



## PROGRAMAÇÃO JOB SHOP APLICADA NO MERCADO OCTG ATRAVÉS DE ALGORITMO DE PROGRAMAÇÃO MATEMÁTICA LINEAR INTEIRA

Ruan Carlos Alves Pereira – UFF – Ruancap@id.uff.br  
Marcos dos Santos - Instituto Militar de Engenharia (IME) - marcosdossantos@ime.eb.br  
Fredjoger B. Mendes – UFF – Fredjoger@gmail.com

### Resumo

O objetivo deste trabalho consiste em realizar a aplicação de um algoritmo baseado em programação matemática linear inteira, desenvolvido através do LINGO (*Branch and Bound*), para otimização da programação de produção em uma *job shop* em uma empresa do segmento OCTG, no mercado de Óleo e Gás, mostrando grande relevância frente as oportunidades de melhoria nos métodos de programação utilizados na empresa alvo e pelo potencial de desenvolvimento do sistema *job shop* de produção no atual cenário econômico do mercado de Óleo e Gás. Para atingir este objetivo, o algoritmo foi desenvolvido levando em consideração aspectos do mercado de Óleo e Gás, além de especificidades da estrutura organizacional da empresa estudada. Através da simulação computacional, desenvolvida utilizando o *software* FLEXSIM, o algoritmo de programação foi testado a validado. Neste ambiente de simulação foi considerado o *layout* exato das máquinas bem como o tipo de equipamentos de transporte e dimensões operacionais que refletem a realidade da empresa onde o trabalho foi aplicado. Destaca-se como principal resultado obtido a disponibilização de um algoritmo (PLI) capaz de realizar a programação de produção de uma *job shop* de forma ótima em um tempo muito menor comparado a programação manual em Excel, trazendo ainda oportunidades de ganhos com a visão clara de ocupação dos recursos e tempo de ociosidade para eventuais intervenções nas máquinas programadas.

**Palavras-Chaves:** *job shop*, OCTG, Otimização, Programação linear inteira e simulação

### 1. Introdução

Conforme Liu et al. (2016), a investigação acerca do controle e programação de produção em *job shops* tem sido efetuada em diferentes perspectivas. Uma das perspectivas refere-se ao problema estático, que pode ser descrito como  $n$  jobs, cada um composto por várias operações que devem ser processadas em  $m$  máquinas, tendo em atenção as devidas restrições, que otimize um determinado critério de desempenho, frequentemente determinado como o tempo total ou *makespan*.

Segundo Santos et al. (2018), a redução do tempo total de produção planejado ou *makespan* é uma necessidade também de empresa do setor produtivo de Óleo e Gás. Este setor apresenta

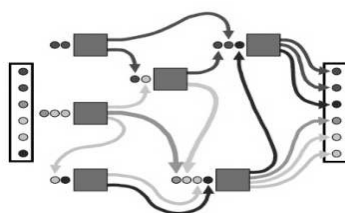
indicadores médios de entrega com perdas médias de 31% devido a atrasos no prazo de entrega. O presente estudo busca oferecer uma solução para programação de produção em um sistema *job shop*, buscando desenvolver um modelo capaz de proporcionar a otimização da programação da produção em uma empresa do setor de Óleo e Gás, quanto ao tempo total programado – *makespan* - para determinado volume de produção de distintos materiais a cada período de tempo, trazendo uma visão clara de ocupação e prazos de finalização dos pedidos.

Contudo, a tarefa de planejar a produção em uma *job shop* se torna cada vez mais complexa à medida que buscamos aproximar nossa visão sobre os fatos à realidade. Estudos sobre planejamento e controle da produção vêm sendo desenvolvido a décadas, como por exemplo, através de Cardenas et al. (2018), um dos primeiros autores a apresentar um método integrado de planejamento e controle da produção em sistemas *job shop*.

É importante ressaltar que, mesmo tendo conhecimento do caminho exato que os *jobs* (ordens de produção) seguirão pela *job shop* e os tempos de processamento em cada máquina, existe uma grande complexidade em prever qual o estado de ocupação de cada máquina e do *shop floor* em geral, sendo esta uma das maiores dificuldades do problema de planejamento e controle das operações em *job shops* (MOREIRA et al., 2019).

A Figura 1 exemplifica um conjunto de fluxos possíveis em uma *job shop* com seis estações de trabalho. Um *job shop* os *routings* possíveis são muito variados, assim como os tempos de processamento nas diversas fases de produção, considerando diversos tipos de materiais a serem processados.

Figura 1: Exemplo de fluxo de uma *job shop*



Fonte: Krajewski et al., 2010

As características expostas sobre o complexo sistema *job shop* e a forma para controlá-las reforça a necessidade de um sistema de planejamento e controle dos *jobs*, que enxergue o *job* desde a entrada até seu último processamento, permitindo identificar em qual etapa de processamento o *job* está e qual a atual previsão para finalização. A complexidade em planejar (*scheduling*) a produção em uma *job shop* está desde a percepção do tamanho dos

*buffers* a serem formados, até a ocupação das máquinas e recursos disponíveis (RIETZ et al., 2015).

Segundo Storer et al. (2015), várias técnicas foram desenvolvidas para problemas de sequenciamento com aplicação à programação *job shop*, como Redes Neurais Artificiais (RNs), Modelagem matemática, Tabu Search (TS) e GRASP. Este trabalho explora técnicas de Programação Matemática Linear Inteira (PLI) para construção da lógica de otimização e sequenciamento das tarefas ao longo do tempo, na *job shop*. Em busca de validar os resultados encontrado através da PLI, um modelo de simulação computacional foi construído, confrontando os valores medidos *in loco* com os valores apresentados pelo modelo de PLI.

Técnicas de simulação vem sendo amplamente utilizadas como uma forma de apresentar uma solução que possa ser aplicada e mostre resultados equiparados aos calculados de otimização. Modelos de simulação trazem para realidade parâmetros muitas das vezes não considerados em modelos matemáticos como taxas de falhas, distâncias entre equipamentos e indisponibilidade de operadores (SHEN et al., 2018).

## **2. Programação matemática linear inteira e simulação computacional**

Para aplicação no presente estudo, será proposto a construção de um modelo matemático de programação linear inteira utilizando o software LINGO. O solver utilizado aplica o método *Branch and Bound* para obter a solução ótima. A aplicação de modelagem matemática está também relacionada à capacidade de analisar situações complexas e com alto grau de dinamismo, conforme o estudo de Hongbum et al. (2014), que trata de um problema de *scheduling* com estruturas de trabalho multiníveis em um ambiente flexível *de job shop*, onde na medida em que o atraso total é considerado como uma função objetivo para concluir as partes nas datas de vencimento definidas, o plano de produção é alterado. Devido à alta complexidade e dinamismo do problema alvo, o resultado otimizado pela modelagem matemática, será integrado a um sistema de simulação, buscando quantificar um índice de eficiência da linha, considerando aspectos e parâmetros não contidos na modelagem.

O modelo de simulação desenvolvido neste trabalho, através do software FLEXSIM, retornará um tempo final de operação maior que o proposto pelo algoritmo modelado. Esta diferença de tempo será traduzida como a eficiência da linha em termos gerais, incluindo aspectos de variabilidade de processamento, falhas, movimentação, espera por operador e outros aspectos não presentes no modelo matemático.

## 2.1. Programação de produção *job shop* – Modelagem Matemática

Nesta seção será apresentado o modelo de programação linear inteira, utilizado para determinar a solução ótima para a programação de produção em um *job shop*, em um ambiente de produção com  $n$  tarefas e  $m$  máquinas, onde cada tarefa  $n$  pode ser processada nas  $m$  máquinas. Assim, o modelo apresentado busca sequenciar as operações de tal forma que minimize a duração total (*makespan*) para completar todos os *jobs* (pedido). Estas vantagens provocam redução de custo unitário para fabricação dos pedidos, tornando a operação mais competitiva e sustentável.

Para modelar matematicamente este problema deve-se definir os seguintes parâmetros e variáveis de decisão:

### Parâmetros:

$PE =$  conjunto de todos os pedidos  $\{k = 1, 2, \dots, nped\}$

$nped =$  número de pedidos

$qitem_k =$  quantidade de itens do pedido  $k$

$ini_k =$  indica o tempo em que o pedido  $k$  pode começar a ser atendido (minutos)

$prazo_k =$  indica o prazo de entrega do pedido  $k$  (minutos)

$PR =$  conjunto de todos os processos  $\{j = 1, 2, \dots, npro\}$

$npro =$  número de processos (corresponde a cada máquina. Ex.: Serra)

$pup_j =$  indica se o processo  $j$  é único numa unidade produtiva (*up*) (binário)

$up_j =$  informa a unidade produtiva a que pertence o processo  $j$  (inteiro)

$pert_{kj} =$  indica se o processo  $j$  pertence ao pedido  $k$  (matriz de dados binários)

$temp_{kj} =$  indica o tempo de cada item do pedido  $k$  no processo  $j$  (minutos)

$setup_{jk_1k_2} =$  informa o tempo de setup entre pedidos  $k_1$  e  $k_2$  (minutos)

$M =$  número muito grande

### Variáveis de decisão:

$x_{k_1k_2j} = \begin{cases} 1 & \text{informa que o pedido } k_1 \text{ é precedente de } k_2 \text{ no processo } j \\ 0 & \text{caso contrário} \end{cases}$

$u_{kj} = \begin{cases} 1 & \text{indica se o pedido } k \text{ realizou o processo } j \\ 0 & \text{caso contrário} \end{cases}$

$tini_{kj} = \text{informa o momento do início do pedido } k \text{ no processo } j$

$tfim_{kj} = \text{informa o momento do fim do pedido } k \text{ no processo } j$

$atraso_k = \text{informa o atraso do pedido } k \text{ (minutos)}$

$C_{MAX} = \text{informa o Makespan (tempo de fabricação), definido pelo tempo de finalização do último pedido deixar o sistema}$

$atraso_{MAX} = \text{informa o maior atraso entre os pedidos (minutos)}$

Função objetivo:

$$MIN = 100 * C_{MAX} + atraso_{MAX} + \sum_{k \in PE} \sum_{j \in PR} tfim_{kj} \quad (1)$$

s. a.

$$C_{MAX} \geq tfim_{kj} \quad \forall k \in PE; \forall j \in PR \mid pert_{kj} = 1 \quad (2)$$

$$atraso_{MAX} \geq atraso_k \quad \forall k \in PE \quad (3)$$

$$atraso_k \geq tfim_{kj} - prazo_k \quad \forall k \in PE; \forall j \in PR \mid pert_{kj} = 1 \quad (4)$$

$$\sum_{\substack{j_2 \in PR \\ up_{j_2} = up_{j_1}}} u_{kj_2} = 1 \quad \forall k \in PE \mid k \neq 1, k \neq nped; \forall j_1 \in PR \mid pert_{kj_1} = 1, pup_j = 0 \quad (5)$$

$$\sum_{\substack{k_2 \in PE \\ k_2 \neq k_1}} x_{k_1k_2j} \leq u_{k_1j} \quad \forall k_1 \in PE; \forall j \in PR \mid pert_{kj} = 1 \quad (6)$$

$$\sum_{\substack{k_2 \in PE \\ k_2 \neq k_1}} x_{k_2k_1j} \leq u_{k_1j} \quad \forall k_1 \in PE; \forall j \in PR \mid pert_{kj} = 1 \quad (7)$$

$$\sum_{\substack{k_2 \in PE \\ k_2 \neq k_1}} x_{k_2k_1j} = u_{k_1j} \quad \forall k_1 \in PE \mid k_1 > 1; \forall j \in PR \quad (8)$$

$$\sum_{\substack{k_2 \in PE \\ k_2 \neq k_1}} x_{k_1k_2j} = u_{k_1j} \quad \forall k_1 \in PE \mid k_1 < nped; \forall j \in PR \quad (9)$$

$$\sum_{\substack{k_2 \in PE \\ k_2 \neq k_1}} x_{k_1 k_2 j} \leq 1 \quad \forall k_1 \in PE; \forall j \in PR \mid pert_{kj} = 1 \quad (10)$$

$$\sum_{\substack{k_2 \in PE \\ k_2 \neq k_1}} x_{k_2 k_1 j} \leq 1 \quad \forall k_1 \in PE; \forall j \in PR \mid pert_{kj} = 1 \quad (11)$$

$$\sum_{k_2 \in PE} x_{k_2 1 j} = 0 \quad \forall j \in PR \quad (12)$$

$$\sum_{k_2 \in PE} x_{nped k_2 j} = 0 \quad \forall j \in PR \quad (13)$$

$$tini_{kj_2} \geq tini_{kj_1} + temp_{kj_1} - (M \times (2 - u_{kj_1} - u_{kj_2}))$$

$$\forall j_1, j_2 \in PR \mid up_{j_1} < up_{j_2}; \forall k \in PE \mid pert_{kj_1} = 1, pert_{kj_2} = 1 \quad (14)$$

$$tfim_{kj_2} \geq tfim_{kj_1} + temp_{kj_1} - (M \times (2 - u_{kj_1} - u_{kj_2}))$$

$$\forall j_1, j_2 \in PR \mid up_{j_1} < up_{j_2}; \forall k \in PE \mid pert_{kj_1} = 1, pert_{kj_2} = 1 \quad (15)$$

$$tfim_{kj} \geq tini_{kj} + (qitem_k \times temp_{kj}) - M \times (1 - u_{kj})$$

$$\forall j \in PR, \forall k \in PE \mid pert_{kj} = 1 \quad (16)$$

$$tini_{1j} \geq 0 \quad \forall j \in PR \quad (17)$$

$$tini_{k_1 j} \geq tfim_{k_2 j} + setup_{j k_2 k_1} - M \times (1 - x_{k_2 k_1 j})$$

$$\forall k_1, k_2 \in PE \mid k_1 \neq k_2; \forall j \in PR \mid pert_{k_1 j} = 1, pert_{k_2 j} = 1 \quad (18)$$

$$tini_{kj} \geq ini_k \quad \forall k \in PE, \forall j \in PR \mid pert_{kj} = 1 \quad (19)$$

$$\sum_{k_2 \in PE} x_{k_1 k_2 j} = 0 \quad \forall k_1 \in PE, \forall j \in PR \mid pert_{k_1 j} = 0 \quad (20)$$

$$\sum_{\substack{k_2 \in PE \\ P}} x_{k_2 k_1 j} = 0 \quad \forall k_1 \in PE, \forall j \in PR \mid pert_{k_1 j} = 0 \quad (21)$$

$$u_{kj} \in [0,1] \quad \forall k \in PE, \forall j \in PR \quad (22)$$

$$x_{k_1 k_2 j} \in [0,1] \quad \forall k_1, \forall k_2 \in PE; \forall j \in PR \quad (23)$$

$$atraso_{max} \geq 0 \quad \forall k \in PE \quad (25)$$

$$atraso_k \geq 0 \quad \forall k \in PE \quad (26)$$



A função objetivo, dada pela equação (1), procura minimizar o *makespan*, o tempo final dos pedidos e o máximo atraso dos pedidos. O coeficiente 100 prioriza a redução do *makespan*, a segunda parcela é responsável por garantir o mínimo atraso e a terceira parcela da função objetivo garante que o tempo fim para realizar cada pedido, seja o menor possível, mesmo que de forma individual. A equação (2) calcula o *makespan* dos pedidos. Já a equação (3), calcula o máximo atraso entre os pedidos. A equação (4) calcula o atraso de cada pedido. A equação (5) garante que cada pedido deve executar todos os processos necessários, declarado no dado de entrada. Já a restrição (6) garante que um pedido só pode ser predecessor em um processo se ele realiza este processo. Igualmente a restrição (7) garante que um pedido só pode ser sucessor em um processo se ele realiza este processo. A equação (8) determina que todo pedido, exceto o primeiro, deve ter um predecessor, enquanto a equação (9) determina que todo pedido, exceto o último, deve ter um sucessor. A desigualdade (10) garante que todo pedido só pode ser predecessor de no máximo um item, entretanto, a desigualdade (11) garante que todo pedido só pode ser sucessor de no máximo um item. A igualdade (12) garante que o primeiro pedido não deve ter um predecessor, enquanto que a igualdade (13) garante que o último pedido não deve ter um sucessor. A restrição (14) determina que o tempo de início dos pedidos em um processo deve começar após um item do pedido nos processos predecessores ser realizado. Já a restrição (15) determina que o tempo de fim dos pedidos em um processo deve terminar após um item do pedido nos processos predecessores ser realizado. A restrição (16) garante que o tempo de fim dos pedidos deve ser superior ao tempo de início acrescido ao tempo total de processamento de seus itens. A restrição (17) determina que o tempo de início do pedido fictício deve ser maior ou igual a 0, para qualquer processo. A desigualdade (18) garante que o tempo de início de um pedido, em um processo, deve ser superior ao final do pedido anterior acrescido de tempo para *setup*. A restrição (19) garante que os processos de um pedido não podem iniciar antes do prazo mínimo. Já a equação (20) e (21), em conjunto, garantem que se um pedido não realiza nenhuma operação, não pode preceder um pedido nem ser precedido por outro. Já as restrições (22), (23), (24), (25) e (26) representam os domínios das variáveis de decisões.

## 2.2. Aplicação do modelo matemático de programação

Para que o algoritmo de programação linear inteira possa ser executado, os dados precisam ser expostos de forma ordenada e padronizada. Para isto, foi utilizada uma aplicação, onde os dados dos pedidos que precisam ser executados são inseridos em uma planilha, que deve ser preenchida em duas etapas.





pedido e a própria base de dados já identificará o tempo de ciclo e tempo de setup para cada tipo de material. No caso dos tornos, caso for possível processar nos dois tornos, deve-se marcar os dois tornos. Para este caso, o modelo irá determinar de forma ótima em qual torno o material deverá ser processado.

Após a inserção dos dados nesta planilha base, para obter o resultado otimizado do plano de produção, é necessário abrir a programação feita no software LINGO. Esta programação é a tradução da modelagem exposta no início deste capítulo, de forma a ser interpretada pelo LINGO.

Para otimização dos 6 pedidos foi utilizado um computador com processador Intel I3 – 16GB RAM, oferecendo a solução ótima em 3 minutos e 15 segundos. O resumo desta solução é exibido logo após a finalização da otimização, conforme Figura 3.

Figura 3: Saída de dados LINGO



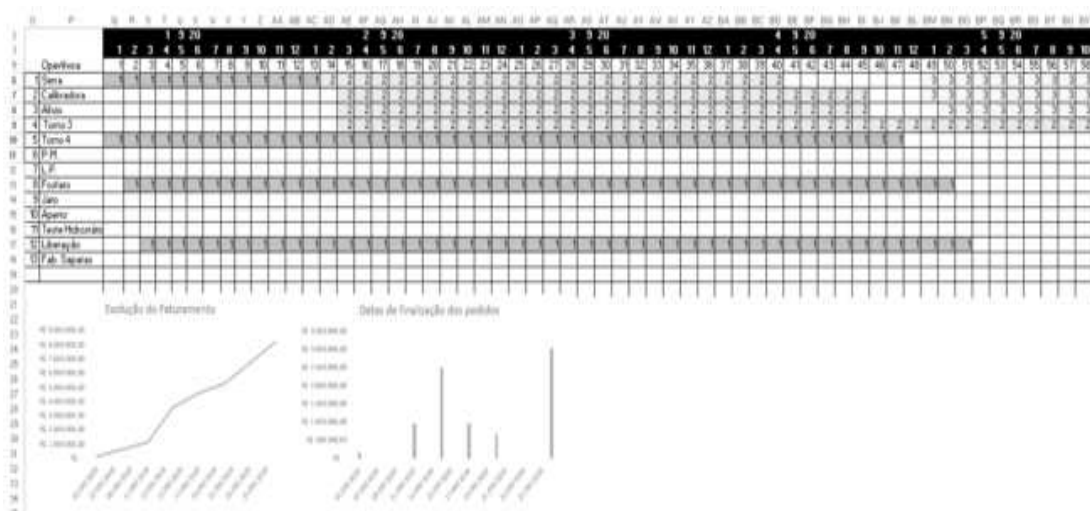
Fonte: O autor, 2022

Além desta saída de dados, que não auxilia de forma prática na atividade de programação da produção, é exportado para planilha base o sequenciamento dos pedidos por tempo e ocupação dos equipamentos. Através da Figura 4 é possível observar que a otimização dos pedidos os quais estão servindo de exemplo, seriam executados fisicamente em 311 horas.

Estes resultados, exibidos em forma de gráfico de Gantt trazem uma oportunidade para simplificar a leitura e acompanhamento de evolução da produção dos pedidos, conforme Figura 4. Através desta tela, sob a ótica do resultado ótimo, gerada através de programação

linear inteira, é possível saber qual pedido acabará primeiro, qual ordenação por tipo de máquina, qual a ociosidade prevista de cada máquina, quais serão os pontos de gargalo para cada tipo de processamento, qual a evolução de faturamento durante o período programado, quais serão as máquinas críticas para os processamentos, quais serão os melhores períodos previstos para intervenções de manutenção ou melhorias e diversas outras oportunidades que dependem de uma visão de integração dos pedidos programados com a *job shop*.

Figura 4: Saída de dados LINGO – EXCEL



Fonte: O autor, 2022

Comparando os resultados obtidos com esta ferramenta proposta e o método atual, podemos perceber grandes oportunidades. Atualmente o plano de produção para a *job shop* denominada Fábrica de acessórios é feito considerando apenas os tornos de forma individual, ou seja, é disponibilizado para produção uma planilha com duas abas, com a sequência de materiais a serem usinados, bem como seus tempos de ciclo para o torno 3 e torno 4.

### 3. Validação do modelo através de simulação no Flexsim

O modelo de programação linear inteira, considerando os 6 pedidos tomados como exemplo, mostrou uma previsão de conclusão de todos os pedidos, 360 peças, em 311 horas, ou seja, 18660 minutos. É verdade que o resultado obtido traz a luz fatos como andamento da execução dos pedidos e taxas de ociosidade / ocupação por máquina, contudo, é necessário avaliar a capacidade da fábrica de acessórios em executar este plano, considerando seus atuais recursos de movimentação, máquinas, taxas de falhas, disponibilidade de operadores, tamanho da fábrica, distância ente os postos, etc.



Para realizar esta tarefa foi realizada a modelagem da fábrica de acessórios no *software* de simulação FLEXSIM, sob a planta baixa da Fábrica de acessórios, respeitando as dimensões e proporções originais da fábrica.

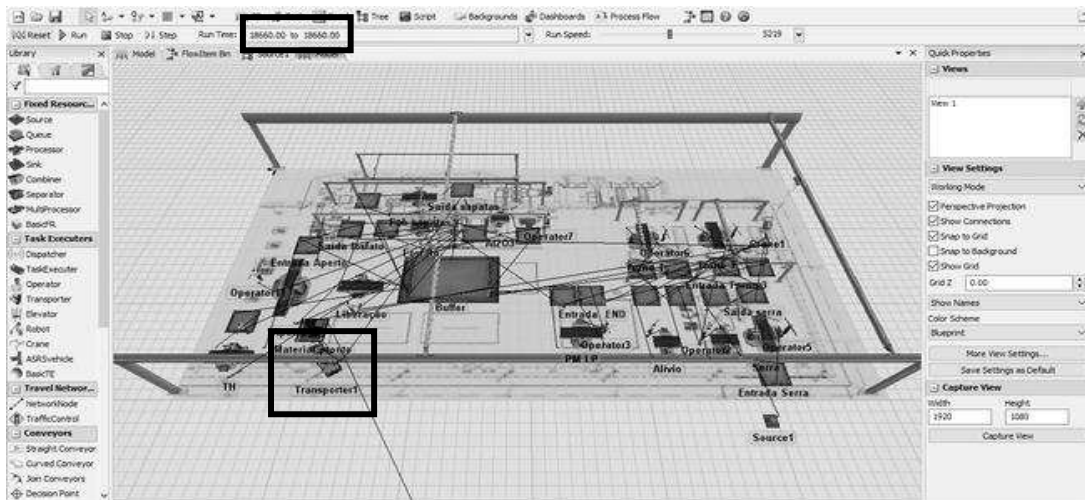
A modelagem foi feita utilizando unidades de tempo, minutos e comprimento, metros. Além da modelagem do *layout* da fábrica, foram vinculados todos os parâmetros de operação da *job shop* como tempos de ciclo por máquina, tempos de *setup*, taxas de falha, velocidade de movimentação das pontes rolantes, prioridades de atendimento à movimentação, taxa de fadiga de operador, distância entre recursos e outros.

Para testar e validar a aderência do modelo de simulação com a realidade foi acompanhado um pedido de produção de 50 peças. Estas peças foram executadas na fábrica em 2420 minutos, considerando todos os processamentos e movimentações. No modelo de simulação estas mesmas 50 peças, com mesma configuração foram executadas em 2418 minutos, refletindo uma aderência de 99,92% entre o modelo de simulação e a realidade.

Para poder validar a construção do modelo de simulação é preciso fazer um *setup* do modelo, equiparando-o ao algoritmo de programação linear inteira, desenvolvido no LINGO. Esta validação foi feita eliminando todos os tempos de movimentação e índices de falhas da fábrica, considerando apenas os valores de tempo de ciclo e *setup* das máquinas. Com este protocolo foi possível obter, para realização completa dos 6 pedidos citados, o mesmo tempo de execução indicado como tempo ótimo pelo algoritmo de programação linear inteira.

A seguir está a Figura 5 com o tempo travado em 18660, com a última peça sendo despachada pela empilhadeira.

Figura 5: Resultado FLEXSIM – Sem movimentações



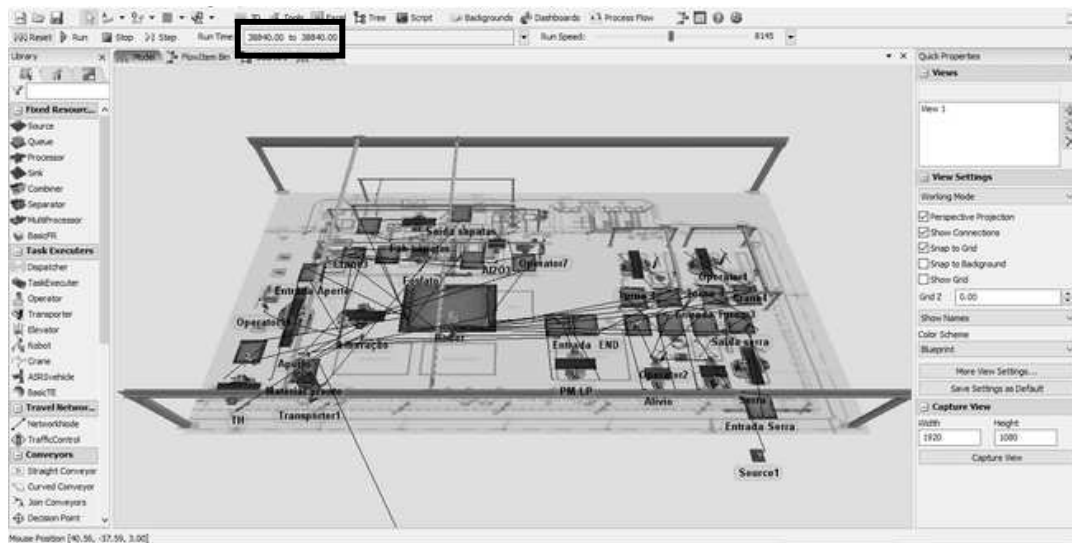
Fonte: O autor, 2022

Como próxima etapa, é necessário atribuir quais serão os recursos de movimentação e taxa de falha para cada entidade do modelo. Na fábrica de acessórios, de forma geral, as movimentações são realizadas com pontes rolantes, pórticos ou talhas. Apenas em casos específicos de movimentações do tubo sobre a banca, é possível ser feita manualmente, ou ainda como exceção, para caso de peças leves é possível realizar a movimentação com um carrinho de movimentação de peças curtas. Desta forma, todas as tarefas de movimentação fora dos postos de trabalho foram atribuídas às pontes, pórticos e talhas.

O *layout* celular em um ambiente *job shop*, como dito anteriormente, costuma trazer certo nível de perda, na produtividade geral na fábrica. Este fato é associado ao tempo de movimentação entre os postos e espera por estas movimentações, visto que não há interligação entre os postos pelos quais os materiais poderiam fluir.

Na Figura 6 temos o resultado de *makespam* da simulação considerando todas as movimentações sendo realizadas por equipamentos de guindar, conforme realidade encontrada na fábrica de acessórios.

Figura 6: Resultado FLEXSIM – Com movimentações



Fonte: O autor, 2022

Na Figura 6 é possível notar que o tempo necessário para finalizar os 6 pedidos, 360 peças, foi de 38840 minutos. Este tempo excede em 20180 minutos o tempo ótimo programado, no LINGO. Este tempo adicional faz com que a fábrica apresente um desempenho de 49%, frente a programação ótima indicada. Esta perda de produtividade está associada à movimentação e taxas de falhas das máquinas e deve ser considerada para cálculo de OTE (*Overall Throughput Effectiveness*) ou *Plant OEE* da fábrica, sendo esta uma variação do OFE (*Overall Factory Effectiveness*), vindo do OEE (*Overall Equipment Effectiveness*).

Para levar a ideia da medição pelo OEE ao nível de toda uma fábrica, existe a proposta do *Overall Factory Effectiveness* (OFE) – ou *Plant OEE*, como chamado por Högfeldt (2005) – de avaliar o desempenho global da mesma admitindo que isso resulte da interação de diferentes máquinas/processos e de decisões/ações tomadas por seus diversos sistemas e subsistemas (OECHSNER et al., 2003).

Uma variante do OFE seria o *Overall Throughput Effectiveness* (OTE), que considera a forma de conexão existente entre os equipamentos. Muthiah (2007) sugerem que os subsistemas que constituem uma fábrica podem assumir quatro tipos básicos de configuração – equipamentos em série, em paralelo, montagem alimentada por múltiplos equipamentos e equipamento que fornece para vários outros –, sendo que no cálculo do OTE para cada tipo de subsistema é preciso observar como sua produção total é limitada pela capacidade dos recursos restritivos e ineficiências encontradas no percurso do fluxo produtivo. Conforme Muthiah (2007), assim como a produção de equipamentos encadeados em série é limitada pelo equipamento-gargalo,

para se obter o numerador global do OFE de uma fábrica é necessário identificar o subsistema-gargalo e verificar como seu OTE restringe as saídas finais.

#### 4. Considerações finais

A ferramenta proposta, baseada em programação linear inteira, foi validada através de simulação computacional, desde que a análise de desempenho através de indicadores de desempenho como OTE e OFE sejam aplicados, afim de trazer para o modelo matemático um valor que represente todas as ineficiências e perdas não computadas dentro do processo produtivo.

Esta ferramenta de programação de produção, baseado em programação linear inteira, é um recurso que traz visão de como um plano de produção pode ser executado de forma ótima, além disso, contribui para trazer a luz o conhecimento sobre diversos aspectos na interface entre o plano de produção e o ambiente de produção, como tempos de ociosidade, oportunidades de melhoria em sistemas de movimentação e indicar necessidades de futuros investimentos.

#### 5. Conclusão

Este trabalho disponibiliza uma ferramenta para programação de produção de uma *job shop*, sob a ótica da *job shop* intitulada fábrica de acessórios, capaz de fornecer uma visão integrada do ambiente produtivo, buscando antecipação nas informações como possíveis atrasos ou necessidades de mudanças de planos. Desta forma, é possível aumentar a geração de valor para os clientes, entregando materiais no prazo e entender novas oportunidades de operações dentro da *job shop*.

A ferramenta desenvolvida permite que, em poucos minutos, mais de 98 milhões de possibilidades de fluxos produtivos sejam testadas e validadas pelo modelo. Este fato é interessante tanto para a empresa produtora, quanto para o mercado capaz de se munir de informações para melhores negociações.

A integração do modelo desenvolvido à estratégia de médio e longo prazo da empresa também é uma oportunidade, visto que é possível termos uma resposta de quantos e quais operadores serão necessários para produção de determinada carteira de médio prazo. Este tipo



de possibilidade traz a luz do conhecimento aspectos que hoje podem gerar ineficiência em custos.

Com esta oportunidade, a empresa conseguirá confrontar os custos esperados para cada tipo de demanda e conciliar o melhor custo, com o menor atraso e com a maior receita possível. Desta forma, os resultados financeiros projetados para empresa ficam mais claros, tornando-se possivelmente atrativa para novos investidores e clientes.

## REFERÊNCIAS

- DURAZO-CARDENAS, I., STARR, A., TURNER, C. J., TIWARI, A., KIRKWOOD, L., BEVILACQUA, M., ... EMMANOUILIDIS, C. (2018). An autonomous system for maintenance scheduling data-rich complex infrastructure: Fusing the railways' condition, planning and cost. *Transportation Research Part C: Emerging Technologies*, 89, 234–253. doi:10.1016/j.trc.2018.02.010
- HÖGFELDT, D. Plant efficiency: a value stream mapping and overall equipment effectiveness study. 2005. 93 f. Master Thesis (Master of Science Programme - Mechanical Engineering)-University of Technology, Lulea, 2005.
- HONGBUM N. e JINWOO P. *Multi-level job scheduling in a flexible job shop environment*. *International Journal of Production Research*, 52:13, 3877-3887. 2014.
- KRAJEWSKI, L. J. e RITZMAN, L. P. (2010). *Operations management: strategy and analysis*. 8ª ed. Addison-Wesley Publishing Company, Massachusetts.
- LIU, M., YI, S., & WEN, P. (2016). Quantum-inspired hybrid algorithm for integrated process planning and scheduling. *Proceedings of the Institution of Mechanical Engineers, Part B: Journal of Engineering Manufacture*, 232(6), 1105–1122. doi:10.1177/0954405416661006.
- MOREIRA M.R. et al. *A methodology for planning and controlling workload in a job-shop: a four-way decision-making problem*. *International Journal of Production Research*, Vol. 47, nº 10, pp. 2805-2821. 2019.
- MUTHIAH, K. M. N.; HUANG, S. H. Overall Throughput Effectiveness (OTE) metric for factory-level performance monitoring and bottleneck detection. *International Journal of Production Research*, v. 45, n. 20, p. 4753-4769, 2007. <http://dx.doi.org/10.1080/00207540600786731> MUTHIAH, K. M. N.; HUANG, S. H.;
- OECHSNER, R. et al. From overall equipment effectiveness to overall Fab effectiveness (OFE). *Materials Science in Semiconductor Processing*, v. 5, n. 4-5, p. 333-339, 2003. [http://dx.doi.org/10.1016/S1369-8001\(03\)00011-8](http://dx.doi.org/10.1016/S1369-8001(03)00011-8)
- RIETZ J., ALVES C., de CARVALHO J.V. Fast Heuristics for Integrated Planning and Scheduling. In: Gervasi O. et al. (eds) *Computational Science and Its Applications -- ICCSA 2015*. ICCSA 2015. Lecture Notes in Computer Science, vol 9156. Springer, Cham. [https://doi.org/10.1007/978-3-319-21407-8\\_30](https://doi.org/10.1007/978-3-319-21407-8_30)BANKS, J.; CARSON J. S. & NELSON B. L. *Discrete-Event System Simulation*. Englewood Cliffs, NJ, USA: Prentice hall, 2015.
- ROBERT H. STORER, S. DAVID Wu e RENZO Vaccari. *New Search Spaces for Sequencing Problems with Application to Job Shop Scheduling*. 2015.
- SANTOS, Welber Lopes; SCHWAB, Marcos Vinicius. First article build process for new Pre-salt equipments *Technical Paper - Rio Oil & Gas 2018 Bloco I Exploração & Produção*, Cód. 38.970. 2018.
- SHEN, Z., & SMALOV, L. Comparative Performance of Genetic Algorithm, Simulated Annealing and Ant Colony Optimisation in solving the Job-shop Scheduling Problem. 2018 26th International Conference on Systems Engineering (ICSEng). doi:10.1109/icseng.2018.8638185. 2018.