

INSTITUTO FEDERAL DE SANTA CATARINA

MAURICIO DE SOUZA GUIMARÃES

**UM MODELO MATEMÁTICO PARA O PROBLEMA DE PROGRAMAÇÃO DA
PRODUÇÃO DE UMA INDÚSTRIA DE MÁQUINAS ESPECIAIS SOB
ENCOMENDA**

Caçador - SC

Dezembro / 2020

MAURÍCIO DE SOUZA GUIMARÃES

**UM MODELO MATEMÁTICO PARA O PROBLEMA DE PROGRAMAÇÃO DA
PRODUÇÃO DE UMA INDÚSTRIA DE MÁQUINAS ESPECIAIS SOB
ENCOMENDA**

Trabalho de conclusão de curso apresentado
ao curso de Engenharia da Produção do
Câmpus Caçador do Instituto Federal de
Santa Catarina para a obtenção do diploma
de Bacharel em Engenharia da Produção

Orientador: Prof. MSc Bruno Santos Vieira

Caçador – SC

Dezembro / 2020

Guimarães, Maurício de Souza

G963m Um modelo matemático para o problema de programação da produção de uma indústria de máquinas especiais sob encomenda / Maurício de Souza Guimarães ; orientador: Bruno Santos Vieira. -- Caçador, SC, 2020.

56 f.

Trabalho de Conclusão de Curso (Graduação)-Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina, Curso de Engenharia de Produção.

Inclui bibliografias

1. Programação da produção. 2. Hora extra Job Shop. 3. GAMS. I. Vieira, Bruno Santos. II. Instituto Federal de Educação, Ciência e Tecnologia de Santa Catarina. Curso de Engenharia de Produção. III. Título.

CDD 658.5

Ficha catalográfica elaborada pela Bibliotecária
Karla Viviane Garcia Moraes – CRB-14/1002

MAURICIO DE SOUZA GUIMARÃES

**UM MODELO MATEMÁTICO PARA O PROBLEMA DE PROGRAMAÇÃO DA
PRODUÇÃO DE UMA INDÚSTRIA DE MÁQUINAS ESPECIAIS SOB
ENCOMENDA**

Este trabalho foi julgado adequado para obtenção do título em (Nome da
Habilitação), pelo Instituto Federal de Educação, Ciência e Tecnologia de Santa
Catarina, e aprovado na sua forma final pela comissão avaliadora
abaixo indicada.

Caçador, 21 de Dezembro de 2020

**BRUNO
SANTOS VIEIRA
07536969910**

Assinado digitalmente por BRUNO SANTOS VIEIRA:
07536969910
DN: C=BR, O=ICP-Brasil, OU=presencial,
OU=34028316000103, OU=Secretaria da Receita
Federal do Brasil - RFB, OU=ARCORREIOS, OU=RFB
*e-CPF A3, CN=BRUNO SANTOS VIEIRA.07536969910
Razão: Eu estou aprovando este documento com minha
assinatura de vinculação legal
Localização: sua localização de assinatura aqui
Data: 2021.02.26 19:13:13-03'00"
Foxit Reader Versão: 10.1.1

Prof. Bruno Santos Vieira Me.

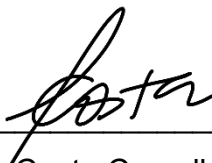
Orientador

Instituto Federal de Santa Catarina



Prof. Lucio Galvão Mendes Me.

Instituto Federal de Santa Catarina



Prof. Eric Costa Carvalho Me.

Instituto Federal de Santa Catarina

RESUMO

O propósito desse trabalho é desenvolver um modelo, programado em GAMS, que faça o sequenciamento de tarefas para uma produção do tipo *Job Shop*, em que seja possível a alocação de tarefas em regime de horas extras, caso necessário, afim de atender os prazos de entrega. O modelo proposto considera a possibilidade de programar tarefas em horas extras e também criação de segundo turno, de forma independente para cada estação de trabalho, com o objetivo de minimizar o custo operacional. Para isso foi utilizada uma abordagem relacionada aos problemas de dimensionamento de lotes chamados *Capacited Lot Sizing Problems*, os quais, quando considerando curtas janelas de tempo, culmina por sequenciar a produção ao invés de dimensionar os lotes. A janela de tempo utilizada foi de um expediente normal de 8,3 horas, que pode ser ampliada a cada dia se necessário, representando trabalho em hora extra ou ainda mais turnos de trabalho, nesse caso indicando a necessidade de terceirizar parte da produção ou aumentar a capacidade. A utilização desse modelo permite não só a programação da produção, como também a simulação de diferentes cenários, melhorando a qualidade do fluxo de informações entre PCP e departamento de vendas durante a negociação dos prazos de entrega. Também auxilia o profissional do PCP quanto à tomada de decisão em relação a programar de fato horas extras, aumentar capacidade produtiva, terceirizar parte da produção, entre outros. Os resultados obtidos ao final do trabalho foram satisfatórios, testando o modelo com alguns problemas encontrados na literatura e também com um problema real com dados reais da empresa estudada, este conseguiu programações sub ótimas, programando poucas atividades em horas extras (entre 0% e 3,9% das horas). Este modelo proposto para sequenciamento tem como benefício a flexibilidade da programação e a minimização de custos operacionais como objetivo.

Palavras-Chave: Sequenciamento. Programação da Produção. Hora extra *Job Shop*. GAMS.

ABSTRACT

The purpose of this work is to develop an algorithm, programmed in GAMS, that sequences tasks in a Job Shop environment, where it is possible to allocate tasks considering overtime work or a second shift, if necessary, to reach delivery deadlines. The proposed algorithm considers the possibility of scheduling overtime work and also creation of a second shift, independently for each workstation, in order to minimize production cost. For this, it was considered an approach related to the problems of lot sizing called Capacited Lot Sizing Problems. This approach, while with short time windows, end up sequencing the production in addition to sizing the lots. The time window used was a normal 8.3-hour work shift, which can be extended each day if necessary, representing overtime work or additional work shifts. The use of this algorithm allows not only to schedule production, but also to simulate different scenarios, improving the quality of the information exchanged between PCP and the sales department during the negotiation of delivery times. It also assists the PCP professional to make decisions in relation of scheduling overtime, increasing production capacity, outsourcing part of production and so on. The results reached in this work were satisfactory, testing the model with some problems found in the literature and also with a real problem with real data, model reached sub optimal schedules, programming few activities in overtime (between 0% and 3.9 % of total hours). This proposed model has the benefit of flexibility in scheduling and minimize operational cost is the objective.

Keywords: Sequencing. Production Schedule. Overtime work. Job Shop. GAMS.

LISTA DE FIGURAS

| | |
|--|----|
| Figura 1 – Representação do <i>Makespan</i> em um gráfico de Gantt | 13 |
| Figura 2 – Fluxograma da metodologia de modelagem | 26 |
| Figura 3 – Picador de cavacos PC130/100 | 31 |
| Figura 4 – Gráfico de Gantt representando tempo contínuo | 32 |
| Figura 5 – Gráfico de Gantt representando a realidade da empresa estudada..... | 32 |
| Figura 6 – Detalhe do modelo | 37 |
| Figura 7 – Gráfico de análise de sensibilidade PC130/100 | 44 |

LISTA DE ABREVIATURAS E SIGLAS

ATO – Assemble-to-Order
CNC – Comando Numérico Computadorizado
EDD – Earliest job Due Date
ERP – Enterprise Resource Planning
ETO – Engineering-to-Order
GAMS – *General Algebraic Modeling System*
JSSP – *Job Shop Scheduling Problem*
LPT – Longest Processing Time
MTO – Make-to-Order
MTS – Make-to-Stock
PCP – Planejamento e Controle da Produção
PMP – Plano Mestre de Produção
SPT – Shortest Processing Time
TI – Tecnologia da informação
WIP – Work in Process

SUMÁRIO

| | |
|--|-----------|
| 1 INTRODUÇÃO | 11 |
| 1.1 Contextualização | 11 |
| 1.2 Justificativa | 12 |
| 1.3 Objetivos | 14 |
| 1.3.1 Objetivo geral | 14 |
| 1.3.2 Objetivos específicos..... | 14 |
| 2 FUNDAMENTAÇÃO TEÓRICA | 15 |
| 2.1 Planejamento e Controle da Produção | 15 |
| 2.1.1 Ambientes de manufatura..... | 15 |
| 2.1.2 Funções do PCP em ambiente de manufatura do tipo sob encomenda..... | 15 |
| 2.1.3 Sequenciamento da Produção em ambiente de manufatura sob encomenda . | 16 |
| 2.2 Modelos matemáticos | 17 |
| 2.2.1 O modelo de Manne | 17 |
| 2.2.2 O modelo de Chen (2001) | 19 |
| 2.2.3 Modelo de dimensionamento de lotes | 20 |
| 2.2.4 GAMS (General Algebraic Modeling System)..... | 22 |
| 2.3 Termos e conceitos utilizados ao longo do trabalho | 24 |
| 2.3.1 Ambientes de sequenciamento | 24 |
| 2.3.2 Conceitos abordados ao longo do trabalho | 24 |
| 3 METODOLOGIA | 26 |
| 3.1 Definição do problema | 27 |
| 3.2 Construção do modelo | 27 |
| 3.3 Solução do modelo | 28 |
| 3.4 Validação do modelo | 28 |
| 3.5 Implementação da solução | 29 |
| 4 DESENVOLVIMENTO, ANÁLISE E DISCUSSÃO DOS RESULTADOS | 30 |
| 4.1 Estruturação inicial do problema | 30 |
| 4.2 Requisitos do modelo | 31 |
| 4.3 O modelo proposto | 34 |
| 4.4 Aplicações e funções do modelo | 38 |
| 4.5 Validação do modelo | 39 |
| 4.6 Implementação do modelo em um caso real | 41 |

| | |
|--|-----------|
| 4.7 Análise de sensibilidade..... | 43 |
| 5 CONCLUSÃO | 45 |
| REFERÊNCIAS..... | 47 |
| APÊNDICE 1..... | 50 |
| APÊNDICE 2..... | 52 |

1 INTRODUÇÃO

1.1 Contextualização

O planejamento da produção é uma atividade desenvolvida tanto por empresas que produzem bens, como também por empresas prestadoras de serviço (TUBINO, 2017). Esta atividade está associada ao departamento de PCP (Planejamento e Controle da Produção) e está intimamente ligada à competitividade das organizações, visto que esta tem como finalidade planejar as atividades realizadas na produção, com o objetivo final de atender uma demanda de produtos, com quantidades e prazos definidos, obedecendo requisitos de qualidade, e utilizando os recursos da empresa da melhor forma possível, minimizando os custos produtivos (PEDROSO, 1996).

A complexidade do planejamento da produção depende de vários fatores, mas especialmente do tipo de sistema produtivo que a empresa opera. Empresas que produzem em processos contínuos ou em massa possuem um planejamento mais focado no abastecimento de matéria-prima nas linhas. Já empresas que produzem em lotes ou bateladas, possuem foco no dimensionamento de lotes de produção e sequenciamento desses lotes. Essas empresas geralmente possuem diferentes produtos em seu portfólio, e a produção precisa ser planejada em função da demanda de cada produto ou mix de produtos (TUBINO, 2017).

As empresas que produzem sob encomenda, ou por projeto, podem ter grande complexidade no planejamento. As demandas dos produtos são baixas, tendendo à unidade e a flexibilidade é alta, ou seja, os projetos geralmente são customizados de acordo com as necessidades dos clientes, geralmente o *Layout* produtivo é organizado por processos e o papel do PCP é complexo em todos os seus níveis. No nível operacional, o PCP precisa alocar tarefas nas máquinas ou departamentos transformadores ao longo do tempo para produzir cada componente que faz parte da estrutura do produto, ou dos diversos produtos que podem estar na produção simultaneamente, tarefa esta, conhecida como sequenciamento da produção (TUBINO, 2017). Diante da complexidade existente na tarefa de sequenciamento da produção em empresas que produzem sob encomenda, surgiu a ideia de desenvolver este modelo.

Este trabalho foi desenvolvido para uma empresa, localizada na cidade de

Caçador/SC, cidade onde o autor reside. A empresa estudada é um fabricante de máquinas especiais para a indústria madeireira. Dentre seus principais produtos é possível citar, máquinas para linha de laminação de madeira, máquinas para linhas de secagem e classificação, picadores de cavacos estacionários e florestais, entre outros. A empresa produz sob encomenda, iniciando o projeto somente após fechamento do contrato com o cliente. As máquinas produzidas pela empresa são máquinas pesadas, composta por centenas ou até milhares de peças. A empresa estudada possui atualmente cerca de 160 colaboradores, seu layout é organizado por processo. As peças que compõe as máquinas são praticamente todas produzidas na própria empresa, exceto acessórios como motores, rolamentos, componentes pneumáticos, hidráulicos, elétricos e etc.

Das peças produzidas a grande maioria inicia sua produção nos setores de corte, onde a empresa conta com serras para cortes de perfis de aço, corte a plasma para chapas grossas e guilhotinas para chapas finas. Em seguida uma parte das peças para pelo setor de caldeiraria e solda, outras vão diretamente para a usinagem, que conta atualmente com tornos, fresas, mandriladoras, plainas, retíficas, furadeiras, tanto máquinas convencionais como CNC's.

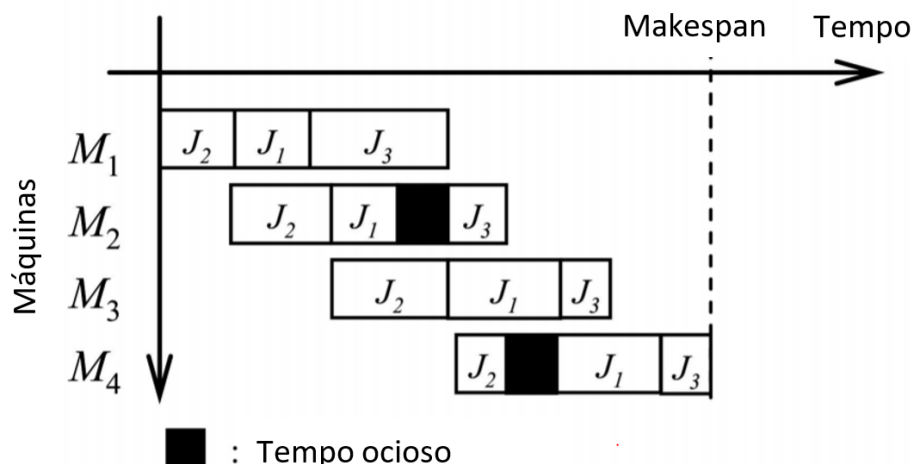
A empresa estudada produz máquinas para a indústria madeireira desde 1950, atualmente o principal mercado onde a empresa vende seus produtos é para Rússia e todo o leste Europeu.

1.2 Justificativa

Existem muitos modelos de algoritmos que fazem o sequenciamento de tarefas. O problema de sequenciamento de tarefas é conhecido como *Job Shop Scheduling Problem* (JSSP), e um objetivo muito comum para os algoritmos existentes é a minimização do *Makespan*, o tempo total para completar todas as tarefas do conjunto de peças que está sendo sequenciada (BLAŻEWICZ; DOMSCHKE; PESCH, 1996).

A Figura 1 exemplifica o *Makespan* em um sistema com 3 peças (Jobs) que são processadas em 4 máquinas diferentes, máquinas estas que podem ser serras, solda, tornos e etc, ou seja, máquinas que irão processar as peças. No eixo horizontal a representação do tempo e no eixo vertical as máquinas ou estações de trabalho que processam cada tarefa das peças.

Figura 1 – Representação do *Makespan* em um gráfico de Gantt



Fonte: Adaptado de Sauvey, Sauer (2020)

Com uma abordagem sob o viés da Pesquisa Operacional, utilizando algoritmos clássicos de sequenciamento, é possível conseguir resultados ótimos quanto à minimização do *Makespan*, porém, essa abordagem utilizando os modelos clássicos muitas vezes descarta soluções interessantes, visto que algumas restrições precisam ser desconsideradas para simplificar o problema (BAPTISTE; LE PAPE; NUIJTEN, 1995). Os algoritmos que resolvem problemas JSSP, em geral, tratam o tempo de forma contínua e consideram que as tarefas são executadas de forma ininterrupta em cada máquina, ou seja, nenhuma tarefa pode ser interrompida para depois ser finalizada em um momento futuro, isso impossibilita a simulação de parada de alguns departamentos no final de um expediente normal e ativação de trabalho sob regime de hora extra em outros departamentos ou estações de trabalho (JAIN; MEERAN, 1998).

Fabricantes de máquinas customizadas, como é o caso da empresa estudada, eventualmente fecham seus pedidos com prazo de entrega não factível de atender executando as tarefas somente sob regime de trabalho normal. Assim sendo, os algoritmos clássicos de *Job Shop Scheduling* não são perfeitamente adaptados à essa realidade, pois, com alguma frequência, ocorrem casos em que é necessário operar sob regime de hora extra em algumas estações de trabalho, ou ainda operar em dois ou três turnos em alguns setores, para conseguir atender os prazos de entrega.

Pesquisar e elaborar um modelo matemático para o problema de sequenciamento da produção, considerando a possibilidade de operar sob regime

especial de trabalho em alguns setores da empresa, e apenas em dias que sejam convenientes para atender os prazos de entrega minimizando custos, é de grande utilidade para a indústria, especialmente para fabricantes de máquinas que trabalham sob encomenda, para os quais a demanda não é constante, e normalmente diferentes projetos estão simultaneamente em produção, com diferentes prazos de entrega. A variação da demanda pode causar ociosidades em alguns departamentos (TUBINO, 2017), porém em outros, é comum ocorrerem sobrecargas em determinados períodos, sendo muitas vezes necessário a execução de tarefas em regime de trabalho especial ou ainda a necessidade de terceirizar parte da produção ou aumentar capacidade produtiva.

A utilização de uma ferramenta que faz esse sequenciamento de tarefas minimizando custos traz diversos benefícios para a indústria, entre eles é possível citar a redução de custos de produção, a possibilidade de simulação de diferentes cenários facilitando a tomada de decisão tanto a nível tático quanto operacional, interação mais eficiente entre PCP e Vendas, devido a facilidade na simulação de carregamentos em função de diferentes datas de entrega, entre outros.

1.3 Objetivos

1.3.1 Objetivo geral

Este trabalho tem como objetivo elaborar um modelo matemático para o problema de sequenciamento da produção em ambiente *Job Shop*, em que o modelo tenha flexibilidade para programar tarefas em regimes especiais de trabalho, de forma a atender prazos de entrega específicos, com o menor custo possível.

1.3.2 Objetivos específicos

Os objetivos específicos deste trabalho são:

- a) Identificar as especificidades da indústria estudada que devem ser levadas em consideração para a construção do modelo;
- b) Elaborar o modelo matemático e escrever o modelo no GAMS;
- c) Testar o modelo proposto com problemas de grandes dimensões e avaliar a viabilidade de usar o modelo proposto para utilização na indústria estudada, com relação à qualidade do resultado obtido e tempo de processamento.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Planejamento e Controle da Produção

2.1.1 Ambientes de manufatura

As atividades e decisões tomadas pelo PCP em relação à produção não são iguais para todas as indústrias, estas variam em função de alguns fatores, dentre estes, é possível citar: O horizonte de planejamento, que pode ser curto, médio e longo prazo; as perguntas que precisam ser respondidas em relação ao que produzir, quando produzir, quanto produzir e com que recursos produzir; decisões em relação ao gerenciamento e controle da demanda, planejamento e controle dos recursos internos e externos, e, também dependem do ambiente de manufatura que pode ser MTS (*Make-to-Stock*), MTO (*Make-to-Order*), ATO (*Assemble-to-Order*) e ETO (*Engineering-to-Order*) (MARTINS; LAUGENI, 2005).

Bremer et al (2000) descrevem os tipos de ambiente de manufatura. No ambiente MTS os produtos são produzidos por antecipação de acordo com uma estimativa de demanda. No ambiente ATO os componentes, subconjuntos e alguns materiais são produzidos e armazenados, aguardando o pedido dos clientes para efetuar a montagem do produto final. No ambiente MTO o autor cita que o projeto básico do produto pode ser feito a partir do contato inicial com o cliente, mas a produção de fato só inicia após recebimento do pedido de compra. Já no ambiente ETO o projeto do produto é desenvolvido baseado nas especificações do cliente.

2.1.2 Funções do PCP em ambiente de manufatura do tipo sob encomenda

Geralmente, a primeira interação do PCP em um projeto ocorre quando o setor de vendas está negociando com o cliente o prazo de entrega de um produto. Nesse momento, o PCP é acionado e precisa verificar, por meio de um sistema de informação, qual é o carregamento dos diversos departamentos da fábrica ao longo do tempo, para em seguida simular a entrada do novo projeto no sistema a fim de avaliar qual é o prazo de entrega que o setor de vendas pode negociar com o cliente (TUBINO, 2017).

A nível estratégico o PCP precisa montar um plano de produção, o qual, baseado em uma estimativa de demanda para seus produtos, irá determinar qual é a

capacidade que seu sistema produtivo precisa ter para atender tal demanda. De acordo com Quezado *et al* (1999) o papel do PCP no nível estratégico está relacionado à tomada de decisões como compra de equipamentos, ampliação ou redução da capacidade produtiva, subcontratação ou demissão da quantidade de recursos humanos, definição das horas-homem disponível, definição do tipo de produto que será produzido, resultando assim no Plano Estratégico de Produção.

A nível tático o PCP desenvolve o Plano Mestre de Produção (PMP) em função dos pedidos em carteira e também de alguma estimativa de médio prazo baseado na capacidade do seu sistema produtivo que já foi previamente dimensionado no Plano Estratégico de Produção. Nessa etapa o PCP irá tomar decisões como determinar quantidade de turnos de operação e terceirizar ou não parte da produção (TUBINO, 2017).

À nível operacional o PCP faz a Programação e Controle da Produção. As tarefas relacionadas ao PCP no nível operacional são sequenciamento da produção, emissão de ordens de compra, ordens de fabricação, ordens de montagem e administração de estoques (QUEZADO, 1999).

2.1.3 Sequenciamento da Produção em ambiente de manufatura sob encomenda

O sequenciamento da produção pode ser executado por meio de regras heurísticas ou ainda podem ser empregadas técnicas de Pesquisa Operacional como Programação Linear, Programação Inteira, Grafos e etc. (TUBINO, 2017), visando otimizar o sequenciamento da produção, maximizando ou minimizando algum ou alguns parâmetros, como por exemplo a minimização do *Makespan*, minimização do atraso máximo, minimização do máximo atraso ponderado, minimização da quantidade de trabalhos atrasados entre outros (ABDOLRAZZAGH-NEZHAD, 2017). Todavia, a minimização do *Makespan* é o objetivo mais utilizado (CHEN; LUH; FANG, 2001).

Os sistemas que fazem o sequenciamento por meio de heurísticas utilizam regras de liberação, ou seja, a cada processo concluído, o sistema faz uma verificação dentre as diversas tarefas que estão disponíveis para serem executadas em um posto de trabalho e define qual é a próxima tarefa a ser executada de acordo com alguma regra pré-estabelecida. De acordo com Pedroso (1996), a literatura contem cerca de uma centena de regras de liberação e, como exemplo de duas regras das mais

conhecidas, pode-se citar a SPT (*Shortest Processing Time*), na qual a tarefa que tem prioridade é sempre aquela com menor tempo de processamento para a estação de trabalho em questão, a LPT (*Longest Processing Time*), se assemelha a SPT no entanto no caso da LPT a tarefa priorizada é sempre aquela com maior tempo de processamento, ou ainda a EDD (*Earliest job Due Date*), na qual a prioridade são as peças que tem prazos de entrega mais curto (TUBINO, 2017).

Como exemplo de sequenciamento da produção utilizando algoritmos desenvolvidos sob o viés da Pesquisa Operacional é possível, entre outros, citar o modelo de Manne (1960, apud MORALES, 2016) que utiliza uma abordagem de tempo contínuo, em que o modelo prescreve o sequenciamento das tarefas minimizando o *Makespan* sem qualquer restrição quanto à prazos de entrega. Diferente de Manne, o modelo proposto por Chen (2001) utiliza uma abordagem baseada em restrições (*Constraint-based*), no qual o problema é modelado por meio de restrições finitas, como por exemplo, a capacidade das máquinas.

2.2 Modelos matemáticos

2.2.1 O modelo de Manne

Alan S. Manne (1960) escreveu em seu artigo intitulado “*On the Job Shop scheduling*” um modelo matemático que resolve os problemas de *Job Shop Scheduling*, utilizando menos variáveis do que alguns modelos propostos até então, facilitando assim o processamento computacional para resolução desses problemas.

O modelo de Manne é formulado a partir da ótica do tempo contínuo e de regras de precedência entre as diversas operações para processamento de cada peça. A definição do problema utilizada por Manne é a de que um conjunto de n peças ($n=1,2,\dots,n$) são processadas em um conjunto de k máquinas ($k=1,2,\dots,k$), cada peça possui sua própria sequência de t operações ($t=1,2,\dots,t$), cada máquina realiza uma operação por vez, a operação sendo iniciada ela não pode ser interrompida e todas as peças e máquinas estão disponíveis no começo do processo de sequenciamento (MANNE, 1960, apud MORALES, 2016).

Os índices, parâmetros e variáveis utilizadas estão listadas logo abaixo e em seguida a formulação matemática de Manne por Liao & You (1992):

i, j : Jobs;

k : Máquinas;

l : Operações;

$p_{i,k}$: Tempo de processamento da peça i na máquina k ;

$r_{i,l,k}$: Indica se a operação l da peça i requer a máquina k ;

M : Número positivo suficientemente grande;

$x_{i,k}$: Instante de início de peça i na máquina k ;

$$y_{i,j,k} \begin{cases} 1 \text{ se a tarefa } J_i \text{ precede } J_j \text{ na máquina } M_k \\ 0 \text{ caso contrário} \end{cases}$$

C_{max} : Momento de término da última tarefa da última peça a ser processada

(*Makespan*);

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

s.a

$$\sum_{k=1}^m r_{i,l,k} (x_{i,k} + p_{i,k}) \leq \sum_{k=1}^m r_{i,l+1,k} x_{i,k} \quad i = 1, 2, \dots, n \quad l = 1, 2, \dots, n - 1 \quad (2)$$

$$(M + p_{j,k}) y_{i,j,k} + (x_{i,k} - x_{j,k}) \geq p_{j,k} \quad 1 \leq i < j \leq n \quad k = 1, 2, \dots, m \quad (3)$$

$$(M + p_{i,k})(1 - y_{i,j,k}) + (x_{j,k} - x_{i,k}) \geq p_{i,k} \quad 1 \leq i < j \leq n \quad k = 1, 2, \dots, m \quad (4)$$

$$(x_{i,k} + p_{i,k}) \leq C_{max} \quad i = 1, 2, \dots, n \quad (5)$$

$$x_{i,k} \geq 0 \quad i = 1, 2, \dots, n \quad k = 1, 2, \dots, m \quad (6)$$

$$y_{i,j,k} \in \{0, 1\} \quad i, j = 1, 2, \dots, n \quad k = 1, 2, \dots, m \quad (7)$$

A função (1) é a definição do objetivo, as inequações (2) definem que para toda a peça i , o instante de início de uma operação ($j+1$) em uma máquina k tem que ser maior ou igual ao instante de início da operação precedente mais o seu tempo de processamento. As inequações (3) e (4), são chamadas restrições disjuntivas, e, determinam que uma máquina k só inicie um processo após terminar de processar uma tarefa precedente, ou seja, garante que duas tarefas diferentes não ocupem o mesmo espaço de tempo em uma mesma máquina. A inequação (5) determina o momento de finalização da última operação de cada peça, o chamado *Makespan*. As inequações (6) definem a variável de instante de início de cada operação como positiva e as equações (7) definem as variáveis que indicam precedência como binária (MORALES, 2016).

2.2.2 O modelo de Chen (2001)

O modelo para resolução de JSSP proposto por Chen *et al* (2001) é um modelo baseado em restrições e considera que n peças serão processadas em h tipos de máquinas com um horizonte de tempo k . O objetivo é minimizar atrasos e adiantamentos durante os processos de fabricação, ou seja, na função objetivo encontram-se penalidades para adiantamentos, que visa minimizar os estoques em processos (*WIP – Work in process*) e penalidades para atrasos no prazo de entrega das peças. Cada tipo de máquina h ($h=1,2,\dots,h$) possui uma quantidade M_{hk} específica disponíveis no tempo k ($k=1,2,3\dots k$), o processamento de cada peça i ($i=1,2,\dots,n$) é composto por uma série de j operações em diferentes máquinas denotadas pelo índice $(i,1), (i,2),\dots,(i,j)$, ou seja, operação 1 da peça i , operação 2 da peça i e assim sucessivamente. Cada operação (i,j) deve ser efetuada em um tipo específico de máquina dado por m_{ij} . Para simplificar, todas as peças são consideradas disponíveis no momento inicial. A formulação matemática proposta por Chen *et al* (2001) pode ser facilmente modificada e incluída a restrição para momento de início de cada operação (i,j) . As variáveis e símbolos utilizados no modelo de Chen *et al* (2001) estão listadas na sequência e em seguida o modelo matemático:

B_i : Momento de início da peça i ;

$b_{i,j}$: Momento de início da operação (i,j) ;

$c_{i,j}$: Momento de finalização da operação (i,j) ;

C_i : Momento de finalização da peça i ;

D_i : Prazo de entrega da peça i ;

E_i : Adiantamento da peça i , definido como $\max(0, S_i - B_i)$;

$M_{h,k}$: Número disponível de máquinas do tipo h no momento k , $1 \leq h \leq H$,

$1 \leq k \leq K$;

$p_{i,j}$: Tempo de processamento da operação (i,j) ;

S_i : Momento desejado de disponibilidade de matéria prima para a peça i ;

T_i : Atraso da peça i , definido como $\max(0, C_i - D_i)$;

w_i : Peso para a penalidade atraso da peça i ;

β_i : Peso para a penalidade adiantamento para a peça i ;

$\delta_{i,j,h,k} \begin{cases} 1 & \text{Se a operação } (i,j) \text{ é executada na máquina } h \text{ no momento } k \\ 0 & \text{caso contrário} \end{cases}$

$$\sum_{i,j} \delta_{i,j,h,k} \leq M_{h,k}, \quad k = 1, 2, \dots, k, h = 1, 2, \dots, h, \quad (8)$$

$$c_{i,j-1} + 1 \leq b_{i,j}, \quad i = 1, 2, \dots, n, j = 1, 2, \dots, n_i, \quad (9)$$

$$c_{i,j} = b_{i,j} + p_{i,j} - 1, \quad i = 1, 2, \dots, n, j = 1, 2, \dots, n_i, \quad (10)$$

$$J = \sum_{i=1}^N w_i T_i + \sum_{i=1}^N \beta_i E_i \quad (11)$$

As inequações (8) representam restrições de capacidade das máquinas e elas nos dizem que a quantidade de tarefas direcionadas para a máquina h no momento k deve ser menor ou igual à quantidade de máquinas h disponíveis no momento k . As inequações (9) são restrições de precedência de operação, em que o momento de início de processamento de uma operação (i,j) deve ser maior ou igual ao momento de finalização do processo anterior $(i,j-1)$. As equações (10) calculam o momento de finalização das operações (i,j) , este é igual ao momento de início da operação (i,j) mais o tempo de processamento de (i,j) . A equação (11) descreve a variável objetivo J como o somatório das penalidades de atraso de todas as peças i mais o somatório de todas as penalidades de adiantamento da peça i , na qual é possível perceber que minimizando J a primeira parcela da equação (11) minimiza o *WIP (Work in process)* e a segunda parcela minimiza os atrasos (CHEN; JUH; FANG, 2001).

2.2.3 Modelo de dimensionamento de lotes

Os modelos de dimensionamento de lotes podem ser classificados de acordo com a quantidade de estágios de produção inerente aos produtos que se deseja dimensionar lotes (*Single-level, Multi-level problems*), os *Single-level* são produtos que tem apenas um estágio de processamento e *Multi-level* aqueles que passam por vários estágios de processamento. Podendo ser classificados ainda de acordo com a quantidade de recursos disponíveis para processar os produtos (*Single-resource, Multi-resource problems*), os *Single-resource* possuem apenas uma máquina para processar a tarefa e os *Multi-resource* possuem várias máquinas paralelas. Ou classificados ainda quanto às janelas de tempo que se deseja dimensionar os lotes (*Small bucket models, Big bucket models*). Nos modelos de

dimensionamento de lotes, existem períodos de tempo pré estabelecidos nos quais os modelos definem qual ou quais produtos serão alocados para processamento em cada período. No caso dos *Small bucket models* esse período de tempo é pequeno de tal forma que apenas um lote pode ser alocado a cada período de tempo, assim, o dimensionamento dos lotes e sequenciamento são feitos ao mesmo tempo. Já nos *Big Bucket models* este período de tempo pré estabelecido é maior, de tal forma que diferentes lotes são alocados a cada período (GICQUEL; MINOUX; DALLERY, 2008).

Os índices, parâmetros e variáveis utilizadas estão listados na sequência e em seguida o modelo matemático para um *Small bucket model*.

i : Produtos;

t : Períodos;

$D_{i,t}$: Demanda do produto i no período t ;

P_t : Capacidade de produção no período t ;

$u_{i,t}$: Capacidade necessária para produzir 1 unid. do produto i no período t ;

h_i : Custo de armazenagem por unid. de produto por período para produto i ;

$c_{i,t}$: Custo de *setup* do produto i no período t ;

$I_{i,t}$: Inventário do produto i no final do período t ;

$x_{i,t}$: Quantidade produzida do produto i no período t ;

$y_{i,t} \begin{cases} 1 & \text{se o recurso produz o produto } i \text{ no período } t \\ 0 & \text{caso contrário} \end{cases}$

$z_{i,t} \begin{cases} 1 & \text{se o produto } i \text{ produzido no período } t \text{ for diferente do produto } i \text{ produzido em } t - 1 \\ 0 & \text{caso contrário} \end{cases}$

Minimizar

$$\sum_{i=1}^N \sum_{t=1}^T (h_{it} I_{it} + c_{it} z_{it}) \quad (12)$$

$$I_{it} = I_{i,t-1} + x_{it} - D_{it} \quad \forall i, \forall t \quad (13)$$

$$u_{it} x_{it} \leq P_t y_{it} \quad \forall i, \forall t \quad (14)$$

$$\sum_{i=1}^N y_{it} \leq 1 \quad \forall t \quad (15)$$

$$z_{it} \geq y_{it} - y_{i,t-1} \quad \forall i, \forall t \quad (16)$$

$$I_{it} \geq 0 \quad \forall i, \forall t \quad (17)$$

$$x_{it} \geq 0 \quad \forall i, \forall t \quad (18)$$

$$y_{it} \in \{0,1\} \quad \forall i, \forall t \quad (19)$$

$$z_{it} \in \{0,1\} \quad \forall i, \forall t \quad (20)$$

A função objetivo (12) é minimizar os custos de estoques e de *setups*. As equações (13) definem o estoque do produto i a cada período t . As restrições (14) limitam a produção do produto i no período t de acordo com a capacidade de produção do período t , a variável binária de setup precisa ser ativada para garantir a capacidade. As restrições (15) garantem que apenas um produto possa ser alocado no período t . As restrições (16) registram troca de *setup*, quando houver troca de produto entre um período t e um período $t-1$. As funções 13 e 17 em conjunto garantem que a demanda de cada produto i no período t seja atendida. As restrições (18) garantem que a quantidade produzida do produto i em cada período t nunca seja negativa. As funções (19) e (20) caracterizam as variáveis binárias.

2.2.4 GAMS (*General Algebraic Modeling System*)

GAMS é um *software* utilizado para fazer a construção e resolução de modelos matemáticos amplos e complexos. O GAMS utiliza uma linguagem de alto nível, simples e intuitiva, que pode ser lida tanto por usuários comuns quanto por programadores, pois GAMS combina linguagem matemática com conceitos de programação (BROOKE, 1997).

De acordo com Brooke (1997) a estrutura básica de um modelo no GAMS é composta de entradas e saída. As entradas são divididas em índices, dados, variáveis, equações e declarações do modelo. O Quadro 1 mostra como é a estrutura básica de entrada para modelagem no GAMS, e apresenta um exemplo para que a linguagem utilizada pelo *software* possa ser visualizada mais facilmente. Vale ressaltar que existem várias formas de programar no GAMS. O exemplo é uma simplificação de uma programação de produção, na qual uma fábrica produz os produtos A, B e C, e, de acordo com o retorno financeiro por cada produto produzido, tempo de processamento em cada máquina e disponibilidade de máquinas, o algoritmo calcula qual é a quantidade dos produtos A, B e C que devem ser produzidos de forma a maximizar o lucro.

Quadro 1 – Estrutura de entrada de um modelo no GAMS

| | | |
|-----------|---|--|
| Índices | Neste primeiro bloco são declarados os conjuntos, ou índices das representações algébricas do modelo | <pre>Set P produtos /A,B,C/ M maquinas /Co,Do,Em/;</pre> |
| Dados | O segundo bloco do modelo é composto pela declaração e valoração dos parâmetros, tabelas e escalares. | <pre>Table Tp(p,m) Tempo de processamento (h) Co Do Em A 8 5 0.7 B 5 10 1 C 2 4 2 ; Parameter Disp(m) Qtd disp. de máq. / Co 3 Do 10 Em 2 / ; Parameter L(p) Lucro por produto (\$) / A 1 B 1.5 C 2 / ; Scalar H Horas trab. por semana /40/;</pre> |
| Variáveis | O terceiro bloco contém a declaração das variáveis do problema. Neste bloco também são determinados os tipos de variáveis, que podem ser livres, positivas, negativas, inteiras ou binárias. | <pre>Variables x(p) Qtd produzida do produto p TUM(m) Tempo utilizado na máq. m TDM(m) Tempo disponível na máq. m z objetivo; Positive variable x, TUM, TDM;</pre> |
| Equações | O quarto bloco é formado pelas equações, ou restrições do modelo. As equações nada mais são do que as relações matemáticas entre variáveis e dados do problema. As equações definem as variáveis ou definem seus limites. | <pre>Equations E1 Tempo disp. em cada tipo de máq. E2 Tempo utilizado em cada máquina E3 Limita o tempo utilizado por máq. E4 função objetivo; E1(m) .. TDM(m)=e=Disp(m)*H; E2(m) .. TUM(m)=e=sum(p,x(p)*Tp(p,m)); E3(m) .. TUM(m)=l=TDM(m); E4 .. z=e=sum(p,x(p)*L(p));</pre> |
| Modelo | Neste bloco os modelos são nomeados, é possível criar mais de um modelo dentro do mesmo programa. As equações que fazem parte do modelo são relacionadas. O tipo de problema, objetivo e Solver utilizado são declarados. Outras funções podem ser escritas, exemplo a função Display, na qual variáveis podem ser listadas para serem exibidas nas telas de saída. | <pre>Model Planej1 /all/; Solve Planej1 maximizing z using LP; Display x.l;</pre> |

Fonte: O autor, adaptado de Brooke (1997)

A segunda parte da estrutura básica do GAMS são os dados de saída. A saída do GAMS fornece muitas informações sobre o modelo, sobre a metodologia que o Solver utilizou para encontrar a solução e mais uma grande quantidade de dados.

Para o entendimento desse trabalho será abordado aqui somente os dados de saída mais importantes.

Dentre os dados da saída, o primeiro que podemos citar é a lista de erros. O programa faz uma varredura nos dados de entrada em busca de inconsistências. Cada tipo de erro tem um código e uma descrição. Os erros encontrados são listados nos dados de saída do programa caso sejam detectados, sempre referenciados à linha do programa onde o erro foi detectado. Outro dado importante de saída do programa é o Mapa de Referências, no qual todos os dados e variáveis do algoritmo são listados, com referência a que tipo de dado ou variável se trata, em que linha foi declarado, em que linha foi valorado e em quais linhas foram encontradas referências. Por último podemos citar o relatório de solução como um todo. Neste o GAMS fornece o valor de cada variável para a solução final encontrada, bem como seu limite inferior e superior e seu valor marginal (BROOKE, 1997).

2.3 Termos e conceitos utilizados ao longo do trabalho

2.3.1 Ambientes de sequenciamento

De acordo com Alharkan (2005) os ambientes para sequenciamento de tarefas podem ser classificados como *Single Machine Shop*, *Flow Shop*, *Job Shop* entre outros ambientes mais específicos. No ambiente *Single Machine Shop*, como o próprio nome sugere, apenas uma máquina processa n produtos. No ambiente *Flow Shop* m máquinas estão em série e os produtos são processados de forma unidirecional, ou seja, entram em uma máquina e seguem o fluxo de máquinas até saírem na outra ponta da linha de produção. E *Job Shop* é o ambiente no qual cada peça tem sua própria sequência de máquinas nas quais precisa passar para processar cada tarefa.

2.3.2 Conceitos abordados ao longo do trabalho

Alguns conceitos foram abordados ao longo do trabalho e serão explicados nesta seção para facilitar o entendimento:

Tempo Padrão ou Tempo de processamento: De acordo com Moreira (2009) é o tempo decorrido desde o início do processamento de uma tarefa até a sua finalização.

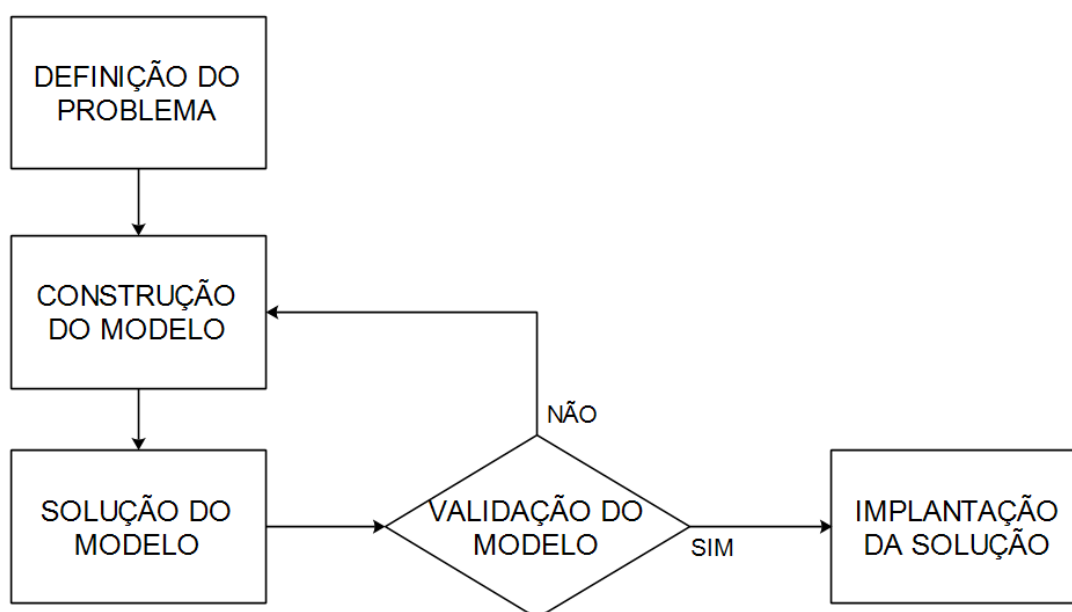
Work in Process (WIP): São os produtos que ficam em estoques intermediários entre o início e o fim de um roteiro de produção (HOPP, 2013).

ERP: Sistemas ERP (*Enterprise Resource Planning*) ou sistemas integrados de gestão empresarial, são sistemas de informação que integram os sistemas produtivos, contábeis, operacionais, administrativos e comerciais de uma organização. Todas as transações realizadas pela empresa precisam ser registradas e o ERP integra todas essas transações, possibilitando um fluxo de informações único e contínuo para toda a organização, sob um único banco de dados (PADILHA, 2005).

3 METODOLOGIA

Este trabalho enquadra-se como uma pesquisa acadêmica aplicada, com objetivo exploratório e abordagem quantitativa, onde será utilizado método de modelagem e simulação (CAUCHIK, 2012). Os dados utilizados como parâmetros para a modelagem são dados reais, quantitativos, coletados no banco de dados da empresa estudada. O método utilizado para desenvolver este trabalho segue a metodologia proposta por Cauchik (2012) representado na Figura 2. Sendo que, a partir da identificação e formulação de um problema, um modelo matemático é construído afim de encontrar a solução para o problema proposto, posteriormente esse modelo passa por um processo de validação, e, enfim implantação.

Figura 2 – Fluxograma da metodologia de modelagem



Fonte: O autor, adaptado de Cauchick (2012)

O foco da metodologia utilizada para desenvolver esse trabalho está em explorar modelos de sequenciamento de tarefas, e confrontar esses modelos com as reais restrições que existem no dia a dia do PCP com relação à tarefa de programação da produção, e assim, propor um modelo que busque atender com mais abrangência as necessidades do setor de PCP. Os tópicos seguintes buscam explicar como se dará cada etapa da pesquisa e desenvolvimento do trabalho.

3.1 Definição do problema

A primeira etapa para a construção desse trabalho consiste em utilizar pesquisas e entrevistas não estruturadas, aplicadas à pessoas de diferentes departamentos da empresa, que de forma direta ou indireta tenham relação com a tarefa de planejamento da produção. As informações coletadas nessas entrevistas servirão de base para a formulação do problema.

A definição do problema é a primeira etapa da pesquisa do tipo modelagem e simulação. Essa etapa é fundamental para que a solução encontrada seja uma boa aproximação da realidade. O pesquisador precisa identificar qual é o objetivo do problema, qual é o escopo que o modelo matemático irá representar, e perceber que, no sistema, existem parâmetros e variáveis que interagem entre si modificando o sistema e conseqüentemente a solução do problema. Cria-se então um entendimento de todo o processo e uma definição clara do problema. O pesquisador precisa compreender claramente o problema para que ao longo da pesquisa ele inclua no modelo matemático todas as variáveis e parâmetros que fornecerão uma solução técnica para o problema (CHIWIF; MEDINA, 2006 *apud* FERNANDES, 2013).

3.2 Construção do modelo

A construção do modelo será realizada no *software* GAMS (*General Algebraic Modeling System*) que, como já citado no referencial teórico, é um *software* utilizado para fazer a construção e resolução de modelos matemáticos.

Antes de iniciar de fato a construção do modelo será feita uma revisão da literatura a fim de identificar possíveis abordagens para o problema. Para desenvolver o modelo é necessário construir uma clara definição do problema, definindo quais são as restrições que são fundamentais para condizer com a situação real e qual será o objetivo do modelo.

Tendo-se construído uma visão clara do problema, passa-se à fase de desenvolvimento do modelo. Testes iniciais serão feitos com as possíveis abordagens identificadas, a fim de visualizar vantagens e desvantagens tanto para a modelagem quanto para a qualidade da saída de dados. Nessa etapa serão identificados os parâmetros e variáveis que influenciam de forma significativa nos dados de saída do modelo (CHIWIF; MEDINA, 2006 *apud* FERNANDES, 2013) e também as relações

matemáticas entre essas variáveis e parâmetros.

Enfim o modelo será construído. A construção normalmente ocorre partindo-se de modelos existentes utilizados na Pesquisa Operacional (CAUCHIK, 2012), alterando e incluindo restrições ou funções que aproximem ao máximo o modelo das necessidades práticas do PCP identificadas na primeira etapa do desenvolvimento.

3.3 Solução do modelo

Após fazer o modelamento do problema é necessário fazer uma verificação quanto à erros no modelo computacional (CHIWIF; MEDINA, 2006 *apud* FERNANDES, 2013). Conforme descrito no referencial teórico, o programa GAMS, nos seus dados de saída, possui uma listagem dos erros. Essa listagem facilita a identificação tanto da localização do erro no modelo, quanto da sua natureza, visto que cada tipo de erro possui um código e uma breve descrição. Com um duplo clique sobre o erro na lista de erros de saída, automaticamente o cursor vai para o modelo em si e se posiciona logo após o componente que apresenta a incoerência no modelo (BROOKE, 1997).

3.4 Validação do modelo

A validação do modelo será realizada em três etapas. A primeira etapa será realizada com um problema de pequenas dimensões, para que seja facilitada a análise dos dados de saída. Conforme relata Cauchick (2012) é comum realizar análises de sensibilidade e de diferentes cenários, o pesquisador levanta questões do tipo *What-if* o que facilita a avaliação do pesquisador se o modelo está respondendo de acordo com o esperado. Como exemplo para o caso de problemas de sequenciamento da produção pode ser perguntado: e se os prazos de entrega forem extremamente longos? E se os prazos de entrega fossem extremamente curtos? E assim avaliar se as saídas do modelo fornecem respostas que são adequadas com o que esses cenários propõem.

A segunda etapa será realizada também com problemas de pequenas dimensões. Nessa etapa o algoritmo proposto será adaptado para ser comparado com o resultado de algoritmos clássicos da Pesquisa Operacional, como o modelo de

Manne, já descrito neste trabalho. Nessa etapa será possível comparar os resultados com relação ao *Makespan* e também será possível comparar o tempo de processamento do modelo de Manne e do modelo proposto, com os mesmos dados de entrada.

A terceira etapa consiste em rodar um problema real de grandes dimensões e comparar o resultado do *Makespan* do modelo proposto com o *Makespan* ótimo fornecido pelo modelo de Manne. Essa comparação com o modelo de Manne é possível adaptando o modelo de Manne à algumas especificidades da empresa, e, no modelo proposto, não restringindo as datas de entrega, deixando-o livre para encontrar a menor data de entrega, minimizando os custos. Também será comparado o resultado do modelo proposto com relação ao resultado que a empresa consegue hoje, com seu próprio algoritmo de sequenciamento.

3.5 Implementação da solução

A implementação da solução dependerá de a empresa investir em *software* que resolva o modelo matemático desenvolvido. Caso a implantação se concretize novos desafios surgirão, como: Converter dados do sistema ERP (*Enterprise Resource Planning*) diretamente para o GAMS e vice-versa, extrair dados de processos em execução na produção e incluir no modelo, visto que a fábrica não pode ser considerada completamente disponível para se realizar um novo planejamento, mas sim, uma estrutura de máquinas e processos que está em constante mutação em relação à carga de trabalho.

De acordo com os resultados obtidos desse trabalho, a empresa terá condições de avaliar a viabilidade de aquisição do *Software* GAMS ou outro em que este modelo possa ser programado. Independente da opção adotada pela empresa, o modelo matemático ficará disponível para a mesma, será feito uma apresentação para os profissionais do TI da empresa, tanto no sentido de explicar como o modelo funciona, como também quais dados de entrada são fundamentais para alimentar o modelo. Dessa forma a empresa poderá ainda optar por utilizar outra linguagem de programação para escrever esse modelo.

4 DESENVOLVIMENTO, ANÁLISE E DISCUSSÃO DOS RESULTADOS

4.1 Estruturação inicial do problema

Inicialmente essa pesquisa teve como objetivo otimizar o sequenciamento da produção. Somente posteriormente foi identificadas algumas limitações dos modelos de sequenciamento e o objetivo da pesquisa passou a ser desenvolver um modelo flexível, que representasse mais fielmente as possibilidades de programação da produção.

Utilizando questionário não estruturado aplicado à equipe do TI (Tecnologia de Informação) e PCP da empresa estudada, foi identificado que o algoritmo desenvolvido internamente para sequenciamento da produção utiliza a regra de despacho LPT (*Longest Processing Time*). A justificativa para a utilização dessa regra é devido à esta privilegiar as peças com longos tempos de processamento, as quais geralmente são as peças maiores estruturais, que são as primeiras que precisam ser entregues ao setor de montagem para iniciar esta etapa. Percebe-se aqui uma oportunidade de utilizar o modelo de Manne (1960) como base e otimizar o sequenciamento minimizando o *Makespan*.

Ainda durante as entrevistas, dois requisitos da empresa foram identificados, os quais o modelo de Manne não contempla em sua construção original. Um deles é que, entre uma tarefa e outra, é necessário considerar um tempo para transporte das peças entre processos. Outro requisito está relacionado com o que a empresa chama de componentes suplementares. Estes são partes de peças que precisam ser pré-processadas antes de serem montadas na peça principal. Ou seja, a primeira tarefa da peça principal só pode iniciar após a finalização da última tarefa de todos os suplementares que fazem parte da estrutura da peça principal.

O modelo de Manne foi inicialmente escrito no GAMS utilizando como base a modelagem de Kalvelagen (2014). Após adaptar o modelo de Manne para deixá-lo equivalente ao modelo da empresa em termos de restrições, ou seja, incluído tempo de transporte e uma lógica para os suplementares, uma máquina foi escolhida para simulação do sequenciamento a fim de avaliar quanto o modelo de Manne poderia melhorar o *Makespan*. Considerando a fábrica totalmente disponível no início da simulação, foi sequenciado nos dois sistemas um picador de cavacos, que é um dos produtos da empresa, o qual pode ser visualizado na Figura 3. O produto escolhido é

composto por 115 peças as quais disputam 43 processos diferentes. A maioria das peças passa por cerca de 5 a 8 processos e a peça que possui mais processos passa por 15 processos. O *Makespan* atingido com o modelo de Manne foi de 220,7 horas e o sequenciamento determinado pelo modelo da empresa foi de 266 horas, ambos considerando 1 hora de tempo de transporte entre cada par de tarefas/processos adjacentes da mesma peça. Esse resultado motivou a continuação do estudo.

Figura 3 – Picador de cavacos PC130/100



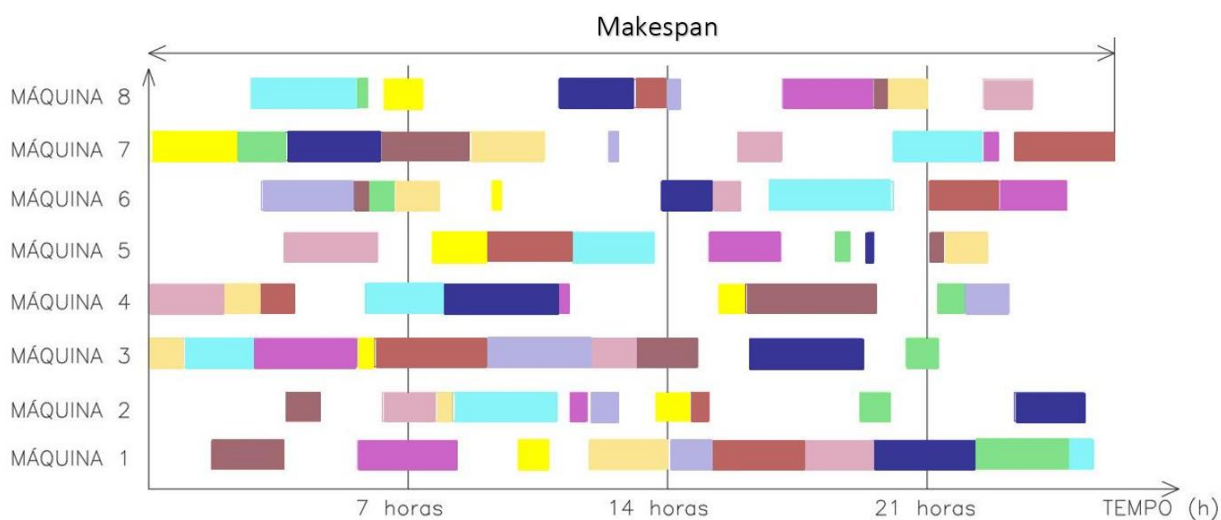
Fonte: Fezer (2021)

4.2 Requisitos do modelo

É fato que, se o sequenciamento determinado pelo modelo de Manne apresentasse um *Makespan* maior do que a data de entrega negociada com o cliente, automaticamente o trabalho em regime de horas extras seria uma alternativa. Porém, como as estações de trabalho possuem carregamentos diferentes, não há necessidade de trabalhar em regime de horas extras em todas estas, e a ativação de regimes de trabalho extra somente nas estações de trabalho com maior carregamento

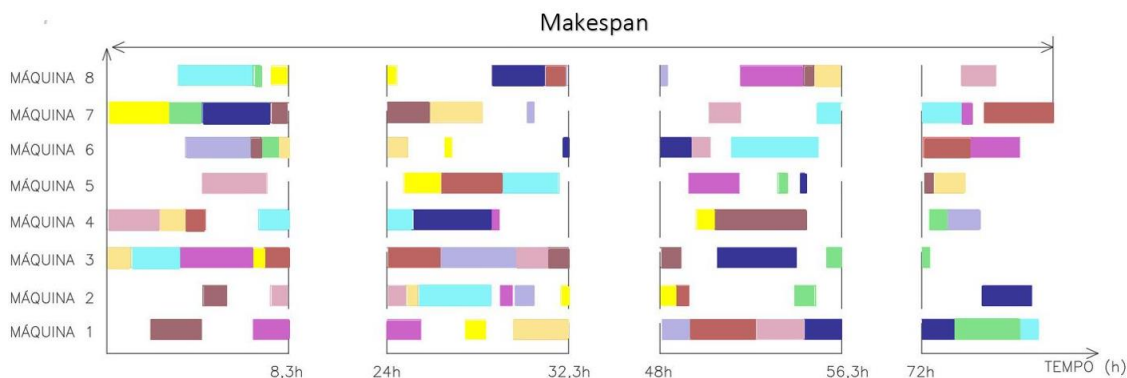
iria distorcer o sequenciamento definido pelo modelo de Manne. A Figura 4 é apenas um exemplo ilustrativo, a qual mostra um gráfico de Gantt que representa o sequenciamento de tarefas em uma janela de tempo contínuo. É possível visualizar no eixo horizontal o tempo, no eixo vertical as máquinas ou estações de trabalho que processam as peças que são representadas pelas diferentes cores. Já a Figura 5 representa melhor a realidade da empresa estudada. Em uma representação de 24h contínuas, a empresa que trabalha apenas em um turno, aloca tarefas, por exemplo, em 8,3h efetivas por dia, ficando as outras 15,7h disponíveis, à priori, sem alocação de tarefas.

Figura 4 – Gráfico de Gantt representando tempo contínuo



Fonte: O autor.

Figura 5 – Gráfico de Gantt representando a realidade da empresa estudada



Fonte: O autor.

O modelo de Manne considera o tempo de forma contínua, e uma das

características é que uma tarefa sendo iniciada ela não pode ser interrompida. Para representar a possibilidade de trabalhar em horas extras em algumas estações de trabalho e em outras não, seria necessário interromper o processamento de algumas tarefas no final de um dia normal de trabalho, enquanto estações de trabalho sobrecarregadas continuariam operando por mais um espaço de tempo (hora extra), diferente do modelo de Manne no qual as tarefas não podem ser interrompidas. Várias tentativas para adaptar o modelo de Manne às novas restrições foram feitas, porém sem sucesso. Então percebeu-se que para resolver esse problema era necessário seccionar o tempo em dias de trabalho, assumir que cada estação de trabalho possui uma capacidade ou disponibilidade de operação por dia, e que essa capacidade/disponibilidade pode ser ampliada operando em regime de hora extra após o final do expediente. Sendo assim, surgiu a necessidade de uma nova abordagem para o problema, utilizando como base uma formulação semelhante à dos modelos de dimensionamento de lotes CLSP (*Capacited Lotsizing Problems*), que divide o tempo em períodos pré-definidos para determinar o tamanho ótimo dos lotes.

Como a empresa estudada possui muitas tarefas de curta duração, e a janela de tempo que faz mais sentido para o problema seria de um dia de trabalho, percebeu-se que, ou o modelo teria que, além de dimensionar os lotes ou tarefas que seriam alocados a cada dia, precisaria também fazer o sequenciamento de tarefas dentro de um dia. Como essa função não foi incluída no modelo, uma restrição de que uma tarefa “t” sendo finalizada no dia “d”, a tarefa “t+1” só poderia ser programada a partir do dia “d+1”, para garantir que não haveria risco de sobreposição de tarefas da mesma peça no mesmo instante de tempo, em máquinas diferentes.

Sendo assim, as seguintes restrições precisavam ser incluídas no modelo:

a) Se a tarefa “t” é executada no dia “d”, o início da tarefa “t+1” pode ocorrer no dia “d+k” com $k \in N^*$;

b) Se uma peça complementar finaliza sua última tarefa em um dia “d”, a primeira tarefa da sua peça principal só pode iniciar no dia “d+k” com $k \in N$;

c) A data de entrega de um conjunto de peças é um parâmetro de entrada;

d) Considerar a possibilidade de programação em regime de horas extras;

e) Caso programação em horas extras não atenda o prazo de entrega é necessário algum indicador de necessidade de terceirizar ou ainda criação de mais turnos de trabalho.

4.3 O modelo proposto

O modelo proposto segue a seguinte lógica: Cada estação de trabalho possui uma capacidade diária em horas, onde tarefas podem ser alocadas. Essa capacidade pode ser ampliada, ativando uma variável binária, que representa trabalho sob regime de hora extra, tal atividade, possui um custo maior do que as atividades alocadas em horário normal.

Dada uma data de entrega para cada grupo de peças que compõe um produto da empresa, o modelo aloca as tarefas para processar essas peças nas estações de trabalho a cada dia, que é a janela de tempo pré-definida, respeitando a sequência de processos que cada peça precisa passar, e respeitando a capacidade diária de cada estação de trabalho. Conhecendo o custo hora máquina e se a máquina está operando em horário normal ou regime de hora extra, o modelo monitora o custo de cada operação, assim, o objetivo do modelo é fazer a alocação das tarefas respeitando os prazos de entrega e minimizando o custo operacional. O modelo proposto não faz o sequenciamento de tarefas a cada dia, mas sim, indica quais tarefas ou quanto de cada tarefa deve ser processado em cada estação de trabalho a cada dia.

O modelo proposto utilizou algumas lógicas dos modelos para CLSP, porém, modificou a estrutura desses modelos consideravelmente. Os modelos para resolver CLSP fazem o monitoramento dos estoques intermediários à cada período de tempo e, baseado nas demandas dos produtos à cada período, fazem o dimensionamento dos lotes de fabricação. No modelo proposto não há a necessidade de monitoramento de estoque, e a demanda não é dada por período, mas sim, representada por uma data de entrega para um produto específico. Como citado anteriormente, os *Small Buckets Models* fazem o sequenciamento das tarefas simultaneamente ao dimensionamento dos lotes, visto que a janela de tempo escolhida é tão pequena, que apenas um tipo de peça/tarefa é alocado em cada janela. Para o modelo proposto não foi adotada uma janela de tempo que coubesse só uma tarefa, visto que algumas tarefas são realmente muito curtas, mas sim uma janela de tempo de um dia, que pode ser ampliada com hora extra ou não. Sendo assim, o modelo proposto acabou utilizando algumas ideias dos modelos de dimensionamento de lotes, mas tornando-se um modelo único, não podendo ser classificado como CLSP.

Dado um conjunto de p peças ($p=1,2,\dots,P$) que são processadas em um conjunto de m máquinas ($m=1,2,\dots,M$), cada peça possui sua própria sequência t de tarefas ($t=1,2,\dots,T$) e cada máquina processa apenas uma tarefa por vez. As tarefas são alocadas por dia d ($d=1,2,\dots,D$) e cada par máquina-dia (m,d) possui uma capacidade pré-definida, que pode ou não ser ampliada representando assim trabalho em regime de hora extra, ou ainda mais turnos de trabalho, até um limite de 24h. O modelo considera que todas as máquinas estão disponíveis no começo da simulação.

Abaixo a lista de índices e variáveis e em seguida a formulação proposta.

p : Peças;

t : Tarefas;

m : Maquinas;

d : Dias;

$M1_p$: Subgrupo de peças p que compõe a máquina 1;

$M2_p$: Subgrupo de peças p que compõe a máquina 2;

$TP_{p,t}$: Tempo de processamento da peça p na tarefa t ;

$MAQ_{p,t}$: Número da máquina onde a tarefa t da peça p deve ser processada;

CHM_m : Custo Hora Máquina da máquina m ;

$MCOP_{m,d}$: Multiplicador do custo de operação da máquina m no dia d ;

CAP_m : Capacidade normal diária da máquina m ;

NOP_m : Número de operadores ou máquinas do mesmo tipo;

$SP_{p,p2}$: Indica se a peça p é suplementar da peça $p2$;

Fs : Multiplicador do custo para abrir segundo turno;

$Fhex$: Multiplicador do custo para operar em hora extra;

CMS : Custo de *Makespan*;

CS : Custo de *Setup*;

$DLM1$: Data limite de entrega das peças da máquina 1;

$DLM2$: Data limite de entrega das peças da máquina 2;

$x_{p,t,d}$: Indica o tempo processado da peça p na tarefa t no dia d ;

CT : Custo total;

$Cop_{m,d}$: Custo de operação da máquina m no dia d ;

ms : *Makespan*;

G : Número positivo suficientemente grande

$$y_{m,d} \begin{cases} 1 \text{ se ativa hora extra na máquina } m \text{ no dia } d \\ 0 \text{ caso contrário} \end{cases}$$

$$z_{m,d} \begin{cases} 1 \text{ se ativa segundo turno na máquina } m \text{ no dia } d \\ 0 \text{ caso contrário} \end{cases}$$

$$s_{p,t,d} \begin{cases} 1 \text{ se ativa setup da peça } p \text{ na tarefa } t \text{ no dia } d \\ 0 \text{ caso contrário} \end{cases}$$

$$f_{p,t,d} \begin{cases} 1 \text{ se finaliza a tarefa } t \text{ da peça } p \text{ no dia } d \\ 0 \text{ caso contrário} \end{cases}$$

ord_d : Ordinário que representa o dia que está sendo considerado.

Minimizar

$$CT = \sum_{m=1}^M \sum_{d=1}^D Cop_{m,d} + (ms * CMS) + (\sum_{p=1}^P \sum_{t=1}^T \sum_{d=1}^D s_{p,t,d} * CS) \quad (12)$$

s.a.

$$\sum_{d=1}^D x_{p,t,d} = TP_{p,t} \quad \forall p = 1,2, \dots, P, t = 1,2, \dots, T \quad (13)$$

$$\sum_{p=1}^P \sum_{t=1}^T x_{p,t,d} \leq NOP_m * CAP_m + (NOP_m * y_{m,d} * 2,34) + (NOP_m * z_{m,d} * 13,36) \quad m = 1,2, \dots, M, d = 1,2, \dots, D \quad (14)$$

$$Cop_{m,d} = (MCOP_{m,d} * CHM_m * NOP_m)(y_{m,d} * Fhex * 2,34 + z_{m,d} * Fs * 13,36) + (\sum_{p=1}^P \sum_{t=1}^T x_{p,t,d} * CHM_m * MCOP_{m,d}) \quad \forall m = 1,2, \dots, M, d = 1,2, \dots, D \quad (15)$$

$$\sum_{d2=1}^D x_{p,t,d2} \leq (1 - f_{p,t,d}) * TP_{p,t} \quad \forall p = 1,2, \dots, P, t = 1,2, \dots, T, d = 1,2, \dots, D, d2 > d \quad (16)$$

$$s_{p,t,d} \leq G * \sum_{d2=1}^D f_{p,t-1,d2} \quad \forall p = 1,2, \dots, P, t = 1,2, \dots, T, d = 1,2, \dots, D, \text{ se } d2 < d \quad (17)$$

$$ms \geq ord_d * f_{p,t,d} \quad \forall p = 1,2, \dots, P, t = 1,2, \dots, T, d = 1,2, \dots, D \quad (18)$$

$$x_{p,t,d} \leq s_{p,t,d} * G \quad \forall p = 1,2, \dots, P, t = 1,2, \dots, T, d = 1,2, \dots, D \quad (19)$$

$$\sum_{d=1}^D f_{p,t,d} = 1 \quad \forall p = 1,2, \dots, P, t = 1,2, \dots, T \quad (20)$$

$$ord_d * f_{p,t,d} \leq DLM1 \quad \forall M1_p, t = 1,2, \dots, T, d = 1,2, \dots, D \quad (21)$$

$$ord_d * f_{p,t,d} \leq DLM2 \quad \forall M2_p, t = 1,2, \dots, T, d = 1,2, \dots, D \quad (22)$$

$$(f_{p,T,d} - 1)G \leq -\sum_{d2}^D s_{p,t,d} \quad \forall p, p2 = 1,2, \dots, P, t = 1,2, \dots, T, d1,2, \dots, D, d2 < d, \text{ e se } SP_{p,p2} = 1 \quad (23)$$

$$f_{p,t,d} \leq s_{p,t,d} \quad \forall p = 1,2, \dots, P, t = 1,2, \dots, T, d = 1,2, \dots, D \quad (24)$$

$$G(s_{p,t,d} + f_{p,t,d-1}) \geq s_{p,t,d-1} \quad \forall p = 1,2, \dots, P, t = 1,2, \dots, T, d = 1,2, \dots, D \quad (25)$$

$$x_{p,t,d} \geq 0 \quad \forall p = 1,2, \dots, P, t = 1,2, \dots, T, d = 1,2, \dots, D \quad (26)$$

$$y_{m,d}, z_{m,d}, s_{p,t,d} \text{ e } f_{p,t,d} \in (0,1) \quad \forall p = 1,2, \dots, P, t = 1,2, \dots, T, d = 1,2, \dots, D \quad (27)$$

A função objetivo a ser minimizada (12) é composta de três parcelas, sendo a primeira referente ao custo operacional, a segunda parcela tem função de minimizar o *Makespan* e a terceira minimizar a quantidade de *setups*. As inequações (13) garantem o tempo de processamento de cada tarefa de cada peça seja representado pela variável “x”. As inequações (14) garantem que a quantidade de horas alocada em cada máquina em cada dia não seja maior que a capacidade em horas. Essa capacidade pode ser aumentada ativando os binários $y_{m,d}$ e $z_{m,d}$. As equações (15) definem o custo operacional de cada máquina a cada dia, o parâmetro multiplicador $MCOP_{m,d}$ foi definido com valor 1 de segunda à sexta, 2 para sábados e 3 para domingos como mostra a Figura 6. Este parâmetro força o modelo a programar tarefas preferencialmente de segunda à sexta-feira, caso necessário utiliza os sábados, caso ainda haja necessidade utiliza os domingos.

Figura 6 – Detalhe do modelo

Table MCOP (m, d) Multiplicador do custo de operação da máquina por dia

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 | 21 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|----|----|----|----|----|----|
| 1 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | |
| 2 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | |
| 3 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | |
| 4 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | |
| 5 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | |
| 6 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | |
| 7 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | |
| 8 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | |
| 9 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | |
| 10 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 2 | 3 | 1 | 1 | 1 | 1 | 1 | 2 | 3 | |

Fonte: O autor.

As inequações (16) garantem que não haverá processamento da peça p na tarefa t após ser finalizada pelo binário f . As inequações (17) garantem a sequência de tarefas, ou seja, só libera setup da peça p na tarefa t se já houve conclusão na tarefa $t-1$ nos dias anteriores. A inequação (18) define o limite inferior para o *Makespan*, porém como ela está na função objetivo, este por fim assume o valor do dia da finalização da última tarefa de todas as tarefas sequenciadas. As inequações (19) forçam a ativação do binário $s_{p,t,d}$ (*setup*) sempre que alguma tarefa for agendada. As equações (20) dizem que toda a tarefa precisa ser finalizada. As inequações (21) e (22) definem o dia limite para finalizar as peças das máquinas 1 e 2. Podem ser criados quantos subgrupos de peças forem necessários, no exemplo deste modelo apenas 2 subgrupos foram sequenciados. As inequações (23) garantem

que a primeira tarefa da peça principal só seja iniciada após finalizada a última tarefa de seus suplementares. Neste exemplo a tarefa “T” representa a última tarefa e “t” qualquer tarefa. As inequações (24), em conjunto com outras equações do modelo, garantem que a ativação do binário $f_{p,t,d}$ ocorra no mesmo dia do último *setup*. As inequações (25) fazem com que o *setup* fique ativado enquanto a peça não é finalizada. As funções (26) e (27) definem o domínio das variáveis.

O modelo proposto escrito no GAMS, com o problema LA19 utilizado para validação do modelo pode ser visto no Apêndice 2. Outras inequações foram incluídas no modelo pois mostraram-se efetivas para melhorar o desempenho computacional.

4.4 Aplicações e funções do modelo

O modelo proposto apresenta como saída quais peças e quantas horas de cada tarefa deve ser processada por máquina a cada dia. O modelo não faz o sequenciamento de tarefas no dia quando mais de uma tarefa é programada para o mesmo dia. Neste caso, a decisão quanto a sequência do dia deve ser tomada pelo operador da máquina ou outro responsável. O modelo tem liberdade para programar tarefas para trabalho em horário normal, sob regime de hora extra, dias de semana e finais de semana e também ativar segundo turno de trabalho se necessário.

O modelo serve para fazer a programação da produção, no entanto, pode ainda ser utilizado para simular a entrada de novas máquinas no sistema considerando diferentes cenários quanto à prazos de entrega, proporcionando mais segurança na fase de negociação com o cliente.

A empresa pode usar o modelo para simular paradas para manutenção programada e períodos de férias, simplesmente alterando os valores do parâmetro $MCOP_{m,d}$ (Multiplicador do custo operacional) para cada máquina individualmente. Valores altos de $MCOP_{m,d}$ fazem o modelo evitar programar tarefas nesses dias, ou seja, somente serão programadas tarefas nesses dias, caso os outros dias restante não sejam suficientes para atender os prazos de entrega.

Uma das características do modelo é que no início da simulação todas as máquinas são consideradas disponíveis, isso é conflitante com a realidade. Nesse caso, sempre que entrar uma nova máquina no sistema, é necessário extrair relatórios das tarefas que faltam processar das peças correntes no sistema, incluir a nova máquina ou máquinas e rodar o modelo novamente.

4.5 Validação do modelo

Após correção dos erros do modelo, iniciou-se a etapa de validação. Para as avaliações iniciais foram utilizados pequenos problemas criados pelo autor.

A técnica utilizada foi a técnica *What-if* já descrita anteriormente. As principais perguntas utilizadas para avaliar o comportamento do modelo foram referentes aos prazos de entrega dos diferentes grupos de peças. Após muitas revisões do modelo este ficou respondendo adequadamente. Com prazos de entrega folgados, o modelo não programava tarefas finais de semana, nem trabalhos em horas extras ou segundo turno. Conforme reduziam-se os prazos de entrega o modelo passava a utilizar como primeiro recurso as horas extras, em seguida os sábados, depois domingos e por fim segundo turno.

A segunda etapa da validação ocorreu com problemas clássicos como FT10, LA2, LA19 e LA21. Esses problemas, são problemas fictícios descritos na literatura. Pesquisadores escrevem modelos matemáticos para resolver problemas de sequenciamento de tarefas e frequentemente utilizam esses problemas clássicos para comparar seus resultados, tanto quanto à obtenção de resultado ótimo, quanto para comparar tempo de processamento computacional. Por exemplo, os problemas LA2, LA19 e LA21 podem ser encontrados em Lawrence (1984), o problema FT10, pode ser encontrado em Fisher & Thompson (1963). De acordo com Vaessens *at al* (1996) os problemas LA2 e LA19 são problemas relativamente mais simples e o problema LA21 o autor descreve como problema em que o processamento computacional é mais desafiador. Os dados para os problemas foram encontrados em Hithub (2020).

Esta etapa consistiu em montar os problemas no GAMS, utilizando modelo de Manne adaptado mostrado no Apêndice 1 deste trabalho. Ao utilizar o parâmetro Tempo de transporte (TT) como “zero” (0), os resultados ótimos quanto ao *Makespan* foram encontrados e comparados com os resultados de Vaessens *at al* (1996), somente para garantir que não ocorreu nenhum erro de digitação nos dados do problema e que o modelo escrito no GAMS funciona adequadamente. Após confirmar resultado ótimo, os tempos de processamento foram escalonados 1:10, para que estes fossem compatíveis em horas com os tempos normalmente encontrados nos processos da empresa estudada.

Sabendo que o modelo proposto possui uma restrição de que a tarefa “t+1” só pode ser iniciada a partir do dia seguinte à finalização da tarefa “t” de uma mesma

peça, o seguinte raciocínio foi seguido de forma a fazer uma comparação mas justa entre o modelo de Manne e o modelo proposto: Imaginou-se um gráfico de Gantt gerado a partir de um sequenciamento utilizando o modelo de Manne com tempo de transporte entre tarefas adjacentes igual a zero, se este gráfico for seccionado de 8,3 em 8,3 horas (1 dia efetivo de trabalho em horário normal), será possível perceber que algumas tarefas finalizarão imediatamente antes da secção, se pensarmos na restrição do modelo proposto de que a próxima tarefa só poderia iniciar a partir da primeira hora do dia posterior, ela estaria à 0 hora de distância desse ponto. Por outro lado, se uma tarefa “t” finalizar logo após uma secção de 8,3 horas, e pensarmos que a tarefa “t+1” tem restrição para iniciar somente a partir do dia seguinte, o fim dessa tarefa “t” estaria no mínimo à 8,3h de distância do início de “t+1”. Sendo assim, pensou-se em utilizar o parâmetro tempo de transporte (TT) com o valor de 4,15h, que seria o tempo médio considerando trabalho somente em horário normal.

Utilizando tempo de transporte de 4,15h no modelo de Manne com o problema escalonado 1:10, o resultado de *Makespan* fornecido pelo modelo de Manne foi utilizado como parâmetro de entrada para o modelo proposto, o qual representa a quantidade de dias úteis que o projeto tem para ser entregue. Assim, espera-se que o modelo proposto não programe tarefas nos finais de semana, nem horas extras ou segundo turno, ou, se programar, que sejam poucas ocorrências. A Tabela 1 mostra os resultados para os diversos problemas testados, todos utilizando a mesma lógica previamente citada. Onde o modelo de Manne é o modelo padrão, o modelo Manne’ é o modelo de Manne escalonado 1:10 e adicionado tempo de transporte de 4,15h entre tarefas. A coluna MS indica o *Makespan* encontrado ou considerado como dado de entrada no caso do modelo proposto. A coluna Y indica a ativação de hora extra e entre parêntesis a quantidade total de horas extras programadas, a coluna SD indica se algum dia houve trabalho sábado ou domingo e entre parêntesis a quantidade total de horas programadas, a coluna Z indica se houve ativação de segundo turno e a coluna % indica o percentual de horas extras programadas em função do somatório dos tempos de processamento.

Tabela 1 – Resultados dos testes

| Problema | Modelo | MS | Y | SD | Z | % |
|----------|----------|--|-----------|----------|---|------|
| FT10 | Manne | 930 | - | - | - | |
| FT10 | Manne' | 123,7h (14,9 dias trabalho normal) | - | - | - | |
| FT10 | Proposto | 15 dias trabalho normal | 4 (7,14h) | 1 (0,2h) | 0 | 1,4 |
| LA2 | Manne | 655 | - | - | - | |
| LA2 | Manne' | 76,8h (9,3 dias trabalho normal) | - | - | - | |
| LA2 | Proposto | 10 dias trabalho normal | 0 | 0 | 0 | 0 |
| LA19 | Manne | 842 | - | - | - | |
| LA19 | Manne' | 113,05h (13,6 dias trabalho normal) | - | - | - | |
| LA19 | Proposto | 14 dias trabalho normal | 1 (0,1h) | 0 | 0 | 0,02 |
| LA21 | Manne | 1048 | - | - | - | |
| LA21 | Manne' | 129,3h (15,6 dias trabalho normal) | - | - | - | |
| LA21 | Proposto | 16 dias trabalho normal | 2(3,9h) | 0 | 0 | 0,5 |

Fonte: O autor.

O resultado do modelo de Manne foi exposto apenas para confirmar que a programação feita no GAMS está funcionando adequadamente. É possível perceber, em função do baixo percentual de horas programadas em regime especial, que o modelo proposto se comporta adequadamente e consegue bons planos de produção no que diz respeito ao *Makespan*. Foi observado durante os testes que, para conseguir bons resultados é necessário rodar o modelo até o resultado ótimo sem nenhuma tolerância. Como o objetivo do modelo é minimizar o custo operacional, sabendo que o custo geralmente é alto, qualquer tolerância em relação ao resultado ótimo faz o modelo programar horas extras.

4.6 Implementação do modelo em um caso real

O mesmo produto da empresa utilizado no início desse projeto, foi utilizado nessa etapa final para avaliar o comportamento do modelo com um problema real de dimensões consideráveis. O picador de cavacos estudado (PC130/100), possui 115

peças, das quais 68% passam por 3 a 6 tarefas, 26% passam por 7 a 11 tarefas e 6% por 12 a 15 tarefas. Os tempos de processamento de cada tarefa variam de 0,01h à 46h, exceção a uma peça que possui um processo com duração de 104h. O tempo de processamento médio, desconsiderando o ponto fora da curva, é de 5,2h. 82% das tarefas possuem tempo de processamento menor do que 8,3h que seria um dia de trabalho normal. Para a simulação foram mantidas as relações de desenhos suplementares de acordo com dados originais do projeto.

Tabela 2 – Resultados dos testes para o problema PC70/70

| Problema | Modelo | MS | Y | SD | Z | % |
|------------------|---------------|-----------------------------|---------------|--------------|----------|----------|
| PC130/100 | Manne | 256,44h | - | - | - | |
| | | 300,5h | - | - | - | |
| PC130/100 | Manne' | (36,2 dias trabalho normal) | | | | |
| PC130/100 | Proposto | 37 dias trabalho normal | 21 (43,8h) | 1 (5,78h) | 0 | 3,87 |

Fonte: O autor.

Para o problema PC130/100 ocorreram 21 ativações de horas extras que totalizam 43,8h, mais uma tarefa programada em um sábado com 5,78h, totalizando 49,6h de um total de 1281 horas do projeto. Ou seja, apenas 3,87% das horas foram programadas em regime especial. Para esse comparativo em relação ao modelo de Manne foi utilizado os dados do PC130/100 e o número de estações de trabalho que a empresa possuía a um ano e meio atrás, quando essa pesquisa iniciou.

O problema PC130/100 foi também comparado com relação ao método de sequenciamento utilizado atualmente na empresa, os resultados podem ser visualizados na Tabela 3. Para essa comparativo os Tempos Padrão das tarefas foram atualizados de acordo com novos tempos definidos pela empresa e também atualizações de projeto, sendo assim, a quantidade total de horas programadas mudou em relação ao projeto testado anteriormente, totalizando nesse caso 1499h programadas.

Tabela 3 – Resultados dos testes para o problema PC130/100

| Problema | Modelo | MS | Y | SD | Z | % |
|-----------|----------|---------|---------|----|---|-----|
| PC130/100 | Fezer | 36 dias | - | - | - | |
| PC130/100 | Proposto | 33 dias | 9 (21h) | - | - | 1,4 |

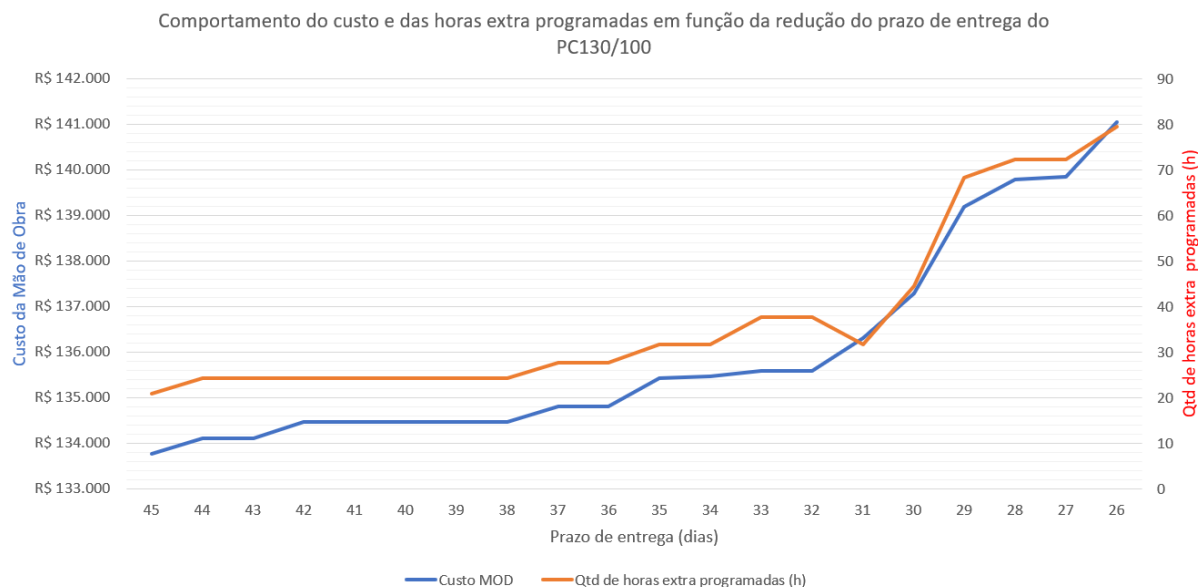
Fonte: O autor.

Para o algoritmo utilizado pela empresa adotou-se 4h de transporte entre tarefas, pois o sistema não permite número fracionado, e o resultado do sequenciamento foi de 36 dias. Utilizando esse parâmetro no modelo proposto como data limite para a entrega, este programou tarefas utilizando 33 dias, programando 9 atividades em horas extras totalizando 21h, nenhum trabalho nos finais de semana e nenhum segundo turno. As atividades em hora extra, representam nesse caso 1,4% do total de horas programadas. Os dados utilizados para fazer o comparativo foram dados reais, porém o algoritmo de sequenciamento da empresa foi modificado durante o último ano e ainda está passando por ajustes, assim, vale ressaltar que os dados de saída do sequenciamento fornecido pela empresa ainda não são precisos.

4.7 Análise de sensibilidade

Algo que pode ser muito interessante para a empresa, tanto no momento da negociação com o cliente, quanto posteriormente ao fazer a programação da produção, é se é interessante antecipar a produção, seja para entrada de novos pedidos, seja por questões de segurança. Sendo assim, com o caso do PC130/100 foi feita uma análise de sensibilidade que pode ser verificada na Figura 7. O tempo representado no eixo horizontal está em dias corridos. Pode ser verificado que a entrega da máquina, para ser feita no 45º dia corrido, programará um total de 21 horas extras, que neste caso representa 1,4% do total de horas, sob um custo operacional total de R\$133.780,00. Ao passo que vai se restringindo cada vez mais o prazo de entrega, observa-se o comportamento do custo operacional total e também da quantidade de horas extras necessárias.

Figura 7 – Gráfico de análise de sensibilidade PC130/100



Fonte: O autor.

Vale ressaltar que o modelo escolhe se programa horas extras nos finais dos expedientes, nos sábados ou domingos. Então percebe-se claramente com a entrega no 31º dia corrido, que houve uma redução na quantidade de horas extras, porém, um aumento no custo. Isso se dá pois o modelo passou a utilizar muitas horas aos sábados ou domingos, que foi considerado com um custo mais alto.

A restrição quanto ao prazo de entrega na análise de sensibilidade foi feita decrementada de um em um dia até que o modelo fizesse a ativação do segundo turno. Ponto esse que representa o menor prazo possível, considerando o investimento em horas extras com a equipe corrente na empresa, visto que a abertura de segundo turno representaria um aumento da capacidade de produção, ou necessidade de terceirização.

Outro ponto interessante observado durante a análise de sensibilidade é que, percebe-se claramente que a ativação de horas extras se dá sempre nas mesmas máquinas. Isso é um indicativo que essas máquinas encontram-se no caminho crítico desse projeto.

5 CONCLUSÃO

O sequenciamento de tarefas é uma atividade que está presente em diversos segmentos, não só da indústria ou serviços, mas também da nossa vida cotidiana. Cada ambiente onde o sequenciamento de tarefas ocorre possui suas próprias especificidades, suas próprias restrições e características. Logo no início dessa pesquisa percebeu-se que, para atender as necessidades da indústria estudada, era necessário incluir restrições e formulações que representassem de forma mais fiel a realidade, especialmente no que diz respeito a segmentação do tempo, representada pelos turnos de trabalho. Optou-se então por alterar o modelo matemático base, e passou-se a modelar o problema utilizando os modelos de dimensionamento de lotes utilizando janelas de tempo do tamanho de um turno de trabalho. Este modelo trouxe flexibilidade para implementar a questão de programação de horas em regimes extraordinários separadamente para cada estação de trabalho de acordo com a necessidade.

O GAMS mostrou-se uma ferramenta de linguagem matemática simples para a construção do modelo, permitindo que a programação não dependesse de um especialista e pudesse ser executada pelo próprio autor. Para a sequência desse trabalho dentro da empresa estudada é necessário ainda criar uma interface entre o sistema ERP (*Enterprise Resource Planning*) da empresa e o GAMS, tanto para entrada de dados quanto para tornar mais amigável o arquivo de saída do GAMS.

Efetuar o sequenciamento das tarefas monitorando o custo operacional dessa programação é algo de grande valia para a indústria e serviços, visto que a grande maioria das empresas buscam redução de custo para aumentarem sua competitividade. Como pode ser verificado por meio das simulações, tanto com os problemas clássicos como com o problema real, o modelo proposto consegue boas programações da produção dentro das restrições impostas, programando um percentual muito pequeno de atividades em horas extras. Além de bons resultados de programação o modelo proposto ainda possui a possibilidade de simular e programar antecipando os prazos de entrega, consegue simular diferentes projetos com diferentes prazos de entrega simultaneamente, proporciona a possibilidade de simulação de diferentes cenários para a produção e permite a simulação considerando paradas para manutenção, férias, ou qualquer outra parada programada. Através da análise de sensibilidade é possível identificar as máquinas que são gargalo para cada

projeto, facilitando a tomada de decisão quanto à quais peças são mais interessantes para terceirização ou ainda, facilitando análise de aumento de capacidade ou de execução de parte das tarefas em outras máquinas compatíveis.

Várias limitações foram identificadas ao longo da construção do modelo. Uma delas está relacionada ao modelo não fazer sequenciamento de tarefas dentro de um dia. Sendo assim, pode-se citar como sugestão de pesquisas futuras, estudar formas de fazer esse sequenciamento dentro do dia para conseguir melhores programações. Outra limitação do modelo está relacionada ao tempo de processamento computacional. Percebe-se que é necessário estender a pesquisa a fim de descobrir mecanismos que tornem o modelo mais eficiente quanto ao tempo de processamento. Em um computador pessoal, um notebook Intel i3 CPU 2GHz e memória RAM de 4GB o problema LA2 com o modelo proposto, achou resultado ótimo em 11s. Já o problema LA21 demorou 49h e 40m para achar o resultado ótimo. O problema FT10 demorou 3h e 48m. O problema PC130/100 não consegue encontrar o resultado ótimo em um neste mesmo notebook, já em um servidor na internet (NEOS – www.neos-server.org), 197 segundos foi o tempo necessário para resolver o problema PC 130/100 com resultado ótimo.

O teste com o picador PC130/100 apresentou resultado satisfatório. Esta máquina é uma das menores máquinas produzidas pela empresa estudada, e a condição de apenas uma máquina estar em produção é inexistente. Normalmente há muitas máquinas/produtos sendo processadas simultaneamente, disputando os mesmos recursos e com diferentes prazos de entrega. Sendo assim, faz-se necessário rodar o modelo proposto com uma condição de mais projetos simultâneos na produção, a fim de verificar o tempo de processamento computacional, e também investir em estudos para melhorar os tempos de processamento computacional. No entanto, mesmo com as limitações de tempo de processamento computacional, o modelo proposto já pode ser utilizado para projetos menores ou ainda planejamento de curtos horizontes de tempo.

A empresa estudada, localizada na cidade de Caçador/SC, é uma empresa pequena, que compete com grandes fabricantes mundiais. O sistema ERP da empresa foi desenvolvido internamente ao longo dos anos, é necessário verificar agora a compatibilidade do GAMS com o sistema da empresa e como montar a interface entre os dois sistemas para que o modelo desenvolvido possa ser utilizado plenamente pelo departamento de PCP.

REFERÊNCIAS

- ABDOLRAZZAGH-NEZHAD, Majid; ABDULLAH, Salwani. **Job shop scheduling: Classification, constraints and objective functions**. Int. J. Comput. Inf. Eng, v. 11, 2017.
- ALHARKAN, Ibrahim M. **Algorithms for sequencing and scheduling**. Industrial Engineering Department, King Saud University, Riyadh, Saudi Arabia, 2005.
- BAPTISTE, Philippe; LE PAPE, Claude; NUIJTEN, Wim. **Incorporating Efficient Operations Research Algorithms in Constraint-Based Scheduling**. In: Proceedings of the First International Joint Workshop on Artificial Intelligence and Operations Research, Timberline Lodge, Oregon. 1995.
- BŁAŻEWICZ, Jacek; DOMSCHKE, Wolfgang; PESCH, Erwin. **The job shop scheduling problem: Conventional and new solution techniques**. European journal of operational research, v. 93, n. 1, p. 1-33, 1996.
- BREMER, Carlos Frederico; LENZA, Rogério de Paula. **Um modelo de referência para gestão da produção em sistemas de produção assembly to order: ato e suas múltiplas aplicações**. Gestão & Produção, v. 7, n. 3, p. 269-282, 2000.
- BROOKE, Anthony; KENDRICK, David A.; MEERAUS, Alexander. **GAMS: Sistema geral de modelagem algébrica**. Edgard Blucher, 1997.
- CAUCHIK, P. A. M.; SOUSA, R. **Metodologia de pesquisa em engenharia de produção e gestão de operações**. 2ª edição. Editora Campus, 2012.
- CHEN, Haoxun; LUH, Peter B.; FANG, Lei. **A time window based approach for job shop scheduling**. In: Proceedings 2001 ICRA. IEEE International Conference on Robotics and Automation (Cat. No. 01CH37164). IEEE, 2001.
- FERNANDES, Laerte José et al. **Planejamento e controle da produção de cilindros para laminação: um estudo de caso quantitativo**. Production, v. 23, n. 1, p. 120-134, 2013.
- Fezer. **Produtos**. Disponível em: <<https://www.fezer.com.br/picador-a-tambor>> Acesso em: 17 jan. 2021.
- FISHER, H.; THOMPSON, G.L. **Probabilistic learning combinations of local job-shop scheduling rules**. Industrial scheduling, p. 225-251, 1963.
- GICQUEL, Céline; MINOUX, Michel; DALLERY, Yves. **Capacitated Lot Sizing models: a literature review**. 2008.
- Github. **JSPLIB/Instances**. Disponível em: <<https://github.com/tamy0612/JSPLIB/>>

tree/master/instances> Acesso em: 10 nov. 2020.

HOPP, Wallace J.; SPEARMAN, Mark L.; MIGLIAVACCA, Paulo Norberto. **A ciência da fábrica**. Bookman, 2013.

JAIN, Anant Singh; MEERAN, Sheik. **A state-of-the-art review of job-shop scheduling techniques. Technical report**, Department of Applied Physics, Electronic and Mechanical Engineering, University of Dundee, Dundee, Scotland, 1998.

KALVELAGEN, Erwin. **Yet Another Math Programming Consultant**. Disponível em: < <http://yetanothermathprogrammingconsultant.blogspot.com/2014/04/playing-with-ft10-job-shop-1.html>> Acesso em: maio. 2019.

LAWRENCE, S. **Resouce constrained project scheduling**: An experimental investigation of heuristic scheduling techniques (Supplement). Graduate School of Industrial Administration, Carnegie-Mellon University, 1984.

LIAO, Ching-Jong; YOU, Chii-Tsuen. **An improved formulation for the job-shop scheduling problem**. Journal of the Operational Research Society, v. 43, n. 11, p. 1047-1054, 1992.

MANNE, Alan S. **On the job-shop scheduling problem**. Operations Research, v. 8, n. 2, p. 219-223, 1960.

MARTINS, Petrônio Garcia; LAUGENI, Fernando Piero. **Administração da produção**. 2005.

MORALES, Sergio Gomez; RONCONI, Débora Pretti. **Formulações matemáticas e estratégias de resolução para o problema job shop clássico**. Production, v. 26, n. 3, 2016.

MOREIRA, Daniel Augusto. **Administração da produção e operações**.—2. ed. rev. e ampl. São Paulo, Cengage Learning, 2009.

PADILHA, Thais Cássia Cabral; MARINS, Fernando Augusto Silva. **Sistemas ERP: características, custos e tendências**. Production, v. 15, n. 1, p. 102-113, 2005.

PEDROSO, Marcelo Caldeira; CORRÊA, Henrique Luiz. **Sistemas de programação da produção com capacidade finita: uma decisão estratégica?** Revista de Administração de Empresas, v. 36, n. 4, 1996.

QUEZADO, Paulo Cesar Augustus Mendes et al. **Programação do fluxo produtivo de máquinas e equipamentos para moinhos sob encomenda utilizando PERT/CPM e heurísticas**. 1999.

SAUVEY, Christophe; SAUER, Nathalie. **Two NEH Heuristic Improvements for Flowshop Scheduling Problem with Makespan Criterion**. Algorithms, 2020.

TUBINO, Dalvio Ferrari. **Planejamento e Controle da Produção: teoria e prática**. Teoria e Prática. 3. ed. São Paulo: Atlas, 2017.

VAESSENS, Robert Johannes Maria; AARTS, Emile HL; LENSTRA, Jan Karel. **Job shop scheduling by local search**. INFORMS Journal on computing, v. 8, n. 3, p. 302-317, 1996.

APÊNDICE 1

Option optcr=0;

Set P Peças /Job1*job10/
 T Tarefas /T1*T10/
 M Máquinas /1*10/;

Table TP(p,t) Tempo de processamento da peça "p" na tarefa "t"

| | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 | t9 | t10 |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| job1 | 4.4 | 0.5 | 5.8 | 9.7 | 0.9 | 8.4 | 7.7 | 9.6 | 5.8 | 8.9 |
| job2 | 1.5 | 3.1 | 8.7 | 5.7 | 7.7 | 8.5 | 8.1 | 3.9 | 7.3 | 2.1 |
| job3 | 8.2 | 2.2 | 1.0 | 7.0 | 4.9 | 4.0 | 3.4 | 4.8 | 8.0 | 7.1 |
| job4 | 9.1 | 1.7 | 6.2 | 7.5 | 4.7 | 1.1 | 0.7 | 7.2 | 3.5 | 5.5 |
| job5 | 7.1 | 9.0 | 7.5 | 6.4 | 9.4 | 1.5 | 1.2 | 6.7 | 2.0 | 5.0 |
| job6 | 7.0 | 9.3 | 7.7 | 2.9 | 5.8 | 9.3 | 6.8 | 5.7 | 0.7 | 5.2 |
| job7 | 8.7 | 6.3 | 2.6 | 0.6 | 8.2 | 2.7 | 5.6 | 4.8 | 3.6 | 9.5 |
| job8 | 3.6 | 1.5 | 4.1 | 7.8 | 7.6 | 8.4 | 3.0 | 7.6 | 3.6 | 0.8 |
| job9 | 8.8 | 8.1 | 1.3 | 8.2 | 5.4 | 1.3 | 2.9 | 4.0 | 7.8 | 7.5 |
| job10 | 8.8 | 5.4 | 6.4 | 3.2 | 5.2 | 0.6 | 5.4 | 8.2 | 0.6 | 2.6 |

Table MAQ(p,t) ID numérico da máquina que cada peça "p" deve processada em cada tarefa "t"

| | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 | t9 | t10 |
|-------|----|----|----|----|----|----|----|----|----|-----|
| job1 | 3 | 4 | 6 | 5 | 1 | 8 | 9 | 10 | 2 | 7 |
| job2 | 5 | 8 | 2 | 9 | 1 | 4 | 3 | 6 | 10 | 7 |
| job3 | 10 | 7 | 5 | 4 | 2 | 1 | 9 | 3 | 8 | 6 |
| job4 | 2 | 3 | 8 | 6 | 9 | 5 | 4 | 7 | 10 | 1 |
| job5 | 7 | 2 | 4 | 1 | 3 | 9 | 5 | 8 | 10 | 6 |
| job6 | 8 | 6 | 9 | 3 | 5 | 7 | 4 | 2 | 10 | 1 |
| job7 | 7 | 2 | 5 | 6 | 3 | 4 | 8 | 9 | 10 | 1 |
| job8 | 1 | 6 | 9 | 10 | 4 | 7 | 5 | 8 | 3 | 2 |
| job9 | 6 | 3 | 4 | 7 | 5 | 8 | 9 | 10 | 2 | 1 |
| job10 | 10 | 5 | 7 | 8 | 1 | 3 | 9 | 6 | 4 | 2 |

alias (p,p2), (t,t2);

Table SP(p,p2) Desenhos suplementares que precisam ser finalizados antes de ser iniciado o processo na peça principal

| | Job1 |
|-------|------|
| Job2 | 0 |
| Job3 | 0 |
| Job4 | 0 |
| Job5 | 0 |
| Job6 | 0 |
| Job7 | 0 |
| Job8 | 0 |
| Job9 | 0 |
| Job10 | 0 |

Scalar TT Tempo de transporte /4.15/;

51

Set NoOverlap(p,t,p2,t2);

NoOverlap(p,t,p2,t2)\$(**ord**(p)<**ord**(p2) **and** MAQ(p,t)=MAQ(p2,t2)
and MAQ(p,t) <> 0) = **yes** ;

Variables x(p,t) Momento de início do processamento
y(p,t,p2,t2) Binário usado para garantir não overlap
z Makespan_Objetivo
End(p,t) Momento de fim de uma tarefa
Pa(p,t) Ativa estação de trabalho

positive variable x, End;

binary variable y;

Scalar G Número grande;

TMAX=**sum**((p,t),TP(p,t));

Equations E1 Função objetivo
E2 Não ocorre tarefas simultâneas na mesma máquina
E3 Não ocorre tarefas simultâneas na mesma máquina
E4 Precedência
E5 calcula o final de cada tarefa
E6 Precedência de suplementares
E7 Ativa estação de trabalho
E8 Inativa estação de trabalho;

E1(p,t) ..z=g=x(p,t)+TP(p,t);
E2(NoOverlap(p,t,p2,t2)) ..x(p2,t2)=G=x(p,t)+TP(p,t)-G*y(p,t,p2,t2);
E3(NoOverlap(p,t,p2,t2)) ..x(p,t)=G=x(p2,t2)+TP(p2,t2)- G*(1-
y(p,t,p2,t2));
E4(p,t)\$(**ord**(t)<**card**(t)) ..x(p,t+1)=G=(x(p,t)+TP(p,t)+ TT*Pa(p,t));
E5(p,t) ..End(p,t)=e=x(p,t)+TP(p,t)+TT*Pa(p,t);
E6(p,p2)\$(SP(p,p2)eq 1) ..x(p,'T1')=G=x(p2,'T10')+ TP(p2,'T10')+TT;
E7(p,t)\$(TP(p,t)<> 0) ..Pa(p,t)=e=1 ;
E8(p,t)\$(TP(p,t)eq 0) ..Pa(p,t)=e=0 ;

Model PlanProd /all/;

Solve PlanProd minimizing z using MIP;

Display x.l, Machine, z.l, End.l, Pa.l;

APÊNDICE 2

Option optcr=0;

Set P Peças /Job1*job10/
 T Tarefas /T1*T10/
 M Maquinas /1*10/
 D Dias /1*22/
 M1(p) Subgrupo MAQ1 /Job1*Job5/
 M2(p) Subgrupo MAQ2 /Job6*Job10/;

Alias (d,d2);

Alias (p,p2);

Table TP(p,t) Tempo de processamento da peça "p" na tarefa "t"

| | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 | t9 | t10 |
|-------|-----|-----|-----|-----|-----|-----|-----|-----|-----|-----|
| job1 | 4.4 | 0.5 | 5.8 | 9.7 | 0.9 | 8.4 | 7.7 | 9.6 | 5.8 | 8.9 |
| job2 | 1.5 | 3.1 | 8.7 | 5.7 | 7.7 | 8.5 | 8.1 | 3.9 | 7.3 | 2.1 |
| job3 | 8.2 | 2.2 | 1.0 | 7.0 | 4.9 | 4.0 | 3.4 | 4.8 | 8.0 | 7.1 |
| job4 | 9.1 | 1.7 | 6.2 | 7.5 | 4.7 | 1.1 | 0.7 | 7.2 | 3.5 | 5.5 |
| job5 | 7.1 | 9.0 | 7.5 | 6.4 | 9.4 | 1.5 | 1.2 | 6.7 | 2.0 | 5.0 |
| job6 | 7.0 | 9.3 | 7.7 | 2.9 | 5.8 | 9.3 | 6.8 | 5.7 | 0.7 | 5.2 |
| job7 | 8.7 | 6.3 | 2.6 | 0.6 | 8.2 | 2.7 | 5.6 | 4.8 | 3.6 | 9.5 |
| job8 | 3.6 | 1.5 | 4.1 | 7.8 | 7.6 | 8.4 | 3.0 | 7.6 | 3.6 | 0.8 |
| job9 | 8.8 | 8.1 | 1.3 | 8.2 | 5.4 | 1.3 | 2.9 | 4.0 | 7.8 | 7.5 |
| job10 | 8.8 | 5.4 | 6.4 | 3.2 | 5.2 | 0.6 | 5.4 | 8.2 | 0.6 | 2.6 |

Table MAQ(p,t) ID numérico da máquina que cada peça "p" deve processada em cada tarefa "t"

| | t1 | t2 | t3 | t4 | t5 | t6 | t7 | t8 | t9 | t10 |
|-------|----|----|----|----|----|----|----|----|----|-----|
| job1 | 3 | 4 | 6 | 5 | 1 | 8 | 9 | 10 | 2 | 7 |
| job2 | 5 | 8 | 2 | 9 | 1 | 4 | 3 | 6 | 10 | 7 |
| job3 | 10 | 7 | 5 | 4 | 2 | 1 | 9 | 3 | 8 | 6 |
| job4 | 2 | 3 | 8 | 6 | 9 | 5 | 4 | 7 | 10 | 1 |
| job5 | 7 | 2 | 4 | 1 | 3 | 9 | 5 | 8 | 10 | 6 |
| job6 | 8 | 6 | 9 | 3 | 5 | 7 | 4 | 2 | 10 | 1 |
| job7 | 7 | 2 | 5 | 6 | 3 | 4 | 8 | 9 | 10 | 1 |
| job8 | 1 | 6 | 9 | 10 | 4 | 7 | 5 | 8 | 3 | 2 |
| job9 | 6 | 3 | 4 | 7 | 5 | 8 | 9 | 10 | 2 | 1 |
| job10 | 10 | 5 | 7 | 8 | 1 | 3 | 9 | 6 | 4 | 2 |

Parameter CHM(m) Custo da Hora Máquina

| | |
|---|-------|
| / | |
| 1 | 77.41 |
| 2 | 77.41 |
| 3 | 77.41 |
| 4 | 77.41 |
| 5 | 77.41 |

```

6      77.41
7      77.41
8      77.41
9      77.41
10     77.41
/

```

Table MCOP (m, d) Multiplicador do custo de operação da máquina por dia

```

      1  2  3  4  5  6  7  8  9  10 11  ... 22
1  1  1  1  1  1  2  3  1  1  1  1  ... 1
2  1  1  1  1  1  2  3  1  1  1  1  ... 1
3  1  1  1  1  1  2  3  1  1  1  1  ... 1
4  1  1  1  1  1  2  3  1  1  1  1  ... 1
5  1  1  1  1  1  2  3  1  1  1  1  ... 1
6  1  1  1  1  1  2  3  1  1  1  1  ... 1
7  1  1  1  1  1  2  3  1  1  1  1  ... 1
8  1  1  1  1  1  2  3  1  1  1  1  ... 1
9  1  1  1  1  1  2  3  1  1  1  1  ... 1
10 1  1  1  1  1  2  3  1  1  1  1  ... 1

```

Parameter CAP (m) Capacidade básica de cada máquina por dia em horas

```

/
1      8.3
2      8.3
3      8.3
4      8.3
5      8.3
6      8.3
7      8.3
8      8.3
9      8.3
10     8.3
/

```

Table SP(p, p2) Indica se a linha é suplementar da coluna

```

      Job1
Job2    0
Job3    0
Job4    0
Job5    0
Job6    0
Job7    0
Job8    0
Job9    0
Job10   0

```

Scalar

Fs fator multiplicador do custo para abrir segundo turno /500/
 Fhex fator multiplicador do custo para abrir hora extra /1.5/
 CMS custo de Makespan /10/
 CS Custo de setup /0.02/
 DLM1 Deadline da MAQ1 /18/
 DLM2 Deadline da MAQ2 /18/
 G Número grande criar parâmetro e declarar como função /500/;

Variables

ct Custo total;

Positive variables

$x(p, t, d)$ Indica o tempo processado da peça p na tarefa t no dia d
 $Cop(m, d)$ Custo de operação da máquina m no dia d
 ms Makespan
 $U(m, d)$ Utilização em horas por máquina por dia
 $Up(m, d)$ Utilização percentual de máquina por dia;

Binary variables

$y(m, d)$ Binário se faz hora extra aquele dia
 $z(m, d)$ Binário se ativa segundo turno naquele dia
 $s(p, t, d)$ Setup da peça p na tarefa t no dia d
 $f(p, t, d)$ 1 se a peça p concluiu a tarefa t no dia d ;

Equations

Obj Função objetivo
 E1 Garante que as tarefas são completas
 E2 Garante o limite de capacidade diário
 E3 Calcula o custo de operação da máquina m no dia d
 E4 Garante que não haverá processamento da peça p na tarefa t após ser finalizada pelo binário f
 E5 Só libera setup da peça p na tarefa t se já houve conclusão na tarefa $t-1$ nos dias anteriores
 E6 Makespan maior que todos os prazos de finalização
 E7 Produzir a peça p na tarefa t no dia d tem que ter setup
 E8 Toda peça p tem que finalizar uma tarefa t
 E9 Restrição de prazo de entrega Máquina 1
 E10 Restrição de prazo de entrega Máquina 2
 E11 Carregamento das máquinas por dia em horas
 E12 Carregamento percentual das máquinas por dia
 E13 Garante primeiro a execução de suplementares
 E14 Fixa a variável f no ultimo dia de processamento
 E15 Mantêm o setup ativo até finalizar para reduzir WIP
 E16 Garante que não haverá setup após finalizado processo
 E17 Igual a x , s e f a zero caso a operação não exista
 E18 Quantidade máxima de setups;

```

Obj
..ct=e=(sum((m,d),Cop(m,d)))+(ms*CMS)+sum((p,t,d),s(p,t,d)*CS)
;
E1(p,t) ..sum(d,x(p,t,d))=e=TP(p,t);
E2(m,d) ..sum((p,t)$ (MAQ(p,t) Eq
ord(M)),x(p,t,d)) =l= Cap(m)+(y(m,d)*2.34)+(z(m,d)*13.36);
E3(m,d)
..Cop(m,d)=e=y(m,d)*Fhex*CHM(m)*2.34*MCOP(m,d) +
z(m,d)*Fs*CHM(m)*13.36*MCOP(m,d) +
sum((p,t),x(p,t,d)*CHM(m)*MCOP(m,d));
E4(p,t,d) ..sum((d2)$ (ord(d2) GT ord
(d)),x(p,t,d2)) =L= (1-f(p,t,d))*TP(p,t);
E5(p,t,d) $(ord(t) GT 1) ..s(p,t,d)=L=G*sum((d2) $(ord(d2)
LT ord(d)),f(p,t-1,d2));
E6(p,t,d) ..ms=G=ord(d)*f(p,t,d);
E7(p,t,d) ..x(p,t,d)=L=s(p,t,d)*G;
E8(p,t)$(MAQ(p,t) <> 0) ..sum(d,f(p,t,d))=E=1;
E9(M1(p),t,d) ..ord(d)*f(p,t,d)=L=DLM1;
E10(M2(p),t,d) ..ord(d)*f(p,t,d)=L=DLM2;
E11(m,d) ..U(m,d)=e=sum((p,t)$(MAQ(p,t) Eq
ord(M)),x(p,t,d));
E12(m,d) ..Up(m,d)=e=(U(m,d)/Cap(m))*100;
E13(p,p2,t,d)$(SP(p,p2)eq 1) ..((f(p,'T15',d)-1)*G) =L= -
sum((d2) $(ord(d2) LT ord(d)),s(p2,'T1',d2));
E14(p,t,d) ..f(p,t,d)=l=s(p,t,d);
E15(p,t,d) ..G*(s(p,t,d) + f(p,t,d-1)) =G=
s(p,t,d-1);
E16(p,t,d) ..sum((d2)$ (ord(d2) GT ord
(d)),s(p,t,d2)) =L= (1-f(p,t,d))*G;
E17(p,t,d)$(MAQ(p,t) eq 0) ..s(p,t,d)+x(p,t,d)=e=0;
E18(p,t)
..sum(d,s(p,t,d))=l=1+(TP(p,t)/8.3);

model lotsizing /All/;
$onecho > cplex.opt
threads 4
$offecho
lotsizing.optFile = 1;
solve lotsizing minizing ct using MIP;
display x.L,f.L,s.L,ms.l,y.l,z.l, MAQ, u.l, up.l, ct.l;

```