9};{4,10};{4,15};{4,19};
5	{5,2};{5,3};{5,4};{5,5};{5,6};{5,7};{5,8};{5,9};{5,10};{5,15};
6	{6,2};{6,3};{6,4};{6,5};{6,6};{6,7};{6,8};{6,9};{6,10};{6,11};
7	{7,2};{7,3};{7,4};{7,5};{7,6};{7,7};{7,8};{7,9};
8	Não Resolve
9	Não Resolve
10	Não Resolve
11	Não Resolve
12	Não Resolve
13	Não Resolve

Tabela 2 - Limites de Programação do Modelo Analisado no GAMS

Observando os limites apresentados nas classes de problemas  $n \in \{2,3,4,5,6,7,8,9,10,11,12,13\}$  e  $m \in \{2,3,4,5,6,7,8,9,10,15,20,25,30,40,60...100\}$ , que correspondem as subclasses  $\{2,98\}, \{3,62\}, \{4,46\}, \{5,35\}, \{6,27\}, \{7,21\}, \{8,17\}, \{9,13\}, \{10,10\}, \{11,7\}, \{12,4\}$  e  $\{13,2\}$ , para o caso da PEIA e as subclasses  $\{2,42\}, \{3,27\}, \{4,19\}, \{5,15\}, \{6,11\}$  e  $\{7,9\}$ , para o caso do GAMS, vê-se que para ambos os casos há uma redução do número de processadores (m) a partir do aumento do número de produtos (p). Comparando esses limites obtidos nos testes com o GAMS e com a PEIA, verificou-se uma amplitude menor do GAMS comparado com a PEIA, isto pode ser melhor visualizado no Quadro 1 e no Gráfico 1.

Classe	Total de Problemas Programados no GAMS	Total de Problemas Programados na PEIA	Classe	Total de Problemas Programados no GAMS	Total de Problemas Programados na PEIA
2	41	97	8	0	16
3	26	61	9	0	12
4	18	43	10	0	9
5	14	34	11	0	6
6	10	26	12	0	3
7	8	20	13	0	1

Quadro 1 – Total de Problemas Programados em Cada Classe no GAMS e na PEIA

A partir dos dados obtidos nos testes de limites de programação da PEIA e do GAMS é possível constatar que para o presente modelo a PEIA desempenha uma papel melhor em relação ao GAMS, isto é, em cada uma das classes analisadas a PEIA apresentou um maior número de possibilidade de programação de produtos e máquinas, veja o Gráfico 1.

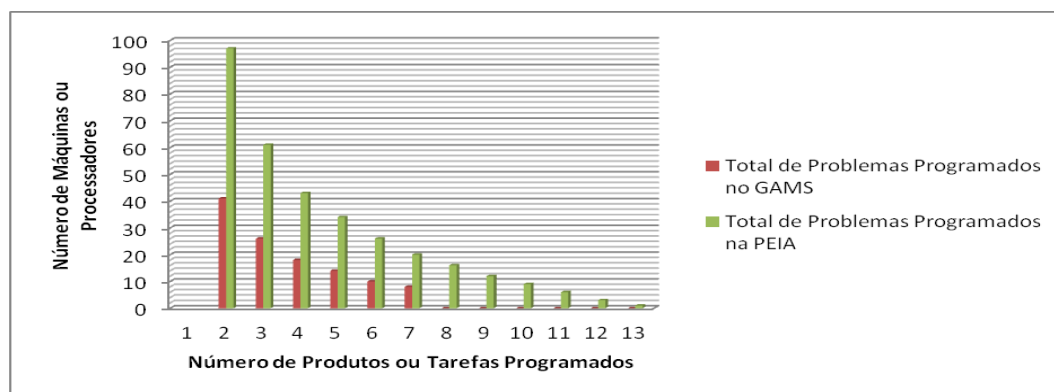


Gráfico 1 – Comparação dos Limites de Programação no GAMS e na PEIA

Após analisar os limites de programação do GAMS e da PEIA, foi realizado testes para analisar o desempenho do GAMS e da PEIA para as classes 2 e 3, no intervalo de tempo 1,99, conforme descrito nas seções seguintes.

### 6.1. Testes do GAMS e da PEIA para a Classe 2 no Intervalo de Tempo 1,99

Foram realizados um total de 200 testes, distribuídos em 100 testes no GAMS e 100 na PEIA. Esses testes consistiram em programar 10 cenários de problemas com intervalos de tempo de 1,99 em cada subclasse de problemas, sendo que o mesmo cenário foi avaliado no GAMS e

na PEIA. Após gerar os cenários e testá-los, analisou-se o desempenho de cada um em cada linguagem de programação.

Os resultados apresentados no Gráfico 2 mostraram um desempenho favorável para a programação realizada no GAMS, uma vez que o Tempo Médio Computacional (TMC) dos 100 problemas analisados desempenharam um tempo médio inferior a um segundo, já para os mesmos problemas programados na PEIA, a média de tempo de resolução no computador foi de 2,6 segundos.

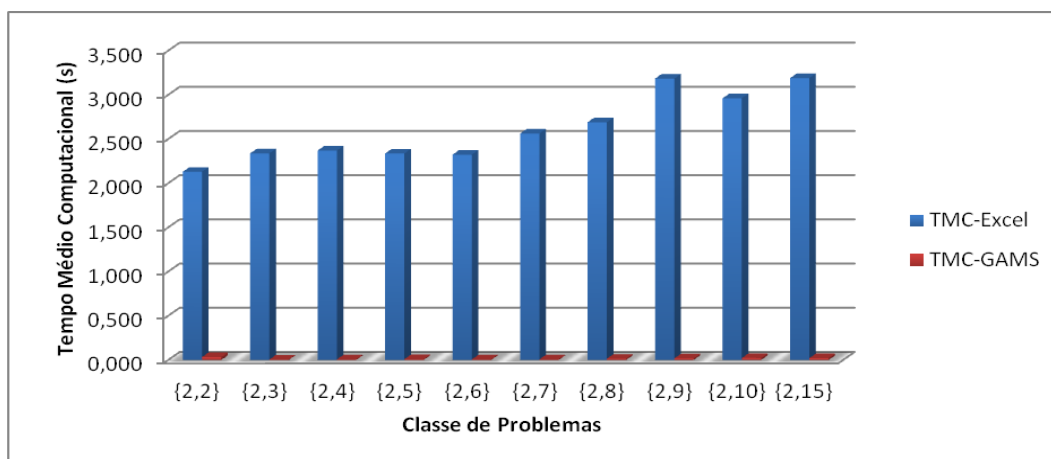


Gráfico 2 – Tempo Médio Computacional do GAMS e do Excel/VBA

Com relação a porcentagem de sucesso, o GAMS obteve o mesmo percentual que a PEIA, isto é, todos os problemas testados obtiveram respostas tanto no GAMS quanto na PEIA. Isto pode ser visualizado no Gráfico 3.

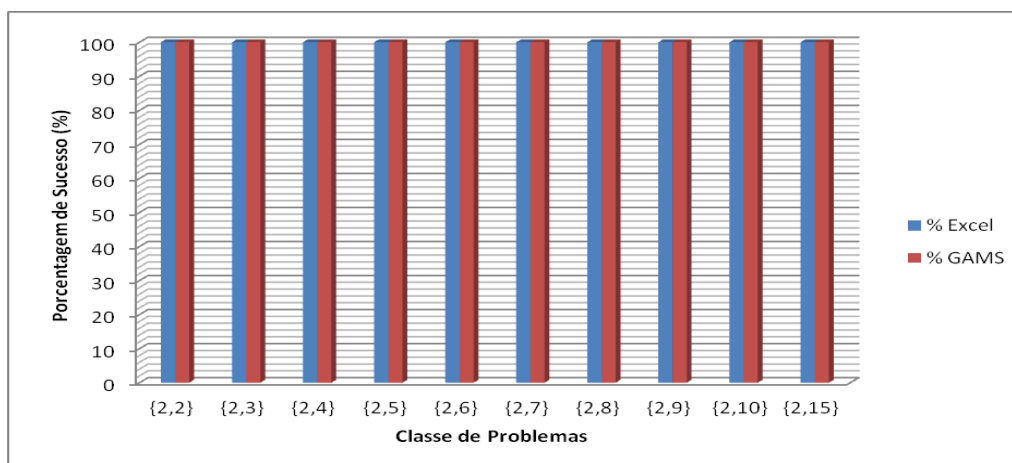


Gráfico 3 – Porcentagem de Sucesso do GAMS e da PEIA

Os resultados do *Makespan* obtido no GAMS comparado com o resultado da PEIA mostraram que 100% dos resultados obtidos no GAMS são iguais as respostas da PEIA, veja o Gráfico 4.



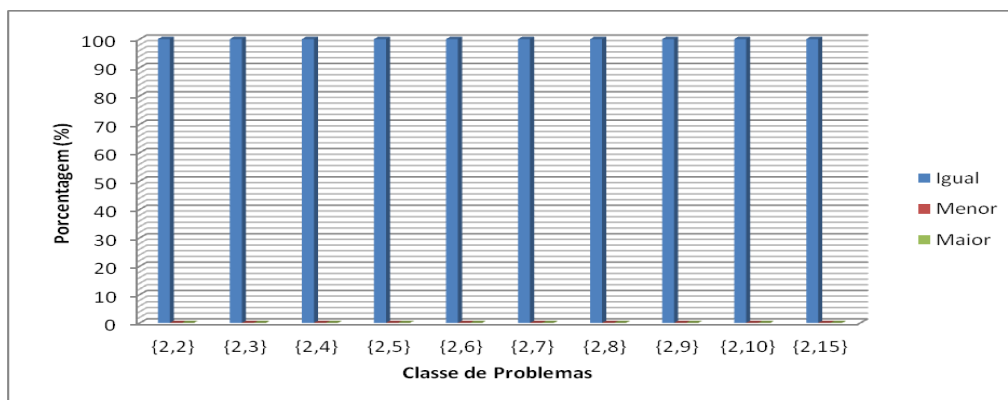


Gráfico 4 – Resultado Comparativo do *Makespan* do GAMS com a PEIA

Os resultados das sequencias de programação da produção encontradas no GAMS e na PEIA mostraram que um total de 100% dos problemas testados no GAMS obtiveram uma sequencia igual com *Makespan* igual ao da PEIA, conforme visualizado no Gráfico 5.

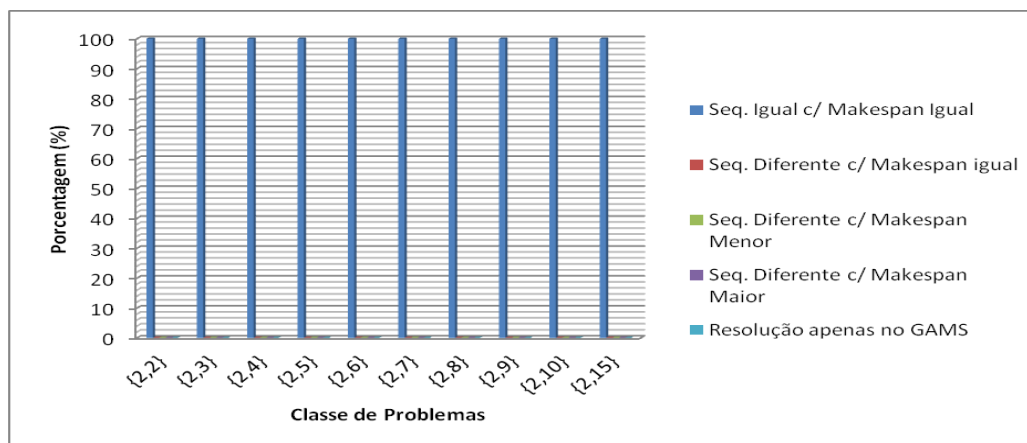


Gráfico 5 – Resultado Comparativo da Sequencia do GAMS com a PEIA

## 6.2. Testes do GAMS e da PEIA para a Classe 3 no Intervalo de Tempo 1,99.

Foram realizados um total de 200 testes, distribuídos em 100 testes no GAMS e 100 na PEIA. Esses testes consistem em programar 10 cenários de problemas com intervalos de tempo de 1,99 em cada subclasse de problemas, sendo que o mesmo cenário é avaliado no GAMS e na PEIA. Após gerar os cenários e testá-los, analisou-se o desempenho de cada um em cada linguagem de programação.

Os resultados apresentados no Gráfico 6 mostraram um desempenho favorável para a programação realizada no GAMS, uma vez que o Tempo Médio Computacional (TMC) dos 100 problemas analisados desempenharam um tempo médio inferior a um segundo, já para os mesmos problemas programados na PEIA, a média de tempo de resolução no computador foi de 3,95 segundos.

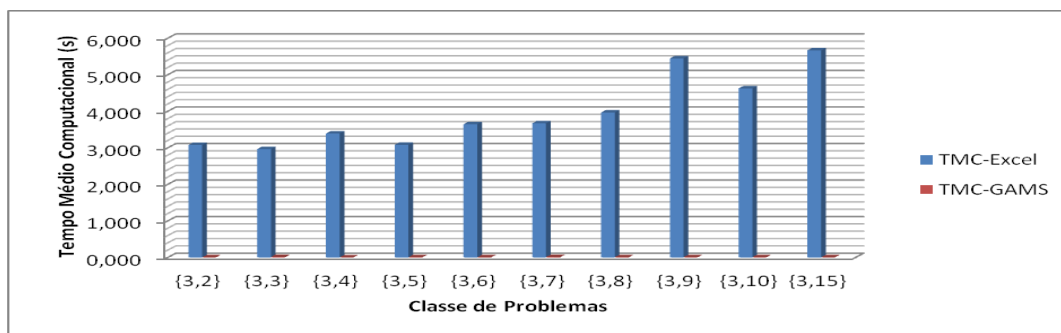


Gráfico 6 – Tempo Médio Computacional do GAMS e da PEIA

Com relação a porcentagem de sucesso, o GAMS obteve o mesmo percentual de sucesso que a PEIA, nas subclasses {3,2}, {3,3}, {3,4}, {3,7}, {3,9}, e {3,15}, já para as classe {3,6}, {3,8} e {3,10}, esse obteve um desempenho superior ao da PEIA, isto é, 100% para o GAMS e 90% para a PEIA. Isto pode ser visualizado no Gráfico 7.

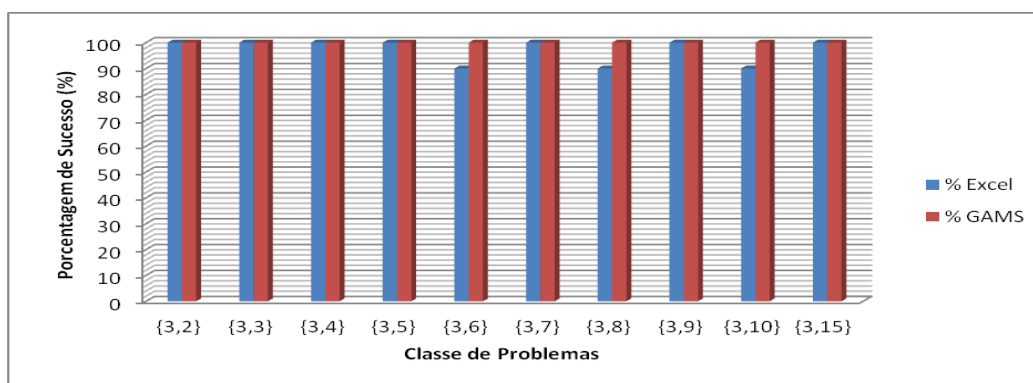


Gráfico 7 – Porcentagem de Sucesso do GAMS e da PEIA

Os resultados do *Makespan* obtido no GAMS para a subclasse [3,2] comparado com o resultado da PEIA mostraram que 70% dos resultados obtidos no GAMS são iguais as respostas da PEIA e 20% é menor. Para as classes {3,3}, {3,6}, {3,7} e {3,10}, essas apresentaram o mesmo percentual para o *Makespan*, isto é, todos apresentaram o mesmo resultado. Com relação as subclasses {3,2}, {3,4}, {3,5}, {3,9} e {3,15}, o GAMS obteve resultados melhores, uma vez que apresentou *Makespan* menores que os da PEIA. Na subclasse {3,8} a PEIA obteve um dos resultados melhores que o GAMS, com *Makespan* menor. Veja o Gráfico 8.

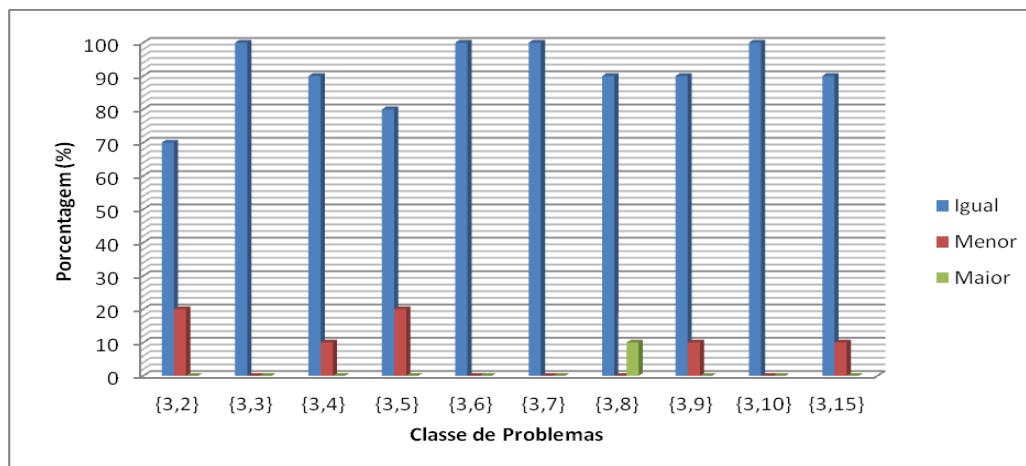


Gráfico 8 – Resultado Comparativo do *Makespan* do GAMS com a PEIA

Os resultados das sequencias de programação da produção encontradas no GAMS e na PEIA mostraram um resultado satisfatório para o GAMS, uma vez que apenas um problema testado apresentou sequencia diferente com *Makespan* maior que o da PEIA. Comparando a PEIA, verifica-se um total de 8 problemas com *Makespan* maior que o do GAMS. Também vale salientar que 3 problemas testados apresentaram resolução apenas no GAMS, mostrando um melhor desempenho para esse. Isto é visualizado no Gráfico 9.

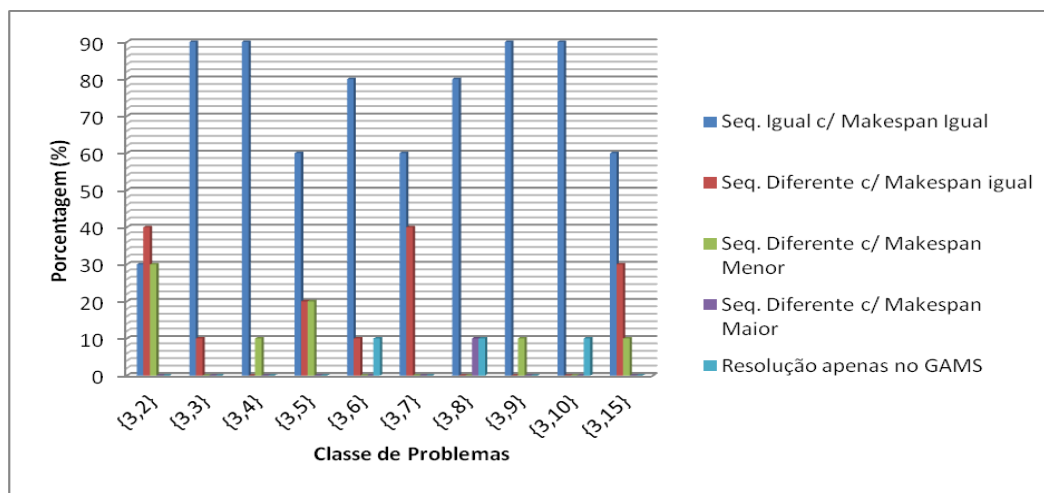


Gráfico 9 – Resultado Comparativo da Sequencia do GAMS com a PEIA

## 7. Considerações Finais

Diante do problema de pesquisa que foi comparar a utilização do GAMS e da Planilha do *Excel* com Interface Amigável (PEIA) para um problema de Programação da Produção em um Processo por Batelada, onde o objetivo da Programação foi de encontrar sequencias de Tarefas que atendam ao objetivo de desempenho de Minimização do Tempo Total de Conclusão das Tarefas (*Makespan*), os resultados apresentados neste artigo mostram que tanto a PEIA como o GAMS podem ser facilmente utilizados para este fim.

O GAMS, quando comparado com a PEIA, conforme os testes realizados para as classes 2 e 3 de problemas, apresentou-se uma superioridade dos resultados, referentes ao quesito porcentagem de sucesso, otimização da função objetivo e tempo médio computacional, no entanto, com relação ao número de classes e subclasses possíveis de programação, a PEIA mostrou-se superior ao GAMS.

De forma geral, também pode-se afirmar que a utilização da linguagem *Visual Basic for Applications* (VBA), para automatização da planilha do *Microsoft Excel*, permitiu de forma simples a criação de macros para a realização de tarefas repetidas. Logo, por meio da VBA, foi possível desenvolver novas funcionalidades dentro do *Excel*, fazendo com que o programa se tornasse uma ferramenta de desenvolvimento. Assim, por meio deste estudo desenvolveu-se uma PEIA de modo a resolver um problema particular de sequenciamento da produção em um processo por batelada. Isto mostra que a mesma pode ser aplicada neste contexto. No entanto, 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.

Vale salientar, que é fundamental uma extensão dos testes de comparação do GAMS com o a PEIA em demais cenários de programação, para que desta forma chega-se a uma conclusão mais precisa, com também, o desenvolvimento de um estudo aplicado.

### Referências Bibliográficas

- ANDRADE, Eduardo Leopoldino de.** *Introdução à pesquisa operacional: métodos e modelos para análise de decisões*. 4.ed. Rio de Janeiro: LTC, 2009.
- BAKER, K. R.** *Introduction to Sequencing and Scheduling*. New York: John Wesley & Sons, Inc., 1974.
- CAIXETA-FILHO, José Vicente.** *Pesquisa Operacional: Técnicas de Otimização Aplicadas a Sistemas Agroindustriais*. 2ª Ed. São Paulo: Atlas, 2004.
- CORRÊA, H. L.; GIANESI, I. G. N.; CAON, M.** *Planejamento, Programação e Controle da Produção*. 4 ed. São Paulo: Atlas, 2001.
- FERNANDES, Flávio C. F.; FILHO, Moacir, G.** *Planejamento e Controle da Produção: dos Fundamentos ao Essencial*. São Paulo: Atlas, 2010.
- HILLER, Frederick S.** *Introdução a Pesquisa Operacional*. 8 ed. Porto Alegre: AMGH, 2010.
- KONDILI, E.; PANTELIDES, C.C. & SARGENT, W.H.:** "A General Algorithm For Short-Term Scheduling Of Batch Operations – I: MILP Formulation", *Computers Chem. Eng.*, 17, pp.211-227, 1993.
- LAESCH, Cláudio; HEIN, Nelson.** *Pesquisa Operacional: Fundamentos e modelos*. São Paulo: Saraiva, 2009.
- LUSTOSA, L.; MESQUITAS, M. A.; QUELHAS, O.; OLIVEIRA, R.** *Planejamento e Controle da Produção*. Rio de Janeiro: Elsevier, 2008.
- MACCARTHY, B. L.; LIU, J. Y.** Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. *International Journal of Production Research*, London, v. 31, n. 1, p. 59-79, 1993.
- MÉNDEZ, Carlos. A.; CERDÁ.; GROSSMANN, Ignácio E.** *State-of-the-art Review of Optimization Methods for Short-term Scheduling of Batch Processes*. *Comp. Chem. Eng.*, 30, 913, 2005.
- MOCCELLIN, J. V. A** *New Heurist Method for the Permutation Flow Shop Scheduling Problem*. *Journal of the Operational Research Society*, Oxford, V. 46, p.883-886, 1995.
- MOREIRA, Daniel Augusto.** *Pesquisa Operacional: Curso Introductorio*. 2ª Ed. São Paulo: Cengage Learning, 2010.
- POLON, Paulo 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.
- PRADO, Darci.** *Programação Linear*. Belo Horizonte, MG: Editora do Desenvolvimento Gerencial, 2003.
- REKLAITIS, G. V.** *Scheduling Approaches for the Batch Process Industries*. *ISA Transactions* 34, 349-358, 1995.
- SÁ MOTTA, I.** *Planejamento e Controle da Produção*. In: MACHILINE, C. et al. *Manual de Administração da Produção*. Rio de Janeiro: FGV, 1972. V.1.
- SCALABRIN, Idionir.; MORES, Claudionor J.; BODANESE, Ronaldo E.; OLIVEIRA, José A. de.** *Programação Linear: Estudo de Caso com Utilização do Solver da Microsoft Excel*. *Revista Universo Contábil*, ISSN 1809-3337, Blumenau, v. 2, n. 2, p. 54-66, maio/ago. 2006.
- SEVERO, Larissa S.** *Aplicação de Modelo de Programação da Produção na Indústria de Couros*. 107 f. Dissertação (Mestrado em Engenharia Química) – PEQ, Programa de Pós Graduação em Engenharia Química – UFRGS, Universidade Federal do Rio Grande do Sul, 2007.
- STEVENSON, William J.** *Administração das Operações de Produção*. 6 ed. Rio de Janeiro: LTC, 2001.
- TAHA, Hamdy A.** *Pesquisa Operacional: Uma Visão Geral*. 8 ed. São Paulo: Pearson Prentice Hall, 2008.
- TAILLARD, E.** *Benchmarks for basic Scheduling Problems*. *Europeans Journal of Operational Research*, Amsterdam, V. 64, p. 278-285, 1993.
- TUBINO, D. F.** *Planejamento e Controle da Produção: Teoria e Prática*. 2 ed. São Paulo: Atlas, 2007.