

Programação da produção em sistema *Flow Shop*: análise hipotética na indústria de bicicletas

Nayara Macedo Vinhal¹
Gislene da Silva Fonseca²

José dos Reis Vieira de
Moura Junior³

Resumo: Há diversas dificuldades presentes em um planejamento de programação da produção, dentre estas, estão a definição de cada fase do planejamento do projeto, bem como a descrição, programação de execução das atividades presentes no mesmo e a sua duração. Nos ambientes industriais existem diversas configurações de máquinas, e neste trabalho será abordado apenas o ambiente *Flow Shop*. Assim, neste trabalho é apresentado o *toolbox* Torsche que auxilia no sequenciamento de tais atividades de programação em um ambiente de *Flow Shop* de uma indústria de bicicletas hipotética, enquanto este ainda possa ser empregado em outros ambientes de problemas de programação da produção.

Palavras-chave: Programação da produção. *Flow Shop*. *Toolbox* Torsche.

1 Introdução

O processo de produção de bens como é conhecido hoje surgiu através da revolução industrial aliado a várias mudanças no contexto político, econômico e social, tornando o sistema de produção em algo complexo (MENEGHELLI, 1999).

-
- 1 Universidade Federal de Goiás – UFG. Regional Catalão, Unidade Acadêmica Especial de Matemática e Tecnologia. Aluna do curso de Matemática Industrial. Contato: nayara.macedov@gmail
 - 2 Universidade Federal de Goiás – UFG. Regional Catalão, Unidade Acadêmica Especial de Engenharia. Aluna do curso de Engenharia de Produção. Contato: gis-sf@hotmail.com
 - 3 Universidade Federal de Goiás – UFG. Regional Catalão, Unidade Acadêmica Especial de Matemática e Tecnologia. Orientador. Contato: zereis@ufg.br

Compreende-se como sistema de produção, um conjunto de recursos físicos, humanos e procedimentais inter-relacionados que foram concebidos de forma a atender necessidades no qual a relação custo-benefício seja compensatória (FERNANDES; FILHO, 2010).

Implementar o planejamento, programação e controle da produção (PPCP) representa ganhos para a empresa, uma vez que o objetivo do PPCP é fazer o planejamento e supervisão de atividades existentes dentro do sistema produtivo da empresa, de modo a otimizar os recursos necessários para produção de bens ou serviços, visando atingir os objetivos e metas da empresa com o menor custo possível. O que, conseqüentemente proporciona a empresa condições de competir com outras empresas no mercado (CORRÊA; GIANESI; CAON, 2011).

O sistema produtivo de uma empresa pode ser dividido em três níveis hierárquicos de atuação: estratégico, tático e operacional. Sendo que cada um destes, é responsável por um tipo de planejamento – planejamento estratégico da produção, planejamento-mestre da produção e programação da produção. E esses planejamentos são baseados em um prazo de execução: longo, médio e curto prazo (TUBINO, 2007).

O bom funcionamento e eficiência do sistema produtivo de uma empresa depende da boa integração entre estes três níveis de planejamento. O qual a programação e sequenciamento da produção tem um papel muito importante, pois esta lida diretamente com a alocação de recursos da empresa ditando o seu desempenho, logo, é de suma importância que esta etapa do processo de fabricação seja feita da melhor maneira possível.

A programação e sequenciamento da produção se refere a alocação de recursos ao longo do tempo, e a maneira que a sequência de operações deve ser ordenada seguindo uma prioridade, para a fabricação de um ou mais produtos. E é baseada em tomadas de decisões que busca otimizar processos industriais, a fim de reduzir custos, tempo de processamento, gasto com transporte, entre outros.

A programação e sequenciamento da produção de um sistema produtivo pode se tornar uma tarefa muito complexa, dependendo das características presentes em seu ambiente produtivo, tais como: produzir grandes quantidades de itens de produtos diferentes, alocar grande quantidade de recursos, lidar com recursos limitados de forma eficiente, considerar tempos de *setup* em uma linha de produção, dentre outros.

Posto isso, o objetivo do presente trabalho é fazer o levantamento bibliográfico acerca do problema de programação da produção (*Scheduling*), para o melhor entendimento dos ambientes de programação da produção, com foco no ambiente do sistema de produção *Flow Shop* e propor o uso do software *toolbox* Torsche para a resolução do mesmo.

2 Revisão bibliográfica

O problema de *scheduling* ganhou destaque em meados da década de 20, com o trabalho de Henry Gantt, no entanto, a primeira obra publicada sobre o tema foi realizada apenas na década de 50. Após este período, o tema vem sendo amplamente estudado tanto em meio acadêmico, como industrial (PINEDO, 2012).

Segundo Pinedo (2012), *scheduling* é um processo de tomada de decisão que lida com alocação de recursos e tarefas ao longo do tempo, visando a otimização de um ou mais objetivos.

Para realizar a programação da produção, é preciso saber quais são os tipos e quantidades de recursos que a empresa dispõe em seu sistema produtivo, para então determinar quais atividades podem ser realizadas, a sua duração, e quando deve começar e terminar o seu processamento (BAKER; TRIETSCH, 2009).

De forma geral, o problema de *scheduling* pode ser definido como sendo um conjunto de n Jobs (tarefas a serem processadas), $J = \{J_1, J_2, \dots, J_n\}$, que são processados por um conjunto de m máquinas disponíveis (recursos que fazem o processamento das tarefas), $M = \{M_1, M_2, \dots, M_m\}$, mediante a um tempo de processamento P_{jm} , onde cada operação segue uma sequência tecnológica. E este tipo de problema pode ser influenciado por diversos fatores, tais como: prioridade das atividades (w_j), *due-date* (prazo de entrega - d_j), tamanho do lote de produção, quantidade e capacidade das máquinas, precedência de operação, dentre outros (RODAMMER; WHITE, 1988).

Os problemas de *scheduling* podem ser classificados como determinístico e estocástico. O determinístico tem a programação definida com base num conjunto finito de elementos já conhecido (tempo de processamento, *due date*, entre outros) do problema. Já o estocástico, o tempo de processamento e outros parâmetros são variáveis aleatórias e possui distribuição de probabilidade (BAKER; TRIETSCH, 2009).

O problema de *scheduling* foi descrito de várias formas na literatura, como pode ser verificado em: (MACCARTHY; LIU, 1993), (LÓPEZ; BARCIA; EYADA, 1995) e (CHEN; POTTS; WOEGINGER, 1998). Sendo que a notação mais usual é expressada como $(\alpha | \beta | \gamma)$, onde: α representa a quantidade de máquinas e o ambiente de produção que o problema está inserido, β representa os tipos de processo e restrições tecnológicas do problema e γ representa a medida de desempenho ou função objetivo do problema.

O ambiente do sistema de produção de uma empresa se refere a qual tipo e quantidade de máquina que está inserida em seu layout de produção. E cada ambiente dos problemas de *scheduling* tem um conjunto de proposições que devem ser respeitadas (BAKER; TRIETSCH, 2009). As possíveis configurações do ambiente dos ambientes de produção estão descritas no quadro 1.

Quadro 1 Ambientes do sistema de produção.

Ambiente	Descrição
Máquina Única (1)	Realiza operações em uma única máquina
Máquinas em Paralelo (P_m, Q_m, R_m)	Realiza operações em m máquinas em paralelo
<i>Flow Shop</i> (F_m)	m quantidade de máquinas em série processa todos os n <i>Jobs</i> em uma sequência pré-determinada
<i>Job Shop</i> (J_m)	Ambiente com m máquinas, e cada <i>Job</i> segue uma sequência de processamento diferente
<i>Open Shop</i> (O_m)	Similar ao <i>Job Shop</i> , onde cada <i>Job</i> segue uma rota distinta, mas neste caso, não há necessidade de precedência de máquina

Fonte: Elaborada pelas autoras, 2016.

O campo β apresenta as possíveis restrições tecnológicas que os problemas de *scheduling* pode ter e estão descritas no quadro 2.

Quadro 2 Restrições presentes em problemas de programação da produção.

Simbologia	Restrição	Descrição
r_j	Tempo de liberação	Tempo em que o <i>Job j</i> está disponível para ser processado
<i>prmp</i>	Preempção	Infere que o <i>Job</i> pode ser interrompido durante o seu processamento
<i>prec</i>	Precedência	Requere que um <i>Job</i> precisa ser concluído para dar início ao processamento do próximo <i>Job</i> que está na sequência
s_{jk}	Sequência de tempo de <i>setup</i>	Representa a sequência de tempo de <i>setup</i> dependente que está incluído entre o processamento dos <i>Jobs j</i> e k . É o tempo de preparação da máquina
<i>brkdw</i>	Breakdown	Indica que a máquina não está continuamente disponível para uso
M_j	Restrição de máquina elegível	Apenas o conjunto de máquinas que pertence a M_j pode processar o <i>Job j</i>
<i>prmu</i>	Permutação	Representa a ordem de execução da fila de espera. Considera-se no caso em que ocorram potenciais elementos prioritários
<i>block</i>	Bloqueio	Permite que uma máquina bloqueie o recebimento de um <i>Job</i> . Geralmente ocorre em casos em que não há estoques intermediários
<i>nwt</i>	Sem espera	O <i>Job</i> é processado ininterruptamente, onde acabando de sair de uma máquina ele entra em outra até completar todo o ciclo

Continua

Quadro 2 Restrições presentes em problemas de programação da produção. (Continuação)

Simbologia	Restrição	Descrição
<i>rcrc</i>	Recirculação	Um <i>Job</i> pode ser processado por uma mesma máquina mais de uma vez
<i>batch (b)</i> <i>batch (b)</i>	Processamento em lote	Uma máquina pode ser apta a processar um número de <i>Jobs b</i> simultaneamente. Cada <i>Job</i> pode ter um tempo diferente, mas o tempo de processamento do lote só ocorre quando o último tenha terminado
<i>fmls</i>	Família de tarefas	<i>n Jobs</i> podem pertencer a <i>F</i> diferentes famílias de tarefas. Os <i>Jobs</i> de uma mesma família podem ter tempos de processamento diferentes, mas o tempo de <i>setup</i> deve ser zero

Fonte: Elaborada pelas autoras, 2016.

O campo γ apresenta as funções objetivos mais comuns de um problema de *scheduling* conforme o quadro 3.

Quadro 3 Funções objetivo ou medida de desempenho.

Simbologia	Função objetivo	Descrição
C_{max}	Makespan	Representa o tempo de conclusão de todos os <i>Jobs</i>
L_{max}	Atraso máximo de pontualidade	Representa o maior atraso de pontualidade dos <i>Jobs</i>
$\sum w_j C_j$	Tempo de conclusão total ponderado	Leva em consideração o maior número de <i>Jobs</i> com maior peso concluídos o mais cedo possível
$\sum w_j T_j$	Tempo de atraso ponderado total	Somatório de todos os tempos de processamento que foram atrasados
$\sum w_j U_j$	Número total de tarefas em atraso ponderadas	Somatório de todas as tarefas em atraso considerando suas importâncias (pesos)

Fonte: Elaborada pelas autoras, 2016.

De acordo com Alharkan (2010), a programação da produção deve garantir que todos os *Jobs* devem ser processados sem ocorrer sobreposição, todas as restrições do problema devem ser respeitadas e os critérios selecionados para a função objetivo devem ser otimizados.

Há vários métodos (algoritmos) para resolver os problemas de *scheduling*, e estes podem ser classificados em duas categorias: Exatos, analisa todas as soluções possíveis e fornece o resultado ótimo, requer grande quantidade de tempo para encontrar a melhor solução e gera grande esforço computacional. E heurísticos (ou aproximados): fornece soluções satisfatórias rapidamente e requer menor

esforço computacional. Fica a critério da pessoa responsável pelo planejamento e programação do sistema de produção escolher qual método usar.

3 Estudo de caso

Neste artigo é apresentado um exemplo hipotético da fabricação de quatro modelos diferentes de bicicletas, que possuem o mesmo processo de fabricação, com tempos de processamento diferentes, com *buffer* (estoque intermediário) limitado e sem consideração de tempos de *setup* (trocas de ferramentas entre mudanças de produtos), em um ambiente *Flow Shop*, para mostrar a aplicação do *toolbox* Torsche em um problema de sequenciamento da produção.

Etapas da fabricação para os quatro modelos:

- a) Montar garfo e mesa – fazer a montagem do garfo e mesa no quadro da bicicleta;
- b) Roda dianteira – colocar a roda dianteira da bicicleta;
- c) Pedal – montar a estrutura do prato pedaleiro e colocar o pedal;
- d) Traseira – montar a estrutura onde será colocada a roda traseira (cassete) e colocar a roda traseira;
- e) Corrente – colocar a corrente na estrutura da bicicleta;
- f) Cabeamento – fazer o cabeamento dos freios dianteiro e traseiro;
- g) Testes – realizar testes na bicicleta que foi fabricada;
- h) Desmontar – desmontagem da mesa e retirada da roda dianteira para embalar a bicicleta;
- i) Embalar – embalagem da bicicleta;
- j) Estoque – estoque do produto acabado pronto para expedição.

Na tabela 1 estão os detalhes sobre o processo de fabricação de cada modelo de bicicleta.

Tabela 1 Dados sobre as etapas de fabricação das bicicletas.

Etapas de fabricação	Buffer	Tempos de processamento para o modelo 1 (segundos)	Tempos de processamento para o modelo 2 (segundos)	Tempos de processamento para o modelo 3 (segundos)	Tempos de processamento para o modelo 4 (segundos)
Montar garfo e mesa	2	50	55	65	65
Roda dianteira	2	35	45	45	40
Pedal	2	50	60	75	70

Continua

Tabela 1 Dados sobre as etapas de fabricação das bicicletas. (Continuação)

Etapas de fabricação	Buffer	Tempos de processamento para o modelo 1 (segundos)	Tempos de processamento para o modelo 2 (segundos)	Tempos de processamento para o modelo 3 (segundos)	Tempos de processamento para o modelo 4 (segundos)
Traseira	1	120	135	125	145
Corrente	2	43	45	50	55
Cabeamento	2	40	50	60	55
Testes	2	60	60	60	60
Desmontar	1	35	50	45	55
Embalar	1	30	40	50	45

Fonte: Elaborada pelas autoras, 2016.

Cada etapa de fabricação é feita em um posto de trabalho, e cada posto de trabalho será representado como sendo uma máquina. Cada modelo de bicicleta é representado como um *Job*. O objetivo do problema é minimizar o tempo de conclusão da fabricação dos quatro modelos de bicicleta, logo, o problema em estudo pode ser formulado como um problema do tipo **$F_9 | \text{block} | C_{max}$** . Neste problema tem-se 9 máquinas dispostas no formato de *Flow Shop* (sequenciais) e que possuem um estoque intermediário limitado, permitindo a interrupção do fluxo de produção, uma vez que o posto de trabalho posterior não tenha *buffer* suficiente para receber mais itens. A função objetivo neste caso é a programação de todos os itens e sua conclusão o mais cedo possível, também conhecido por *Makespan*.

A programação e sequenciamento da produção de um sistema produtivo pode se tornar uma tarefa muito complexa, dependendo das características presentes em seu ambiente produtivo, tais como, produzir grandes quantidades de itens de produtos diferentes, alocar grande quantidade de recursos, lidar com recursos limitados de forma eficiente, considerar tempos de *setup* em uma linha de produção, dentre outros fatores.

E mediante a essa perspectiva, o uso de ferramentas computacionais (*softwares*) que auxiliam em tomada de decisões, acerca do problema de programação e sequenciamento da produção, representa vantagens para as empresas, pois, essas ferramentas permitem fazer simulações de situações reais acerca do problema. Sendo possível avaliar o desempenho do ambiente produtivo da empresa antes da

situação real vir acontecer, o que ajuda a conter prováveis desperdícios e melhorar a capacidade produtiva da mesma.

A medida de desempenho utilizada neste trabalho foi a minimização do tempo de conclusão das atividades.

Por se tratar de um problema de *Flow Shop*, todas as operações do sistema produtivo devem seguir a mesma rota de fabricação. Onde as operações de determinado modelo de bicicleta só podem passar para próxima etapa se a anterior tiver sido concluída (restrição block).

A seguir estão os resultados fornecidos pelo *toolbox* Torsche para o problema proposto no estudo de caso descrito acima.

4 Discussão e resultados

A solução da programação da produção do primeiro teste do processo de fabricação descrito no estudo de caso está ilustrada na Figura 1, e a sua taxa de utilização do *buffer* para cada etapa do processo na Figura 2.

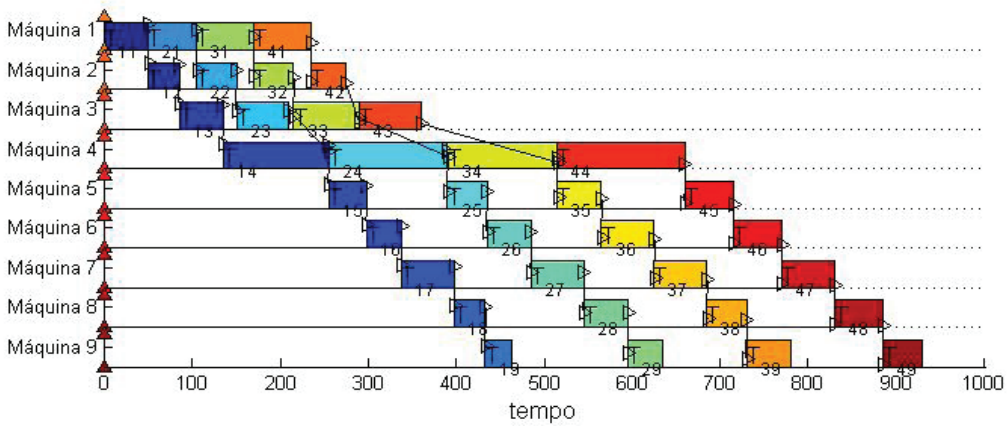


Figura 1 Primeiro teste.

Na Figura 1, o primeiro modelo de bicicleta foi representado pela cor azul escuro, o segundo modelo por azul claro, o terceiro por verde e o quarto modelo pela cor laranja.

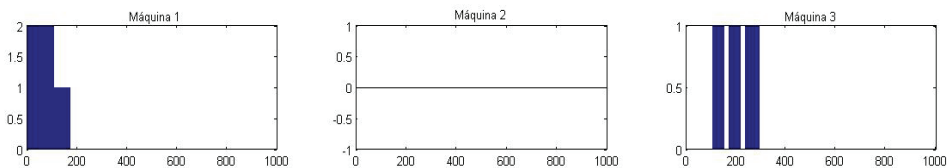


Figura 2 Taxa de utilização dos *buffers* do primeiro teste. (Continua)

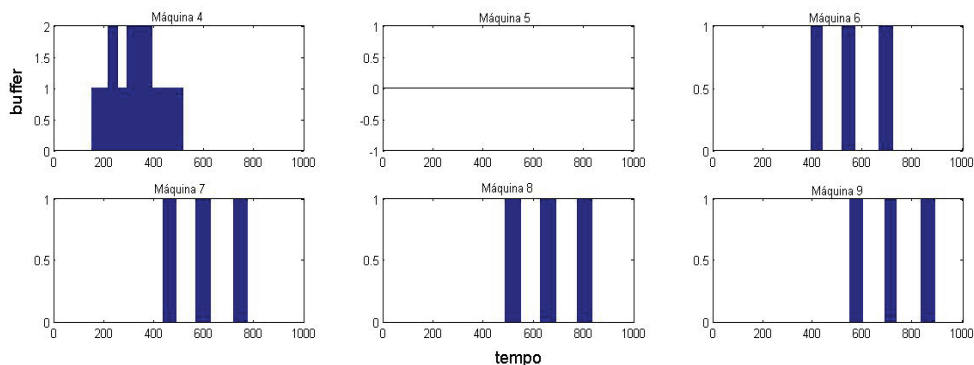


Figura 2 Taxa de utilização dos buffers do primeiro teste. (Continuação)

Através deste modelo é possível ver o número máximo de 2 elementos em fila nas estações de trabalho 1 e 4 (por períodos de tempos consideráveis), o que indica que nestas etapas a quantidade de *buffer* precisa ser alterado para melhorar o processo, enquanto as estações 6 a 9 intercalaram seus *buffers* entre 0 e 1 elementos. As estações 2 e 5 não formaram filas em nenhum instante (elemento no *buffer*).

Para o segundo teste do processo de fabricação, os valores do *buffer* de todas as etapas do processo foram modificados, utilizando um *buffer* com valor 10 para todas as operações. A solução do sequenciamento deste teste está ilustrada na Figura 3 (segundo a mesma sequência de processamento do primeiro teste: modelo 1 -> modelo 2 -> modelo 3 -> modelo 4), e as taxas de utilização do *buffer* estão representadas na Figura 4.

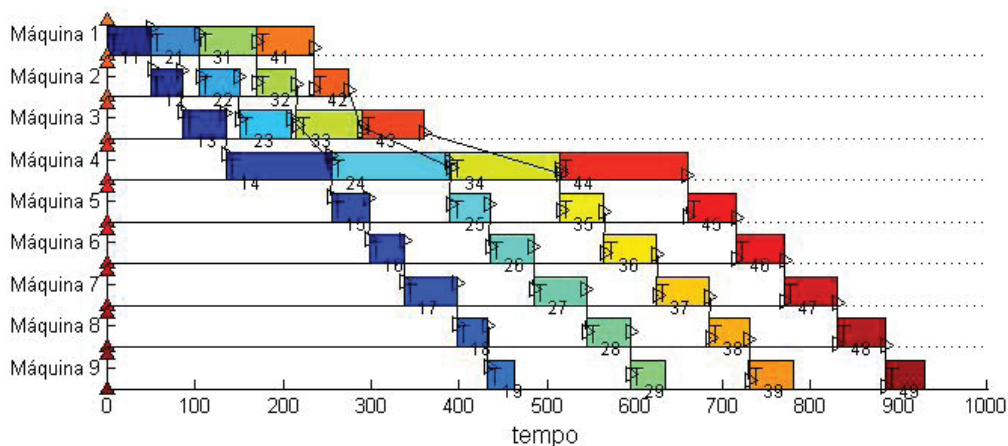


Figura 3 Segundo teste.

Como a sequência de operações foi mantida igual ao primeiro teste, nota-se que o tempo gasto para finalizar o processamento de todas as etapas foi o mesmo.

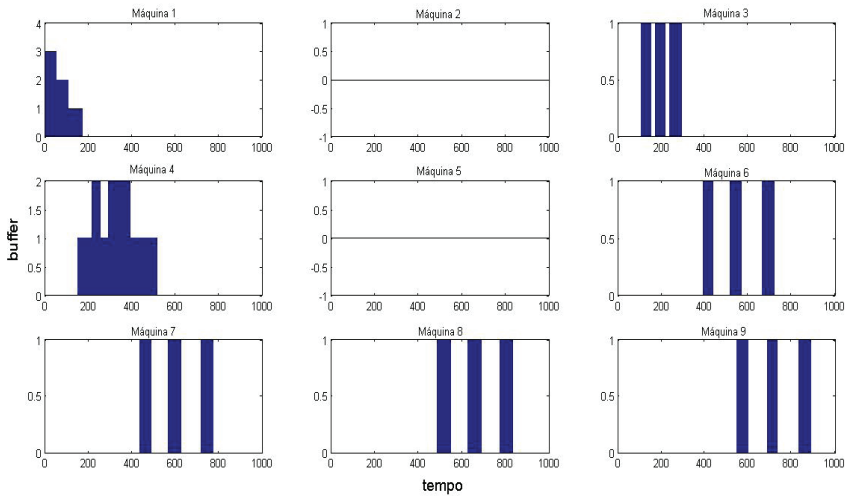


Figura 4 Taxa de utilização dos buffers do segundo teste.

Quanto a taxa de utilização do *buffer* do segundo cenário simulado, foi possível perceber o acúmulo em fila de no máximo 4 elementos na estação de trabalho 1 e que não impactou na formação de filas na estação de trabalho 4 em que acumula 2 elementos no *buffer* por um tempo considerável. As estações de trabalho 3, 6, 7, 8 e 9 intercalam na fila 1 elemento e 0. As estações de trabalho 2 e 5 em nenhum momento utilizaram o seu respectivo *buffer*.

No terceiro teste realizado (ilustrado na Figura 5), a ordem do processamento das operações foi alterada para: modelo 4 (representado pela cor azul escuro) -> modelo 1 (representado pela cor azul claro) -> modelo 2 (representado pela cor verde) -> modelo 3 (representado pela cor laranja).

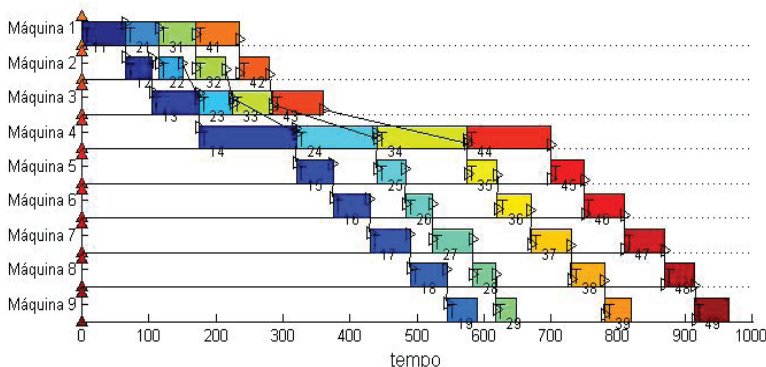


Figura 5 Terceiro teste.

O valor do *buffer* utilizado neste teste (ilustrado na Figura 6) foi o mesmo que o descrito no estudo de caso. Este teste foi realizado para comparar o efeito da troca de sequência com relação ao *buffer*.

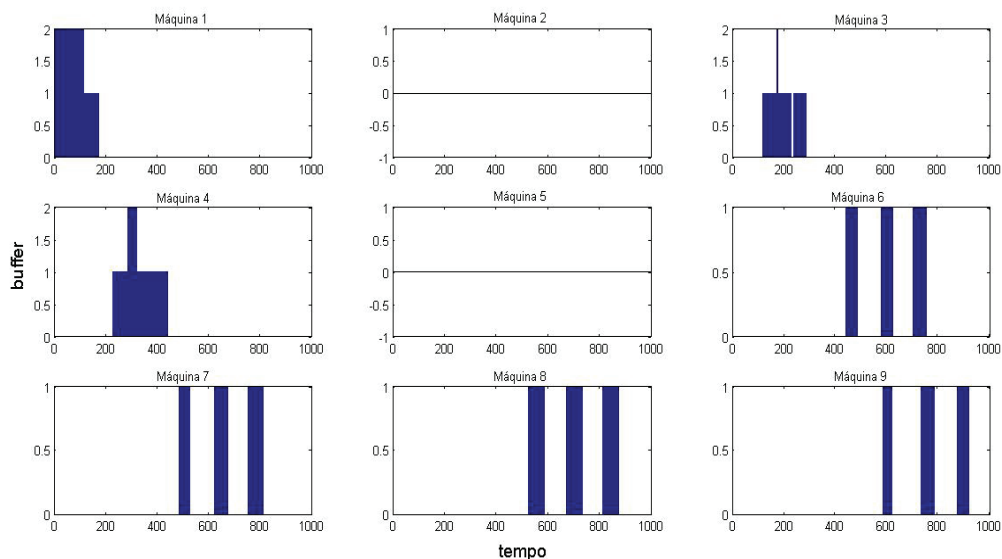


Figura 6 Taxa de utilização dos *buffers* do terceiro teste.

A partir da visualização deste gráfico pode-se observar que o *buffer* das estações de trabalho número 1, 3 e 4 alcançaram, em algum momento, o valor de pico de 2 elementos em fila. No entanto, nas estações 3 e 4 o intervalo de tempo com 2 elementos foi mínimo. Para as estações de trabalho 6 a 9 houveram momentos intercalados de 1 elemento em fila e nenhum, enquanto as estações 2 e 5 não houveram elementos em fila em nenhum instante de tempo.

Comparando o cenário simulado 1 em relação ao 2, aumentando-se os valores de *buffer* de 2 para 10, foi possível perceber que, além da taxa de utilização do *buffer* da primeira etapa do processo (representado pela máquina 1), nenhum outro *buffer* do processo sofre modificação. Por se tratar de um estudo hipotético, mas com dados de processamento reais de uma indústria, foi possível verificar a possibilidade de mudanças no *buffer* intermediário das estações de trabalho. Esta comparação demonstra que, muitas vezes, na prática não são necessários *buffers* intermediários de grande capacidade, eliminando um conceito amplamente explorado pelas linhas de produção em massa e bem desenvolvidos em sistemas puxados de produção enxuta baseados em Kanban.

Comparando o cenário 1 em relação ao cenário 3, no qual são alternadas as ordens de processamento dos Jobs, percebe-se que, embora os padrões de uso de *buffer* nas estações de trabalho 1, 2, 5, 6, 7, 8 e 9 não tenham sido modificados, a

estação 3 passa a utilizar momentaneamente 2 elementos no *buffer* não alterando demais aspectos de fila. Por outro lado, a estação 4 recebe uma menor solicitação de *buffer*. Com isto, pode-se perceber a potencialidade de balanceamento de carga de *buffers* pelo ajuste da melhor sequência de processamento das tarefas em um sistema *Flow Shop*. No entanto, é importante ressaltar que este sequenciamento é capaz de otimizar o uso de um sistema de *buffers*, mas que podem afetar padrões de respostas globais do sistema como atrasos e tempos totais de processamento dos Jobs C_{max} . Assim, apesar de otimizar o uso dos *buffers* que permitam maior flexibilidade de uma planta industrial, em contrapartida, tem-se um aumento no tempo de conclusão total das tarefas.

Comparando estes dois cenários estudados, apesar de um melhor uso dos *buffers* indo do cenário 1 para o cenário 3, o tempo de conclusão total aumenta de $C_{max}=930$ segundos do cenário 1 indo para $C_{max}=965$ segundos do cenário 3.

5 Conclusões

A pesquisa descrita neste artigo teve como finalidade, mostrar a aplicação do *toolbox* Torsche para a resolução de problemas de Flow Shop de uma fábrica hipotética de bicicleta e realizar diferentes testes para avaliar o desempenho do sistema produtivo que foram simulados.

Através deste estudo foi possível compreender os aspectos básicos de uma programação da produção de um sistema Flow Shop com estoques intermediários. Além de possibilitar a compreensão acerca do impacto dos estoques intermediários no processo produtivo no que diz respeito à programação da produção e a influência do sequenciamento da produção na solução do problema.

Como proposta para trabalhos futuros, pretende-se considerar outros aspectos que estão presentes em um sistema de produção para tornar o modelo o mais próximo possível de um sistema real. Como por exemplo: tempo de *setup*, *duo date*, dentre outros.

Referências

- ALHARKAN, I. M. *Algorithms for sequencing and scheduling*. Saudi Arabia: Industrial Engineering Department - King Saud University, [2010]. Disponível em: <http://faculty.ksu.edu.sa/ialharkan/IE428/Algorithms_for_Sequencing_and_Scheduling1.pdf>. Acesso em: 05 de jul. 2016.
- BAKER, K. R.; TRIETSCH, D. *Principles of sequencing and scheduling*. Nova Jersey: Wiley. 2009. Disponível em: <<http://onlinelibrary.wiley.com/doi/10.1002/9780470451793.fmatter/pdf>>. Acesso em: 05 ago. 2016.

- CHEN, B.; POTTS, C. N.; WOEGINGER, G. J. **A review of machine scheduling: complexity, algorithms and approximability.** In: Du, D. -Z.; PARDALOS, P. M. Handbook of combinatorial optimization. Kluwer Academic Publisher. 1998. p. 21-169.
- CORRÊA, H. L.; GIANESI, I. G. N.; CAON, M. **Planejamento, promoção e controle da produção: MRP II/ERP: conceitos, uso e implementação: base para SAP, Oracle Applications e outros softwares integrados de gestão.** 5. ed. São Paulo: Atlas, 2011.
- FERNANDES, F. C. F.; FILHO, M. G. **Planejamento e controle da produção: dos fundamentos ao essencial.** São Paulo: Atlas, 2010.
- LÓPEZ, O. C.; BARCIA, R. M.; EYADA, O. Problema de programação da produção um esquema de classificação. *Prod.* São Paulo, v. 5, no.2, 145-168, jul./dez. 1995.
- MACCARTHY, B. L.; LIU, J. Addressing the gap in scheduling research: a review of optimization and heuristic methods in production scheduling. *International Journal of Production Research.* v. 31, n.1, p. 59-79. 1993.
- MENEGHELLI, L. **O ambiente das organizações na era da globalização.** Instituto Catarinense de PósGraduação. 1999. Disponível em: <<http://www.posunias-selvi.com.br/artigos/rev01-03.pdf>>. Acesso em: 07 ago. 2016.
- PINEDO, M. L. **Scheduling: theory, algorithms and systems.** 4. ed. Nova York: Springer, 2012.
- RODAMMER, F. A.; WHITE, Jr. K. P. A recent survey of production scheduling. *IEEE Transaction on Systems, Man and Cybernetics.* v. 18, n. 6, p. 841-851. 1988.
- TUBINO, D. F. **Planejamento e controle da produção.** São Paulo: Atlas, 2007.

